# Table of Contents

## Get Started with Prisma Cloud

- Prisma Cloud .......................................................... 11
  - Cloud Security Posture Management with Prisma Cloud ................. 11
  - Cloud Workload Protection with Prisma Cloud .................................. 13
- Prisma Cloud License Types ........................................... 14
- Prisma Cloud—How it Works .......................................... 16
- Get Prisma Cloud From the Palo Alto Networks Marketplace ............ 18
- Get Prisma Cloud From the AWS Marketplace ............................... 20
  - Annual or Longer Term Subscription ............................................. 20
  - Hourly PAYG Subscription .......................................................... 22
- Get Prisma Cloud From the GCP Marketplace ............................... 25
- Access Prisma Cloud ..................................................... 28
- Prisma Cloud—First Look ................................................ 30
- Prisma Cloud—Next Steps ............................................... 31
- NAT Gateway IP Addresses for Prisma Cloud ............................... 32
- Access the Prisma Cloud REST API ......................................... 35
- Prisma Cloud FAQs ......................................................... 37
  - Account Onboarding and SSO ................................................ 37
  - Password and Help ............................................................. 37
  - Policy and Alerts ............................................................... 38

## Connect Your Cloud Platform to Prisma Cloud

- Cloud Account Onboarding ............................................. 45
- Onboard Your AWS Account ............................................. 46
  - Add an AWS Cloud Account on Prisma Cloud ............................ 46
  - Add an AWS Organization to Prisma Cloud .................................. 53
  - Update an Onboarded AWS Account .......................................... 61
  - Set Up the Prisma Cloud Role for AWS—Manual ....................... 65
  - Prisma Cloud on AWS China .................................................. 69
  - AWS APIs Ingested by Prisma Cloud ........................................ 70
- Onboard Your Azure Account ........................................... 75
  - Azure Cloud Account Onboarding Checklist ............................... 75
  - Add an Azure Subscription on Prisma Cloud ............................. 77
  - Update an Onboarded Azure Account ....................................... 88
  - Add an Azure Active Directory Account on Prisma Cloud .......... 88
  - Set Up Your Azure Subscription for Prisma Cloud ...................... 91
  - Create a Custom Role on Azure to Enable Prisma Cloud to Access Flow Logs .................................................. 97
  - Use the Azure PowerShell Script to Add an Azure Account .......... 99
  - Troubleshoot Azure Account Onboarding ................................ 101
  - Microsoft Azure APIs Ingested by Prisma Cloud ........................ 104
- Onboard Your Google Cloud Platform (GCP) Account .................. 106
  - Permissions and Roles for GCP Account on Prisma Cloud ........... 106
  - Add Your GCP Project to Prisma Cloud ..................................... 114
  - Update an Onboarded Google Cloud Account ........................... 119
  - Enable Flow Logs for GCP Projects ........................................ 120
  - Flow Log Compression on GCP ............................................. 122
  - Add Your GCP Organization to Prisma Cloud ............................ 124
  - Enable Flow Logs for GCP Organization ................................... 130
  - Create a Service Account With a Custom Role for GCP ............... 131

---

**TABLE OF CONTENTS**  iii
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCP APIs Ingested by Prisma Cloud</td>
<td>134</td>
</tr>
<tr>
<td>Onboard Your Alibaba Cloud Account</td>
<td>137</td>
</tr>
<tr>
<td>Set Up Your Alibaba Account</td>
<td>137</td>
</tr>
<tr>
<td>Add an Alibaba Cloud Account on Prisma Cloud</td>
<td>141</td>
</tr>
<tr>
<td>Alibaba APIs Ingested by Prisma Cloud</td>
<td>146</td>
</tr>
<tr>
<td>Cloud Service Provider Regions on Prisma Cloud</td>
<td>147</td>
</tr>
<tr>
<td>AWS Regions</td>
<td>147</td>
</tr>
<tr>
<td>Azure</td>
<td>148</td>
</tr>
<tr>
<td>GCP</td>
<td>150</td>
</tr>
<tr>
<td>Alibaba Cloud</td>
<td>151</td>
</tr>
<tr>
<td>Manage Prisma Cloud Administrators</td>
<td>153</td>
</tr>
<tr>
<td>Prisma Cloud Administrator Roles</td>
<td>155</td>
</tr>
<tr>
<td>Create and Manage Account Groups on Prisma Cloud</td>
<td>156</td>
</tr>
<tr>
<td>Create an Account Group</td>
<td>156</td>
</tr>
<tr>
<td>Manage Account Groups</td>
<td>156</td>
</tr>
<tr>
<td>Create Prisma Cloud Roles</td>
<td>157</td>
</tr>
<tr>
<td>Prisma Cloud Administrator Permissions</td>
<td>159</td>
</tr>
<tr>
<td>Manage Roles in Prisma Cloud</td>
<td>166</td>
</tr>
<tr>
<td>Add Administrative Users On Prisma Cloud</td>
<td>167</td>
</tr>
<tr>
<td>Create and Manage Access Keys</td>
<td>169</td>
</tr>
<tr>
<td>Manage your Prisma Cloud Profile</td>
<td>172</td>
</tr>
<tr>
<td>Set up SSO Integration on Prisma Cloud</td>
<td>173</td>
</tr>
<tr>
<td>Set up Just-in-Time Provisioning on Okta</td>
<td>181</td>
</tr>
<tr>
<td>View Audit Logs</td>
<td>184</td>
</tr>
<tr>
<td>Define Prisma Cloud Enterprise and Anomaly Settings</td>
<td>185</td>
</tr>
<tr>
<td>Set Up Inactivity Timeout</td>
<td>185</td>
</tr>
<tr>
<td>Set Up Global Settings for Policy and Alerts</td>
<td>185</td>
</tr>
<tr>
<td>Set Up Anomaly Policy Thresholds</td>
<td>186</td>
</tr>
<tr>
<td>Manage Prisma Cloud Alerts</td>
<td>189</td>
</tr>
<tr>
<td>Prisma Cloud Alerts and Notifications</td>
<td>191</td>
</tr>
<tr>
<td>Trusted IP Addresses on Prisma Cloud</td>
<td>192</td>
</tr>
<tr>
<td>Enable Prisma Cloud Alerts</td>
<td>194</td>
</tr>
<tr>
<td>Create an Alert Rule</td>
<td>195</td>
</tr>
<tr>
<td>Configure Prisma Cloud to Automatically RemEDIATE Alerts</td>
<td>199</td>
</tr>
<tr>
<td>Send Prisma Cloud Alert Notifications to Third-Party Tools</td>
<td>203</td>
</tr>
<tr>
<td>Send Alert Notifications to Amazon SQS</td>
<td>203</td>
</tr>
<tr>
<td>Send Alert Notifications to Azure Service Bus Queue</td>
<td>204</td>
</tr>
<tr>
<td>Send Alert Notifications Through Email</td>
<td>205</td>
</tr>
<tr>
<td>Send Alert Notifications to a Slack Channel</td>
<td>207</td>
</tr>
<tr>
<td>Send Alert Notifications to Splunk</td>
<td>208</td>
</tr>
<tr>
<td>Send Alert Notifications to Jira</td>
<td>208</td>
</tr>
<tr>
<td>Send Alert Notifications to Google Cloud SCC</td>
<td>209</td>
</tr>
<tr>
<td>Send Alert Notifications to ServiceNow</td>
<td>210</td>
</tr>
<tr>
<td>Send Alert Notifications to Webhooks</td>
<td>211</td>
</tr>
<tr>
<td>Send Alert Notifications to PagerDuty</td>
<td>211</td>
</tr>
<tr>
<td>Send Alert Notifications to AWS Security Hub</td>
<td>212</td>
</tr>
<tr>
<td>Send Alert Notifications to Microsoft Teams</td>
<td>212</td>
</tr>
<tr>
<td>Send Alert Notifications to Cortex XSOAR</td>
<td>214</td>
</tr>
<tr>
<td>View and Respond to Prisma Cloud Alerts</td>
<td>215</td>
</tr>
<tr>
<td>Generate Reports on Prisma Cloud Alerts</td>
<td>219</td>
</tr>
<tr>
<td>Alert Payload</td>
<td>222</td>
</tr>
</tbody>
</table>
**Prisma Cloud DevOps Security** ................................................................................ 365

Secure Your Infrastructure Automation ........................................................................ 367

Prisma Cloud Plugins.................................................................................................. 368

Set Up Your Prisma Cloud Configuration File for IaC Scan ........................................ 370

- Configure IaC Scan to Support Terraform................................................................. 370
- Configure IaC Scan to Support AWS CloudFormation .............................................. 371
- Configure IaC Scan to Support Kubernetes.............................................................. 372
- Configure Prisma Cloud Tags.................................................................................... 372

Use the Prisma Cloud Extension for AWS DevOps........................................................ 373

- Set Up IaC Scanning with AWS CodePipeline ......................................................... 373
- Set Up Container Image Scanning with AWS CodeBuild ........................................... 382
- poll.sh....................................................................................................................... 385

Use the Prisma Cloud Extension for Azure DevOps..................................................... 395

- Install and Configure the Prisma Cloud Extensions .................................................... 395
- Set up a Custom Task for IaC Scanning .................................................................... 398
- Set Up Container Image Scanning ............................................................................ 401
- Set Up RASP Defender.............................................................................................. 404
- Sample YAML File..................................................................................................... 405

Use the Prisma Cloud Plugin for CircleCI.................................................................. 406

Use the Prisma Cloud Plugin for IntelliJ IDEA .......................................................... 415

- Install the Prisma Cloud Plugin for IntelliJ ............................................................... 415
- Configure the Prisma Cloud Plugin for IntelliJ ......................................................... 416
- Scan Using the Prisma Cloud Plugin for IntelliJ ....................................................... 417

Use the Prisma Cloud App for GitHub ....................................................................... 422

- Set up the Prisma Cloud App Files for GitHub ....................................................... 422
- Install the Prisma Cloud App for GitHub ................................................................. 425

Use the Prisma Cloud Extension for GitLab ................................................................ 429

- Use the Prisma Cloud Extension for the GitLab CI/CD Pipeline ................................ 429
- Use the Prisma Cloud Extension for GitLab SCM ..................................................... 443

Use the Prisma Cloud Extension for Visual Studio Code ........................................... 449

- Install Prisma Cloud Extension for Visual Studio Code ........................................... 449
- Configure the Prisma Cloud Extension for VS Code ................................................. 449
- Scan Using the Prisma Cloud VS Code Extension ..................................................... 451

Use the Prisma Cloud IaC Scan REST API .................................................................. 453

- REST API Authentication......................................................................................... 453
- REST API Base URL................................................................................................. 453
- IaC Scan API Version 2............................................................................................ 453
- REST API Request to Scan Terraform Files (Deprecated) ........................................ 460
- REST API Request to Scan AWS CloudFormation Templates (Deprecated) .............. 465
- REST API Request to Scan Kubernetes Templates (Deprecated) .............................. 467
# Prisma Cloud Data Security

What is Included with Prisma Cloud Data Security ............................................................................. 473
Enable the Prisma Cloud Data Security Module .................................................................................... 474
  Add a New AWS Account and Enable Data Security ........................................................................... 476
  Edit an AWS Account Onboarded on Prisma Cloud to Enable Data Security ...................................... 483
Monitor Data Security Scan Results on Prisma Cloud ....................................................................... 493
  Use the Data Policies to Scan ............................................................................................................. 493
  Data Security settings .......................................................................................................................... 494
  Data Dashboard .................................................................................................................................. 496
  Data Inventory .................................................................................................................................... 499
  Resource Explorer ............................................................................................................................... 502
Object Explorer ..................................................................................................................................... 503
Exposure Evaluation ............................................................................................................................... 504
Supported File Extensions—Prisma Cloud Data Security .................................................................... 506
Supported Data Profiles & Patterns ....................................................................................................... 507
Disable Prisma Cloud Data Security and Offboard AWS account ....................................................... 515
Guidelines for Optimizing Data Security Cost on Prisma Cloud ............................................................. 517
  Cost Implications and Control ............................................................................................................ 517
API Throttling and Egress Implications ................................................................................................. 517

Legacy Prisma Cloud IaC Scan REST API (Deprecated) ......................................................................... 468
Get Started with Prisma Cloud

Prisma™ Cloud is an API-based cloud service that connects to your cloud environments in just minutes and aggregates volumes of raw configuration data, user activity information, and network traffic to analyze and produce concise and actionable insights.

- Prisma Cloud
- Prisma Cloud—How it Works
- Prisma Cloud License Types
- Get Prisma Cloud From the Palo Alto Networks Marketplace
- Get Prisma Cloud From the AWS Marketplace
- Get Prisma Cloud From the GCP Marketplace
- Access Prisma Cloud
- Prisma Cloud—First Look
- Prisma Cloud—How it Works
- NAT Gateway IP Addresses for Prisma Cloud
- Access the Prisma Cloud REST API
- Prisma Cloud FAQs
Prisma Cloud

Prisma™ Cloud is a cloud infrastructure security solution and a Security Operations Center (SOC) enablement tool that enables you to address risks and secure your workloads in a heterogeneous environment (hybrid and multicloud) from a single console. It provides complete visibility and control over risks within your public cloud infrastructure and enables you to manage vulnerabilities, detect anomalies, ensure compliance, and provide runtime defense in heterogeneous environments, such as Windows, Linux, Kubernetes, Red Hat OpenShift, AWS Lambda, Azure Functions, and GCP Cloud Functions. The main capabilities are:

- Continuous security assessment of your configuration, compliance monitoring, and integration with external services for incident management and remediation to address issues identified on your resources in the public cloud. These capabilities are completely API-based and you can configure these capabilities using the different tabs on the Prisma Cloud administrative console. For an overview, see Cloud Security Posture Management with Prisma Cloud.

- Consistent visibility and runtime defense with least-privilege microsegmentation for physical machines, virtual machines, containers, and serverless workloads—regardless of location. These capabilities require an agent and the API. Use the Compute tab on the Prisma Cloud administrative console to set up and monitor this functionality. For an overview, see Cloud Workload Protection with Prisma Cloud.

Cloud Security Posture Management with Prisma Cloud

The API-based service enables granular visibility into your resources deployed on public cloud platforms—Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure—and in to the network traffic flows to these resources from the internet and between instances. Prisma™ Cloud also provides threat detection and response for resource misconfigurations and workload vulnerabilities and provides visibility into user activity within each cloud environment. Tracking user activity helps you identify account compromises, escalation of privileges with privileged accounts, and insider threats from malicious users, unauthorized activity, and inadvertent errors. Prisma Cloud continuously monitors your cloud environments to help ensure that your cloud infrastructure is protected from these security threats.

In addition to providing visibility and reducing risks, Prisma Cloud facilitates Security Operations Center (SOC) enablement and adherence to compliance standards. As the service automatically discovers and monitors compliance for new resources that are deployed in your cloud environment, it enables you to implement policy guardrails to ensure that resource configurations adhere to industry standards and helps you integrate configuration change alerts into DevSecOps workflows that automatically resolve issues as they are discovered. This capability streamlines the process of identifying issues and detecting and responding to a list of prioritized risks to maintain an agile development process and operational efficiency.
Here are some highlights of Prisma Cloud:

- **Comprehensive Visibility**— Enables you to view your resources—deployed on multiple cloud infrastructure platforms—from a single console. In addition to providing a consolidated view of the resources across the cloud platforms, Prisma Cloud integrates with threat intelligence feeds, vulnerability scanners, and Security Information and Event Management (SIEM) solutions to help you build a contextual view of your cloud deployments.

- **Policy Monitoring**— Enables you to use Prisma Cloud, which includes Security policies based on industry standards, to continuously monitor for violations. Because cloud platforms enable agility and your users can create, modify, and destroy resources on-demand, these user actions often occur without any security oversight. Prisma Cloud provides hundreds of out-of-the-box policies for common security and compliance standards, such as GDPR, PCI, CIS, and HIPAA. You can also create custom policy rules to address specific needs or to customize the default policy rules.

- **Anomaly Detection**— Automatically detects suspicious user and network behavior using machine learning. Prisma Cloud consumes data about your AWS resources from AWS CloudTrail, AWS Inspector, and Amazon GuardDuty to detect account compromises and insider threats. This service uses machine learning to score the risk level for each cloud resource based on the severity of business impact, policy violations, and anomalous behavior. Risk scores are then aggregated so that you can prioritize your alerts and benchmark risk postures across your entire environment.

- **Contextual Alerting**— Leverages highly contextual alerts for prioritization and rapid response. Because Prisma Cloud also integrates with external vulnerability services, such as AWS Inspector, Tenable.io, and Qualys, to continuously scan your environment, it has additional context to identify unexpected and potentially unauthorized and malicious activity. For example, the service scans for unpatched hosts, escalation of privileges, and use of exposed credentials, and also scans communication for malicious IP addresses, URLs, and domains.

- **Cloud Forensics**— Enables you to go back to any point in time and investigate an issue within seconds. To help you identify security blind spots and investigate issues, Prisma Cloud monitors network traffic from sources such as AWS VPC flow logs, Azure flow logs, GCP flow logs, Amazon GuardDuty, and user activity from AWS CloudTrail and Azure.

- **Compliance Reporting**— Reports your risk posture to your management team, to your board of directors, and to auditors.

- **Limited GA** Data Security—Scans data stored on AWS S3 buckets and provides visibility on the scan results directly on the Prisma Cloud dashboard. The data security capabilities include predefined data policies and associated data classification profiles such as PII, Financial, or Healthcare & Intellectual Property that scan your objects stored in the S3 bucket to identify exposure—how sensitive information
is kept private, or exposed or shared externally, or allows unauthorized access. It also uses the WildFire service to detect known and unknown malware in these objects.

Cloud Workload Protection with Prisma Cloud

Prisma™ Cloud offers cloud workload protection, as either a SaaS option or a self-hosted solution that you deploy and manage (review your options).

The SaaS option, available with the Prisma Cloud Enterprise Edition, offers a single management console for threat detection, prevention, and response for your heterogeneous environment where your teams are leveraging public cloud platforms and a rich set of microservices to rapidly build and deliver applications. The Compute tab on the Prisma Cloud administrative console enables you to define policy and to monitor and protect the hosts, containers, and serverless functions within your environment.

To monitor the workloads, you must deploy Prisma Cloud Defenders: the agents. All Defenders, regardless of their type, connect back to the console using WebSocket over port 8084 to retrieve policies and enforce vulnerability and compliance blocking rules to the environments where they are deployed, and to send data back to the Compute tab within the Prisma Cloud administrative console. For documentation on how to get started with deploying Defenders, configuring policies, viewing alerts, and interpreting the data on Radar, see the Prisma Cloud Administrator’s Guide (Compute). For administrative user management, such as integrating single sign-on, setting up custom roles, and creating access keys, use the Settings tab on the Prisma Cloud administrative console outlined in this document.
Prisma Cloud License Types

Prisma Cloud is available as a one-, two-, or three-year subscription in the following three editions:

- **Prisma Cloud Business Edition**—License includes configuration security posture management (CSPM), compliance reporting, automated remediation, custom policy creation, and a standard success plan. The Business edition is tailored for investigating resource misconfigurations and verifying adherence to compliance standards so that you can take steps to implement policies and regulations that enable you to secure your public cloud deployments and comply with security standards. The Business edition is powered entirely through the application programming interface (API) and delivered exclusively as a software-as-a-service (SaaS) model.

- **Prisma Cloud Compute Edition**—License includes workload protection for hosts, containers, and serverless deployments in any cloud or on-premises environment. This is an agent-based approach to protect resources across the application lifecycle. Unlike the Prisma Cloud Business or Enterprise editions, this is a self-operated software solution that you download and run in your own environments—whether public, private, or hybrid clouds—including entirely air-gapped environments.

- **Prisma Cloud Enterprise Edition**—License includes all features included in the Business edition and Compute edition licenses (including a standard success plan). The Enterprise edition—delivered as a SaaS model—combines API- and agent-based approaches to deliver comprehensive host, container, serverless, IaaS, PaaS, IAM, network, and storage security for cloud and on-premises environments. It protects you from the most sophisticated threats with advanced machine learning capabilities, network security monitoring, user entity behavior analysis (UEBA) for detecting location- and activity-based anomalies, and integration with host vulnerability management tools.

You can optionally upgrade to a premium success plan that includes 24x7 access to customer success experts, custom workshops, implementation assistance, configuration and planning, online documentation, and best practices guides.

Each of these editions has a different capacity unit and unit price in Prisma Cloud Credits. The number of credits required to secure your assets can vary across the different Prisma Cloud modules such as Visibility, Compliance & Governance, Compute Security, or Network Security. Refer to the Prisma Cloud Licensing and Editions Guide for details.

Licensing is sold in increments of 100 credits and you estimate the number of units you need to monitor and protect. Your usage data is based on the number of capacity units that you are consuming for each Prisma Cloud module every hour, and the **Time Range** is averaged for daily, weekly, monthly and quarterly usage to prevent overages based on short-term bursts.

After you Connect Your Cloud Platform to Prisma Cloud or deploy Prisma Cloud Defenders, you can review the actual number of licensable assets that you’re securing with the service. The **Settings > Licensing** tab displays a resource count in a tabular format and a graphical view of the resource trend as a line graph. Use the License Usage graph to review the average number of billable resources monitored across your overall cloud environments or view the pattern for each cloud type, your actual usage trends, and the number of
Prisma Cloud credits you have purchased. While the table depicts the total number of resources for which credits are applied, the graph represents the actual credits consumed over the time range and is averaged across the time period.

If you have environments where you have deployed Prisma Cloud Defenders either on-premises or on private or public cloud platforms that are not being monitored by Prisma Cloud, such as on OpenShift, on-prem Kubernetes clusters, or AWS Fargate, select Non-onboarded Cloud Account Resources to view the details on credits used towards resources deployed on cloud environments that are not onboarded on Prisma Cloud.

The default time range is three months but you can select a time range of your choice. You can also download and share the licensing information in a zip file that includes information on each cloud platform in CSV file format. For a time period of 3 days or less, you can download hourly usage data, and data on daily usage for a time period greater than 3 days.

For details on how credits are calculated for Prisma Cloud Defenders, see Prisma Cloud Compute—Licensing.
Prisma Cloud—How it Works

As a Security Operations Center (SOC) enablement tool, Prisma™ Cloud helps you identify issues in your cloud deployments and then respond to a list of prioritized risks so that you can maintain an agile development process and operational efficiency.

When you add a cloud account to Prisma Cloud, the IaaS Integration Services module ingests data from flow logs, configuration logs, and audit logs in your cloud environment over an encrypted connection and stores the encrypted metadata in RDS and Redshift instances within the Prisma Cloud AWS Services module. You then use the Prisma Cloud administrative console or the APIs to interact with this data to configure policies, to investigate and resolve alerts, to set up external integrations, and to forward alert notifications. The Enterprise Integration Services module enables you to leverage Prisma Cloud as your cloud orchestration and monitoring tool and to feed relevant information to existing SOC workflows. The integration service ingests information from your existing single sign-on (SSO) identity management system and allows you to feed information back in to your existing SIEM tools and to your collaboration and helpdesk workflows.

To ensure the security of your data and high availability of Prisma Cloud, Palo Alto Networks makes Security a priority at every step. The Prisma Cloud architecture uses Cloudflare for DNS resolution of web requests and for protection against distributed denial-of-service (DDoS) attacks. The following diagram represents the infrastructure within a region:
For data redundancy of stateful components, such as RDS and Redshift, and of stateless components, such as the application stack and Redis (used primarily as a cache), the service uses native AWS capabilities for automated snapshots or has set up automation scripts using AWS Lambda and SNS for saving copies to S3 buckets.

Additionally, to ensure that these snapshots and other data at rest are safe, Prisma Cloud uses AWS Key Management Service (KMS) to encrypt and decrypt the data. To protect data in transit, the infrastructure terminates the TLS connection at the Elastic Load Balancer (ELB) and secures traffic between components within the data center using an internal certificate until it is terminated at the application node. This ensures that data in transit is encrypted using SSL. And, lastly, for workload isolation and micro segmentation, the built-in VPC security controls in AWS securely connect and monitor traffic between application workloads on AWS.
Get Prisma Cloud From the Palo Alto Networks Marketplace

Purchase or try Prisma™ Cloud from Palo Alto Networks Marketplace. Within 24 hours of your purchase, you will get access to the Prisma Cloud tenant that is provisioned for you.

STEP 1 | Go to Palo Alto Networks Marketplace.

STEP 2 | Create an account.

Required only if you do not have a Palo Alto Networks Customer Support Portal (CSP) account.

1. Enter the personal and company information requested in the form. Required fields are indicated with red asterisks.
2. Accept the privacy agreement and Create an account.
3. Look for the welcome email in your inbox and click the link in that email to continue the activation process.
4. Click here on the page that displays and enter your account credentials on the Palo Alto Networks single sign-on (SSO) page.
5. Answer the security questions and Save your changes.

You will be logged in to Palo Alto Networks Marketplace.

STEP 3 | Scroll down and select the Prisma Cloud app on Palo Alto Networks Marketplace and then View app.

STEP 4 | Select Free Trial or Buy Now.

The Enterprise edition is available in either the Trial or Buy Now option; the Business edition is available only as a Buy Now subscription. (See Prisma Cloud License Types for details.)

- **Buy Now:**
  1. Select the license edition—Business or Enterprise.
  2. Select the Term—1 year.
  3. Select Billing cycle—yearly or monthly.
  4. Select the Number of workloads you need to secure.

    The range is 100 to 900 workloads; contact sales if you need to secure more than 900 workloads.
  5. Opt in to purchase the Premium Success Plan.

    If you do not opt in, you will receive the standard success plan.
  6. Select the region in which you want to provision your Prisma Cloud instance.

    Available regions are Australia, Germany, GovCloud, and US.
  7. Click Next and review the summary.
  8. Enter your Billing information.
  9. Review and Accept the terms and conditions of the EULA and click Next.

    You will see a success message on screen. Then, within the next 24 hours, you will receive two emails: the first is an order confirmation email and the second is a welcome email that includes a link you will use to log in to your Prisma Cloud tenant. You can also access your Prisma Cloud tenant from the Palo Alto Networks hub using the Prisma Cloud tile.
  10. Enable Auto Renewal to ensure that you have uninterrupted monitoring and protection for your public cloud deployments.
• Free trial:
  1. Select the region in which you want to provision your Prisma Cloud instance. Available regions are Australia, Germany, GovCloud, and US.
  2. Accept the EULA and click Next.

  A license confirmation displays.

  Within the next 24 hours, you will receive a welcome email that includes a link to log in to the Prisma Cloud tenant that is provisioned for you. You can directly access the Prisma Cloud instance or log in to the hub and click the Prisma tile to log in to your Prisma Cloud tenant.
Get Prisma Cloud From the AWS Marketplace

Purchase Prisma™ Cloud as a SaaS subscription directly from the AWS Marketplace. You can choose a 1-, 3-, or 5-year subscription for the Prisma Cloud Business or Enterprise Edition, or as a PAYG subscription based on hourly usage, in the Prisma Cloud Enterprise Edition. The PAYG model offers you a 15-day trial and is available in the Enterprise Edition only.

Within 24 hours of your purchase, you will get access to the Prisma Cloud tenant that is provisioned for you.

- Annual or Longer Term Subscription
- Hourly PAYG Subscription

Annual or Longer Term Subscription

STEP 1 | Go to AWS Marketplace.

STEP 2 | Search for Palo Alto Networks on AWS Marketplace and select Prisma Cloud Threat Defense and Compliance Platform.

STEP 3 | Continue to Subscribe.

STEP 4 | Enter the details to get a Prisma Cloud subscription.

The subscription is for a 12-month period.
1. Select your autorenewal preference.

   The default for **Renewal Settings** is **Yes** so that your contract is automatically renewed before it expires. This ensures that you have uninterrupted coverage for securing your public cloud deployments.

2. Select the license edition and the number of units.

   The **Prisma Cloud License Types** are Business or Enterprise with the standard success plan. If you want to protect 1,000 workloads, enter 10 units (1 unit is 100 workloads).

3. **Create contract**.

4. **Confirm the contract and Pay Now**.

5. **Set up your account** to continue setting up Prisma Cloud.

---

*This link has a temporary access token that expires in 24 hours. If you want to come back later and provide the details to provision your Prisma Cloud instance, you must log in to your AWS account. You can then use the View or Modify link for the active AWS Marketplace software subscription on Your Software page of AWS Marketplace. You can then Click here to setup your account.*
STEP 5 | Provide the details for provisioning your Prisma Cloud instance.

1. Enter the personal and company information requested in the form.
2. Select the region where you want your Prisma Cloud tenant provisioned.
3. Select Register.

   A message informs you whether the registration was successful. Look for the welcome email in your inbox and click the link in that email to log in to the Palo Alto Networks hub.

4. Log in to the hub and click the Prisma Cloud tile to start using Prisma Cloud.

   You are now ready for Prisma Cloud—First Look and Prisma Cloud—Next Steps.

Hourly PAYG Subscription

STEP 1 | Go to AWS Marketplace.

STEP 2 | Search for Palo Alto Networks on AWS Marketplace and select Prisma Cloud Enterprise Edition.

STEP 3 | Continue to Subscribe.
STEP 4 | **Subscribe** to Prisma Cloud.

The first 15 days of the subscription are free, and you will be charged for the subscription hourly after the free period expires.
STEP 5 | Provide the details for provisioning your Prisma Cloud instance.
1. Enter the personal and company information requested in the form.
2. Select the region where you want your Prisma Cloud tenant provisioned.
3. Select Register.

A message informs you whether the registration was successful. Look for the welcome email in your inbox and click the link in that email to log in to the Palo Alto Networks hub.

4. Log in to the hub and click the Prisma Cloud tile to start using Prisma Cloud.

You are now ready for Prisma Cloud—First Look and Prisma Cloud—Next Steps.
Get Prisma Cloud From the GCP Marketplace

Purchase Prisma™ Cloud directly from Google Cloud Platform (GCP) Marketplace. Within 24 hours of your purchase, you will get access to the Prisma Cloud tenant that is provisioned for you.

**STEP 1 |** Go to GCP Marketplace and search for Prisma Cloud.

**STEP 2 |** View All Plans and choose the one to which you want to Subscribe.

View the available plans and pick the one that best meets the security and compliance requirements for your enterprise (see Prisma Cloud License Types).

**STEP 3 |** Activate the subscription you selected.

The subscription from the marketplace is for 100 workloads for a 12-month period. For a longer term or to secure more workloads, please contact your Palo Alto Networks Sales Representative.

**STEP 4 |** Provide the details for provisioning your Prisma Cloud instance.
1. Enter the personal and company information requested in the form.
2. Provide the **Tenant Name** that allows you to identify your Prisma Cloud instance.
   
   The name you enter here is displayed in the GCP marketplace subscription details and on the Prisma Cloud console.
3. Select the region where you want your Prisma Cloud tenant provisioned.
4. Select **Register**.
   
   A message informs you whether the registration was successful. After you successfully register, Palo Alto Networks sends you two emails: the first email is titled *Welcome to Prisma Cloud Support* and enables you to **Set Password**; the second email is titled *Welcome to Prisma Cloud* and it includes a link to **Get Started**. Use this link to navigate to the Palo Alto Networks hub and log in using the registered email address and the new password that you configured.

5. Verify your subscription details on GCP Marketplace.

---

*When the subscription cycle ends, your subscription will not be automatically renewed. You can upgrade your subscription at any time. If you want to change your plan to*
a more limited set of capabilities, the change is in effect at the end of the currently committed subscription term.

**STEP 5 |** Log in to the hub and click the Prisma Cloud tile to start using Prisma Cloud.

You are now ready for Prisma Cloud—First Look and Prisma Cloud—Next Steps.
Access Prisma Cloud

The welcome email you receive from Palo Alto Networks (noreply@prismacloud.paloaltonetworks.com) includes a link to where you can access your instance of Prisma Cloud. If you are the first registered user, a Palo Alto Networks Customer Support Portal (CSP) account is created for you and you can log in to Prisma Cloud to start securing your cloud deployments. For all other Prisma Cloud users, when your Prisma Cloud system administrator adds you to the tenant, you receive two emails. Use the Welcome to Palo Alto Networks Support email to activate the CSP account and set a password to access the Palo Alto Networks Support portal before you click Get Started in the Welcome to Prisma Cloud email to log in to your Prisma Cloud instance.

The link in your welcome email varies depending on whether you are using Palo Alto Networks Customer Support Portal (CSP) credentials to log in or if you are using a third-party identity provider (IdP) for Single Sign-On (SSO):

If you have a Palo Alto Networks CSP account and are not using a third-party IdP, the link enables you to log in directly to Prisma Cloud using the email address and password registered with your CSP account.

If you are using a third-party IdP and the login URL is configured on Prisma Cloud, the link redirects you to login page for your IdP and you can log in using your IdP credentials.

If you are using a third-party IdP but the login URL is not configured on Prisma Cloud, you must navigate to your IdP and click the Prisma Cloud tile there to log in using the credentials set up on your IdP.

Browser Support—To access the Prisma Cloud administrator console, Chrome version 72 or later provides the optimal user experience. The Prisma Cloud console is not explicitly tested on other browsers and, though we expect it to perform with graceful degradation, it is not guaranteed to work on other browsers.

Status Updates—Use the Prisma Cloud status page to view operational status and subscribe to updates about the service.

STEP 1 | Launch a web browser and access the URL for Prisma Cloud or go to the Palo Alto Networks hub to access the app.

The URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. Your welcome email will include one of the following URLs that is specific to the tenant provisioned for you:

- https://app.prismacloud.io
- https://app2.prismacloud.io
- https://app3.prismacloud.io
- https://app4.prismacloud.io
- https://app.ca.prismacloud.io
- https://app.eu.prismacloud.io
- https://app2.eu.prismacloud.io
On the hub, if you can see the Prisma Cloud tile in to the app because of a SAML error, it likely means that you do not have an account on that Prisma cloud instance. Contact your system administrator for an account to access that instance.

STEP 2 | Accept the EULA.

After you accept the terms and conditions of use, use the get started guide to learn the basics.

STEP 3 | Select Licensing and verify that you have the correct Prisma Cloud License Types or get started with Prisma Cloud—First Look.

STEP 4 | Switch between Prisma Cloud instances.

If you are responsible for monitoring clouds belonging to different organizations (tenants), you can use your Palo Alto Networks login credentials to access all tenants from the hub. To enable multitenant login access, your system administrator must add your email address on each Prisma Cloud tenant (see Add Administrative Users On Prisma Cloud). You will then receive an email from Palo Alto Networks to get started. By default, an administrator on Prisma cloud is designated as an Instance Administrator for that tenant only on the hub. If you want to change a role or enable access to other apps, see roles.

With Palo Alto Networks CSP credentials, you can click the app switcher on the hub and then select a different tenant to which to switch between instances.

If you see the serial number for your instance and want to change it to a descriptive label, navigate to the Settings page using gear > Manage Apps in the upper-right. Click directly on the serial number and rename it. This new name displays only on the hub and it does not automatically apply to your Prisma Cloud instance name.
When you Access Prisma Cloud, you first see the Alerts. You can then use the following tabs to interact with the data and visualize the traffic flow and connection details to and from the different resources in your cloud deployment; review the default policy rules and compliance standards; and explore how the web interface is organized to help you and DevSecOps teams to monitor cloud resources.

- **Dashboard**—Get a graphical view of the health cloud (security and compliance posture) of your assets deployed across multiple public cloud environments.
- **Investigate**—Identify security threats and vulnerabilities, create and save investigative queries, and analyze impacted resources.
- **Policies**—Configure policies to maintain compliance and security.
- **Compliance**—Monitor your cloud accounts against compliance standards (such as NIST, SOC 2, HIPAA, PCI, GDPR, ISO 27001:2013), create custom policies, and generate reports for offline viewing.
- **Alerts**—View the list of discovered violations and anomalies, drill in to the details and look up remediation options, and create alert rules and notification templates.
- **Compute**—Deploy the Prisma Cloud Defender in hosts, containers, and serverless environments to identify vulnerabilities, detect anomalous behavior, and provide least privilege micro-segmentation and runtime defense across the entire application lifecycle from CICD to runtime. You can deploy Prisma Cloud Defenders on heterogeneous environments, including Windows, Linux, Kubernetes, OpenShift, AWS Lambda, Azure Functions, GCP Cloud Functions, and more. Defenders can create IP table rules on the host to observe network traffic and to enforce both CNNF and CNAF firewall policies. For details see the Prisma Cloud Administrator's Guide (Compute).
- **Settings**—Add new cloud accounts and set up Prisma Cloud administrative users. You can also set up account groups, create users, associate roles and permissions, add external integrations including SAML integration (SSO), view audit logs, add trusted IP addresses, view license and usage reporting, and modify the browser session timeout.
- **Profile**—Maintain your profile, manage your credentials, and change the UI display language.
Prisma Cloud—Next Steps

Now that you have familiarized yourself with Prisma™ Cloud, here are some things to consider next so that you can begin protecting your cloud resources:

- Add the NAT Gateway IP Addresses for Prisma Cloud so that you can access Prisma Cloud from your network.
- Create and Manage Account Groups on Prisma Cloud.
- Connect Your Cloud Platform to Prisma Cloud.
- Add Administrative Users On Prisma Cloud.
- Create an Alert Rule.
- Prisma Cloud Integrations.
- Prisma Cloud Dashboards.
- Investigate Incidents on Prisma Cloud.
- Create a Policy on Prisma Cloud.
- Add a New Compliance Report.
- ☐ Deploy Prisma Cloud Defenders for securing host, container, and serverless functions.
NAT Gateway IP Addresses for Prisma Cloud

Prisma™ Cloud uses the following NAT gateway IP addresses. To ensure that you can access Prisma Cloud and the API for any integrations that you enabled between Prisma Cloud and your incidence response workflows, or for your Prisma Cloud Defenders to communicate with the Prisma Cloud Compute Console, review the list and update the IP addresses in your allow lists.

<table>
<thead>
<tr>
<th>URL</th>
<th>Source IP Address to Allow</th>
</tr>
</thead>
</table>
| app.prismacloud.io | 3.217.51.44  
|               | 3.218.144.244  
|               | 34.199.10.120  
|               | 34.205.176.82  
|               | 34.228.96.118  
|               | 52.201.19.205  |
| app2.prismacloud.io | 3.16.7.30  
|               | 13.59.164.228  
|               | 18.191.115.70  
|               | 18.218.243.39  
|               | 18.221.72.80  
|               | 18.223.141.221  |
| app3.prismacloud.io | 34.208.190.79  
|               | 52.24.59.168  
|               | 52.39.60.41  
|               | 52.26.142.61  
|               | 54.213.143.171  
|               | 54.218.131.166  |
| app4.prismacloud.io | 13.52.27.189  
|               | 13.52.105.217  
|               | 13.52.157.154  
|               | 13.52.175.228  
|               | 52.52.50.152  
|               | 52.52.110.223  |
| app5.prismacloud.io | 3.128.141.242/32  
|               | 3.129.241.104/32  
|               | 3.130.104.173/32  
<p>|               | 3.136.191.187/32  |</p>
<table>
<thead>
<tr>
<th>URL</th>
<th>Source IP Address to Allow</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13.59.109.178/32</td>
</tr>
<tr>
<td></td>
<td>18.190.115.80/32</td>
</tr>
<tr>
<td>app.anz.prismacloud.io</td>
<td>3.104.252.91</td>
</tr>
<tr>
<td></td>
<td>13.210.254.18</td>
</tr>
<tr>
<td></td>
<td>13.239.110.68</td>
</tr>
<tr>
<td></td>
<td>52.62.75.140</td>
</tr>
<tr>
<td></td>
<td>52.62.194.176</td>
</tr>
<tr>
<td></td>
<td>54.66.215.148</td>
</tr>
<tr>
<td>app.ca.prismacloud.io</td>
<td>15.223.59.158</td>
</tr>
<tr>
<td></td>
<td>15.223.96.201</td>
</tr>
<tr>
<td></td>
<td>15.223.127.111</td>
</tr>
<tr>
<td></td>
<td>52.60.127.179</td>
</tr>
<tr>
<td></td>
<td>99.79.30.121</td>
</tr>
<tr>
<td></td>
<td>35.182.209.121</td>
</tr>
<tr>
<td>app.prismacloud.cn</td>
<td>52.82.89.61</td>
</tr>
<tr>
<td></td>
<td>52.82.102.153</td>
</tr>
<tr>
<td></td>
<td>52.82.104.173</td>
</tr>
<tr>
<td></td>
<td>52.83.179.1</td>
</tr>
<tr>
<td></td>
<td>52.83.70.13</td>
</tr>
<tr>
<td></td>
<td>52.83.77.73</td>
</tr>
<tr>
<td>app.eu.prismacloud.io</td>
<td>3.121.64.255</td>
</tr>
<tr>
<td></td>
<td>3.121.248.165</td>
</tr>
<tr>
<td></td>
<td>3.121.107.154</td>
</tr>
<tr>
<td></td>
<td>18.184.105.224</td>
</tr>
<tr>
<td></td>
<td>18.185.81.104</td>
</tr>
<tr>
<td></td>
<td>52.29.141.235</td>
</tr>
<tr>
<td>app2.eu.prismacloud.io</td>
<td>18.200.200.125</td>
</tr>
<tr>
<td></td>
<td>3.248.26.245</td>
</tr>
<tr>
<td></td>
<td>99.81.226.57</td>
</tr>
<tr>
<td></td>
<td>52.208.244.121</td>
</tr>
<tr>
<td></td>
<td>18.200.207.86</td>
</tr>
<tr>
<td></td>
<td>63.32.161.197</td>
</tr>
<tr>
<td>app.gov.prismacloud.io</td>
<td>15.200.20.182</td>
</tr>
<tr>
<td></td>
<td>15.200.89.211</td>
</tr>
<tr>
<td>URL</td>
<td>Source IP Address to Allow</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>52.222.38.70</td>
</tr>
<tr>
<td></td>
<td>52.61.207.0</td>
</tr>
<tr>
<td></td>
<td>15.200.68.21</td>
</tr>
<tr>
<td></td>
<td>15.200.146.166</td>
</tr>
<tr>
<td>app.sg.prismacloud.io</td>
<td>13.250.248.219</td>
</tr>
<tr>
<td></td>
<td>18.139.183.196</td>
</tr>
<tr>
<td></td>
<td>52.76.28.40</td>
</tr>
<tr>
<td></td>
<td>52.76.70.227</td>
</tr>
<tr>
<td></td>
<td>52.221.36.124</td>
</tr>
<tr>
<td></td>
<td>52.221.157.53</td>
</tr>
<tr>
<td>Prisma Cloud Compute Console</td>
<td>On the Compute &gt; Manage &gt; System &gt; Downloads, find the region in the URL for Path to Console. Use that region to identify the destination IP address, which you must allow or add as trusted to access the Prisma Cloud Compute console.</td>
</tr>
<tr>
<td></td>
<td>us-east1—35.196.73.150</td>
</tr>
<tr>
<td></td>
<td>us-west1—35.233.225.166</td>
</tr>
<tr>
<td></td>
<td>asia-northeast1—34.84.195.213</td>
</tr>
<tr>
<td></td>
<td>europe-west3—34.89.249.72</td>
</tr>
</tbody>
</table>

*These IP addresses are not the outgoing address for alerts or cloud discovery.*
Access the Prisma Cloud REST API

Prisma Cloud has a REST API that enables you configure custom integrations for your cloud security needs. You can, for example, use it to automate sending alert notifications to an in-house tool you use or to extend the DevOps security capabilities for a tool that does not have an extension or plugin for Prisma Cloud. Most actions supported on the Prisma Cloud web interface are available with the REST API. See the Prisma Cloud REST API reference for details about the REST API.

Watch this!

Prisma Cloud requires an API access key to enable programmatic access to the REST API. By default, only the System Admin has API access and can enable API access for other administrators. To generate an access key, see Create and Manage Access Keys. After you obtain an access key, you can submit it in a REST API request to generate a JSON Web Token (JWT). The JWT is then used to authenticate all subsequent REST API requests on Prisma Cloud.

STEP 1 | Obtain a JWT to authenticate API requests.

The following is an example of a cURL call for a REST API request that returns a JWT. Your Prisma Cloud access key provides the request parameters. Note that an access key is made up of two parts: an Access Key ID and a Secret Key. In the body parameters, specify your Access Key ID as the string value for the `userName` and your Secret Key as the string value for the `password`.

```bash
curl -X POST \  https://api.prismacloud.io/login \  -H 'Content-Type: application/json' \  -d '{"username":"<Access Key ID>","password":"<Secret Key>"}''
```

The following shows the response for a successful request.

```json
{
  "token": "<JWT>",
  "message": "login_successful",
  "customerNames": [
    {
      "customerName": "Test",
      "tosAccepted": true
    }
  ]
}
```

The value for `token` in the response is the JWT you will use to authorize your subsequent REST API requests.

STEP 2 | Authenticate Using the JWT.
Specify the JWT in an HTTP header parameter for every Prisma Cloud REST API request. The following table shows the details of the header parameter.

<table>
<thead>
<tr>
<th>HTTP Header Parameter Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>&lt;your JWT&gt;</td>
</tr>
</tbody>
</table>

**STEP 3 | Refresh the JWT.**

The JWT is valid for 10 minutes, so you must refresh the token for continued access to the Prisma Cloud API. If you make an API request with an expired JWT request, you will receive an HTTP 401 (Unauthorized) response. The following example is a cURL call that makes an API request to obtain a refreshed JWT.

```
curl -X GET
https://api.prismacloud.io/auth_token/extend
-H 'Content-Type: application/json'
-H 'x-redlock-auth:<current JWT>'
```

The following is an example of the response to a successful request to refresh a JWT.

```
{
    "token": "<JWT>",
    "message": "login_successful",
    "customerNames": [
        {
            "customerName": "Test",
            "tosAccepted": true
        }
    ]
}
```

The value for `token` in the response is the new JWT you must specify in the HTTP header parameter of subsequent REST API requests.
Explain VPC flow logs

VPC flow logs provide a unidirectional record of network traffic that inform you about how packets flowed from A to B and from B to A (as in a separate record). They provide no direct insight into which endpoint is the server in any conversation or which endpoint initiated a conversation. You could look at which flow record has a lower timestamp and assume that the source in that record is the client but, in the case of VPC flow logs, log collection is aggregated over several-minute windows, which removes the precision required to make this a reliable indicator. Additionally, long-lived connections and connections that appear on the boundaries of batches of logs will defeat this heuristic. There are other factors to consider, such as source port vs destination port. You can also compare the count of distinct peers for a given endpoint IP address and port. Prisma™ Cloud evaluates all of these conditions, plus others, with a weight given to each measure and a historical bias. However, these measures are heuristics and are therefore not perfect.

Why do we need to list permissions for Key Vault in Azure?

To support policies based on Azure Key Vault, Prisma Cloud needs to ingest Key Vault. Prisma Cloud does not ingest the keys or secrets only their IDs and other metadata.

Which SAML binding should I use for SSO?

The Prisma Cloud SAML endpoint supports HTTP POST binding.

Password and Help

What are the rules for password similarity and reuse in Prisma Cloud?

When you create a new password, we check for its similarity with the current password. The measure of similarity between the new and the current and old password strings is determined by the minimum number of single-character edits, such as insertions, deletions, substitutions that are required to change one word in to another. We do not accept the new password string if the similarity with the current or old passwords is 70% or higher.

Example: If your current password is MenloPark.123!, then you cannot use M3nl0P@rk.123! but you could use ParkMenlo.123!.

What are the complexity requirements for creating Prisma Cloud passwords?

Passwords must have a minimum of 8 characters and a maximum of 30 characters and include one or more of each of the following: an uppercase letter, a lowercase letter, a digit (0 to 9), and a special character (“~” “!” “@” “#” “$” “%” “^” “&” “*” “(” “)” “_” “+” “=” “[” “]” “\” “|” “:” “;” “'” “<” “>” “?” and “/”).

Help icon

Check to ensure that your browser allows pop-ups. Check whether your advertisement blocking software is blocking Prisma Cloud and, if so, add the URL for your Prisma Cloud instance and app.pendo.io to the allow list. Check and disable any local firewall rules or proxies that are blocking either or both of these URLs.

Where do I submit documentation requests or report errors in the documentation?
Please let us know how we are doing at documentation@paloaltonetworks.com. When writing to us about a documentation error, please include the URL for the page where you see the issue.

- **Where do I find documentation for the Compute tab for securing host, container, and serverless functions?**

  If you are using the Prisma Cloud Enterprise edition license, see Prisma Cloud Administrator’s Guide (Compute). If you are using Prisma Cloud Compute Edition license and are deploying and hosting it on your own, see Prisma Cloud Compute Edition Administrator’s Guide.

- **How do I get technical help or open a support case?**

  Check the discussions on the Palo Alto Networks Live Community, and to open a support case, log in to the Customer Support Portal.

### Policy and Alerts

- **What happens when I have two alert rules for the same conditions—one with and one without auto remediation?**

  The Alert Rule with auto remediation enabled takes precedence are automatically resolved.

- **With which threat intelligence feeds does Prisma Cloud integrate?**

  Prisma Cloud integrates with the threat intelligence feeds from AutoFocus, Facebook ThreatExchange, AWS GuardDuty, and AWS Inspector to help you turn your threat data in to actionable intelligence and to focus on what is relevant to your deployments.

  Each threat intelligence feed provides a classification for each of the IP addresses they include, and Prisma Cloud uses this data to identify bad actors. Some IP addresses that have been known to launch Command and Control traffic or DDOS attacks, are classified as outright malicious. Other IP addresses are listed as suspicious, and these have demonstrated patterns of association with other malicious sites or have indicators—file properties, behaviors, and activities— that are suspicious or highly suspicious. For details on AutoFocus, see AutoFocus artifacts.

- **How often does Prisma Cloud retrieve data from cloud services that provide automated security assessment or threat detection?**

  If you set up an integration with Qualys, Tenable, Amazon GuardDuty, or AWS Inspector for additional context on risks in the cloud, Prisma Cloud retrieves data from these services periodically. The data from Qualys and Tenable is retrieved every hour; the data from AWS Inspector and Amazon GuardDuty is retrieved at every ingestion cycle.

- **After I update a config policy query, how long does it take to automatically resolve alerts that no longer match this policy?**

  When a Config-based policy query is changed, all the alerts generated by this policy are re-evaluated at the next scan. Alerts that are no longer valid because of the policy change are automatically resolved.

- **What is the list of web applications that Prisma Cloud automatically classifies?**

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Application Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ICMP</td>
</tr>
<tr>
<td>21</td>
<td>FTP</td>
</tr>
<tr>
<td>22</td>
<td>SSH</td>
</tr>
<tr>
<td>23</td>
<td>TELNET</td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
</tr>
<tr>
<td>Port Number</td>
<td>Application Classification</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>53</td>
<td>DNS</td>
</tr>
<tr>
<td>80</td>
<td>Web (80)</td>
</tr>
<tr>
<td>88</td>
<td>Kerberos</td>
</tr>
<tr>
<td>111</td>
<td>RPC (111)</td>
</tr>
<tr>
<td>135</td>
<td>RPC (135)</td>
</tr>
<tr>
<td>143</td>
<td>IMAP</td>
</tr>
<tr>
<td>389</td>
<td>LDAP</td>
</tr>
<tr>
<td>443</td>
<td>Web (443)</td>
</tr>
<tr>
<td>444</td>
<td>SNPP</td>
</tr>
<tr>
<td>445</td>
<td>Generic (445)</td>
</tr>
<tr>
<td>514</td>
<td>Syslog</td>
</tr>
<tr>
<td>587</td>
<td>SMTP</td>
</tr>
<tr>
<td>636</td>
<td>LDAP (TLS)</td>
</tr>
<tr>
<td>995</td>
<td>IMAP</td>
</tr>
<tr>
<td>1433</td>
<td>SQL Server</td>
</tr>
<tr>
<td>1515</td>
<td>OSSEC</td>
</tr>
<tr>
<td>1521</td>
<td>Oracle</td>
</tr>
<tr>
<td>2376</td>
<td>Docker TLS</td>
</tr>
<tr>
<td>3128</td>
<td>Web Proxy</td>
</tr>
<tr>
<td>3268</td>
<td>Active Directory (GC)</td>
</tr>
<tr>
<td>3306</td>
<td>My SQL</td>
</tr>
<tr>
<td>3389</td>
<td>RDP</td>
</tr>
<tr>
<td>5050</td>
<td>Mesos Server</td>
</tr>
<tr>
<td>5432</td>
<td>Postgres</td>
</tr>
<tr>
<td>5439</td>
<td>Redshift</td>
</tr>
<tr>
<td>5671</td>
<td>RabbitMQ</td>
</tr>
<tr>
<td>Port Number</td>
<td>Application Classification</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>5672</td>
<td>RabbitMQ</td>
</tr>
<tr>
<td>5900</td>
<td>VNC</td>
</tr>
<tr>
<td>6168</td>
<td>Generic (6168)</td>
</tr>
<tr>
<td>6379</td>
<td>Redis</td>
</tr>
<tr>
<td>7200</td>
<td>Generic (7200)</td>
</tr>
<tr>
<td>7205</td>
<td>Generic (7205)</td>
</tr>
<tr>
<td>7210</td>
<td>MaxDB</td>
</tr>
<tr>
<td>8000</td>
<td>HTTP (8000)</td>
</tr>
<tr>
<td>8080</td>
<td>HTTP (8080)</td>
</tr>
<tr>
<td>8140</td>
<td>Puppet</td>
</tr>
<tr>
<td>8332</td>
<td>Bitcoin</td>
</tr>
<tr>
<td>8333</td>
<td>Bitcoin</td>
</tr>
<tr>
<td>8443</td>
<td>HTTP (8443)</td>
</tr>
<tr>
<td>8545</td>
<td>Ethereum (8545)</td>
</tr>
<tr>
<td>8888</td>
<td>HTTP (8888)</td>
</tr>
<tr>
<td>9000</td>
<td>Generic (9000)</td>
</tr>
<tr>
<td>9006</td>
<td>Web (9006)</td>
</tr>
<tr>
<td>9092</td>
<td>Kafka</td>
</tr>
<tr>
<td>9300</td>
<td>Elastic Search</td>
</tr>
<tr>
<td>9997</td>
<td>Splunk Logger</td>
</tr>
<tr>
<td>15671</td>
<td>RabbitMQ WebUI</td>
</tr>
<tr>
<td>15672</td>
<td>RabbitMQ WebUI</td>
</tr>
<tr>
<td>27017</td>
<td>MongoDB</td>
</tr>
<tr>
<td>29418</td>
<td>Git</td>
</tr>
<tr>
<td>30000</td>
<td>Generic (30000)</td>
</tr>
<tr>
<td>30303</td>
<td>Ethereum (30303)</td>
</tr>
<tr>
<td>Port Number</td>
<td>Application Classification</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>52049</td>
<td>NFS</td>
</tr>
<tr>
<td>55514</td>
<td>Syslog</td>
</tr>
<tr>
<td>60000</td>
<td>Generic</td>
</tr>
<tr>
<td>61420</td>
<td>Minuteman LB</td>
</tr>
<tr>
<td>61421</td>
<td>Minuteman LB</td>
</tr>
<tr>
<td>61668</td>
<td>Generic (61668)</td>
</tr>
</tbody>
</table>
To begin monitoring the resources on your cloud infrastructure, you must first connect your public cloud accounts to Prisma™ Cloud. When you add your cloud account to Prisma Cloud, the API integration between your cloud infrastructure provider and Prisma Cloud is established and you can begin monitoring the resources and identify potential security risks in your infrastructure.

- Cloud Account Onboarding
- Onboard Your AWS Account
- Onboard Your Azure Account
- Onboard Your Google Cloud Platform (GCP) Account
- Onboard Your Alibaba Cloud Account
- Cloud Service Provider Regions on Prisma Cloud
Cloud Account Onboarding

To get the most out of your investment in Prisma™ Cloud, you first need to add your cloud accounts to Prisma Cloud. This process requires that you have the correct permissions to authenticate and authorize the connection and retrieval of data.

Prisma Cloud administrators with the System Administrator and Cloud Provisioning Administrator roles can use the cloud account onboarding guided tour for a good first-run experience with all supported cloud platforms—Alibaba Cloud, AWS, Azure, and Google Cloud. The workflow provides the context you need to make decisions based on your own security and compliance requirements and it uses automation scripts—Cloud Formation templates for AWS or Terraform templates for Azure and GCP—to create the custom roles and enable the permissions required to add a cloud account.

When you log in to Prisma Cloud for the first-time, the guided tour displays after the welcome tour and prompts you to pick a cloud platform to add to Prisma Cloud.

You will make a few choices and provide basic account details to retrieve configuration logs and get started with Prisma Cloud for monitoring and visibility. If you want to ingest data from event logs and flow logs, you need to perform additional tasks.

- Onboard Your AWS Account
- Onboard Your Azure Account
- Onboard Your Google Cloud Platform (GCP) Account
- Onboard Your Alibaba Cloud Account
Onboard Your AWS Account

To connect your AWS Organizations (only supported on public AWS) or AWS accounts on the public AWS, AWS China, AWS GovCloud account to Prisma™ Cloud, you must complete some tasks on the AWS management console and some on Prisma Cloud. The onboarding workflow enables you to create a Prisma Cloud role with either read-only access to your traffic flow logs or with limited read-write access to remediate incidents. With the correct permissions, Prisma Cloud can successfully connect to and access your AWS account(s).

In addition to scanning your AWS resources against Prisma Cloud policies for compliance and governance issues, you can also scan objects in AWS S3 buckets for data security issues. The data security capabilities include predefined data policies and associated data classification profiles such as PII, Financial, or Healthcare & Intellectual Property that scan your objects stored in the S3 bucket to identify exposure—how sensitive information is kept private, or exposed or shared externally, or allows unauthorized access. Prisma Cloud Data Security capability is in Limited GA and available to select Prisma Cloud Enterprise Edition customers only.

- Add an AWS Cloud Account on Prisma Cloud.
- Add an AWS Organization to Prisma Cloud
- Update an Onboarded AWS Account
- Set Up the Prisma Cloud Role for AWS—Manual
- AWS APIs Ingested by Prisma Cloud

Add an AWS Cloud Account on Prisma Cloud

Use the following workflow to add your AWS public, AWS China, or AWS GovCloud accounts to Prisma™ Cloud. To add AWS Organizations on Prisma Cloud, see Add an AWS Organization to Prisma Cloud.

If you want to download and review the CloudFormation templates, get the S3 URLs from here:

<table>
<thead>
<tr>
<th>Role</th>
<th>S3 Template URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS Public Cloud</strong>—AWS account and AWS Organization; master account</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template</a></td>
</tr>
<tr>
<td>For member accounts within AWS Organizations—Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template</a></td>
</tr>
<tr>
<td>For member accounts within AWS Organizations—Read-Write (Limited)</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template</a></td>
</tr>
</tbody>
</table>
### STEP 1 | Before you begin.

If you would like Prisma Cloud to ingest VPC flow logs and any other integrations, such as Amazon GuardDuty or AWS Inspector, you must enable these services on the AWS management console. The CFT enables the ingestion of configuration data and AWS CloudTrail logs (audit events) only. VPC flow logs and any other integrations, such as Amazon GuardDuty or AWS Inspector are retrieved only if you previously enabled these services for the AWS account that you are onboarding.

1. **Decide whether you want to manually create the roles to authorize permissions for Prisma Cloud.**
   
   The onboarding flow automates the process of creating the Prisma Cloud role and adding the permissions required to monitor and/or protect your AWS account. If you want to create these roles manually instead, see [Set Up the Prisma Cloud Role for AWS—Manual](#).

2. **Create a CloudWatch log group.**
   
   The CloudWatch log group defines where the log streams are recorded.
   
   1. Select Services > CloudWatch > Logs > Actions > Create log group.
   2. Enter a name for the log group and click Create log group.
   3. **Enable flow logs.**
      
      1. Select Services > VPC > Your VPCs.
      2. Select the VPC to enable flow logs for and select Actions > Create flow log.
      3. Set the Filter to **Accept** or **All**.

      Setting the filter to **All** enables Prisma Cloud to retrieve accepted and rejected traffic from the flow logs. Setting the filter to **Accept** retrieves Accepted traffic only. If you set the filter to **Reject**, Prisma Cloud will not retrieve any flow log data.

   4. **Verify that the Destination is configured to Send to CloudWatch Logs.**
      
      If you set the destination as Amazon S3 bucket, Prisma Cloud will be unable to retrieve the data.

   5. **Select the Destination log group** you created above.

   6. **Create new or use existing IAM role** to publish flow logs to the CloudWatch log group.

<table>
<thead>
<tr>
<th>Role</th>
<th>S3 Template URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS GovCloud</strong></td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template">https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template</a></td>
</tr>
<tr>
<td><strong>AWS China</strong></td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template">https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template</a></td>
</tr>
</tbody>
</table>
If you are an existing IAM role to publish logs to the CloudWatch log group, you must edit the IAM role to include the following permissions.

```json
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

You will also need to Enable trust relationships so that the IAM role can access the CloudWatch Log group.

**STEP 2 |** Access Prisma Cloud and select Settings > Cloud Accounts > Add New.

**STEP 3 |** Select AWS as the Cloud to Protect.

**STEP 4 |** Enter a Cloud Account Name.

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies your AWS account on Prisma™ Cloud.
STEP 5 | Select the Mode.

Decide whether to enable permissions to only monitor (read-only access) or to monitor and protect (read-write access) the resources in your cloud account. Your selection determines which AWS Cloud Formation Template (CFT) is used to automate the process of creating the custom role required for Prisma Cloud.

STEP 6 | Set up the Prisma Cloud role on AWS.

To automate the process of creating the Prisma Cloud role that is trusted and has the permissions required to retrieve data on your AWS deployment, Prisma Cloud uses a CFT. The CFT enables the ingestion of configuration data and AWS CloudTrail logs (audit events) only, and it does not support the ability to enable VPC flow logs for your AWS account or any other integrations, such as Amazon GuardDuty or AWS Inspector.

1. Open a new tab on your browser and sign in to your AWS account for your AWS public cloud or AWS GovCloud deployment that you want to protect using Prisma Cloud.
   
   To onboard an AWS GovCloud account, your Prisma Cloud instance must be on https://app.gov.prismacloud.io
   
2. Click back to the Prisma Cloud console, and in the onboarding flow, select Create Stack.
   
   You will be directed to the AWS CloudFormation stack for your AWS public or AWS GovCloud environment, and the following details are automatically filled in for you:
   
   - **Stack Name**—The default name for the stack is PrismaCloudApp.
   - **External ID**—The Prisma Cloud ID, a randomly generated UUID that is used to enable the trust relationship in the role’s trust policy.
   - **Prisma Cloud Role Name**—The name of the role that will be used by Prisma Cloud to authenticate and access the resources in your AWS account.
3. Accept the IAM acknowledgment for resource creation and select **Create Stack**. The stack creation is initiated. Wait for the CREATE_COMPLETE status.

4. Select **Outputs** and copy the value of the Prisma CloudARN. The Prisma Cloud ARN has the External ID and permissions required for enabling authentication between Prisma Cloud and your AWS account.
5. Paste the Role ARN and click Next.

STEP 7 | Select one or more account groups and click Next.

You must assign each cloud account to an account group and Create an Alert Rule to associate with that account group to generate alerts when a policy violation occurs.
STEP 8 | Review the onboarding Status of your AWS account on Prisma Cloud.

The status check verifies that VPC flow logs are enabled on at least 1 VPC in your account, and audit events are available in at least one region on AWS CloudTrail.

If you have services that are not enabled on your AWS account, the status screen provides you some details.
Add an AWS Organization to Prisma Cloud

If you have consolidated access to AWS services and resources across your company within AWS Organizations, you can onboard the AWS master account on Prisma Cloud. When enable the AWS Organizations on the AWS management console and add the root or master account that has the role of a payer account that is responsible for paying all charges accrued by the accounts in its organization, all member accounts within the hierarchy are added in one streamlined operation on Prisma Cloud.

Figure 1: Image from AWS documentation

In this workflow, you first deploy a CloudFormation template in the master account to create the Prisma Cloud role to monitor, or monitor and protect your resources deployed on the master account. And then, you use CloudFormation StackSets to automate the creation of the Prisma Cloud role, which authorizes Prisma Cloud to access each member account. When you then add a new member account to your AWS organization, it is onboarded automatically on Prisma Cloud within a few (up to six) hours.

- If you want to exclude one or more Organizational Units (OUs) and all the member accounts it includes, you can manually disable individual member accounts on Prisma Cloud after they are onboarded. Alternatively, to onboard a subset of accounts, you can exclude the OUs when deploying the StackSet so that the PrismaCloud role is only created in the OUs for which you want to onboard accounts.
• If you had previously onboarded your AWS master account as a standalone or individual account, you must re-add the account as an Organization. All your existing data on assets monitored, alerts generated, or account groups created are left unchanged.

After you onboard your account as an AWS Organization, you cannot roll back. To add the account as a standalone or individual account, you must delete the Organization on Prisma Cloud and use the instructions to Add an AWS Cloud Account on Prisma Cloud.

• If you had previously onboarded an AWS account that is a member of the AWS Organization that you now add on Prisma Cloud, all your existing data on assets monitored, alerts generated, or account groups created are left unchanged. On Prisma Cloud, the member account will be logically grouped under the AWS Organization.

When you delete the AWS Organization on Prisma Cloud, you can recover all the existing data related to these accounts if you re-onboarded within 24 hours. After 24 hours, the data is deleted from Prisma Cloud.

- Add a New AWS Organization Account on Prisma Cloud
- Update an Onboarded AWS Organization

Add a New AWS Organization Account on Prisma Cloud

Add your AWS Organization on Prisma Cloud.

STEP 1 | Access Prisma Cloud and select **Settings > Cloud Accounts > Add New**.

STEP 2 | Select **AWS** as the **Cloud to Protect**.

STEP 3 | Enter a **Cloud Account Name** and onboard **Organization**.

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies your AWS Organization on Prisma™ Cloud.
STEP 4 | Select the Mode.

Decide whether to enable permissions to only monitor (read-only access) or to monitor and protect (read-write access) the resources in your cloud account. Your selection determines which AWS Cloud Formation Template (CFT) is used to automate the process of creating the custom role required for Prisma Cloud.

STEP 5 | Set up the Prisma Cloud role on the AWS master account.

To automate the process of creating the Prisma Cloud role that is trusted and has the permissions required to retrieve data on your AWS deployment, Prisma Cloud uses a CFT. The CFT enables the ingestion of configuration data and AWS CloudTrail logs (audit events) only, and it does not support the ability to enable VPC flow logs for your AWS account or any other integrations, such as Amazon GuardDuty or AWS Inspector.

1. Open a new tab on your browser and sign in to the AWS master account that you want to add on Prisma Cloud.
2. Click back to the Prisma Cloud console, and in the onboarding flow, select Create Stack.

   You will be directed to the AWS CloudFormation stack for your AWS environment, and the following details are automatically filled in for you:
   - **Stack Name**—The default name for the stack is PrismaCloudApp.
   - **External ID**—The Prisma Cloud ID, a randomly generated UUID that is used to enable the trust relationship in the role’s trust policy.
   - **Prisma Cloud Role Name**—The name of the role that will be used by Prisma Cloud to authenticate and access the resources in your AWS account.

3. Accept the IAM acknowledgment for resource creation and select Create Stack.

   The stack creation is initiated. Wait for the CREATE_COMPLETE status.

4. Select Outputs and copy the value of the Prisma CloudARN.
The Prisma Cloud ARN has the External ID and permissions required for enabling authentication between Prisma Cloud and your AWS account.

5. Paste the Master Role ARN and click Next.

STEP 6 | Create a StackSet to create the Prisma Cloud role within each member account.

AWS StackSets enables you to automate the process of creating the Prisma Cloud role across multiple accounts in a single operation.

1. Download the template file.
   Get the template file:
   - For member accounts with read-only access permissions (Monitor mode)—https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template
   - For member accounts with the read-write access permissions (Monitor & Protect mode)—https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template

2. On the AWS management console, select Services > CloudFormation > StackSets > Create StackSet.
   Verify that you are logged in to the AWS master account.
3. Upload the template file and click Next, then enter a StackSet Name.
4. In Parameters, enter the values for PrismaCloudRoleName and ExternalId.
   The PrismaCloudRoleName must include org within the string.
5. Click Next and select Service managed permissions.
6. Click Next and select Deploy to organization under Deployment targets.
If you do not want to onboard all member accounts, you can select **Deploy to organization unit OUs** and deploy the Stackset only to selected OUs only.

7. Set Automatic deployment **Enabled**, and Account removal behavior **Delete stacks**.
8. In Specify regions, select a region.
9. In Deployment Options, Maximum concurrent accounts, select **Percentage** and set it to **100**.
10. In Deployment Options, Failure tolerance, select **Percentage** and set it to **100**.
11. Click Next, and review the configuration.

12. Select **I acknowledge that AWS CloudFormation might create IAM resources with custom names** and **Submit**.

The StackSet creation is initiated. Wait for the SUCCEEDED status. When the process completes, each member account where the role was created is listed under **Stack instances** on the AWS management console.
13. Select **Parameters** and copy the values for PrismaCloudRoleName and ExternalId.

**STEP 7 | Configure the member account role details on Prisma Cloud.**

Use the details you copied from the previous step to set up the trust relationship and retrieve data from the member accounts.

1. Paste the **Member Role Name** and **Member External ID**.
2. Select **I confirm the stackset has created Prisma roles in member accounts successfully** and click **Next**.

If you have a large number of member accounts, it may take a while to create the role in each account and list it for verification. If you want to verify that the role was created in all accounts, do not select the checkbox. You can edit the cloud account settings later and onboard the member accounts. If you do not select the checkbox, only the master account will be onboarded to Prisma Cloud.

**STEP 8 | Select an account group and click **Next**.**

During initial onboarding, you must assign all the member cloud accounts with the AWS Organization hierarchy to an account group. Then, **Create an Alert Rule** to associate with that account group so that alerts are generated when a policy violation occurs.

*If you would like to selectively assign AWS member accounts to different account groups on Prisma Cloud, you can edit the cloud account settings later.*

**STEP 9 | Review the onboarding **Status** of your AWS Organization on Prisma Cloud.**
The status check verifies that VPC flow logs are enabled on at least 1 VPC in your master account, and audit events are available in at least one region on AWS CloudTrail. It also displays the number of member accounts that are provisioned with the Prisma Cloud role.

If you did not select the I confirm the stackset has created Prisma roles in member accounts successfully checkbox, the status screen displays the onboarding status of the master account but does not list the number of member accounts.

Update an Onboarded AWS Organization

In addition to updating the CFT stack for enabling permissions for new services, you can use this workflow to update the account groups that are secured with Prisma Cloud, change the protection mode from Monitor to Monitor & Protect or the reverse way, and redeploy the Prisma Cloud role in member accounts. You can opt to onboard all member accounts under Organizations hierarchy, or selectively add the OUs whose member accounts you want to onboard on Prisma Cloud.

STEP 1 | Provision the Prisma Cloud role on the AWS master account.

1. Download the template file.
   Get the template file for your needs:
   - For master accounts with the read-only access for Monitor mode—https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template
   - For member accounts with the read-write access for Monitor & Protect mode—https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write.template
2. Log in to your master account on the AWS management console.
3. Select Services > CloudFormation > Stacks.
4. Select PrismaCloudApp Stack and click Update Stack.
5. Replace the existing template with the template you downloaded earlier.
6. Click **Next**, review the configuration.

7. Select **I acknowledge that AWS CloudFormation might create IAM resources with custom names and Submit.**

**STEP 2 | Configure the member accounts.**

1. Log in to your Master Account on the AWS management console.
2. Select **Services > CloudFormation > StackSets.**
3. Select the Prisma stack set and **Edit StackSet Details.**
4. Replace the current template with the downloaded template.
5. Click **Next** and enter values for **PrismaCloudRoleName** and **ExternalId.**
6. Click **Next** and verify **Service managed permissions** is selected.
7. Select **Deploy To Organizational units (OUs)**, and Under Organizational units (OUs), select all the OUs that are displayed, or enter the AWS OU ID.
   
   To enter your Organization Root ID use the format r-[0-9a-z]{4,32}. For example, r-6usb.

8. In **Specify regions**, select a region from the drop-down.

9. In **Deployment Options**, Maximum concurrent accounts, select **Percentage** and set it to **100**.

10. In **Deployment Options**, Failure tolerance, select **Percentage** and set it to **100**.

11. Click **Next**, and review the configuration.

12. Select **I acknowledge that AWS CloudFormation might create IAM resources with custom names and Submit.**

   The StackSet creation is initiated. Wait for the SUCCEEDED status. When the process completes, each member account where the role was created is listed under **Stack instances.**

13. Select **Parameters** and copy the values for PrismaCloudRoleName and ExternalId.

**STEP 3 | Access Prisma Cloud** and select the AWS Organization account you want to modify.

1. Select **Settings > Cloud Accounts** and select the account.

2. **(Optional)** Select a different **account group** and click **Next.**
During initial onboarding, you must assign all the member cloud accounts with the organization hierarchy to one account group.

You can now edit to selectively assign AWS member accounts to different account groups on Prisma Cloud.

**STEP 4 | Review the onboarding Status of your AWS organization on Prisma Cloud.**

The status check verifies that VPC flow logs are enabled on at least 1 VPC in your master account, and audit events are available in at least one region on AWS CloudTrail. It also displays the number of member accounts that are provisioned with the Prisma Cloud role.

---

**Update an Onboarded AWS Account**

After you add your cloud account to Prisma Cloud, you may need to update the PrismaCloud stack to provide additional permissions for new policies that are frequently added to help you monitor your cloud account and ensure that have a good security posture. When you update the CFT stack, Prisma Cloud can ingest data on new services that are supported. These CFTs are available directly from the Prisma Cloud administrative console and are also accessible from the S3 bucket. For instruction on updating your AWS Organization, see Add an AWS Organization to Prisma Cloud.
<table>
<thead>
<tr>
<th>Role</th>
<th>S3 Template URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS Public Cloud</strong>—AWS account and AWS Organization, master account</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template</a></td>
</tr>
<tr>
<td>For member accounts within AWS Organizations Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template</a></td>
</tr>
<tr>
<td>For member accounts within AWS Organizations Read-Write (Limited)</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template">https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template</a></td>
</tr>
<tr>
<td><strong>AWS GovCloud</strong></td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template">https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template</a></td>
</tr>
<tr>
<td><strong>AWS China</strong></td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><a href="https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template">https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template</a></td>
</tr>
</tbody>
</table>

In addition to updating the CFT stack for enabling permissions for new services, you can use this workflow to update the account groups that are secured with Prisma Cloud or to change the protection mode from Monitor to Monitor & Protect or the reverse way.

**STEP 1** | Log in to the Prisma Cloud administrative console.

**STEP 2** | Select the AWS cloud account you want to modify.

Select **Settings > Cloud Accounts** and click on the name of the cloud account to manage from the list of cloud accounts.
STEP 3 | (Optional) Change the account groups you want to monitor.

STEP 4 | (To change permissions for the Prisma Cloud role) Update the Prisma Cloud App using the CloudFormation template (CFT).

1. Click the link to download the latest template and follow the instructions to update the stack.
2. Update the stack either using the AWS console or using the AWS CLI.

   - Log in to AWS console.
   - Select Services > CloudFormation > Stacks.
   - Select the PrismaCloudApp stack to update and select Update.

Select Replace current template and Upload a template file you downloaded earlier; you can optionally provide the Amazon S3 URL listed in the table above.

If you decide to create a new stack instead of updating the existing stack, you must copy the ExternalID and PrismaCloudRoleARN values from the CFT outputs.

   - Configure stack options.
   - Click Next and verify the settings.
   - Preview your changes to the CloudFormation template for the role you updated.
   - Update your CFT.

If you created a new stack, you must log in to the Prisma Cloud administrative console and select your cloud account on Settings > Cloud Accounts to enter the ExternalID and PrismaCloudRoleARN values from the CFT outputs.
- Check the Status to verify that Prisma Cloud can successfully retrieve information on your cloud resources.

- Use AWS Command Line Interface to deploy the updated Prisma Cloud App stack.
- Using the AWS CLI tool, enter the following command to retrieve the latest CloudFormation template.

<table>
<thead>
<tr>
<th>Role</th>
<th>CLI Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Public cloud</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><code>wget https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template --quiet -O /tmp/rl-read-only.template</code></td>
</tr>
<tr>
<td>AWS GovCloud</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td><code>wget https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template --quiet -O /tmp/redlock-govcloud-read-only.template</code></td>
</tr>
<tr>
<td>Role</td>
<td>CLI Command</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AWS China</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td>wget <a href="https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template">https://s3.amazonaws.com/redlock-public/cft/rl-cn-read-only.template</a> --quiet -O /tmp/rl-cn-read-only.template</td>
</tr>
</tbody>
</table>

- Enter the following command to deploy the updated CloudFormation template.

Replace with the correct name for the CloudFormation template, current stack name, role ARN, and External ID to overwrite the current stack or enter new values to create a new stack.

- **Read-Only**—aws cloudformation deploy --template-file /tmp/<RedLock-cloudformation-template-name> --stack-name <Stack Name> --parameter-overrides RedlockRoleARN=<Role ARN> ExternalID=<xxxxxxxxxx> --capabilities CAPABILITY_NAMED_IAM
- **Read-Write (Limited)**—aws cloudformation deploy --template-file /tmp/<RedLock-cloudformation-template-name> --stack-name <Stack Name> --parameter-overrides RedlockRoleARN=<Role ARN> ExternalID=<xxxxxxxxxx> --capabilities CAPABILITY_NAMED_IAM

Set Up the Prisma Cloud Role for AWS—Manual

If you do not want to use the guided onboarding flow that automates the process of creating the roles required for Prisma™ Cloud to monitor or monitor and protect your accounts on AWS, you must create the roles manually. In order to monitor your AWS account, you must create a role that grants Prisma Cloud access to your flow logs and read-only access (to retrieve and view the traffic log data) or a limited read-write access (to retrieve traffic log data and remediate incidents). To authorize permission, you must copy the policies from the relevant template and attach it to the role. Event logs associated with the monitored cloud account are automatically retrieved on Prisma Cloud.

**STEP 1** | Log in to the AWS Management Console to create a role for Prisma Cloud.

Refer to the AWS documentation for instructions. Create the role in the same region as your AWS account, and use the following values and options when creating the role:

- **Type of trusted entity**: Another AWS Account and enter the Account ID*: **188619942792**
- Select **Require external ID**, which is a unique alphanumeric string. You can generate a secure UUIDv4 at https://www.uuidgenerator.net/version4.
- Do not enable MFA. Verify that **Require MFA** is not selected.
• Click Next and add the AWS Managed Policy for Security Audit.

Then, add a role name and create the role. In this workflow, later, you will create the granular policies and edit the role to attach the additional policies.

STEP 2 | Get the granular permissions from the AWS CloudFormation template for your AWS environment.

The Prisma Cloud S3 bucket has read-only templates and read-and-write templates for the public AWS and AWS GovCloud environments.

1. Download the template you need.
   • AWS public cloud
     
     https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template
     

   • AWS GovCloud
     
     
     https://s3.amazonaws.com/redlock-public/cft/redlock-govcloud-read-only.template

2. Identify the permissions you need to copy.
To create the policy manually, you will need to add the required permissions inline using the JSON editor. From the read-only template you can get the granular permissions for the PrismaCloud-IAM-ReadOnly-Policy, and the read-write template lists the granular permissions for the PrismaCloud-IAM-ReadOnly-Policy and the PrismaCloud-IAM-Remediation-Policy.

1. Open the appropriate template using a text editor.
2. Find the policies you need and copy it to your clipboard.

   Copy the details for one or both permissions, and make sure to include the open and close brackets for valid syntax, as shown below.

   ![JSON template](image1)

   ![JSON template](image2)

**STEP 3 | Create the policy that defines the permissions for the Prisma Cloud role.**

Both the read-only role and the read-write roles require the AWS Managed Policy SecurityAudit Policy. In addition, you will need to enable granular permissions for the PrismaCloud-IAM-ReadOnly-Policy for the read-only role, or for the read-write role add the PrismaCloud-IAM-ReadOnly-Policy and the limited permissions for PrismaCloud-IAM-Remediation-Policy.

1. Select IAM on the AWS Management Console.
2. In the navigation pane on the left, choose Access Management > Policies > Create policy.
3. Select the JSON tab.

   Paste the JSON policies that you copied from the template within the square brackets for Statement.
If you are enabling read and read-write permissions, make sure to append the read-write permissions within the same Action statement.

4. Review and create the policy.

**STEP 4** Edit the role you created in **Step 1** and attach the policy to the role.

**STEP 5** Required only if you want to use the same role to access your CloudWatch log group Update the trust policy to allow access to the CloudWatch log group.
Edit the Trust Relationships to add the permissions listed below. This allow you to ensure that your role has a trust relationship for the flow logs service to assume the role and publish logs to the CloudWatch log group.

```json
{
   "Effect": "Allow",
   "Principal": {
      "Service": "vpc-flow-logs.amazonaws.com"
   },
   "Action": "sts:AssumeRole"
}
```

**STEP 6 | Copy the Role ARN.**

**STEP 7 | Resume with the account onboarding flow at Paste the Role ARN in Add an AWS Cloud Account on Prisma Cloud**

**Prisma Cloud on AWS China**

To use Prisma Cloud to monitor or monitor and protect your deployments in the AWS China regions of Ningxia and Beijing, you require a Prisma Cloud instance in China. See Add an AWS Cloud Account on Prisma Cloud for getting started with the configuration logs for your resources.

Review the following sections to know what is not currently supported and available for the AWS China deployments:

- Prisma Cloud Compute, which enables you to secure containers and serverless functions is not available on the Prisma Cloud instance in China.
- Prisma Cloud does not support the following services:
  - Amazon GuardDuty
  - AWS Inspector
  - AWS Organization
- AWS does not support the following services:
  - Amazon AppStream
  - Amazon CloudSearch
  - Amazon Data Pipeline
• AWS Glue
• Amazon Route53
• Amazon Simple Email Service
• AWS WAF
• AWS WAFv2
• Resource and alert attribution

AWS APIs Ingested by Prisma Cloud

The following are AWS APIs that are ingested by Prisma Cloud.

<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
</table>
| API Gateway | • aws-apigateway-get-rest-apis  
• aws-apigateway-get-stages  
• aws-apigateway-domain-name  
• aws-apigateway-base-path-mapping  
• aws-apigateway-method  
• aws-apigateway-client-certificates |
| AWS AutoScaling | • aws-describe-auto-scaling-groups  
• aws-ec2-autoscaling-launch-configuration |
| AWS Certificate Manager | aws-acm-describe-certificate |
| Amazon Elastic Container Service (ECS) | • aws-ecs-describe-task-definition  
• aws-ecs-service |
| AWS CloudFormation | aws-cloudformation-describe-stacks |
| AWS CloudFront | aws-cloudfront-list-distributions |
| Amazon CloudSearch | aws-cloudsearch-domain |
| AWS CloudTrail | • aws-cloudtrail-describe-trails  
• aws-cloudtrail-get-event-selectors  
• aws-cloudtrail-get-trail-status |
| AWS CloudWatch | • aws-cloudwatch-describe-alarms  
• aws-cloudwatch-log-group  
• aws-logs-describe-metric-filters |
| Amazon Cognito | • aws-cognito-identity-pool  
• aws-cognito-user-pool |
| AWS Direct Connect Gateway | • aws-directconnect-describe-gateway |
| Amazon EC2 | • aws-describe-account-attributes  
• aws-ec2-describe-instances  
• aws-ec2-describe-images  
• aws-ec2-describe-snapshots |
<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>• aws-ec2-describe-network-interfaces</td>
<td></td>
</tr>
<tr>
<td>• aws-ec2-key-pair</td>
<td></td>
</tr>
<tr>
<td>• aws-ec2-describe-volumes</td>
<td></td>
</tr>
<tr>
<td>• aws-ec2-elastic-address</td>
<td></td>
</tr>
<tr>
<td>Amazon MQ</td>
<td>aws-mq-broker</td>
</tr>
<tr>
<td>Amazon SageMaker</td>
<td>aws-sagemaker-notebook-instance</td>
</tr>
<tr>
<td></td>
<td>aws-sagemaker-endpoint</td>
</tr>
<tr>
<td>AWS Config</td>
<td>aws-configservice-describe-configuration-recorders</td>
</tr>
<tr>
<td>Delivery Channels</td>
<td>aws-describe-delivery-channels</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>aws-dynamodb-describe-table</td>
</tr>
<tr>
<td>AWS Database Migration Service</td>
<td>• aws-dms-endpoint</td>
</tr>
<tr>
<td></td>
<td>• aws-dms-replication-instance</td>
</tr>
<tr>
<td>AWS Elastic Beanstalk</td>
<td>• aws-elasticbeanstalk-environment</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticbeanstalk-configuration-settings</td>
</tr>
<tr>
<td>Amazon Elastic Container Registry (ECR)</td>
<td>• aws-ecr-image</td>
</tr>
<tr>
<td></td>
<td>• aws-ecr-get-repository-policy</td>
</tr>
<tr>
<td>AWS Elastic File System (EFS)</td>
<td>aws-describe-mount-targets</td>
</tr>
<tr>
<td>Amazon Elastic Container Service for Kubernetes (EKS)</td>
<td>aws-eks-describe-cluster</td>
</tr>
<tr>
<td>ElastiCache</td>
<td>• aws-cache-engine-versions</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticache-cache-clusters</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticache-describe-replication-groups</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticache-reserved-cache-nodes</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticache-subnet-groups</td>
</tr>
<tr>
<td></td>
<td>• aws-elasticache-snapshots</td>
</tr>
<tr>
<td>Amazon Elastic Load Balancing</td>
<td>• aws-elb-describe-load-balancers</td>
</tr>
<tr>
<td></td>
<td>• aws-describe-ssl-policies</td>
</tr>
<tr>
<td></td>
<td>• aws-elbv2-describe-load-balancers</td>
</tr>
<tr>
<td></td>
<td>• aws-elbv2-target-group</td>
</tr>
<tr>
<td>Amazon ElasticSearch Service</td>
<td>aws-es-describe-elasticsearch-domain</td>
</tr>
<tr>
<td>Amazon Elastic MapReduce (EMR)</td>
<td>• aws-emr-describe-cluster</td>
</tr>
<tr>
<td></td>
<td>• aws-emr-public-access-block</td>
</tr>
<tr>
<td>Amazon S3 Glacier</td>
<td>• aws-glacier-get-vault-access-policy</td>
</tr>
<tr>
<td></td>
<td>• aws-glacier-get-vault-lock</td>
</tr>
<tr>
<td>SERVICE</td>
<td>API NAME IN PRISMA CLOUD</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Amazon GuardDuty</td>
<td>aws-guardduty-detector</td>
</tr>
<tr>
<td>AWS Glue</td>
<td>aws-glue-security-configuration</td>
</tr>
</tbody>
</table>
| AWS Identity and Access Management (IAM)     | - aws-iam-list-access-keys  
|                                              | - aws-iam-get-account-summary  
|                                              | - aws-iam-list-server-certificates  
|                                              | - aws-iam-get-credential-report  
|                                              | - aws-iam-list-mfa-devices  
|                                              | - aws-iam-list-virtual-mfa-devices  
|                                              | - aws-iam-get-account-password-policy  
|                                              | - aws-iam-get-policy-version  
|                                              | - aws-iam-list-users  
|                                              | - aws-iam-list-user-policies  
|                                              | - aws-iam-list-roles  
|                                              | - aws-iam-list-groups  
|                                              | - aws-iam-list-attached-user-policies  
|                                              | - aws-iam-list-ssh-public-keys  
|                                              | - aws-iam-saml-provider  
|                                              | - aws-iam-service-last-accessed-details  |
| AWS Key Management Service (KMS)             | aws-kms-get-key-rotation-status |
| Amazon Kinesis                               | aws-kinesis-list-streams  
|                                              | aws-kinesis-firehose-delivery-stream  |
| AWS Lambda                                   | - aws-lambda-list-functions  
|                                              | - aws-lambda-get-region-summary  |
| AWS Organization                             | - aws-organization-account  
|                                              | - aws-organization-ou  
|                                              | - aws-organization-root  
|                                              | - aws-organization-scp  
|                                              | - aws-organization-tag-policy  |
| AWS Resource Access Manager (RAM)            | - aws-ram-principal  
|                                              | - aws-rds-describe-db-instances  
|                                              | - aws-rds-describe-db-snapshots  
|                                              | - aws-rds-describe-event-subscriptions  
|                                              | - aws-rds-db-cluster-snapshots  
<p>|                                              | - aws-rds-db-clusters  |
| Amazon Relational Database Service (RDS)     | aws-redshift-describe-clusters |
| Amazon RedShift                              |                          |</p>
<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
</table>
| AWS Route53                         | • aws-route53-list-hosted-zones  
                                       • aws-route53-domain |
| Amazon RDS                          | aws-rds-describe-db-parameter-groups |
| AWS Secrets Manager                 | aws-secretsmanager-describe-secret |
| AWS Systems Manager                 | aws-ssm-parameter |
| Amazon S3                           | • aws-s3control-public-access-block  
                                       • aws-s3api-get-bucket-acl  
   The list of APIs associated with this API name are:  
   • listBuckets  
   • getS3AccountOwner  
   • getRegionName  
   • getBucketLocation  
   • getBucketAcl  
   • getBucketPolicy  
   • getBucketPolicyStatus  
   • getBucketVersioningConfiguration |
| Amazon Simple Notification Service (SNS) | • aws-sns-get-subscription-attributes  
                                           • aws-sns-get-topic-attributes  
                                           • aws-sns-platform-application |
| Amazon Simple Queue Service (SQS)    | aws-sqs-get-queue-attributes |
| Amazon VPC                          | • aws-ec2-describe-security-groups  
                                       • aws-ec2-describe-route-tables  
                                       • aws-ec2-describe-subnets  
                                       • aws-ec2-describe-vpcs  
                                       • aws-ec2-describe-vpc-peering-connections  
                                       • aws-describe-vpc-endpoints  
                                       • aws-ec2-vpn-connections-summary  
                                       • aws-ec2-describe-vpn-connections  
                                       • aws-ec2-describe-vpn-gateways  
                                       • aws-ec2-describe-vpn-gateways-summary  
                                       • aws-vpc-dhcp-options  
                                       • aws-vpc-nat-gateway  
                                       • aws-ec2-describe-flow-logs  
                                       • aws-ec2-describe-internet-gateways  
                                       • aws-ec2-describe-network-acls  
                                       • aws-ecr-get-repository-policy  
                                       • aws-vpc-transit-gateway |
<p>| AWS Web Application Firewall (WAF)   | • aws-waf-web-acl-resources |</p>
<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon WorkSpaces</td>
<td>• aws-waf-classic-web-acl-resource</td>
</tr>
<tr>
<td></td>
<td>• aws-describe-workspace-directories</td>
</tr>
<tr>
<td></td>
<td>• aws-workspaces-describe-workspaces</td>
</tr>
</tbody>
</table>
Onboard Your Azure Account

To begin monitoring and identifying compliance violations and vulnerabilities on your Azure commercial or Government environment, you must add your Azure subscriptions to Prisma™ Cloud. To successfully add your Azure subscriptions, you must enable authentication between Prisma Cloud and your Azure resources and configure the permissions required to read configuration data, flow logs, and audit logs, and to remediate issues that are identified on your Azure resources.

And, if you want to retrieve information on your users who access resources deployed within your Azure subscription, add your Azure Active Directory account to Prisma Cloud.

- Azure Cloud Account Onboarding Checklist.
- Add an Azure Subscription on Prisma Cloud.
- Add an Azure Active Directory Account on Prisma Cloud.
- Set Up Your Azure Cloud Account Manually or Use the Azure PowerShell Script.
- Troubleshoot Azure Account Onboarding.

Azure Cloud Account Onboarding Checklist

Prisma™ Cloud supports both Azure commercial and Azure Government. For Azure commercial, the onboarding flow enables you to provide your subscription details as inputs and generates a Terraform template, which you can download and run from the Azure Cloud Shell. The workflow automates the process of setting up the Prisma Cloud application on Azure Active Directory and enabling the permissions for read-only or read-write access to your Azure subscription. You will however, need to review and manually enable the permissions for Prisma Cloud to retrieve network traffic data from network security group (NSG) flow logs.

To successfully onboard and monitor the resources within your Azure Government subscription, use the following checklist to authorize the correct set of access rights to Prisma Cloud.

- Collect your Azure Subscription ID from the Azure portal.

  Get your Azure Subscription ID. Make sure that you have Account Owner or Contributor privileges so that you can add Prisma Cloud as an application on your Azure Active Directory. To onboard your Azure subscription on Prisma Cloud, set up an Active Directory application object (Application ID) and a Service Principal object (Object ID) that, together, enable API access. The process of setting up Prisma Cloud on Azure Active Directory will provide you with the keys and IDs that are required to establish an identity for sign-in and access to resources in your Azure subscription. The Service Principal ID defines the permissions and scope that is assumed by Prisma Cloud.

- Review the roles and associated permissions required:

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader</td>
<td>The Reader role at the subscription level is required for Prisma Cloud to monitor the configuration of existing Azure resources within your Azure subscription. Prisma Cloud requires this role to ingest configuration and activity logs.</td>
</tr>
<tr>
<td>Reader and Data Access</td>
<td>The Reader and Data Access role at the subscription level is required for Prisma Cloud to fetch flow logs and storage account attributes so that you can use Prisma Cloud policies that assess risks in your storage account. This role includes the permissions to access</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>the storage account keys and authenticate to the storage account to access the data.</td>
</tr>
<tr>
<td></td>
<td>• For Prisma Cloud to access flow logs stored in storage accounts that belong to subscriptions that are not monitored by Prisma Cloud, you must provide Reader and Data Access role on the storage accounts.</td>
</tr>
<tr>
<td></td>
<td>• The Reader and Data Access role is not a superset of the Reader role. Although this role has read-write access, Prisma Cloud only uses these permissions to access and read the flow log from the storage account.</td>
</tr>
<tr>
<td>Network Contributor or a custom role to query flow log status</td>
<td>The built-in <strong>Network Contributor</strong> role can manage network data necessary to access and read flow logs settings for all network security groups (NSGs) along with the details on the storage account to which the flow logs are written. It also enables auto-remediation of network-related incidents.</td>
</tr>
<tr>
<td></td>
<td>You can use the built-in role or create a custom role to allow Prisma Cloud to fetch flow log status. As a best practice, Create a Custom Role on Azure to Enable Prisma Cloud to Access Flow Logs and use the least privilege principal to enable access only to the required permissions. The network contributor built-in role provides a much broader set of permissions than required by Prisma Cloud.</td>
</tr>
<tr>
<td></td>
<td><strong>To create a custom role</strong>, you must have the <code>Microsoft.Authorization/roleDefinitions/write</code> permission on all AssignableScopes, such as Owner or User Access Administrator.</td>
</tr>
<tr>
<td></td>
<td>You can then use the Azure CLI to create a custom role with the <code>Microsoft.Network/networkWatchers/queryFlowLogStatus/action</code> permission to query the status of flow logs.</td>
</tr>
<tr>
<td>Storage Account Contributor</td>
<td><em>(Optional but required if you want to enable auto-remediation)</em> The <strong>Storage Account Contributor</strong> role</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Custom role with permissions</td>
<td>(Optional but required if you want to enable ingestion of the listed services) Create a custom role with the following permissions:</td>
</tr>
<tr>
<td></td>
<td>• Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action—To ingest data from Azure Container Registry webhooks that are triggered when a container image or Helm chart is pushed to or deleted from a registry.</td>
</tr>
<tr>
<td></td>
<td>• Microsoft.Web/sites/config/list/action—To ingest Authentication/Authorization data from Azure App Service that hosts websites and web applications. The Reader Role listed earlier is adequate to ingest configuration data from the Azure App Service.</td>
</tr>
</tbody>
</table>

Prisma Cloud provides a JSON file that makes it easier for you to create a custom role with the read-only permissions required to monitor your Azure resources.

- Enable Prisma Cloud to obtain network traffic data from network security group (NSG) flow logs: NSG flow logs are a feature of Network Watcher, which allows you to view information about ingress and egress IP traffic through an NSG.
  - Create one or more network security groups if you have none.
  - Create Azure Network Watcher instances for the virtual networks in every region where you collect NSG flow logs.
  
  Network Watcher enables you to monitor, diagnose, and view metricsenable and disable logs for resources in an Azure virtual network.
  
  - Create storage accounts. You must have a storage account in each region where you have NSGs because flow logs are written to the same region as the NSGs. As a best practice, configure a single storage account to collect flow logs from all NSGs in a region.
  
  Microsoft.Insights is the resource provider namespace for Azure Monitor, which provides features such as metrics, diagnostic logs, and activity logs.
  
  - Enable NSG flow logs version 1 or 2. Azure does not support NSG flow logs version 2 in all regions.
  - Verify that you can view the flow logs.

Continue to Add an Azure Subscription on Prisma Cloud.

Add an Azure Subscription on Prisma Cloud

Connecting Prisma™ Cloud to your Azure cloud account enables you to analyze and monitor traffic logs, and detect potential malicious network activity or compliance issues. To enable API access between Prisma Cloud and your Microsoft Azure Subscription, you need to gather account information about your subscription and Azure Active Directory so that Prisma Cloud can monitor the resources in your cloud account, and add one subscription at a time.

If you are adding an Azure commercial account, this workflow uses Terraform templates to streamline the set up process. The template automates the process of creating and registering Prisma Cloud as an
application on your Active Directory and creating the Service Principal and associating the roles required to enable authentication.

If you are adding an Azure Government or Azure China subscription, you must complete some tasks manually on the Azure portal.

- Add Azure Commercial Subscription to Prisma Cloud
- Add Azure Government Subscription to Prisma Cloud
- Add an Azure China Subscription on Prisma Cloud
- Create a Custom Role for Prisma Cloud (Not required if you use the Terraform template)

**Add Azure Commercial Subscription to Prisma Cloud**

**STEP 1 | Access Prisma Cloud** and select **Settings > Cloud Accounts > Add New**.

**STEP 2 | Select Azure as the Cloud to Protect.**

**STEP 3 | Enter a Cloud Account Name.**

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies your Azure subscription on Prisma Cloud.
**STEP 4** | Select the **Mode**.

Decide whether to enable permissions to only monitor (read-only access) or to monitor and protect (read-write access) the resources in your cloud account. Your selection determines which Terraform template is used to automate the process of creating the custom role required for Prisma Cloud.

**STEP 5** | Register Prisma Cloud as an application on your Azure Active Directory.

Prisma Cloud requires your Azure **Subscription ID** so that it can identify your Azure cloud account and retrieve the storage account and key vault information. Prisma Cloud also needs the **Directory ID**, **Application ID**, **Application Key**, and **Service Principal ID** to establish the connection between Prisma Cloud and Azure Active Directory so that it can access the resources in your subscription.

1. Fill out the details to set up Prisma Cloud on your Azure subscription and click **Next**.

   From the Azure portal, get your Azure Active Directory ID, that is referred to as Tenant ID on Prisma Cloud, and your Azure Subscription ID. You must also choose a Service Principal password. Use a strong password that meets the Azure password complexity guidelines. If you later decide to change this password on the Azure portal, you must manually update the password on Prisma Cloud.

   The Terraform template uses the value you enter as inputs to automate the process of setting up the custom role with the associated permissions for the **Monitor** or **Monitor & Protect** mode you selected earlier.

2. Download the Terraform template.

   Prisma Cloud recommends that you create a directory to store the Terraform template you download. This allows you to manage the templates when you add a different Azure subscription to Prisma Cloud. Give the directory a name that uniquely identifies the subscription for which you’re using it (for example, onboard-<subscription-name>).

3. Login to the Azure portal Cloud Shell (Bash).

4. Upload the template to the Cloud Shell.

   ![Cloud Onboarding Setup](image-url)
5. Run the following Terraform commands.
   1. `terraform init`
   2. `terraform apply`

6. Copy the details after applying the Terraform template.
   Get the Application Key, Application ID and the Service Principal Object ID from the output file.

   **STEP 6** Select **Ingest & Monitor Network Security Group flow logs** and click **Next**.

   Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG. You must first configure **Network Security Groups** on Azure and assign a storage account to enable Flow log ingestion on Prisma Cloud. Make sure that Azure Flow logs are stored within a storage account in the same region as the NSG. If you want to enable flow log ingestion, you must complete the tasks outlined in Steps 6 to Step 10 in **Set Up Your Azure Subscription for Prisma Cloud**. If you enable this option without setting it up on the Azure portal, Prisma Cloud will not be able to retrieve any Flow logs.

   **STEP 7** Select the **Account Groups** you want to add and click **Next**.
You must assign each cloud account to an account group, and Create an Alert Rule to associate the account group with it to generate alerts when a policy violation occurs.

**STEP 8 | Verify the Status and click Done.**

> If Prisma Cloud is able to successfully connect to your Azure subscription and retrieve information, the status is displayed with a green check mark. If Prisma Cloud is unable to retrieve the logs, the error message indicates what failed. See Troubleshoot Azure Account Onboarding for help.

**STEP 9 | Verify that you can view the information on your Azure resources on Prisma Cloud.**

Wait for approximately 10-24 hours after you onboard the Azure subscription to Prisma Cloud, to review the data that was collected about your Azure resources. After Prisma Cloud ingests data, the information is available for compliance checks, configuration review, audit history, and network visualization.

> It takes about four to six hours before you can view flow logs in Prisma Cloud. Prisma Cloud ingests flow logs from the previous seven days from when you onboard the account.

1. Log in to Prisma Cloud.
2. Select Investigate and enter the following RQL query.
This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

```sql
network where cloud.account = '{{cloud account name}}' AND source.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 0
```

**Add Azure Government Subscription to Prisma Cloud**

Connect your Azure Government subscription on Prisma Cloud to monitor resources for potential security and compliance issues.

**STEP 1 | Set Up Your Azure Subscription for Prisma Cloud.**

**STEP 2 | Add your Azure subscription on Prisma Cloud.**

1. Access Prisma Cloud and select **Settings > Cloud Accounts > Add New**.
2. Enter a **Cloud Account Name**.
3. Select **Cloud Type Azure** and the **Government** environment where your resources are deployed, click **Next**.
4. Enter your Azure **Subscription ID**, **Active Directory ID**, **Application ID**, **Application Key** and **Service Principal ID**.
5. Select **Ingest & Monitor Network Security Group flow logs** and click **Next**.

Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG. Make sure that Azure Flow logs are stored within a storage account in the same region as the NSG. See **Azure Cloud Account Onboarding Checklist** for the set up details to ensure that Prisma Cloud can successfully ingest NSG flow logs.

6. Select the **Account Groups** you want to add and click **Next**.
You must assign each cloud account to an account group, and create an alert rule to associate the account group with it to generate alerts when a policy violation occurs.

7. Verify the status and done to save your changes.

If Prisma Cloud was able to successfully make an API request to retrieve the Azure flow logs, the status is displayed with a green check mark. If Prisma Cloud is unable to retrieve the logs, the error message indicates what failed. See Troubleshoot Azure Account Onboarding for help.

STEP 3 | Verify that you can view the information on your Azure resources on Prisma Cloud.

Wait for approximately 10-24 hours after you onboard the Azure subscription to Prisma Cloud, to review the data that was collected about your Azure resources. After Prisma Cloud ingests data, the information is available for compliance checks, configuration review, audit history, and network visualization.

It takes about four to six hours before you can view flow logs in Prisma Cloud. Prisma Cloud ingests flow logs from the previous seven days from when you onboard the account.

1. Log in to Prisma Cloud.
2. Select Investigate and enter the following RQL query.

   This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

   ```
   network where cloud.account = '{${cloud account name}}' AND source.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 0
   ```
Add an Azure China Subscription on Prisma Cloud

With the Prisma Cloud Business Edition license on app.prismacloud.cn, you can monitor your Microsoft Azure China subscriptions. To get started, gather the details listed in Set Up Your Azure Subscription for Prisma Cloud from the Azure China portal and connect your subscription to Prisma Cloud. When you add the subscription, Prisma Cloud monitors the configuration metadata for your IaaS and PaaS services, and identifies potential resource misconfiguration and improper exposure.

Network flow logs is in beta, and event logs are not monitored currently.

STEP 1 | Add your Azure subscription on Prisma Cloud.
1. Log in to Prisma Cloud.
2. Select Settings > Cloud Accounts > Add New
3. Select Cloud Type Azure and click Next.
4. Enter a Cloud Account Name.

5. Enter your Azure Subscription ID, Active Directory ID, Application ID, Application Key and Service Principal ID.

These are the details you collected from the Azure portal.

   Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG. Make sure that Azure Flow logs are stored within a storage account in the same region as the NSG. See Azure Cloud Account Onboarding Checklist for the set up details to ensure that Prisma Cloud can successfully ingest NSG flow logs.

7. Select the Account Groups you want to add and click Next.

   You must assign each cloud account to an account group, and Create an Alert Rule to associate the account group with it to generate alerts when a policy violation occurs.

8. Verify the Status and Save your changes.

   If Prisma Cloud was able to successfully make an API request to retrieve the configuration metadata, the status is displayed with a green check mark. If Prisma Cloud is unable to retrieve the logs, the error message indicates what failed.
Review the details for the account you added on Settings > Cloud Accounts. The cloud account owner name is displayed for you.

**STEP 2 |** Verify that you can view the information on your Azure resources on Prisma Cloud.

Wait for approximately 10-24 hours after you onboard the Azure subscription to Prisma Cloud, to review the data that was collected about your Azure resources. After Prisma Cloud ingests data, the information is available for asset inventory, compliance checks and configuration review.

1. Log in to Prisma Cloud.
2. Select Inventory > Assets.

View a snapshot of the current state of all cloud resources or assets that you are monitoring and securing using Prisma Cloud.
Create a Custom Role for Prisma Cloud

If you want to manually create the role and review all the permissions required for monitoring your Azure subscription, instead of using the Terraform template, you can use the custom role JSON file from Prisma Cloud. To create a custom role on Azure, you must have an Azure Active Directory Premium 1 or Premium 2 license plan.

STEP 1 | Download and save the JSON file from here.
Save the JSON file on your local machine or laptop.

If you are using this file for onboarding your Azure China subscription, remove the permission for

Microsoft.Databricks/workspaces/read

because Databricks is not available on Azure China.

STEP 2 | Install the Azure CLI and log in to Azure.

STEP 3 | Go to the directory where you stored the JSON file.

STEP 4 | Enter the following Azure CLI command.

If you renamed the file, you will need to replace the JSON filename to match that in the following command.

```
az role definition create --role-definition "azure_prisma_cloud_lp_read_only.json"
```
For services that are not available in the Azure environment where you are creating the role, the following error message displays `New-AzRoleDefinition : The resource provider referenced in the action is not returned in the list of providers from Azure Resource Manager`. You must edit the JSON file to remove the permissions for services that are not available.

Update an Onboarded Azure Account

For an Azure subscription that you have already added to Prisma Cloud, you can enable flow log ingestion after you manually set it up starting at 6.

**STEP 1 |** Log in to your Prisma Cloud administrative console.

**STEP 2 |**

Add an Azure Active Directory Account on Prisma Cloud

Connecting Prisma™ Cloud to monitor your Azure Active Directory enables you to retrieve information on your users who access resources deployed within your Azure subscription.

This feature is available as a Limited GA, and to try it please contact Palo Alto Networks Customer Success.

**STEP 1 |** Authorize an Azure Active Directory application to read user profile information.

**Watch it.**

1. Log in to the Azure portal.
2. Select Azure Active Directory > App Registrations.
3. Select the application from the list and select API Permissions > Add a Permission.
4. Select **Microsoft Graph > Application Permissions**
   This allows you to allow the application to access the Microsoft Graph APIs.

5. Select the permission **user.read.all**, and **Add Permission**.
   The permission you added allows the app to read user profiles without a signed in user.

   This permission requires admin consent. An Azure AD tenant administrator must grant these permissions by making a call to the admin consent endpoint.

---

**STEP 2 | Collect the details for the Active Directory app.**

1. Log in to the Azure portal.
2. Get the **Active Directory ID**.
3. Get the **Application ID** and **Object ID**.
   You must enter the Object ID as the Service Principal ID in Step 3-d.

4. Get the **Application Key**.
STEP 3 | Add your Azure Active Directory on Prisma Cloud.

2. Enter a Cloud Account Name.
4. Select Cloud Type Azure and the Commercial or Government environment where your AD resources are deployed, click Next.
5. Enter your Azure Active Directory ID, Application ID, Application Key and Service Principal ID.
6. Select the Account Groups you want to add and click Next.

You must assign each cloud account to an account group, and Create an Alert Rule to associate the account group with it to generate alerts when a policy violation occurs.
7. Verify the Status and Done to save your changes.

If Prisma Cloud is able to validate the credentials by making an authentication call using the credentials provided in the previous step, it displays a green check mark.

STEP 4 | Verify that you can view the information on your Azure Active Directory users on Prisma Cloud.

1. Log in to Prisma Cloud.
2. Select **Investigate** and enter the following RQL query to view details on your Azure Active Directory users.

```
config where cloud.type = 'azure' AND api.name = 'azure-active-directory-user' AND json.rule = userType equals "Guest"
```

---

### Set Up Your Azure Subscription for Prisma Cloud

Connect Prisma™ Cloud to your Azure cloud environment so that you can monitor for threats and compliance violations, enable auto-remediation of incidents, and identify hosts and containers that contain vulnerabilities. Before Prisma Cloud can monitor the resources within your Azure Government subscription, you must add Prisma Cloud as an application to your Azure Active Directory and configure your Azure subscription to allow Prisma Cloud to analyze flow log data.

> You do not need to complete this workflow for Azure commercial because the onboarding flow uses Terraform template to automate this process. Start with Add an Azure Subscription on Prisma Cloud

Prisma Cloud requires your Azure **Subscription ID** so that it can identify your Azure cloud account and retrieve the storage account and key vault information. Prisma Cloud also needs the **Directory ID**, **Application ID**, **Application Key**, and **Service Principal ID** to establish the connection between Prisma Cloud and Azure Active Directory so that it can access the resources in your subscription.

**STEP 1 | Locate and copy your Azure subscription ID.**

Prisma Cloud requires the **Subscription ID** so that it can identify your Azure cloud account and can retrieve the storage account and key vault information.

1. Log in to **Microsoft Azure**, select **All services > Subscriptions**, select your subscription, and copy the **Subscription ID**.
**STEP 2 | Add Prisma Cloud as a new application on Azure Active Directory.**

Registering Prisma Cloud as an application on Azure AD generates an Application ID. You need this ID and an Application key to authenticate Prisma Cloud on Azure and to maintain a secure connection.

1. Log in to Microsoft Azure and select **Azure Active Directory > App registrations > New application registration.**

2. Enter a **Name** to identify the Prisma Cloud application, select the **Supported account types** that can use the application as **Accounts in this organizational directory only**, enter your login URL for Prisma Cloud as the **Redirect URI**, and then click **Register.**

   The log in URL for Prisma Cloud is the URL you received in your order confirmation email, and it varies depending on your region.

3. Generate a **client secret** for the Prisma Cloud application.

   The client secret is the application password for Prisma Cloud.

   1. Select **Azure Active Directory > App Registrations > All Applications** and select the Prisma Cloud application.
   2. Add a client secret or the application password (**Certificates & Secrets > New client secret**).
   3. Enter a **Description** and select a **Duration**, which is the term for which the key is valid.
   4. Add the new client secret and then copy the value of that new client secret for your records because you cannot view this key after you close this dialog. You will need this new client secret application key when you **Add an Azure Subscription on Prisma Cloud.**
STEP 3 | Copy the information on the Prisma Cloud application from Azure Active Directory.

For Prisma Cloud to interact with the Azure APIs and collect information on your Azure resources, you must capture the following values.

1. Select **Azure Active Directory > App Registrations > All Applications**, find the Prisma Cloud application you created, and copy the **Directory ID**.

2. Select **Azure Active Directory > Enterprise Applications**.

3. Select your Prisma Cloud application **Properties** and copy the **Application ID** and **Object ID**.

    You must enter the Object ID as the Service Principal ID in the next step.

    Make sure that you get the Object ID for the Prisma Cloud application from **Enterprise Applications > All applications on the Azure portal**—not from **App Registrations**.
STEP 4 | Grant permissions for the Prisma Cloud application to access information at the Azure Subscription level.

To assign roles, you must have Owner or User Access Administrator privileges on your Azure Subscription.

1. Select All Services > Subscriptions.
2. Select your subscription and Add role assignment (Access Control (IAM)).
3. Select the Role, verify that Azure AD user, group, or service principal is selected (Assign access to), and select the Prisma Cloud app to assign the roles.

Review the Azure Cloud Account Onboarding Checklist for a description of the roles and permissions that are required at the subscription level. Then decide which roles you must add for your security and monitoring needs—Reader Role, Reader and Data Access Role, Create a Custom Role on Azure to Enable Prisma Cloud to Access Flow Logs, Network Contributor Role, or Storage Account Contributor Role.
STEP 5 | (Optional) Grant permission for the Prisma Cloud application to access the Azure Key Vault service.

If you use Azure Key Vault to safeguard and manage cryptographic keys and secrets used by your cloud applications and services, Prisma Cloud needs permission to ingest this key vault data.

1. From Azure, select All Services > Key Vaults.
2. Select your Key vault name and Access Policies.
3. Select (add) your Prisma Cloud application (Add new > Select Principal).
4. Select List for Key permissions and for Secret permissions, select both List and List Certificate Authorities for Certification permissions, and then click OK.

STEP 6 | On the Azure portal, Enable Network Watcher and register Insights provider.

STEP 7 | On the Azure portal, Create a storage account on Azure for NSG flow logs.

Your Azure storage account stores the flow logs that are required for Prisma Cloud to monitor and analyze network traffic. When Prisma Cloud ingests the data in these logs, you can interact with the information in Prisma Cloud. For example, you can run queries against the data, visualize network topology, and investigate traffic flows between two instances. You can also apply network policies to this traffic.

If you do not have regulatory guidelines that specify a log retention period to which you must adhere, we recommend you set retention to at least 15 days.

STEP 8 | On the Azure portal, Enable NSG flow logs.

STEP 9 | Configure Prisma Cloud Reader and Data Access role for your Azure storage account.
To ingest Azure flow logs, you have to grant access to the storage account in which the logs are stored. The Reader and Data Access role provides the ability to view everything and allows read/write access to all data contained in a storage account using the associated storage account keys. If your flow logs are stored in storage accounts that belong to one or more subscriptions that are not monitored by Prisma Cloud, you must configure the Prisma Cloud application with the Reader and Data Access role for each storage account.

1. After creating your storage account, select **Access control (IAM) > Add role assignment.**

![Add role assignment](image)

2. Select **Reader and Data Access** as the **Role**, **Select** the administrative user to whom you want to assign the role, and **Save** your changes.

STEP 10 | Add an Azure Subscription on Prisma Cloud.

Create a Custom Role on Azure to Enable Prisma Cloud to Access Flow Logs

To enable Prisma™ Cloud to access Azure flow logs and monitor flow-related data (such as volume of traffic generated by a host, top sources of traffic to the host, or to identify which ports are in use), you must provide the required permissions. While you can use the built-in Network Contributor role that enables a much broader set of permissions, it is a best practice to create a custom role so that you follow the principle of least privilege and limit access rights to the bare-minimum. Use the Azure Cloud Account Onboarding Checklist to verify on which services you want to ingest data and manually assign the permissions for this custom role that includes the permissions required. To create a custom role, install Azure CLI and create a limited role named **Prisma Cloud - Flow Logs Setting Reader** and then enable the role to access flow logs.
To create a custom role on Azure, you must have an Azure Active Directory Premium 1 or Premium 2 license plan.

STEP 1 | Create a custom role using Azure CLI.

If you already assigned a Network Contributor Role to an Azure user, you can skip this step.

- Manually create a custom role JSON file for flow logs only.

  The permissions required are:
  ```json
  ```

1. Install the Azure CLI and log in to Azure.
2. Open a text editor (such as notepad) and enter the following command in the JSON format to create a custom role. You can create custom roles using Azure PowerShell, Azure CLI, or the REST API. These instructions use the Azure CLI command (run on PowerShell or on the DOS command prompt) to create the custom role with queryFlowLogStatus permission. Make sure to provide your Azure Subscription ID in the last line.
3. Save the JSON file on your local Windows system and give it a descriptive name, such as `ad-role-cli.json`.
4. Log in to the Azure portal from the same Windows system and complete the following steps:
   1. Open a PowerShell window (or a DOS Command Prompt Window)
   2. Go to the directory where you stored the JSON file.
   3. Enter the following Azure CLI command (replace the JSON filename to match the name you specified when you saved your custom role JSON file.

```
az role definition create --role-definition "ad-role-cli.json"
```

The output is as follows:

```json
```

STEP 2 | Configure the custom role to access the flow logs.

1. Log in to the Microsoft Azure Portal.
2. Select your subscription (All services > Subscriptions)
3. Select Access control (IAM) > Add role assignment.
Verify that you can see the new custom role you created in the **Role** drop-down.

4. Assign the **Role** to Prisma Cloud, enable the permission to query flow log status, and **Save** your changes.

---

**Use the Azure PowerShell Script to Add an Azure Account**

To save time and reduce the likelihood of human error, you can use the Azure PowerShell Az module to automate the process of setting up the Prisma Cloud application on your Microsoft Azure cloud environment. Because this script adds the Prisma Cloud application on your Azure Active Directory, you must be an administrator with authorization to create roles, and add an application to manage your Azure subscription.

**STEP 1** | Download the **RedLock Azure Onboarding script**.

**STEP 2** | Launch Azure Cloud Shell.

Verify that you are in PowerShell.

**STEP 3** | Upload the RedLock Azure Onboarding script.

**STEP 4** | Verify the script is in the local directory and then enter `./RedLock-Azure-Onboarding-Beta-V5.ps1`

By default, the file is uploaded to your home directory and you might need to use `cd ~` and then enter `./RedLock-Azure-Onboarding-Beta-V5.ps1`

**STEP 5** | When prompted, enter your Azure **SubscriptionID**, a name to identify the RedLock application for example, `APPNAME-RedLock-V2`, and the URL that you use to access Prisma Cloud, for example `https://app.redlock.io`.
STEP 6 | When prompted, open the link on your browser.

You should be prompted with a browser window, login to Azure with user who has permissions to create Application, Roles, ServicePrincipal and change Application permissions.

WARNING: To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code AXR9C to authenticate.

STEP 7 | Log in to Windows Azure PowerShell.

Log in to Azure using an account with Owner or Admin access.

STEP 8 | Enter the authentication code and click Continue to log in to Windows Azure PowerShell.

STEP 9 | Verify the details on successful completion of the process.

The Azure PowerShell window displays the results of the script. It is also saved to a file named Subscription-ApplicationName.txt

1. The Application ID.

2. The Reader role.
STEP 10 | Generate the Application Key (client secret) for the Prisma Cloud application.

The script does not create the application key. To generate an application key (Step 2-3), you must log in to the Azure portal.

STEP 11 | Copy the Prisma Cloud application details from Azure Active Directory.

Prisma Cloud requires the Directory ID, Application ID, Application Key, and Service Principal ID to establish the connection between Prisma Cloud and Azure Active Directory and access the resources in your subscription.

STEP 12 | Continue to Add an Azure Subscription on Prisma Cloud.

Troubleshoot Azure Account Onboarding

After you have completed onboarding your Azure subscription to Prisma Cloud, use these checks to resolve issues if Prisma Cloud cannot retrieve logs and metadata including resource configurations, user activities, network traffic, host vulnerabilities/activities on your Azure resources. Without enabling the correct permissions and configuration on the Azure portal and on Prisma Cloud, you’ll be unable to identify, detect, and remediate issues to manage the risks in your environment.

- Verify that the cloud account permissions are accurate on Settings > Cloud Accounts > Cloud_Account_Name > Status.

At every ingestion cycle when Prisma Cloud connects to the Azure subscription to retrieve and process the data, the service validates that you continue to have the permissions required to continue monitoring the resources within your Azure subscription. Periodically, review the status of these checks to learn of any changes or modifications that limit your ability to maintain visibility and security governance over the resources within your Azure subscription.

- Reader role, and Reader and Data Access role at the subscription level.

If you see the error Subscription does not have Reader role(s) assigned, verify that you have entered the correct Service Principal ID. On the Azure portal, the Object ID for the Prisma Cloud application is what you must provide as the Service Principal ID.
Checks for the Network Contributor role or Custom role that is required to query flow log status.

Checks for the Reader and Data Access role on the storage accounts to access the flow logs stored in storage buckets in subscriptions that are not monitored by Prisma Cloud.

Checks for the Storage Account Contributor role (optional and required only for remediation) that is required for auto remediation of policy violations related to storage accounts.

Checks whether flow logs are published to the storage account.

Check that Azure flow logs are being generated and written to the storage account.
1. Log in to the Azure portal.
2. Select **Storage Accounts** and select the storage account that you want to check.
3. Select **Blobs > Blob Service** and navigate through the folders to find the *.json files.
   These are the flow logs that Prisma Cloud ingests.

On the Azure portal, check that you have created storage accounts in the same regions as the Network Security Groups.

Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG. Azure flow logs must be stored within a storage account in the same region as the NSG.

1. Log in to Prisma Cloud.
2. Select **Investigate** and enter the following RQL query
   
   \[ \text{network where source.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 0} \]

   This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

On the Azure portal, verify that you have enabled Network Watcher instance.

The Network Watcher is required to generate flow logs on Azure.

1. Log in to the Azure portal and select **Network Watcher > Overview** and verify that the status is **Enabled**.
2. Log in to Prisma Cloud.
3. Select Investigate and enter the following RQL query:
   ```rql
   config where cloud type = 'azure'
   AND api.name = 'azure-network-nsg-list'
   addcolumn provisioningState.
   ```

- On the Azure portal, check that you have enabled flow logs on the NSGs.
  1. Log in to the Azure portal, and select Network Watcher > NSG Flow Logs and verify that the status is Enabled.

2. Log in to Prisma Cloud.
3. Select Investigate and enter the following RQL query:
   ```rql
   network where source.publicNetwork
   IN ('Internet IPs', 'Suspicious IPs') AND bytes > 0
   ```
   This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

- The cloud account status displays red and includes the error message "Authentication failed. Azure Subscription not found.

When the Azure subscription is deleted or disabled on the Azure portal and Prisma Cloud cannot monitor the subscription, the cloud account status displays red and includes the error message "Authentication failed. Azure Subscription not found."
Microsoft Azure APIs Ingested by Prisma Cloud

The following APIs are ingested by Prisma Cloud:

<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure Application Gateway</td>
<td>azure-application-gateway</td>
</tr>
<tr>
<td>Azure Active Directory</td>
<td>• azure-active-directory-user</td>
</tr>
<tr>
<td></td>
<td>• azure-active-directory-conditional-access-policy</td>
</tr>
<tr>
<td></td>
<td>• azure-active-directory-named-location</td>
</tr>
<tr>
<td>Azure API Management</td>
<td>azure-api-management-service</td>
</tr>
<tr>
<td>Azure Cache</td>
<td>azure-redis-cache</td>
</tr>
<tr>
<td>Azure Compute</td>
<td>• azure-vm-list</td>
</tr>
<tr>
<td></td>
<td>• azure-disk-list</td>
</tr>
<tr>
<td>Azure Container Registry</td>
<td>azure-container-registry</td>
</tr>
<tr>
<td>Azure Cosmos DB</td>
<td>azure-cosmos-db</td>
</tr>
<tr>
<td>Azure Data Factory</td>
<td>• azure-data-factory-v1</td>
</tr>
<tr>
<td></td>
<td>• azure-data-factory-v2</td>
</tr>
<tr>
<td>Azure Data Bricks</td>
<td>azure-databricks-workspace</td>
</tr>
<tr>
<td>Azure Database for MySQL</td>
<td>azure-mysql-server</td>
</tr>
<tr>
<td>Azure Event Hubs</td>
<td>azure-event-hub-namespace</td>
</tr>
<tr>
<td>Azure Key Vault</td>
<td>• azure-key-vault-list</td>
</tr>
<tr>
<td>Azure Load Balancer</td>
<td>azure-network-lb-list</td>
</tr>
<tr>
<td>Azure Logic Apps</td>
<td>• azure-logic-app-workflow</td>
</tr>
<tr>
<td></td>
<td>• azure-logic-app-custom-connector</td>
</tr>
<tr>
<td>Azure Resource Group</td>
<td>• azure-resource-group</td>
</tr>
<tr>
<td>Azure Resource Manager</td>
<td>azure-role-assignment</td>
</tr>
<tr>
<td>SERVICE</td>
<td>API NAME IN PRISMA CLOUD</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>Azure Policy</td>
<td>azure-policy-assignments</td>
</tr>
<tr>
<td></td>
<td>azure-policy-definition</td>
</tr>
<tr>
<td>Azure Security Center</td>
<td>azure-security-center-settings</td>
</tr>
<tr>
<td>Azure Virtual Network</td>
<td>• azure-network-public-ip-address</td>
</tr>
<tr>
<td></td>
<td>• azure-network-usage</td>
</tr>
<tr>
<td></td>
<td>• azure-network-application-security-group</td>
</tr>
<tr>
<td></td>
<td>• azure-network-vnet-list</td>
</tr>
<tr>
<td></td>
<td>• azure-network-subnet-list</td>
</tr>
<tr>
<td></td>
<td>• azure-network-route-table</td>
</tr>
<tr>
<td></td>
<td>• azure-network-lb-list</td>
</tr>
<tr>
<td></td>
<td>• azure-network-peering</td>
</tr>
<tr>
<td></td>
<td>• azure-network-nic-list</td>
</tr>
<tr>
<td></td>
<td>• azure-network-nsg-list</td>
</tr>
<tr>
<td></td>
<td>• azure-network-firewall</td>
</tr>
<tr>
<td>Azure SQL Database</td>
<td>• azure-sql-db-list</td>
</tr>
<tr>
<td></td>
<td>• azure-sql-server-list</td>
</tr>
<tr>
<td></td>
<td>• azure-sql-managed-instance</td>
</tr>
<tr>
<td>Azure Database for PostgreSQL</td>
<td>azure-postgresql-server</td>
</tr>
<tr>
<td>Azure Monitor</td>
<td>• azure-activity-log-alerts</td>
</tr>
<tr>
<td></td>
<td>• azure-monitor-log-profiles-list</td>
</tr>
<tr>
<td>Azure Network Watcher</td>
<td>azure-network-watcher-list</td>
</tr>
<tr>
<td>Storage Account</td>
<td>azure-storage-account-list</td>
</tr>
<tr>
<td>Subnets</td>
<td>azure-network-subnet-list</td>
</tr>
<tr>
<td>Azure Kubernetes Service</td>
<td>azure-kubernetes-cluster</td>
</tr>
<tr>
<td>App service</td>
<td>azure-app-service</td>
</tr>
<tr>
<td>Azure VPN Gateway</td>
<td>azure-network-vpn-connection-list</td>
</tr>
</tbody>
</table>
Onboard Your Google Cloud Platform (GCP) Account

To enable Prisma™ Cloud to retrieve data on your Google Cloud Platform (GCP) resources and identify potential security risks and compliance issues, you must connect your GCP accounts to Prisma Cloud. In keeping with the GCP resource hierarchy, you can choose whether you want Prisma Cloud to monitor one or more GCP Projects or all projects that are under your GCP Organization. Regardless of what you choose, the process of onboarding automates the process of creating a service account, creating and associating roles with the service account, and enabling specific APIs.

- Permissions and Roles for GCP Account on Prisma Cloud.
- Add Your GCP Project to Prisma Cloud.
- Add Your GCP Organization to Prisma Cloud.
- Create a Service Account With a Custom Role for GCP
- Flow Log Compression on GCP
- Enable Flow Logs for GCP Projects
- Enable Flow Logs for GCP Organization
- GCP APIs Ingested by Prisma Cloud

Permissions and Roles for GCP Account on Prisma Cloud

In order to analyze and monitor your Google Cloud Platform (GCP) account, Prisma Cloud requires access to specific APIs and a service account which is an authorized identity that enables authentication between Prisma Cloud and GCP. A combination of custom, predefined and primitive roles grant the service account the permissions it needs to complete specific actions on the resources in your GCP project or organization. Depending on your cloud protection needs, the service account requires the following roles for read or read-write access:

- Viewer—Primitive role on GCP.
- Prisma Cloud Viewer— Custom role. Prisma Cloud needs this custom role to grant cloud storage bucket permission to read storage bucket metadata and update bucket IAM policies. This role requires storage.buckets.get to retrieve your list of storage buckets, and storage.buckets.getIampolicy to retrieve the IAM policy for the specified bucket.
- Compute Security Admin—Predefined role on GCP. An optional privilege that is required only if you want to enable auto-remediation.
- Organization Role Viewer—Predefined role on GCP. This role is required for onboarding a GCP Organization.
- Dataflow Admin—Predefined role on GCP. An optional privilege that is required for dataflow log compression using the Dataflow service. See Flow Log Compression on GCP for details.
- Folder Viewer—Predefined role on GCP. An optional privilege that is required only if you are onboarding Folders in the GCP resource hierarchy.

Prisma Cloud can ingest data from several GCP APIs. In the GCP project where you create the service account, you must enable the Stackdriver Logging API (logging.googleapis.com) to monitor audit logs, and any other APIs for which you want Prisma Cloud to monitor resources. When you use the Terraform template that Prisma Cloud provides to automate the onboarding of your GCP project or organization, the required permissions are automatically enabled for you.

The following table lists the APIs and associated granular permissions if you want to create a custom role to onboard your GCP account. When the APIs are enabled and the service account has the correct set of
roles and associated permissions, Prisma Cloud can retrieve data about your GCP resources and identify potential security risks and compliance issues across your cloud accounts. To create a custom role for the service account, see Create a Service Account With a Custom Role for GCP before you continue to Add Your GCP Project to Prisma Cloud or Add Your GCP Organization to Prisma Cloud.

To enable the APIs that enable Prisma Cloud to monitor your GCP projects, use it as shown in this example (that uses some of the APIs below):

```
```

Verify the APIs that you have enabled with:

```
gcloud services list
```

<table>
<thead>
<tr>
<th>API</th>
<th>Service Name</th>
<th>Description</th>
<th>Role Name</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Engine API</td>
<td>appengine.googleapis.com</td>
<td>Allows you to access App Engine, which is a fully managed serverless platform on GCP.</td>
<td>App Engine Viewer</td>
<td>appengine.applications.get</td>
</tr>
<tr>
<td>Access Context Manager API</td>
<td>accesscontextmanager.googleapis.com</td>
<td>Read access to policies, access levels, and access zones.</td>
<td>Access Context Manager Reader</td>
<td>accesscontextmanager.accessPolicies.list</td>
</tr>
<tr>
<td>BigQuery API</td>
<td>bigquery.googleapis.com</td>
<td>Allows you to create, manage, share, and query data.</td>
<td>BigQuery Metadata Viewer</td>
<td>bigquery.datasets.get</td>
</tr>
<tr>
<td>Cloud Functions</td>
<td>cloudfunctions.googleapis.com</td>
<td>Cloud Functions is Google Cloud’s</td>
<td>Cloud Functions Viewer</td>
<td>cloudfunctions.functions.getIamPolicy</td>
</tr>
<tr>
<td>API</td>
<td>Service Name</td>
<td>Description</td>
<td>Role Name</td>
<td>Permissions</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>Cloud DataFlow API</td>
<td>dataflow.googleapis.com</td>
<td>Manages Google Cloud Dataflow projects.</td>
<td>Dataflow Admin</td>
<td>resourcemanager.projects.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>storage.buckets.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>storage.objects.create</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>storage.objects.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>storage.objects.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>See Flow Log Compression on GCP</td>
</tr>
<tr>
<td>Cloud DNS API</td>
<td>dns.googleapis.com</td>
<td>Cloud DNS translates requests for domain names into IP addresses and manages and publishes DNS zones and records.</td>
<td>DNS Reader</td>
<td>dns.dnsKeys.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dns.managedZones.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dns.managedZones.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dns.projects.get</td>
</tr>
<tr>
<td>Cloud Pub/Sub</td>
<td>pubsub.googleapis.com</td>
<td>Real-time messaging service that allows you to send and receive messages between independent applications.</td>
<td>Custom Role</td>
<td>pubsub.topics.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pubsub.topics.getIamPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pubsub.subscriptions.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pubsub.subscriptions.getIamPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pubsub.snapshots.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pubsub.snapshots.getIamPolicy</td>
</tr>
<tr>
<td><strong>API</strong></td>
<td><strong>Service Name</strong></td>
<td><strong>Description</strong></td>
<td><strong>Role Name</strong></td>
<td><strong>Permissions</strong></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cloud Resource Manager API</td>
<td>cloudresourcemanager.googleapis.com</td>
<td>Creates, reads, and updates metadata for Google Cloud Platform resource containers.</td>
<td>Role Viewer</td>
<td>resourcemanager.projects.getIamPolicy</td>
</tr>
<tr>
<td>Cloud Key Management Service (KMS) API</td>
<td>cloudkms.googleapis.com</td>
<td>Google Cloud KMS allows customers to manage encryption keys and perform cryptographic operations with those keys.</td>
<td>Custom Role</td>
<td>cloudkms.cryptoKeys.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cloudkms.keyRings.list</td>
</tr>
<tr>
<td>Cloud Service Usage</td>
<td>serviceusage.googleapis.com</td>
<td>API that lists the available or enabled services, or disables services that service consumers no longer use on GCP.</td>
<td>Role Viewer</td>
<td>serviceusage.services.list</td>
</tr>
<tr>
<td>Cloud SQL Admin API</td>
<td>sqladmin.googleapis.com</td>
<td>API for Cloud SQL database instance management.</td>
<td>Custom Role</td>
<td>cloudsql.instances.list</td>
</tr>
<tr>
<td>Compute Engine API</td>
<td>compute.googleapis.com</td>
<td>Creates and runs virtual machines on the Google Cloud Platform.</td>
<td>Compute Network Viewer</td>
<td>compute.backendServices.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.disks.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.firewalls.list</td>
</tr>
<tr>
<td>API</td>
<td>Service Name</td>
<td>Description</td>
<td>Role Name</td>
<td>Permissions</td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
<td>-------------</td>
<td>-----------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.forwardingRules.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.globalForwardingRules.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.images.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.images.getIamPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.instanceGroups.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.instances.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.instanceGroups.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.networks.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.projects.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.regionBackendServices.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.routers.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.routes.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.sslPolicies.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.sslPolicies.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.subnetworks.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.targetHttpProxies.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.targetHttpsProxies.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compute.targetPools.list</td>
</tr>
<tr>
<td>API</td>
<td>Service Name</td>
<td>Description</td>
<td>Role Name</td>
<td>Permissions</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Google API Key</td>
<td>apikeys.googleapis.com</td>
<td>Google lets you manage your project's API keys. This service is in Alpha.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloud Bigtable API</td>
<td>bigtableadmin.googleapis.com</td>
<td>Bigtable is a NoSQL Big Data database service. Custom Role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Cloud Storage API</td>
<td>storage-component.googleapis.com</td>
<td>Cloud Storage is a RESTful service for storing and accessing your data on Google's infrastructure. Custom Role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Dataproc Clusters API</td>
<td>dataproc.googleapis.com</td>
<td>Dataproc is a managed service for creating clusters of Project Viewer, or a custom role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>Service Name</td>
<td>Description</td>
<td>Role Name</td>
<td>Permissions</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Google Recommendations API</td>
<td>recomender.googleapis.com</td>
<td>A service that can be used to run Hadoop and Spark applications.</td>
<td>IAM Recommender Viewer</td>
<td>recommender.iamPolicyRecommendations.list</td>
</tr>
<tr>
<td>GCP IAM Recommender</td>
<td>recommender.googleapis.com</td>
<td>Provides usage recommendations for Google Cloud resources.</td>
<td>IAM Recommender Viewer</td>
<td></td>
</tr>
<tr>
<td>Google Cloud Run</td>
<td>run.googleapis.com</td>
<td>Cloud Run is a managed service for deploy and manage user provided container images</td>
<td>Role Viewer</td>
<td>run.locations.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>run.services.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>run.services.getIamPolicy</td>
</tr>
<tr>
<td>Identity and Access Management (IAM) API</td>
<td>iam.googleapis.com</td>
<td>Manages identity and access control for GCP resources, including the creation of service accounts, which you can use to authenticate to Google and make API calls.</td>
<td>Role Viewer</td>
<td>iam.roles.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>iam.roles.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>iam.serviceAccountKeys.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>iam.serviceAccounts.list</td>
</tr>
<tr>
<td>API</td>
<td>Service Name</td>
<td>Description</td>
<td>Role Name</td>
<td>Permissions</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Kubernetes Engine API</td>
<td>container.googleapis.com</td>
<td>Builds and manages container-based applications, powered by the open source Kubernetes technology.</td>
<td>Kubernetes Engine Cluster Viewer</td>
<td>container.clusters.get</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>container.clusters.list</td>
</tr>
<tr>
<td>Services Usage API</td>
<td>serviceusage.googleapis.com</td>
<td>List all services available to the specified GCP project, and the current state of those services with respect to the project.</td>
<td>N/A</td>
<td>ServiceUsage.Services.List</td>
</tr>
<tr>
<td>Stackdriver Monitoring API</td>
<td>monitoring.googleapis.com</td>
<td>Manages your Stackdriver Monitoring data and configurations.</td>
<td>Monitoring Viewer</td>
<td>monitoring.alertPolicies.list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>monitoring.metricDescriptors.get</td>
</tr>
</tbody>
</table>
### Add Your GCP Project to Prisma Cloud

Begin here to add a GCP project to Prisma Cloud. If you want to add multiple projects, you must either repeat this process for each project you want to onboard, or you allow Prisma Cloud to automatically monitor all GCP projects—current and future—that use the Service Account attached to the project you are adding to Prisma Cloud. Prisma Cloud refers to this service account as a **Master Service Account**.

> **After you start monitoring your project using Prisma Cloud, if you delete the project on GCP, Prisma Cloud learns about it and automatically deletes the account from the list of monitored accounts on Settings > Cloud Accounts. To track the automatic deletion of the project, an audit log is generated with information on the name of the deleted account and the date that the action was performed.**

**STEP 1 |** Access Prisma Cloud and select **Settings > Cloud Accounts > Add New**.

**STEP 2 |** Select **Google Cloud** as the **Cloud to Protect**.

**STEP 3 |** Enter a **Cloud Account Name**.

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies your GCP project on Prisma™ Cloud.
STEP 4 | Select the **Mode**.

Decide whether to enable permissions to only monitor (read-only access) or to monitor and protect (read-write access) the resources in your cloud account. Your selection determines which Terraform template is used to automate the process of creating the service account and attaching the roles required for Prisma Cloud.

STEP 5 | Select **Project for Onboard Using** and enter your **Project ID** and the name of your **Flow Log Storage Bucket**.

The Terraform template does not enable flow logs, and you must complete the workflow in Enable Flow Logs for GCP Projects for Prisma Cloud to retrieve flow logs. Additionally, if you want to enable flow log compression on Prisma Cloud and address the lack of native compression support for flow logs sink setup on GCP, you must do it manually too. When you enable **Use Dataflow to generate compressed logs**, Prisma Cloud sets up the network and compute resources required for flow log compression and this can take up to five minutes.

When you enable flow logs, the service ingests flow log data for the last seven days. Then if flow logs become unavailable for any reason such as if you manually disabled...
flow logs, modified API permissions, or an internal error occurred, when access is restored, logs from the preceding seven days only are ingested.

STEP 6 | (Optional) Allow Prisma Cloud to monitor all current and future GCP projects associated with the service account.

If you have multiple GCP projects, enable **Automatically onboard projects that are accessible by this service account** to allow Prisma Cloud to monitor all current and future GCP projects associated with the Service Account.

STEP 7 | Set up the Service Account for Prisma Cloud.

1. Download the Terraform template for the mode you selected.

   Prisma Cloud recommends that you create a directory to store the Terraform template you download. This allows you to manage the templates when you add a different Google project to Prisma Cloud. Give the directory a name that uniquely identifies the subscription for which you’re using it (for example, onboard-<subscription-name>).

2. Open a new tab on your browser and sign in to the Google Cloud Shell.

3. Upload the template to the Google Cloud Shell.

4. Run the following Terraform commands to generate the Service Account.

   1. `terraform init`
   2. `terraform apply`

5. Upload your **Service Account Key (JSON)** file, review the GCP onboarding configuration displayed on screen to verify that it is correct, and click **Next**.

   The service account security key is used for service-to-service authentication within GCP. The private key file is required to authenticate API calls between your GCP projects and Prisma Cloud.

   If you are on a PC, when you copy the JSON file output from Google Cloud Shell the content is formatted as text instead of JSON. When you upload this file to Prisma Cloud, the **Invalid JSON file** error displays. To fix the error, use a JSON formatting tool such as Sublime or Atom to find the errors (for example, the certificate value should be a single line) and validate the format before you upload the file on Prisma Cloud.
6. Enable the GCP APIs on all projects.

You must enable the Stackdriver Logging API (logging.googleapis.com) to monitor audit logs and any other GCP APIs for which you want Prisma Cloud to monitor resources, on all GCP projects; enabling it only of the project that hosts the service account is not adequate. For example, in the Google Cloud Shell, enter:

```
gcloud services enable serviceusage.googleapis.com
appengine.googleapis.com bigquery.googleapis.com
cloudfunctions.googleapis.com dataflow.googleapis.com
dns.googleapis.com dataproc.googleapis.com
cloudresourcemanager.googleapis.com cloudkms.googleapis.com
sqladmin.googleapis.com compute.googleapis.com storage-component.googleapis.com
recommender.googleapis.com
iam.googleapis.com container.googleapis.com monitoring.googleapis.com
logging.googleapis.com
```

**STEP 8** | Select the account groups to associate to your project and click Next.

You must assign each cloud account to an account group, and Create an Alert Rule to associate the account group with it to generate alerts when a policy violation occurs.
STEP 9 | Verify the onboarding Status of your GCP project to Prisma Cloud and click Done.

You can review the status and take necessary actions to resolve any issues encountered during the onboarding process by viewing the Cloud Accounts page. It takes between 4-24 hours for the flow log data to be exported and analyzed before you can review it on Prisma Cloud. To verify if the flow log data from your GCP project has been analyzed, you can run a network query on the Investigate page.

1. Go to Cloud Accounts, locate your GCP project and view the status.

If Prisma Cloud GCP IAM role does not have adequate permissions to ingest data on the monitored resources within your project, the status icon displays as red or amber and it lists the permissions that are missing.
2. Go to **Investigate**, replace the name with your GCP Cloud Account name and enter the following network query.

This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

```
network where cloud.account = '{{cloud account name}}' AND source.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 0
```

---

**Update an Onboarded Google Cloud Account**

For a Google Cloud project or organization that you have already added to Prisma Cloud, you can update the following options.

**STEP 1** | Log in to the Prisma Cloud administrative console.

**STEP 2** | Select the Google Cloud account you want to modify.

- Select **Settings > Cloud Accounts** and click on the name of the cloud account to manage from the list of cloud accounts.
- Change the account groups you want to monitor.
- If you have onboarded the GCP Organization, select which folders and projects to monitor, or monitor and protect.

On the Google Cloud console, verify that the IAM permissions for the service account includes the Folder Viewer role.
If this role does not have adequate permissions, the following error displays

- Update the flow log bucket name.
- Enable Flow Log Compression on GCP and select Use Dataflow to generate compressed logs (significantly reduces network egress costs).

Enable Flow Logs for GCP Projects

With VPC flow logs, Prisma Cloud helps you visualize flow information for resources deployed in your GCP projects. VPC flow logs on GCP provide flow-level network information of packets going to and from network interfaces that are part of a VPC, including a record of packets flowing to a source port and destination port, the number of distinct peers connecting to an endpoint IP address and port, so that you can monitor your applications from the perspective of your network. On the Investigate page, you can view the traffic flow between virtual machines in different service-projects and/or host-projects that are using shared VPC network and firewall rules.

VPC flow logs are supported on VPC networks only, and are not available for legacy networks on GCP.

To analyze these logs on Prisma Cloud you must enable VPC flow logs for each VPC subnet and export the logs to a sink that holds a copy of each log entry. Prisma Cloud requires you to export the flow logs to a single Cloud Storage bucket, which functions as the sink destination that holds all VPC flow logs in your environment. When you then configure Prisma Cloud to ingest these logs, the service can analyze this data and provide visibility into your network traffic and detect potential network threats such as crypto mining, data exfiltration, and host compromises.

Prisma Cloud automates VPC flow log compression using the Google Cloud Dataflow service, and saves them to your Storage bucket for ingestion. Consider enabling the Google Cloud Dataflow Service and enabling log compression because transferring raw GCP Flow logs from your storage bucket to Prisma Cloud can add to your data cost. See Flow Log Compression on GCP to make sure that you have the permissions to create and run pipelines for a Cloud Dataflow job.

Enabling flow logs will incur high network egress costs. Palo Alto Networks strongly recommends that you enable Flow Log Compression on GCP to significantly reduce the network egress costs associated with sending uncompressed GCP logs to the Prisma Cloud infrastructure.

STEP 1 | Enable flow logs for your VPC networks on GCP.
To analyze your network traffic, you must enable flow logs for each project you want Prisma Cloud to monitor.

1. Log in to GCP console and select your project.

3. Select VPC network and click EDIT.
4. Select Flow logs On to enable flow logs.
5. Set the Aggregation Interval to 15 min.
6. Set the Sample rate to 100%.
As a best practice, setting the aggregate interval and the sample rate as recommended above generates alerts faster on Prisma Cloud and reduces network costs you incur.

7. Save your changes.

STEP 2 | Create a Sink to export flow logs.

You must create a sink and specify a Cloud Storage bucket as the export destination for VPC flow logs. You must configure a sink for every project that you want Prisma Cloud to monitor and configure a single Cloud Storage bucket as the sink destination for all projects. When you Add Your GCP Project to Prisma Cloud, you must provide the Cloud Storage bucket from which the service can ingest VPC flow logs. As a cost reduction best practice, set a lifecycle to delete logs from your Cloud Storage bucket.

1. Select Navigation menu > Logging > Logs Viewer > Create Sink

2. Select GCE Subnetwork.
3. Change All logs to compute.googleapis.com/vpc_flows and click OK.

4. Enter a name and select Cloud Storage as the Sink Service.
5. Select an existing Cloud Storage bucket or create a new Cloud Storage bucket as the Sink Destination, and click Create Sink.
6. Add a lifecycle rule to limit the number of days you store flow logs on the Cloud Storage bucket.

   By default, logs are never deleted. To manage cost, specify the threshold (in number of days) for which you want to store logs.

   2. Select the Lifecycle link for the storage bucket you want to modify.

   3. Add rule and Select object conditions to set Age to 30 days and Select Action as Delete.

   Logs that are stored on your Cloud Storage bucket will be deleted in 30 days.

4. Select Continue and Save your changes.

STEP 3 | Add the name of the Cloud Storage bucket you referenced above in Flow Logs Storage Bucket when you Add Your GCP Project to Prisma Cloud.

Flow Log Compression on GCP

Prisma Cloud enables you to automate the compression of flow logs using the Google Cloud Dataflow service.
This additional automation on Prisma cloud addresses the lack of native compression support for flow logs sink setup on GCP, and helps reduce the egress costs associated with transferring large volume of logs to the Prisma Cloud infrastructure. Therefore, Prisma Cloud recommends that you enable flowlog compression.

When you enable dataflow compression on Prisma Cloud, the dataflow pipeline resources are created in the same GCP project associated with the Google Cloud Storage bucket to which your VPC Flow logs are sent, and it saves the compressed logs also to the Cloud Storage bucket. Therefore, if you are onboarding a GCP Organization and enabling dataflow compression to it or enabling dataflow compression to an existing GCP Organization that has been added to Prisma cloud, make sure that the Dataflow-enabled Project ID is the same Google Cloud Storage bucket to which you send VPC flow logs.

In order to launch the dataflow job and create and stage the compressed files, the following permissions are required:

- Enable the Dataflow APIs.
  The API is dataflow.googleapis.com.
- Grant the service account with permissions to:
  - Run and examine jobs—Dataflow Admin role
  - Create a network, subnetwork, and firewall rules within your VPC—compute.networks.create, compute.subnetworks.create, compute.firewalls.create, compute.networks.updatepolicy

To enable connectivity with the Dataflow pipeline resources and the compute instances that perform log compression within your VPC, Prisma Cloud creates a network, subnetwork, and firewall rules in your VPC. You can view the compute instance that are spun up with the RQL config where api.name='gcloud-compute-instances-list' AND json.rule = name starts with "prisma-compress"

For details on enabling the APIs, see Permissions and Roles for GCP Account on Prisma Cloud.

The Cloud Dataflow service spins up short lived compute instances to handle the compression jobs and you may have associated costs with the service. Palo Alto Networks recommends keeping your Cloud Storage bucket in the same project in which you have enabled the Dataflow service. Based on the location of your Cloud Storage bucket, Prisma Cloud launches the Cloud Dataflow jobs in the following regions:

<table>
<thead>
<tr>
<th>Storage Bucket Region</th>
<th>Region Where the Dataflow is Launched</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-central1</td>
<td>us-central1</td>
</tr>
<tr>
<td>us-east1</td>
<td>us-east1</td>
</tr>
<tr>
<td>us-west1</td>
<td>us-west1</td>
</tr>
<tr>
<td>europe-west1</td>
<td>europe-west1</td>
</tr>
<tr>
<td>europe-west4</td>
<td>europe-west4</td>
</tr>
<tr>
<td>asia-east1</td>
<td>asia-east1</td>
</tr>
<tr>
<td>asia-northeast1</td>
<td>asia-northeast1</td>
</tr>
<tr>
<td>eur4 or eu</td>
<td>europe-west4</td>
</tr>
<tr>
<td>asia</td>
<td>asia-east1</td>
</tr>
</tbody>
</table>
### Add Your GCP Organization to Prisma Cloud

Begin here to add a GCP Organization and folders to Prisma Cloud. If you have added a GCP project to Prisma Cloud and you now want to add the GCP Organization to which the project belongs, the existing GCP project is moved under the Organization in Prisma Cloud.

When you add the GCP Organization to Prisma Cloud, you can specify which folders and projects to include or exclude under the organization resource hierarchy.

**STEP 1 | Review the best practices for onboarding your GCP Organization to Prisma Cloud.**

1. **Enable the GCP APIs on each GCP project.**
   
   For the cloud services that you want Prisma Cloud to monitor or monitor and protect, you must enable the APIs listed in Permissions and Roles for GCP Account on Prisma Cloud. If a cloud service API is not enabled on a GCP project, Prisma Cloud skips the ingestion for the respective service; you must, however, ensure that Service Usage API is enabled on each GCP project that you want Prisma Cloud to monitor or monitor and protect under your GCP Organization hierarchy.

   To skip ingestion for a cycle, Prisma cloud watches the response from the Service Usage API for the details on which cloud services are enabled in a GCP project. For example, if you have not enabled cloud functions in one or more GCP projects within the GCP Organization, Prisma cloud can learn about it and skip the ingestion cycle for this cloud service.

2. **Create the service account in a dedicated GCP project.**

   GCP enforces a limit on the API calls allowed to a GCP project/IAM service account. When you create the service account in a dedicated GCP project, you can ensure that the API calls that Prisma Cloud makes do not interfere with any quota limits against your production workloads and services hosted in the separate GCP project.

3. **Verify that you have granted all the required permissions to the Prisma Cloud service account.**

   If the service account does not have the IAM permissions required to retrieve data, Prisma Cloud skips ingestion of the respective cloud.service(s) for your onboarded account.

**STEP 2 | Access Prisma Cloud** and select **Settings > Cloud Accounts > Add New.**

**STEP 3 | Select Google Cloud as the Cloud to Protect.**

**STEP 4 | Enter a Cloud Account Name.**

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies this GCP organization on Prisma™ Cloud.
STEP 5 | Select the Mode.

Decide whether to enable permissions to only monitor (read-only access) or to monitor and protect (read-write access) the resources in your cloud account. Your selection determines which Terraform template is used to automate the process of creating the service account and attaching the roles required for Prisma Cloud.

STEP 6 | Select Organization for Onboard Using and enter additional details.

1. Enter your **Organization Name** and **Organization ID**.

   All the GCP projects under the Organization hierarchy—current and future—will be monitored by Prisma Cloud. To find your GCP Organization ID, log in to the **GCP console** and select your organization.

2. Enter your **Project ID** and the name of your **Flow Log Storage Bucket**.

3. (Optional) Enable **Use Dataflow to generate compressed logs**.
The Terraform template does not enable flow logs, and you must complete the workflow in Enable Flow Logs for GCP Organization for Prisma Cloud to retrieve flow logs. Additionally, if you want to enable flow log compression on Prisma cloud and address the lack of native compression support for flow logs sink setup on GCP, you must do it manually too. When you enable log compression, Prisma Cloud sets up the network and compute resources required for flow log compression and this can take up to five minutes.

When you enable flow logs, the service ingests flow log data for the last seven days. Then, if flow logs become unavailable for any reason such as if you manually disabled flow logs, modified API permissions, or an internal error occurred, when access is restored, logs from the preceding seven days only are ingested.

4. Enter the Project ID where you enabled the Cloud Dataflow service and click Next.

It is best if this project is where you send your VPC flow logs too.

STEP 7 | Set up the Service Account for Prisma Cloud.

A service account is an identity to which you can grant granular permissions instead of creating individual user accounts. To monitor all the GCP projects that are within the GCP Organizational hierarchy, the service account requires four roles. Of the four roles, three are common for granting permissions at the GCP project level too; the Organization Role Viewer and Folder Viewer roles are additionally required to grant access to the Organization's properties:

- Viewer—Primitive role.
- (Required for Prisma Cloud Compute, Optional for Prisma Cloud) Compute Security Admin—Predefined role.
- RedLock Viewer—Custom role.
- Organization Role Viewer—Predefined role.
- Folder Viewer—Predefined role.

1. Download the Terraform template for the mode you selected.

Prisma Cloud recommends that you create a directory to store the Terraform template you download. This allows you to manage the templates when you add a different Google organization to Prisma Cloud. Give the directory a name that uniquely identifies the subscription for which you’re using it (for example, onboard-<subscription-name>).

2. Open a new tab on your browser and sign in to the Google Cloud Shell.

3. Upload the template to the Google Cloud Shell.

4. Run the following Terraform commands to generate the Service Account.

1. `terraform init`
2. `terraform apply`

5. Upload your Service Account Key (JSON) file, review the GCP onboarding configuration displayed on screen to verify that it is correct, and click Next.

The service account security key is used for service-to-service authentication within GCP. The private key file is required to authenticate API calls between your GCP projects and Prisma Cloud.

If you are on a PC, when you copy the JSON file output from Google Cloud Shell the content is formatted as text instead of JSON. When you upload this file to Prisma Cloud, the Invalid JSON file error displays. To fix the error, use a JSON formatting tool such as Sublime or Atom (for example, the certificate value should be a single line), and validate the formatting before you upload the file on Prisma Cloud.
6. Select the projects you want to add to Prisma Cloud.

You can choose to include:

- **All projects** included within the organization hierarchy.
- **Include a subset** or **Exclude a subset** of projects. Select the relevant tab and choose the projects to include or exclude.

When you select a folder, all existing projects within that folder or sub-folder are onboarded to Prisma Cloud. The periodic sync also checks for any new projects and sub-folders that you subsequently add on the cloud platform and adds them to Prisma Cloud.

7. Resolve any missing permissions or errors.

If the service account does not have adequate permissions, the following error displays.
and if there are issues with the following message indicates that there is an issue with the service account. This error occurs when the service account is deleted, or disabled or when the key is deleted on the Google Cloud console.

8. Enable the GCP APIs.

In the GCP project where you created the service account, you must enable the Stackdriver Logging API (logging.googleapis.com) to monitor audit logs, and any other GCP APIs for which you want Prisma Cloud to monitor resources. For example, in the Google Cloud Shell, enter:

```bash
```

**STEP 8** | Select the account groups to associate to your GCP project and click **Next**.

You must assign each cloud account to an account group, and Create an Alert Rule to associate the account group with it to generate alerts when a policy violation occurs.

**STEP 9** | Verify the onboarding **Status** of your GCP Organization to Prisma Cloud and click **Done**.

If you are missing permissions for the GCP IAM role to successfully ingest data from your GCP Organization, the icon displays red or amber and the details of the permission gaps display on screen.

You can review the status and take necessary actions to resolve any issues encountered during the onboarding process by viewing the **Cloud Accounts** page. It takes between 4-24 hours for the flow log
data to be exported and analyzed before you can review it on Prisma Cloud. To verify if the flow log data from your GCP Organization have been analyzed, you can run a network query on the Investigate page.

- After you add the GCP Organization to Prisma Cloud, you must create a support request to delete the GCP Organization or the projects within your GCP Organization. You cannot delete the account from Prisma Cloud.
- Because Prisma Cloud has access to all projects associated with a Service Account, if you want to remove access to a project that is associated with the Service Account, you must remove the project from the Service Account on the GCP IAM console. In the next scanning cycle, the project is excluded and Prisma Cloud no longer has access to the project.

1. Go to Cloud Accounts, locate your GCP account and view the status.

2. Verify the projects that are onboarded to Prisma Cloud.
   Select the cloud account name and review the list of projects to verify the include/exclude selections you made earlier.

3. Go to Investigate, replace the name with your GCP Cloud Account name and enter the following network query.
   This query allows you to list all network traffic from the Internet or from Suspicious IP addresses with over 0 bytes of data transferred to a network interface on any resource on any cloud environment.

\[
\text{network where cloud.account} = '{{\text{cloud account name}}}'} \text{ AND source.publicnetwork IN (\text{\textquote{Internet IPs}, \textquote{Suspicious IPs}})} \text{ AND bytes} > 0
\]
Enable Flow Logs for GCP Organization

Prisma Cloud uses the traffic data in flow logs for your GCP organization or folder resource hierarchy to detect network threats such as cryptomining, data exfiltration, and host compromises. Before Prisma Cloud can analyze your flow log data, you must create a sink to export the flow logs to a Cloud Storage bucket. To configure a sink for your whole GCP organization or folder, use the gcloud command line tool.

*Enabling flow logs will incur high network egress costs. Palo Alto Networks strongly recommends that you enable Flow Log Compression on GCP to significantly reduce the network egress costs associated with sending uncompressed GCP logs to the Prisma Cloud infrastructure.*

**STEP 1** | Gather the following information from your GCP account:
  - Cloud Storage bucket name
  - Organization ID

**STEP 2** | Download and install the Google Cloud SDK.

During the SDK install, you must log in to your GCP account. This account must have these three permissions enabled at the organization level:
  - Billing Account Administrator
  - Logging Admin
  - Organization Administrator

**STEP 3** | Run this command to create a service account needed to configure the sink for your Cloud Storage bucket but replace the **Bucket-name** with your Cloud Storage bucket name and **Organization ID** with your organization ID.

```
$ gcloud logging sinks create <sink-name> storage.googleapis.com/<bucket-name> --include-children --organization=<organisation-id> --log-filter="resource.type="gce_subnetwork" AND logName:"logs/compute.googleapis.com%2Fvpc_flows"
```
If you are onboarding a GCP folder, you must have the Folder Viewer role and can use the command 

```bash
$ gcloud logging sinks create <sink-name> storage.googleapis.com/<bucket-name> --include-children --folder=<folder-id> --log-filter="resource.type="gce_subnetwork" AND logName:"logs/compute.googleapis.com%2Fvpc_flows"
```

to create a service account needed to configure the sink for your Cloud Storage bucket.

**STEP 4** | Grant the service account permission to access your Cloud Storage bucket.

1. Select **Navigation menu > Storage** and select your Cloud Storage bucket.
2. Select **Permissions > Add members**.
3. Add the service account email address for **Members**, select **Storage > Storage Admin** and select **Add**.

**STEP 5** | Add the name of Cloud Storage bucket you created above in **Flow Logs Storage Bucket** when you **Add Your GCP Organization to Prisma Cloud**.

**STEP 6** | *(Optional)* Enable **Flow Log Compression on GCP**.

Enable flow log compression on Prisma Cloud to automate the compression of flow logs using the Google Cloud Dataflow service. When enabled the compressed logs are stored to the same Storage bucket as your flow logs and mitigates the network egress costs associated with sending uncompressed GCP logs to the Prisma Cloud infrastructure.

**Create a Service Account With a Custom Role for GCP**

If you prefer to create a service account with more granular permissions to **Add Your GCP Organization to Prisma Cloud** or **Add Your GCP Project to Prisma Cloud**, instead of using the Terraform template which grants the Viewer (primitive) role for read-only access to resources in your GCP account, use the following instructions.
If you enable granular permissions, you must update the custom role and add additional permissions that may be required to ingest data from any new service that is added on Prisma Cloud.

To enable dataflow log compression using the Dataflow service, you must enable additional permissions. See Flow Log Compression on GCP for details on ingesting network log data.

**STEP 1** | Create a YAML file with the custom permissions.

1. Create a YAML file and add the granular permissions for the custom role.

   Use this YAML format as an example. You must add the permissions for onboarding your GCP project or organization, from the link above, to this file:

   ```yaml
   title: prisma-custom-role
   description: prisma-custom-role
   stage: beta
   includedPermissions:
     - compute.networks.list
     - compute.backendServices.list
   ```

**STEP 2** | Create the custom role.

When creating a service role, you must select a GCP project because GCP does not allow the service account to belong directly under the GCP Organization.

1. Select the GCP project in which you want to create the custom role.
2. Upload the YAML file to the Cloud Shell.
3. Run the `gcloud` command:
   ```bash
   gcloud iam roles create <prisma customrole name> --project <project-ID> --file <YAML file name>
   ```

**STEP 3** | Create a Service Account and attach the custom role to it.

2. Create Service Account and add the role you created earlier to it.
3. Create a key and download the private key.

**STEP 4** | Continue to Add Your GCP Project to Prisma Cloud and use the private key for the service account to complete onboarding.

**STEP 5** | (For onboarding GCP Organization only) Create the custom role in the GCP Organization level.

1. Select your GCP Organization.
2. Verify that the YAML file you created in Step 1 includes the additional permissions for GCP Organization.

   Run the `gcloud` command:
   ```bash
   gcloud iam roles create <prisma customrole name> --organization <org ID> --file <YAML File name>
   ```
STEP 6 | (For onboarding GCP Organization only) Set up your Service Account to monitor all the GCP folders and projects within the GCP Organization.

You must associate the Service account you created in the project in Step 3 to the GCP Organization-level and add the custom role you created in the previous step. Additionally, you must add the predefined role for Organization Viewer to the service account. All these tasks together enable the service account to monitor all the GCP projects that are within the GCP Organizational hierarchy.

1. Copy the service account member address.

   Select the project that you used to create the service account, and select IAM & admin > IAM to copy the service account member address.

2. Select your Organization, select IAM & Admin > IAM to Add members to the service account.
3. Paste the service account member address you copied as New members, and Select a role.
4. Select the custom role you created in Step 4, and click + ADD ANOTHER ROLE.
5. Select Resource Manager > Organization Role Viewer, and Folder Viewer role, and Save.

   The Organization Viewer role enables permissions to view the Organization name without granting access to all resources in the Organization. The Folder Viewer roles is also required to onboard your GCP folders.
STEP 7 | (For onboarding GCP Organization only) Continue to Add Your GCP Organization to Prisma Cloud and use the private key associated with your service account to complete onboarding.

GCP APIs Ingested by Prisma Cloud

The following are GCP APIs that have been ingested by Prisma Cloud.

<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Alpha release) Google API Key</td>
<td>gcloud-api-key</td>
</tr>
<tr>
<td>Google Compute Engine (GCE)</td>
<td>• gcp-compute-disk-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-instances-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-interfaces-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-project-info</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-nat</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-route</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-router</td>
</tr>
<tr>
<td></td>
<td>• gcloud-virtual-network-interface</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-image</td>
</tr>
<tr>
<td>Google VPC</td>
<td>• gcloud-compute-networks-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-networks-subnets-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-firewall-rules-list</td>
</tr>
<tr>
<td>SERVICE</td>
<td>API NAME IN PRISMA CLOUD</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>Google Access Context Manager</td>
<td>gcloud-access-policy</td>
</tr>
<tr>
<td>Google App Engine</td>
<td>• gcloud-app-engine-firewall-rule</td>
</tr>
<tr>
<td></td>
<td>• gcloud-app-engine-application</td>
</tr>
<tr>
<td>Google Bigtable</td>
<td>• gcloud-bigtable-instance-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-bigtable-table</td>
</tr>
<tr>
<td>Google Cloud Functions</td>
<td>gcloud-cloud-function</td>
</tr>
<tr>
<td>Google Stackdriver</td>
<td>• gcloud-monitoring-policies-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-logging-metric</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>gcloud-sql-instances-list</td>
</tr>
<tr>
<td>Google Cloud Storage</td>
<td>gcloud-storage-buckets-list</td>
</tr>
<tr>
<td>Google Recommendations for GCP IAM Recommender</td>
<td>gcloud-iam-policy-recommendation-list</td>
</tr>
<tr>
<td>Cloud Identity &amp; Access Management (Cloud IAM)</td>
<td>• gcloud-iam-service-accounts-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-iam-service-accounts-keys-list</td>
</tr>
<tr>
<td></td>
<td>• gcloud-iam-get-audit-config</td>
</tr>
<tr>
<td></td>
<td>• gcloud-project-iam-role</td>
</tr>
<tr>
<td></td>
<td>• gcloud-organization-iam-role</td>
</tr>
<tr>
<td>Google Cloud Resource Manager</td>
<td>• gcloud-organization-iam-policy</td>
</tr>
<tr>
<td></td>
<td>• gcloud-projects-get-iam-user</td>
</tr>
<tr>
<td></td>
<td>• gcloud-projects-get-iam-policy</td>
</tr>
<tr>
<td>Cloud SQL</td>
<td>gcloud-sql-instances-list</td>
</tr>
<tr>
<td>BigQuery</td>
<td>gcloud-bigquery-dataset-list</td>
</tr>
<tr>
<td>Google Kubernetes Engine</td>
<td>gcloud-container-describe-clusters</td>
</tr>
<tr>
<td>DNS</td>
<td>• gcloud-dns-project-info</td>
</tr>
<tr>
<td></td>
<td>• gcloud-dns-managed-zone</td>
</tr>
<tr>
<td>Google Cloud Key Management Service (KMS)</td>
<td>gcloud-kms-keyring-list</td>
</tr>
<tr>
<td>Google Cloud Memorystore</td>
<td>gcloud-redis-instances-list</td>
</tr>
<tr>
<td>Google Service Usage</td>
<td>gcloud-services-list</td>
</tr>
<tr>
<td>Google Stackdriver Logging</td>
<td>gcloud-logging-sinks-list</td>
</tr>
<tr>
<td></td>
<td>gcloud-events-logging-sinks-list</td>
</tr>
<tr>
<td>Google Cloud Load Balancing</td>
<td>• gcloud-compute-internal-lb-backend-service</td>
</tr>
<tr>
<td></td>
<td>• gcloud-lb-virtual-network-interface</td>
</tr>
<tr>
<td>SERVICE</td>
<td>API NAME IN PRISMA CLOUD</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-target-pools</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-target-https-proxies</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-url-maps</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-global-forwarding-rule</td>
</tr>
<tr>
<td></td>
<td>• gcloud-compute-load-balancer</td>
</tr>
<tr>
<td>Google PubSub</td>
<td>• gcloud-pubsub-topic</td>
</tr>
<tr>
<td></td>
<td>• gcloud-pubsub-subscription</td>
</tr>
<tr>
<td></td>
<td>• gcloud-pubsub-snapshot</td>
</tr>
</tbody>
</table>
Onboard Your Alibaba Cloud Account

Use Prisma™ Cloud for monitoring your Alibaba Cloud infrastructure in Mainland China and International regions.

To begin monitoring the resources on your cloud infrastructure, you must first create a role on Alibaba Cloud and then connect your Alibaba cloud account to Prisma Cloud. When you add your cloud account to Prisma Cloud, the API integration between your cloud infrastructure provider and Prisma Cloud is established and you can begin monitoring the resources and identify potential security risks in your infrastructure.

- Set Up Your Alibaba Account
- Add an Alibaba Cloud Account on Prisma Cloud
- Alibaba APIs Ingested by Prisma Cloud

Set Up Your Alibaba Account

Prisma Cloud is available for visibility and monitoring of your Alibaba Cloud infrastructure in Mainland China and International regions. The first step to start monitoring your resources on Alibaba Cloud is to grant Prisma Cloud access to your account. To do this, you must create a role and attach policies that enable permissions to authorize access to the assets deployed within the account. You can choose to create a custom policy with granular permissions or use the Alibaba Cloud system policy to enable ReadOnlyAccess. After you create the role and enable permissions, you can add the Alibaba Cloud Resource Name (ARN) on Prisma Cloud so that it can assume the role to monitor your Alibaba Cloud account.

**STEP 1** | (Required if you want to enable granular access permissions) Create a custom policy.

Creating a custom policy allows you to use the principle of least privilege and enable the bare-minimum permissions that Prisma Cloud currently requires to monitor your account. If you do not want to update these permissions periodically, you can skip ahead to **Step 2** and use the Alibaba Cloud system policy to enable ReadOnlyAccess permissions to all aliyun services.

1. Download the permissions for Alibaba China.
   - The JSON file includes the required permissions.

2. Log in to the Alibaba Cloud console for China region.
4. Enter a new Policy Name and select Script.
5. Paste the contents in to the Policy Document and click OK.
STEP 2 | Create a RAM role.

You must create a RAM role and attach policies to authorize API access to Prisma Cloud. You can attach the custom policy with granular permissions or use the Alibaba Cloud system policy to enable `ReadOnlyAccess`.

1. On the Alibaba Cloud console, select **Product > Resource Access Management**.
2. Select **RAM Roles > Create RAM Role**.
3. Select Trusted entity type as Alibaba Cloud Account and Next.
4. Enter a RAM Role Name.
5. Enter the Prisma Cloud Account ID as a trusted Alibaba Cloud account.

If your Prisma Cloud instance is on https://app.prismacloud.cn, the Prisma Cloud Account ID is 1306560418200997. Otherwise, the Prisma Cloud Account ID is 5770382605230796.

Enter the appropriate account ID in Select Trusted Alibaba Cloud Account > Other Alibaba Cloud Account and click OK.

6. Select Add Permissions to RAM Role.

Either attach the permissions associated with the custom policy (if you created one), or use the system policy.
- Custom Policy
• System Policy

7. Click Finished.

STEP 3 | Copy the Alibaba Cloud Resource Name (ARN).

You need the ARN to add the Alibaba cloud account on Prisma Cloud.

1. Select RAM Roles and search for the name you entered earlier.
2. Note the ARN.

**STEP 4 | Add an Alibaba Cloud Account on Prisma Cloud.**

Add an Alibaba Cloud Account on Prisma Cloud

After you create a RAM role with permissions that enable Prisma Cloud programmatic access to your cloud resources on Alibaba Cloud, all you now need to do for visibility into changes on the cloud infrastructure is to add the account you want to monitor.

**STEP 1 | Access Prisma Cloud.**

If you have not already activated your account, check your email for the welcome to Prisma Cloud email that includes your username, and create a new password to log in.
New User Activation

Hello

Congratulations! You're just a few steps away from experiencing the most comprehensive cloud monitoring and compliance service available today. Get started by activating your account with the following link and this username: @paloaltonetworks.com.

Activate My Account

Don’t recognize this? It's possible the administrator for your organization’s Prisma Cloud® subscription created this account for you. For further help or clarification, please contact npc-help@paloaltonetworks.com.

Thank you!
The Prisma Cloud Team

On first-time login to Prisma Cloud in the Alibaba Mainland China region, you must accept the EULA. When you click the EULA, a new page displays where you can review the content. As a temporary work around, to Agree and Submit you must refresh the page and log in again with your credentials.

STEP 2 | Select Settings > Cloud Accounts > Add New.

STEP 3 | Select Alibaba Cloud as the Cloud to Protect.
STEP 4 | Enter a Cloud Account Name.

A cloud account name is auto-populated for you. You can replace it with a cloud account name that uniquely identifies your Alibaba Cloud account on Prisma™ Cloud.

STEP 5 | Enter the Alibaba Cloud Resource Name (ARN) as RAM Role and click Next.

The ARN is the unique identifier for the RAM role you created to authorize API access for Prisma Cloud. When you enter the ARN, the Alibaba Cloud Account ID gets added automatically.
STEP 6 | Select one or more account groups and click Next.

You must assign each cloud account to an account group, and create an alert rule to associate the account group with it to generate alerts when a policy violation occurs.

STEP 7 | Verify the onboarding status.

If you have set up the RAM role and policies properly, the onboarding process should be successful.
**STEP 8 | Select Done.**

**STEP 9 | Next Steps:**

- Review the Prisma Cloud default Policies for Alibaba Cloud.

  Select Policies, set the Cloud Type filter as Alibaba Cloud and view all the Config policies that are available to detect any misconfigurations in your infrastructure.
- Start using the **Prisma Cloud Asset Inventory** for visibility.

**Alibaba APIs Ingested by Prisma Cloud**

Prisma Cloud ingests the following Alibaba Cloud APIs to retrieve metadata on your resources in the Alibaba Cloud environment.

<table>
<thead>
<tr>
<th>SERVICE</th>
<th>API NAME IN PRISMA CLOUD</th>
</tr>
</thead>
</table>
| Alibaba Resource Access Management | • alibaba-cloud-ram-password-policy  
• alibaba-cloud-ram-group  
• alibaba-cloud-ram-policy  
• alibaba-cloud-ram-role  
• alibaba-cloud-ram-user |
| Action Trail             | alibaba-cloud-action-trail |
| Elastic Compute Service  | • alibaba-cloud-ecs-disk  
• alibaba-cloud-ecs-instance  
• alibaba-cloud-ecs-security-group |
| Object Storage Services  | alibaba-cloud-oss-bucket-info |
| VPC                      | alibaba-cloud-vpc |
| RDS                      | alibaba-cloud-rds-instance |
| Server Load Balancer     | alibaba-cloud-load-balancer |
Cloud Service Provider Regions on Prisma Cloud

View the list of all cloud regions supported on Prisma Cloud.

- AWS Regions
- Azure
- GCP
- Alibaba Cloud

### AWS Regions

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap-east-1</td>
<td>AWS Hong Kong</td>
</tr>
<tr>
<td>ap-northeast-1</td>
<td>AWS Tokyo</td>
</tr>
<tr>
<td>ap-northeast-2</td>
<td>AWS Seoul</td>
</tr>
<tr>
<td>ap-south-1</td>
<td>AWS Mumbai</td>
</tr>
<tr>
<td>ap-southeast-1</td>
<td>AWS Singapore</td>
</tr>
<tr>
<td>ap-southeast-2</td>
<td>AWS Sydney</td>
</tr>
<tr>
<td>ca-central-1</td>
<td>AWS Canada</td>
</tr>
<tr>
<td>eu-central-1</td>
<td>AWS Frankfurt</td>
</tr>
<tr>
<td>eu-north-1</td>
<td>AWS Stockholm</td>
</tr>
<tr>
<td>eu-west-1</td>
<td>AWS Ireland</td>
</tr>
<tr>
<td>eu-west-2</td>
<td>AWS London</td>
</tr>
<tr>
<td>eu-west-3</td>
<td>AWS Paris</td>
</tr>
<tr>
<td>me-south-1</td>
<td>AWS Bahrain</td>
</tr>
<tr>
<td>sa-east-1</td>
<td>AWS Sao Paulo</td>
</tr>
<tr>
<td>us-east-1</td>
<td>AWS Virginia</td>
</tr>
<tr>
<td>us-east-2</td>
<td>AWS Ohio</td>
</tr>
<tr>
<td>us-west-1</td>
<td>AWS California</td>
</tr>
<tr>
<td>Region ID</td>
<td>Region Name</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>us-west-2</td>
<td>AWS Oregon</td>
</tr>
<tr>
<td>us-gov-east-1</td>
<td>AWS GovCloud (US-East)</td>
</tr>
<tr>
<td>us-gov-west-1</td>
<td>AWS GovCloud (US-West)</td>
</tr>
</tbody>
</table>

**AWS China**

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn-north-1</td>
<td>AWS Beijing</td>
</tr>
<tr>
<td>cn-northwest-1</td>
<td>AWS Ningixa</td>
</tr>
</tbody>
</table>

**Azure**

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>australiacentral</td>
<td>Azure Australia Central</td>
</tr>
<tr>
<td>australiacentral2</td>
<td>Azure Australia Central 2</td>
</tr>
<tr>
<td>australiaeast</td>
<td>Azure Australia East</td>
</tr>
<tr>
<td>australiasoutheast</td>
<td>Azure Australia Southeast</td>
</tr>
<tr>
<td>brazilsouth</td>
<td>Azure Brazil South</td>
</tr>
<tr>
<td>canadacentral</td>
<td>Azure Canada Central</td>
</tr>
<tr>
<td>canadaeast</td>
<td>Azure Canada East</td>
</tr>
<tr>
<td>centralindia</td>
<td>Azure Central India</td>
</tr>
<tr>
<td>centralus</td>
<td>Azure Central US</td>
</tr>
<tr>
<td>chineast</td>
<td>Azure China East</td>
</tr>
<tr>
<td>chineast2</td>
<td>Azure China East 2</td>
</tr>
<tr>
<td>chinanorth</td>
<td>Azure China North</td>
</tr>
<tr>
<td>chinanorth2</td>
<td>Azure China North 2</td>
</tr>
<tr>
<td>eastasia</td>
<td>Azure East Asia</td>
</tr>
<tr>
<td>eastus</td>
<td>Azure East US</td>
</tr>
<tr>
<td>Region ID</td>
<td>Region Name</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>eastus2</td>
<td>Azure East US 2</td>
</tr>
<tr>
<td>francecentral</td>
<td>Azure France Central</td>
</tr>
<tr>
<td>francesouth</td>
<td>Azure France South</td>
</tr>
<tr>
<td>germanycentral</td>
<td>Azure Germany Central</td>
</tr>
<tr>
<td>germannorth</td>
<td>Azure Germany North</td>
</tr>
<tr>
<td>germannortheast</td>
<td>Azure Germany Northeast</td>
</tr>
<tr>
<td>germanywestcentral</td>
<td>Azure Germany West Central</td>
</tr>
<tr>
<td>japaneast</td>
<td>Azure Japan East</td>
</tr>
<tr>
<td>japanwest</td>
<td>Azure Japan West</td>
</tr>
<tr>
<td>koreacentral</td>
<td>Azure Korea Central</td>
</tr>
<tr>
<td>koreasouth</td>
<td>Azure Korea South</td>
</tr>
<tr>
<td>northcentralus</td>
<td>Azure North Central US</td>
</tr>
<tr>
<td>northeurope</td>
<td>Azure North Europe</td>
</tr>
<tr>
<td>norwayeast</td>
<td>Azure Norway East</td>
</tr>
<tr>
<td>norwaywest</td>
<td>Azure Norway West</td>
</tr>
<tr>
<td>southafricanorth</td>
<td>Azure South Africa North</td>
</tr>
<tr>
<td>southafricawest</td>
<td>Azure South Africa West</td>
</tr>
<tr>
<td>southcentralus</td>
<td>Azure South Central US</td>
</tr>
<tr>
<td>southeastasia</td>
<td>Azure Southeast Asia</td>
</tr>
<tr>
<td>southindia</td>
<td>Azure South India</td>
</tr>
<tr>
<td>switzerlandnorth</td>
<td>Azure Switzerland North</td>
</tr>
<tr>
<td>switzerlandwest</td>
<td>Azure Switzerland West</td>
</tr>
<tr>
<td>uaeentral</td>
<td>Azure UAE Central</td>
</tr>
<tr>
<td>uaeenorth</td>
<td>Azure UAE North</td>
</tr>
<tr>
<td>uksouth</td>
<td>Azure UK South</td>
</tr>
<tr>
<td>ukwest</td>
<td>Azure UK West</td>
</tr>
<tr>
<td>Region ID</td>
<td>Region Name</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>westcentralus</td>
<td>Azure West Central US</td>
</tr>
<tr>
<td>westeurope</td>
<td>Azure West Europe</td>
</tr>
<tr>
<td>westindia</td>
<td>Azure West India</td>
</tr>
<tr>
<td>westus</td>
<td>Azure West US</td>
</tr>
<tr>
<td>westus2</td>
<td>Azure West US 2</td>
</tr>
<tr>
<td></td>
<td><strong>Azure Government</strong></td>
</tr>
<tr>
<td>usgovarizona</td>
<td>Azure Gov Arizona (US)</td>
</tr>
<tr>
<td>usgoviowa</td>
<td>Azure Gov Iowa (US)</td>
</tr>
<tr>
<td>usgovtexas</td>
<td>Azure Gov Texas (US)</td>
</tr>
<tr>
<td>usgovvirginia</td>
<td>Azure Gov Virginia (US)</td>
</tr>
</tbody>
</table>

**GCP**

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>asia</td>
<td>GCP Asia Pacific</td>
</tr>
<tr>
<td>asia-east1</td>
<td>GCP Taiwan</td>
</tr>
<tr>
<td>asia-east2</td>
<td>GCP Hong Kong</td>
</tr>
<tr>
<td>asia-northeast1</td>
<td>GCP Tokyo</td>
</tr>
<tr>
<td>asia-northeast2</td>
<td>GCP Osaka</td>
</tr>
<tr>
<td>asia-northeast3</td>
<td>GCP Seoul</td>
</tr>
<tr>
<td>asia-south1</td>
<td>GCP Mumbai</td>
</tr>
<tr>
<td>asia-southeast1</td>
<td>GCP Singapore</td>
</tr>
<tr>
<td>australia-southeast1</td>
<td>GCP Sydney</td>
</tr>
<tr>
<td>eu</td>
<td>GCP European Union</td>
</tr>
<tr>
<td>eur4</td>
<td>GCP Finland and Netherlands</td>
</tr>
<tr>
<td>europe</td>
<td>GCP Europe</td>
</tr>
<tr>
<td>europe-north1</td>
<td>GCP Finland</td>
</tr>
</tbody>
</table>
### Region ID vs Region Name

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>europe-west1</td>
<td>GCP Belgium</td>
</tr>
<tr>
<td>europe-west2</td>
<td>GCP London</td>
</tr>
<tr>
<td>europe-west3</td>
<td>GCP Frankfurt</td>
</tr>
<tr>
<td>europe-west4</td>
<td>GCP Netherlands</td>
</tr>
<tr>
<td>europe-west6</td>
<td>GCP Switzerland</td>
</tr>
<tr>
<td>nam4</td>
<td>GCP Iowa and South Carolina</td>
</tr>
<tr>
<td>northamerica-northeast1</td>
<td>GCP Montreal</td>
</tr>
<tr>
<td>southamerica-east1</td>
<td>GCP Sao Paulo</td>
</tr>
<tr>
<td>us</td>
<td>GCP United States</td>
</tr>
<tr>
<td>us-central1</td>
<td>GCP Iowa</td>
</tr>
<tr>
<td>us-east1</td>
<td>GCP South Carolina</td>
</tr>
<tr>
<td>us-east4</td>
<td>GCP Northern Virginia</td>
</tr>
<tr>
<td>us-west1</td>
<td>GCP Oregon</td>
</tr>
<tr>
<td>us-west2</td>
<td>GCP Los Angeles</td>
</tr>
<tr>
<td>asia-northeast3</td>
<td>GCP Seoul</td>
</tr>
<tr>
<td>us-west4</td>
<td>GCP Las Vegas</td>
</tr>
<tr>
<td>asia-southeast2</td>
<td>GCP Jakarta</td>
</tr>
<tr>
<td>us-west3</td>
<td>GCP Salt Lake City</td>
</tr>
</tbody>
</table>

### Alibaba Cloud

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibaba China</td>
<td></td>
</tr>
<tr>
<td>ali.cn.cn-beijing</td>
<td>Alibaba Cloud Beijing</td>
</tr>
<tr>
<td>ali.cn.cn-chengdu</td>
<td>Alibaba Cloud Chengdu</td>
</tr>
<tr>
<td>ali.cn.cn-hangzhou</td>
<td>Alibaba Cloud Hangzhou</td>
</tr>
<tr>
<td>ali.cn.cn-huhehaote</td>
<td>Alibaba Cloud Hohhot</td>
</tr>
<tr>
<td>Region ID</td>
<td>Region Name</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>ali.cn.cn-qingdao</td>
<td>Alibaba Cloud Qingdao</td>
</tr>
<tr>
<td>ali.cn.cn-shanghai</td>
<td>Alibaba Cloud Shanghai</td>
</tr>
<tr>
<td>ali.cn.cn-shenzhen</td>
<td>Alibaba Cloud Shenzhen</td>
</tr>
<tr>
<td>ali.cn.cn-zhangjiakou</td>
<td>Alibaba Cloud Zhangjiakou</td>
</tr>
<tr>
<td><strong>Alibaba International</strong></td>
<td></td>
</tr>
<tr>
<td>ali.int.ap-northeast-1</td>
<td>Alibaba Cloud Tokyo</td>
</tr>
<tr>
<td>ali.int.ap-south-1</td>
<td>Alibaba Cloud Mumbai</td>
</tr>
<tr>
<td>ali.int.ap-southeast-1</td>
<td>Alibaba Cloud Singapore</td>
</tr>
<tr>
<td>ali.int.ap-southeast-2</td>
<td>Alibaba Cloud Sydney</td>
</tr>
<tr>
<td>ali.int.ap-southeast-3</td>
<td>Alibaba Cloud Kuala Lumpur</td>
</tr>
<tr>
<td>ali.int.ap-southeast-5</td>
<td>Alibaba Cloud Jakarta</td>
</tr>
<tr>
<td>ali.int.cn-hongkong</td>
<td>Alibaba Cloud Hong Kong</td>
</tr>
<tr>
<td>ali.int.eu-central-1</td>
<td>Alibaba Cloud Frankfurt</td>
</tr>
<tr>
<td>ali.int.eu-west-1</td>
<td>Alibaba Cloud London</td>
</tr>
<tr>
<td>ali.int.me-east-1</td>
<td>Alibaba Cloud Dubai</td>
</tr>
<tr>
<td>ali.int.us-east-1</td>
<td>Alibaba Cloud Virginia</td>
</tr>
<tr>
<td>ali.int.us-west-1</td>
<td>Alibaba Cloud Silicon Valley</td>
</tr>
</tbody>
</table>
Manage Prisma Cloud Administrators

Role-based access controls allow you to restrict access to the cloud accounts based on a user’s role in the organization. For example, you can assign groups of accounts to a line of business cloudOps, DevOps, and SecOps owners to restrict their access to Prisma Cloud for the accounts they own.

- Prisma Cloud Administrator Roles
- Create and Manage Account Groups on Prisma Cloud
- Create Prisma Cloud Roles
- Prisma Cloud Administrator Permissions
- Manage Roles in Prisma Cloud
- Add Administrative Users On Prisma Cloud
- Create and Manage Access Keys
- Manage your Prisma Cloud Profile
- Set up SSO Integration on Prisma Cloud
- Define Prisma Cloud Enterprise and Anomaly Settings
Prisma Cloud Administrator Roles

A user on Prisma Cloud is someone who has been assigned administrative privileges, and a role defines the type of access that the administrator has on the service. When you define a role, you specify the permission group and the account groups that the administrator can manage. Prisma Cloud has four types of permission groups built-in for administrators.

- **System Admin**—Full control (read/write permissions) to the service, and they can create, edit, or delete account groups or cloud accounts. Only System administrators have access to all Settings on Prisma Cloud and can view audit logs to analyze actions performed by other users who have been assigned administrative privileges.

  If you use the System Admin role with Only for Compute capabilities enabled, the administrator only has full control (read/write permissions) to the Compute tab and APIs on Prisma Cloud, and does not have access to the rest of Prisma Cloud capabilities.

- **Account Group Admin**—Read/write permissions for the cloud accounts and account groups to which they are granted access.

  An account group administrator can only view resources deployed within the cloud accounts to which they have access. Resources deployed on other cloud accounts that Prisma Cloud monitors are excluded from the search or investigation results.

- **Account Group Read Only**—Read only permissions to view designated sections of Prisma Cloud. This role does not have permissions to modify any settings.

- **Account and Cloud Provisioning Admin**—Combines the permissions for the Account Group Admin and the Cloud Provisioning Admin to enable an administrator who is responsible for a line of business. With this role, in addition to being able to onboard cloud accounts, the administrator can access the dashboard, manage the security policies, investigate issues, view alerts and compliance details for the designated accounts only.

- **Cloud Provisioning Admin**—Permissions to onboard and manage cloud accounts from Prisma Cloud and the APIs, and the ability to create and manage the account groups. With this role access is limited to Settings > Cloud Accounts and Settings > Account Groups on the admin console.

- **Build and Deploy Security**—Restricted permissions to DevOps users who need access to a subset of Compute capabilities and/or API access to run IDE, SCM and CI/CD plugins for Infrastructure as Code and image vulnerabilities scans. For example, the Build and Deploy Security role enables read-only permissions to review vulnerability and compliance scan reports on Compute and to manage and download utilities such as Defender images, plugins and twistcli.

  And if you use the Build and Deploy Security role with Access key only enabled, the administrator can create one access key to use the Prisma Cloud Compute APIs.

  See Prisma Cloud Compute Roles for more details for the roles and associated permissions.

Add Administrative Users On Prisma Cloud. You can View permissions associated with each role on Settings > Roles > +Add New
Create and Manage Account Groups on Prisma Cloud

You can use Account Groups to combine access to multiple cloud accounts with similar or different applications that span multiple divisions or business units, so that you can manage administrative access to these accounts from Prisma Cloud.

When you onboard a cloud account to Prisma Cloud, you can assign the cloud account to one or more account groups, and then assign the account group to Prisma Cloud Administrator Roles. Assigning an account group to an administrative user on Prisma Cloud allows you to restrict access only to the resources and data that pertains to the cloud account(s) within an account group. Alerts on Prisma Cloud are applied at the cloud account group level, which means you can setup separate alert rules and notification flows for different cloud environments.

Create an Account Group

**STEP 1** | Select Settings > Account Groups and click + Add New.

**STEP 2** | Enter a Name and Description for the Account Group.

**STEP 3** | Select the cloud accounts that you want to group together in this account group and click Save.

Manage Account Groups

To view and manage account groups:

**STEP 1** | Select Settings > Account Groups.

**STEP 2** | To edit the details of an Account Group, click the record and change any details.

**STEP 3** | To clone an Account Group, hover over the account group and click Clone.

Cloning an account group is creating a copy of an existing account group. Cloning serves as a quick method of creating a new account group if you choose to change few details of the source account group.

**STEP 4** | To delete an Account Group, hover over the account group and click Delete.
Create Prisma Cloud Roles

Roles on Prisma Cloud enable you to specify what permissions an administrator has, and to which cloud accounts they have access, what policies they can apply, and how they interact with alerts and reports, for example.

When you create a cloud account, you can assign one or more cloud account to account group(s) and then attach the account group to the role you create. This flow allows you to ensure the administrator who has the role can access the information related to only the cloud account(s) to which you have authorized access.

STEP 1 | Select Settings > Roles and click + Add New.

STEP 2 | Enter a name and a description for the role.

STEP 3 | Select a permission group.

See Prisma Cloud Administrator Roles for a description of the following permission groups.

- Select System Admin to allow full access and control over all sections of Prisma Cloud including overall administration settings and permissions management. To limit access to the Compute capabilities and APIs that enable you to secure your host, serverless, and container deployments, select Only for Compute capabilities.

- Select Account Group Admin to allow full access over designated accounts including a subset of administration settings and permissions management for the designated Account Groups you specify. By default an Account Group Admin has the ability to dismiss, snooze, and resolve alerts that are generated against all policies included in an alert rule defined by the System Admin. You can, however, restrict the ability dismiss or resolve alerts.

- Select Account ReadOnly to allow read access to designated accounts across Prisma Cloud administrative console, excluding any administration settings and permissions management.

- Select Account and Cloud Provisioning Admin—to enable an administrator who is responsible for a line of business. With this role the permissions allow you to onboard cloud accounts, and access the dashboard, manage the security policies, investigate issues, view alerts and compliance details for the designated Account Groups you specify. By default an Account and Cloud Provisioning Admin has the ability to dismiss, snooze, and resolve alerts that are generated against all policies included in an alert rule defined by the System Admin. You can, however, restrict the ability dismiss or resolve alerts.

- Select Cloud Provisioning Admin to onboard and manage cloud accounts from the admin console and through APIs. They can also create and manage Account Groups. They do not have access to any other Prisma Cloud features.

- Select Build and Deploy Security to enable DevOps users who need access to a subset of Compute capabilities and/or API access to run IDE, SCM and CI/CD plugins for Infrastructure as Code and image vulnerabilities scans.

STEP 4 | Click View Permissions to see the permissions associated with every permission group. The Permission matrix shows what permissions each permission group has within Prisma Cloud.
STEP 5 | *(Optional)* Restrict the ability dismiss or resolve alerts.

If you would like to ensure that only the System Admin role can manage alerts associated with a policy defined by a system administrator, select **Restrict alert dismissal**. When enabled, an administrator with any other role such as Account Group Admin or Account and Cloud Provisioning Admin roles cannot dismiss or resolve alerts.

STEP 6 | *(Optional)* Enable access to data collected from Prisma Cloud Defenders.

Select **Non-onboarded cloud accounts access**, if you would like to ensure that the roles—Account Group Admin, Account Group Read Only, and Account and Cloud Provisioning Admin—can view data sent from Prisma Cloud Defenders deployed on resources hosted on-premises or on cloud platforms deployed on private or public cloud platforms that are not being monitored by Prisma Cloud, such as on OpenShift, on-prem Kubernetes clusters, or AWS Fargate. When you enable this option, administrators who are assigned the role can view data collected from Defenders on the **Compute** tab.

STEP 7 | Select Account Groups that you want to associate with this role and click **Save**.
# Prisma Cloud Administrator Permissions

The following table provides a list of the access privileges associated with each role for different parts of the Prisma Cloud administrative console. For details on permissions for Prisma Cloud Compute roles, see Prisma Cloud Compute roles.

<table>
<thead>
<tr>
<th>Compute Role</th>
<th>Sys Admin</th>
<th>System Admin (Only allow compute access)</th>
<th>Auditor</th>
<th>Defender Manager</th>
<th>Auditor</th>
<th>DevOps</th>
<th>CI</th>
<th>DevSecOps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Account Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
<td>Account Group Read Only</td>
</tr>
<tr>
<td><strong>Dashboard</strong></td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td><strong>Inventory</strong></td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td><strong>Investigate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Queries</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td>Save Searches</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Edit / Delete Saved Search</td>
<td>Yes</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Policies</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Policy</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Create Policy</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Add/ Edit CLI Remediation in Policy</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Auditor</td>
<td>Auditor (Only allow compute access)</td>
<td>Defender Manager</td>
<td>Auditor (Cloud Provisioning Admin)</td>
<td>DevOps</td>
<td>CI</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>--------------------------------------</td>
<td>---------</td>
<td>-----------------------------------</td>
<td>-----------------</td>
<td>---------------------------------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td>Compute Role</td>
<td>Sys Admin</td>
<td>System Admin</td>
<td>Account Group Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
<td>Account Group Read Only</td>
</tr>
<tr>
<td>Edit / Delete / Disable Policy</td>
<td>Yes</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compliance</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td></td>
</tr>
<tr>
<td>Create / Edit Reports</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Download Reports</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td></td>
</tr>
<tr>
<td>Delete Reports</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Create / Edit / Delete Compliance Standards</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View Compliance Standards</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Alerts</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td></td>
</tr>
<tr>
<td>View / Search Alerts</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td></td>
</tr>
<tr>
<td>Dismiss / Resolve / Snooze Alerts</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin (Allow compute access)</td>
<td>Auditor Admin</td>
<td>Defender Manager</td>
<td>Auditor Admin</td>
<td>DevOps Admin</td>
<td>CI Admin</td>
<td>DevSecOps Admin</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------------------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>--------------</td>
<td>-------------</td>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>Compute Role</td>
<td>Sys Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Auditor Group Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
<td>Account Group Read Only</td>
</tr>
<tr>
<td>Save Alert Filter(s)</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Delete Alert Filter(s)</td>
<td>Yes</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>Users in this role</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create Report</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Download Reports</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td>Delete Reports</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View Alert Rules</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td>Create / Edit / Delete / Disable Alert Rules</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View Notification Templates</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Create / Edit / Delete Notification Templates</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin (Only allow compute access)</td>
<td>Auditor Manager</td>
<td>Defender Manager</td>
<td>Auditor Manager</td>
<td>DevOps Administrator</td>
<td>CI Administrator</td>
<td>DevSecOps User</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>------------------------------------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>---------------------</td>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Compute</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes - Auditor</td>
<td>Yes - Defender Manager</td>
<td>Yes - Auditor</td>
<td>No</td>
<td>Yes - DevOps</td>
<td>Yes - DevSecOps User</td>
</tr>
<tr>
<td>Radar</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes read-only access to data relevant to the account in account group</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Defend</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes read-only access to all data</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Monitor</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes read-only access to data relevant to account in</td>
<td>Monitor Vulnerabilities/Compliance but only CI tab under Images/Functions</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Compute Role</td>
<td>Sys Admin</td>
<td>System Admin (Only allow compute access)</td>
<td>Auditor</td>
<td>Defender Manager</td>
<td>Auditor</td>
<td>DevOps</td>
<td>CI</td>
<td>DevSecOps</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
<td>--------</td>
<td>------------------</td>
<td>--------</td>
<td>--------</td>
<td>----</td>
<td>-----------</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Account Group Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
<td>Account Group Read Only</td>
</tr>
<tr>
<td>Manage</td>
<td>Yes</td>
<td>Yes</td>
<td>View All Logs, Defenders - Manage deployed to account group, Alerts - View, Collections and Tags - Read Only, Authentication - Read Only, System - Downloads - Jenkins Plugin and twistcli</td>
<td>View All Logs, Defenders - Manage current defenders and deploy new ones, Authentication - view user certificates, System - Access all downloads</td>
<td>View All Logs, Defenders - Manage deployed to account group, Alerts - View, Collections and Tags - Read Only, Authentication - Read Only, System - Downloads - Jenkins Plugin and twistcli, path to console</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Settings**

<table>
<thead>
<tr>
<th>View Accounts</th>
<th>All accounts</th>
<th>No</th>
<th>Designated accounts</th>
<th>Designated accounts</th>
<th>Designated No accounts</th>
<th>No</th>
<th>Designated accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Account Details</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create / Edit / Delete /</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compute Role</td>
<td>Sys Admin</td>
<td>System Admin (Only allow compute access)</td>
<td>Auditor</td>
<td>Defender Manager</td>
<td>Auditor</td>
<td>DevOps</td>
<td>CI</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------------------------------</td>
<td>---------</td>
<td>------------------</td>
<td>---------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Account Group Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
</tr>
<tr>
<td>Disable Accounts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Account Groups</td>
<td>All accounts</td>
<td>No</td>
<td>Designated accounts</td>
<td>No</td>
<td>Designated No accounts</td>
<td>No</td>
<td>Designated accounts</td>
</tr>
<tr>
<td>Create / Edit / Delete Account Groups</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create / View / Edit / Delete User Roles</td>
<td>Yes</td>
<td>Read-only access</td>
<td>Read-only access</td>
<td>Read-only access</td>
<td>Read-only access</td>
<td>Read-only access</td>
<td>Read-only access</td>
</tr>
<tr>
<td>Create / View / Edit / Delete / Disable Users</td>
<td>Yes</td>
<td>Read-only access to view the roles assigned for self</td>
<td>Read-only access to view the roles assigned for self</td>
<td>Read-only access to view the roles assigned for self</td>
<td>Read-only access to view the roles assigned for self</td>
<td>Read-only access to view the roles assigned for self</td>
<td>Read-only access to view the roles assigned for self</td>
</tr>
<tr>
<td>Add/Activate/Deactivate/Delete Access Keys</td>
<td>Yes; Can manage access keys for other roles also.</td>
<td>Yes</td>
<td>Can manage access keys for self</td>
<td>Yes</td>
<td>Can manage access keys for self</td>
<td>Yes</td>
<td>Can manage access keys for self</td>
</tr>
<tr>
<td>View / Edit SSO Settings</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compute Role</td>
<td>Sys Admin</td>
<td>System Admin (Only allow compute access)</td>
<td>Auditor</td>
<td>Defender Manager</td>
<td>Auditor</td>
<td>DevOps</td>
<td>CI</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------------------------------------------</td>
<td>---------</td>
<td>-----------------</td>
<td>---------</td>
<td>--------</td>
<td>---</td>
</tr>
<tr>
<td>Prisma Cloud Role</td>
<td>System Admin</td>
<td>System Admin with Compute Access Only</td>
<td>Account Group Admin</td>
<td>Cloud Provisioning Admin</td>
<td>Account and Cloud Provisioning Admin</td>
<td>Build and Deploy Security</td>
<td>Build and Deploy Security</td>
</tr>
<tr>
<td>Create / View / Edit / Delete / Disable Integrations</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View/ Edit Trusted IP Addresses</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View Licensing Info</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View Prisma Cloud Audit Logs</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View/ Edit AnomalySettings</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>View/ Edit Enterprise Settings</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Manage Roles in Prisma Cloud

Use roles to define the permissions for a specific account group.

**STEP 1** | To view roles, select **Settings > Roles**.

**STEP 2** | To edit the details of a role, click the record and change any details.

**STEP 3** | To clone a role, hover over the role and click **Clone**.

Cloning a role is creating a copy of an existing role and then updating it to meet your requirements quickly. Only the System Admin role can clone a role.

**STEP 4** | To delete a role, hover over the role and click **Delete**.
Add Administrative Users On Prisma Cloud

To provide administrative access to Prisma Cloud—admin and API—you must add users locally on Prisma Cloud. You can choose whether you want these administrators to use Palo Alto Networks Customer Support Portal (CSP) credentials to log in or SSO using a third-party Identity Service Provider.

If you want to use Palo Alto Networks Customer Support Portal (CSP) credentials, when you add the email address for a user who already has a support account with Palo Alto Networks, they can just log in to Prisma Cloud using the Prisma Cloud URL or from the Prisma Cloud tile on hub. If you Set up SSO Integration on Prisma Cloud with an Identity Service Provider who supports SAML, you can configure Just-in-Time Provisioning (JIT) to create a local account on the fly, instead of creating the account in advance on Prisma Cloud. With JIT, you do not need to manually create a local user account.

The following instructions are for manually adding a local user account on Prisma Cloud.

**STEP 1** | Select **Settings > Users** and click + Add New.

**STEP 2** | Enter **First Name**, **Last Name**, and **Email** of the user.

For a user who has a Palo Alto Networks CSP account, you must enter the email address that is associated with the CSP account so that they can log in as soon as you save the changes. If the user does not have a CSP account, as soon as you add them here and save your changes, they will receive two emails. One email to activate the CSP account and another email with a link to get started with Prisma Cloud.

**STEP 3** | **Assign Roles** to the user.

You can assign up to five roles to a user, and must select one as the **Default Role**. See Prisma Cloud Administrator Roles for the different permission groups and associated permissions. Users with multiple roles can use the Profile to switch between roles. The default role is marked with a star.

The role assumed by the user is tied to policies, saved searches, saved alert filters, and recurring compliance reports that do not have a cloud account selected. These objects are available to any other user who has the same role, and it is not tied to the specific user.

**STEP 4** | Specify a **Time Zone** for the user and click **Save**.

**STEP 5** | Decide whether to **Allow user to create API Access Keys**.

By default, API access is enabled for the System Admin role only. When you add a new administrator, decide whether or not you want to enable API access for the other roles; the key icon in the API Access
column indicates that the administrator has API access, and can create up to two access keys per role on Prisma Cloud. See Create and Manage Access Keys for more information.

STEP 6 | After you add an administrator, you can edit or delete the user or modify permissions to add additional roles.

When you delete an administrator or modify the role, all the access keys associated with the user and role are deleted immediately.

- To edit the details of an user, click the record and change the details.
- To disable an user, toggle the Status of the user.
- To delete an user, hover over the user and click Delete.

STEP 7 | Change the password for an administrative user.

If you want to set a new password to periodically change it or if you are unable to log in because you forgot your password. As a security measure, if you enter an incorrect password five times, your account is locked and you must reset your password.

1. Access the URL for your Prisma Cloud instance.
2. Click the Forgot password link.

You will receive an email at the registered email address (Step 2 above). Use the link in the email to set a new password.
Create and Manage Access Keys

Access Keys are a secure way to enable programmatic access to the Prisma Cloud API, if you are setting up an external integration or automation. By default, only the System Admin has API access and can enable API access for other administrators.

You can enable API access either when you Add Administrative Users On Prisma Cloud, you can modify the user permissions to enable API access. If you have API access, you can create up to two access keys per role for most roles; some roles such the Build and Deploy Security role can generate one access key only. When you create an access key, the key is tied to the role with which you logged in.

Create an access key for a limited time period and regenerate your API keys periodically to minimize exposure and follow security best practices. On the Settings > Audit Logs, you can view a record of all access key related activities such as an update to extend its validity, deletion, or a revocation.

Watch this!

**STEP 1 | Select Settings > Access Keys > + Add New**
If you do not see the option to add a new key, it means that you do not have the permissions to create access keys.

**STEP 2 | Enter a descriptive Name for the key.**

**STEP 3 | Set the Key Expiry term.**

Select the checkbox and specify a term—date and time for the key validity—that adheres to your corporate compliance standards. If you do not select key expiry, the key is set to never expire; if you select it, but do not specify a date, the key expires in a month. In the event a key is compromised, you can administratively disable (Make Inactive) the key.

**STEP 4 | Create the key.**
If you have multiple roles, you must switch roles to create an access key for each role.
Copy or download the Access Key ID and the Secret Key as a CSV file. After you close the window, you cannot view the secret key again, and must delete the existing key and create a new key.

**STEP 5 | View the details for your keys.**
You can verify the expiry date for the key and can update it here, review when it was last used and the status —Active or Expired.

If you have multiple roles, the access key details display only for the role with which you are logged in.
Manage your Prisma Cloud Profile

Manage your Prisma Cloud profile.

STEP 1 | To view your profile information, click the icon on the Right hand top corner.

STEP 2 | Edit your Name, Last Name, or Time zone and click Save.
Set up SSO Integration on Prisma Cloud

On Prisma Cloud, you can enable single sign-on (SSO) using an Identity Provider (IdP) that supports Security Assertion Markup Language (SAML), such as Okta, Azure Active Directory, or PingID. You can configure only one IdP for all the cloud accounts that Prisma Cloud monitors.

To access Prisma Cloud using SSO, every administrative user requires a local account on Prisma Cloud. You can either Add Administrative Users On Prisma Cloud to create the local account in advance of enabling SSO, or use Just-In-Time (JIT) Provisioning on the SSO configuration on Prisma Cloud if you prefer to create the local account automatically. With JIT Provisioning, the first time a user logs in and successfully authenticates with your SSO IdP, the SAML assertions are used to create a local user account on Prisma Cloud.

To enable SSO, you must first complete the setup on the IdP. Then, log in to Prisma Cloud using an account with System Admin privileges to configure SSO and redirect login requests to the IdP’s login page, so that your Prisma Cloud administrative users can log in using SSO. After you enable SSO, you must access Prisma Cloud from the IdP's portal. Prisma Cloud supports IdP initiated SSO, and it’s SAML endpoint supports the POST method only.

As a best practice, enable a couple administrative users with both local authentication credentials on Prisma Cloud and SSO access so that they can log in to the administrative console and modify the SSO configuration when needed, without risk of account lockout. Make sure that each administrator has activated their Palo Alto Networks Customer Support Portal (CSP) account using the Welcome to Palo Alto Networks Support email and set a password to access the portal.

Also, any administrator who needs to access the Prisma Cloud API cannot use SSO and must authenticate directly to Prisma Cloud using the email address and password registered with Prisma Cloud.

**STEP 1** | Decide whether you want to first add Add Administrative Users On Prisma Cloud or you prefer to add users on the fly with JIT Provisioning when you Configure SSO on Prisma Cloud.

If you want to enable JIT provisioning for users, Create Prisma Cloud Roles before you continue to the next step. When you configure SSO on the IdP, you must attach this role to the user’s profile so that the user has the appropriate permissions and can monitor the assigned cloud accounts on Prisma Cloud.

**STEP 2** | Copy the Audience URI, for Prisma Cloud, which users need to access from the IdP.

1. Log in to Prisma Cloud and select **Settings > SSO**.
2. Copy the **Audience URI (SP Entity ID)** value. This is a read-only field in the format: https://app.prismacloud.io?customer=<string> to uniquely identify your instance of Prisma Cloud. You require this value when you configure SAML on your IdP.

**STEP 3** | Set up the Identity Provider for SSO.

1. This workflow uses Okta as the IdP. Before you begin to set up Okta configuration, login to your Prisma Cloud instance and copy the Audience URI (SP Entity ID) from Prisma Cloud. See For example: https://app.prismacloud.io/settings/sso.
2. Login to Okta as an Administrator and click **Admin**.
3. Click **Add Applications**.

4. Search for **Prisma Cloud** and **Add**.

5. On **Create a New Application Integration**, select **Web** for Platform and **SAML 2.0** for Sign on method.
6. Click Create.
7. On General Settings, use these values and click Next.
   - **App Name** - Prisma Cloud SSO app
   - **App Logo** - Use the Prisma Cloud logo
   - **App Visibility** - Do not check these options

8. To **Configure SAML**, specify the **Sign On URL**.
   The format for Sign On URL uses the URL for Prisma Cloud, but you must replace app with api and add saml at the end. For example, if you access Prisma Cloud at https://app2.prismacloud.io,
your Sign On URL should be https://api2.prismacloud.io/saml and if it is https://app.eu.prismacloud.io, it should be https://api.eu.prismacloud.io/saml.

9. For Audience URI - Use the value displayed on Prisma Cloud Settings > SSO that you copied in the first step.

10. Select Name ID format as EmailAddress and Application username as Email.

11. For Advanced Section, select Response as Unsigned, Assertion Signature as Signed, Assertion Encryption as UnEncrypted.
12. Assign users who can use the Prisma Cloud SSO app to log in to Prisma Cloud.

13. (Required only for JIT provisioning of a local user account automatically on Prisma Cloud) Specify the attributes to send with the SAML assertion.
   For more details, see Set up Just-in-Time Provisioning on Okta.

14. (Required only for JIT provisioning of a local user account automatically on Prisma Cloud) Assign the role you created on Prisma Cloud to the user profile.
   You have now successfully created an application for the SAML integration. This application will have the details of the **IdP URL** and **Certificate** which you’ll need to add on Prisma Cloud to complete the SSO integration.

**STEP 4 | Configure SSO on Prisma Cloud.**
1. Log in to Prisma Cloud and select **Settings > SSO**.
2. **Enable SSO**.
3. Enter the value for your **Identity Provider Issuer**.
This is the URL of a trusted provider such as Google, Salesforce, Okta, or Ping who act as your IdP in the authentication flow. On Okta, for example, you can find the Identity Provider issuer URL at Applications > Sign On > View Setup Instructions.

In the setup instructions, you have Identity Provider Issuer and Prisma Cloud Access SAML URL.

How to Configure SAML 2.0 for RedLock Application

The following is needed to configure RedLock:

1. Identity Provider Single Sign-On URL: RedLock Access SAML URL
   - https://www.example.com/identity-provider/app/idp.saml

2. Identity Provider Issuer: Identity Provider Issuer
   - http://www.okta.com

4. Enter the Identity Provider Logout URL to which a user is redirected to, when Prisma Cloud times out or when the user logs out.

5. Enter your IdP Certificate in the standard X.509 format.
   - You must copy and paste this from your IdP.
6. Enter the Prisma Cloud Access SAML URL configured in your IdP settings.

For example, on Okta this is the Identity Provider Single Sign-On URL. When you click this URL, after authentication with your IdP, you are redirected to Prisma Cloud. This link along with the Relay State Parameter is used for all the redirection links embedded in notifications like email, slack, SQS, and compliance reports.

7. **Relay State Param name** is SAML specific Relay State parameter name. If you provide this parameter along with Prisma Cloud Access SAML URL, all notification links in Splunk, Slack, SQS, email, and reports can link directly to the Prisma Cloud application. The relay state parameter or value is specific to your Identity Provider. For example, this value is RelayState for Okta.

   *When using RelayState functionality, make sure your Prisma Cloud Access SAML URL corresponds to Identity Provider Single Sign-On URL ending in `/sso/saml`.*

8. (Optional) Clear the **Enforce DNS Resolution for Prisma Cloud Access SAML URL**.

   By default, Prisma Cloud performs a DNS look up to resolve the Prisma Cloud SAML Access URL you entered earlier. If your IdP is on your internal network, and you do not need to perform a DNS look up, you can clear this option to bypass the DNS lookup.


   **Enable JIT Provisioning,** if you want to create a local account for users who are authenticated by the IdP. With JIT, the user is provisioned with the first five roles mapped to the user’s profile on the IdP.

10. Provide the user attributes in the SAML assertion or claim that Prisma Cloud can use to create the local user account.

    You must provide the email, role, first name, and last name for each user. Timezone is optional.
The role that you specify for the user’s profile on the IdP must match what you created on Prisma Cloud in Step 1.

11. Select **Allow select users to authenticate directly with Prisma Cloud** to configure some users to access Prisma Cloud directly using their email address and password registered with Prisma Cloud, in addition to logging in via the SSO provider.

When you enable SSO, make sure to select a few users who can also access Prisma Cloud directly using the email and password that is registered locally on Prisma Cloud to ensure that you are not locked out of the console in the event you have misconfigured SSO and need to modify the IdP settings. For accessing data through APIs, you need to authenticate directly to Prisma Cloud.

12. Select the **Users** who can access Prisma Cloud either using local authentication credentials on Prisma Cloud or using SSO.

The users listed in the allow list can log in using SSO and also using a local account username and password that you have created on Prisma Cloud.

13. **Save** your changes.

14. **Verify access using SSO.**

Administrative users for whom you have enabled SSO, must access Prisma Cloud from the Identity Provider’s portal. For example, if you have integrated Prisma Cloud with Okta, administrative users must login to Okta and then click on the Prisma Cloud app icon to be logged in to Prisma Cloud.

15. **Using View last SSO login failures**, you can see details of last five login issues or errors for SSO authentication for any users.
If the user is logged in already using a username/password and then logs in using SSO, the authentication token in the browser's local storage is replaced with the latest token.

Set up Just-in-Time Provisioning on Okta

To successfully set up local administrators on the fly with Just-in-Time (JIT) provisioning, you need to configure the Prisma Cloud app for Okta to provide the SAML claims or assertions that enable Prisma Cloud to add the authenticated SSO user on Prisma Cloud. Then, to ensure that the SSO user has the correct access privileges on Prisma Cloud, you need to assign a Prisma Cloud role to the user; if this role is not a default role on Prisma Cloud, you must define the role before you assign the role to the user on Okta.

**STEP 1** | Create the Prisma Cloud App for Okta.

**STEP 2** | For JIT, create a custom attribute on the Prisma Cloud Okta app.

1. Go to Directory > Profile Editor > Apps.
2. Find the Prisma Cloud app and select Profile, and Add Attribute.
Enter a display name, a variable name, and an attribute length that is long enough to accommodate the role names on Prisma Cloud.

**STEP 3** | Configure the **Attribute Statements** on the Prisma Cloud Okta app.
Specify the user attributes in the SAML assertion or claim that Prisma Cloud can use to create the local user account.

1. Select Applications > Applications
2. Select the Prisma Cloud app, **General** and edit the **SAML Settings** to add the attribute statements.

   You must provide the email, role, first name, and last name for each user. Timezone is optional.

   ![Attribute Statements](image)

**STEP 4** | Assign the Prisma Cloud role for each SSO user.
Each SSO user who is granted access to Prisma Cloud, can have between one to five Prisma Cloud roles assigned. Each role determines the permissions and account groups that the user can access on Prisma Cloud.

1. Select Applications > Applications
2. Select the Prisma Cloud app and Assignments.
For existing users, click the pencil icon to add the Prisma Cloud Role you want to give this user. For example, System Admin.

For new users, select **Assign > Assign to People**, click **Assign** for the user you want to give access to Prisma Cloud and define the Prisma Cloud Role you want to give this user.

**STEP 5 | Continue with 4.**
View Audit Logs

As part of compliance requirement for organizations, companies need to demonstrate they are pro-actively tracking security issues and taking steps to remediate issues as they occur. Prisma Cloud Audit Logs section enables companies to prepare for such audits and demonstrates compliance. The Audit logs section lists out the actions performed by the users in the system.

**STEP 1** | Select a **Time Range** to view the activity details by users in the system.

**STEP 2** | Select the columns in the table and **Download** all administrator activity.

The user activity details are in a CSV format.
Define Prisma Cloud Enterprise and Anomaly Settings

Set the enterprise level settings to build standard training models for anomaly detection, alert disposition, and some other global settings such as the timeout before the user is looked out for inactivity, and user attribution for alerts.

- Set Up Inactivity Timeout
- Set Up Global Settings for Policy and Alerts
- Set Up Anomaly Policy Thresholds

Set Up Inactivity Timeout

Specify a timeout period after which an inactive administrative user will be automatically logged out of Prisma Cloud. An inactive user is one who does not interact with the UI using their keyboard and mouse within the specified time period.

STEP 1 | Select Settings > Enterprise Settings.

STEP 2 | User Idle Timeout

If you modify the timeout period, the new value is in effect for all administrative users who log in after you make the change; the previous timeout applies for all currently logged in users.

Set Up Global Settings for Policy and Alerts

These settings apply to all Prisma Cloud policies. For Anomaly policies, you have more customizable settings, see Set Up Anomaly Policy Thresholds.

- Auto enable new default policies of the type.
  1. Select Settings > Enterprise Settings.
  2. Granularly enable new Default policies of severity High, Medium or Low.

While some high severity policies are enabled to provide the best security outcomes, by default, policies of medium or low severity are in a disabled state. When you select the checkbox to auto enable policies of a specific severity, you can either retroactively enable all policies that match the severity or only enable policies that are added to Prisma Cloud going forward.

If you enable policies of a specific severity, when you then clear the checkbox, the policies that were enabled previously are not disabled; going forward, policies that match the severity you cleared are no longer enabled to scan your cloud resources and generate alerts.

If you want to disable the policies that are currently active, you must disable the status of each policy on the Policies page.
• Enable **Make Alert Dismissal Note Mandatory**, to mandate the users to dismiss alerts only after specifying a reason.

• **Enable Populate User Attribution in Alerts Notifications**

  User attribution data provides you with context on who created or modified the resource that triggered the alert. Select this option to make sure that the alerts include user attribution data in the alert payload, so that it is sent as part of the JSON data to notification channels such as SQS or Splunk. Enabling this option can result in a delay of up to two hours in the generation of alerts because the relevant user information may not be instantly available from the cloud provider.

**Set Up Anomaly Policy Thresholds**

Prisma Cloud allows you to define different thresholds for anomaly detection for Unusual Entity Behavior Analysis (UEBA) that correspond to policies which analyze audit events, and for unusual network activity that correspond to policies which analyze network flow logs. You can also define your preference for when you want to alert notifications based on the severity assigned to the anomaly policy.

• For UEBA policies:
  1. Select **Settings > Anomaly Settings > Alerts and Thresholds**.
  2. Select a policy.
  3. Define the **Training Model Threshold**.

    The Training Model Threshold informs Prisma Cloud on the values to use for setting the baseline for the machine learning (ML) models.

    *For production environments, set the Training Model Threshold to High so that you allow for more time and have more data to analyze for determining the baseline.*

    **For unusual user activity:**
    1. Low: The behavioral models are based on observing at least 25 events over 7 days.
    2. Medium: The behavioral models are based on observing at least 100 events over 30 days.
    3. High: The behavioral models are based on observing at least 300 events over 90 days.

    **For account hijacking:**
    1. Low: The behavioral models are based on observing at least 10 events over 7 days.
    2. Medium: The behavioral models are based on observing at least 25 events over 15 days.
    3. High: The behavioral models are based on observing at least 50 events over 30 days.

  4. Define your **Alert Disposition**.
Alert Disposition is your preference on when you want to be notified of an alert, based on the severity of the issue —low, medium, high. The alert severity is based on the severity associated with the policy that triggers an alert.

You can profile every activity by location or user activity. The activity-based anomalies identify any activities which have not been consistently performed in the past. The location based anomalies identify locations from which activities have not been performed in the past.

Choose the disposition (in some cases you may only have two to choose from):

1. **Conservative:**
   - For unusual user activity—Report on unknown location and service to classify an anomaly.
   - For account hijacking—Reports on location and activity to login under travel conditions that are not possible, such as logging in from India and US within 8 hours.

2. **Moderate:**
   - For unusual user activity—Report on unknown location, or both unknown location and service to classify an anomaly.

3. **Aggressive:**
   - For unusual user activity—Report on either unknown location or service to classify an anomaly.
   - For account hijacking—Report on unknown browser and Operating System, or impossible time travel.

**Set the Alert Disposition to Conservative to reduce false positives.**

When you change Training Model Threshold or Alert Disposition the existing alerts are resolved and new ones are regenerated based on the new setting. It might take a while for the new anomaly alerts to show on the Alerts page.

- For unusual network activity.

   For anomalies policies that help you detect network incidents, such as unusual protocols or port used to access a server on your network, you can customize the following for each policy.

   1. Select **Settings > Anomaly Settings > Alerts and Thresholds**.
   2. Select a policy.

   ![Policy settings](image)

   3. Define the **Training Model Threshold**.
The Training Model Threshold informs Prisma Cloud on the values to use for various parameters such as number of days and packets for creating the ML models. These thresholds are available only for the policies that require model building such as Unusual server port activity and Spambot activity.

1. Low: The behavioral models are based on observing at least 10K packets over 7 days.
2. Medium: The behavioral models are based on observing at least 100k packets over 14 days.
3. High: The behavioral models are based on observing at least 1M packets over 28 days.

4. Define your **Alert Disposition**.

Alert Disposition is your preference on when you want to be notified of an alert, based on the severity of the issue —low, medium, high. The alert severity is based on the severity associated with the policy that triggers an alert. You can choose from three dispositions based on the number of ports, hosts or the volume of traffic generated to a port or host on a resource:

1. **Aggressive**: Reports High, Medium, and Low severity alerts.
   - For example, a Spambot policy that sees 250MB traffic to a resource, or a port sweep policy that scans 10 hosts.

2. **Moderate**: Reports High and Medium severity alerts.
   - For example, a Spambot policy that sees 500MB traffic to a resource, or a port sweep policy that scans 25 hosts.

3. **Conservative**: Report on High severity alerts only.
   - For example, a Spambot policy that sees 1GB traffic to a resource, or a port sweep policy that scans 40 hosts.
Manage Prisma Cloud Alerts

- Prisma Cloud Alerts and Notifications
- Trusted IP Addresses on Prisma Cloud
- Enable Prisma Cloud Alerts
- Create an Alert Rule
- Configure Prisma Cloud to Automatically RemEDIATE Alerts
- Send Prisma Cloud Alert Notifications to Third-Party Tools
- View and Respond to Prisma Cloud Alerts
- Generate Reports on Prisma Cloud Alerts
- Alert Payload
- Prisma Cloud Risk Ratings
Prisma Cloud Alerts and Notifications

Prisma™ Cloud continually monitors all of your cloud environments to detect misconfigurations (such as exposed cloud storage instances), advanced network threats (such as cryptojacking and data exfiltration), potentially compromised accounts (such as stolen access keys), and vulnerable hosts. Prisma Cloud then correlates configuration data with user behavior and network traffic to provide context around misconfigurations and threats in the form of actionable alerts.

Although Prisma Cloud begins monitoring and correlating data as soon as you onboard the cloud account, there are tasks you need to perform before you see alerts generated by policy violations in your cloud environments. The first task to Enable Prisma Cloud Alerts is to add the cloud account to an account group during onboarding. Next, create an alert rule that associates all of the cloud accounts in an account group with the set of policies for which you want Prisma Cloud to generate alerts. You can view the alerts for all of your cloud environments directly from Prisma Cloud and drill down in to each to view specific policy violations. If you have internal networks that you want to exclude from being flagged in an alert, you can Trusted IP Addresses on Prisma Cloud.

In addition, Prisma Cloud provides out-of-box ability to Configure External Integrations on Prisma Cloud with third-party technologies, such as SIEM platforms, ticketing systems, messaging systems, and automation frameworks so that you can continue using your existing operational, escalation, and notification tools. To monitor your cloud infrastructures more efficiently and provide visibility in to actionable events across all your cloud workloads, you can also:

- **Generate Reports on Prisma Cloud Alerts**—on-demand or scheduled reports—on open alerts and email them to your stakeholders.
- **Send the Alert Payload** to a third-party tool.
Trusted IP Addresses on Prisma Cloud

Prisma™ Cloud enables you to specify IP addresses or CIDR ranges for:

- **Trusted Login IP Addresses**—Restrict access to the Prisma Cloud administrator console and API to only the specified source IP addresses.
- **Trusted Alert IP Addresses**—If you have internal networks that connect to your public cloud infrastructure, you can add these IP address ranges (or CIDR blocks) as trusted on Prisma Cloud. When you add IP addresses to this list, you can create a label to identify your internal networks that are not in the private IP address space to make alert analysis easier. When you visualize network traffic on the Prisma Cloud Investigate tab, instead of flagging your internal IP addresses as internet or external IP addresses, the service can identify these networks with the labels you provide.

Prisma Cloud default network policies that look for internet exposed instances also do not generate alerts when the source IP address is included in the trusted IP address list and the account hijacking anomaly policy filters out activities from known IP addresses. Also, when you use RQL to query network traffic, you can filter out traffic from known networks that are included in the trusted IP address list.

- **Anomaly Trusted List**—Exclude trusted IP addresses when conducting tests for PCI compliance or penetration testing on your network. Any addresses included in this list do not generate alerts against the Prisma Cloud Anomaly Policies that detect unusual network activity such as the policies that detect internal port scan and port sweep activity, which are enabled by default.

To add an IP address to the trusted list:

- **Add an Alert IP address.**
  1. Select Settings > Trusted Alert IP Addresses > + Add New
     - You must have the System Administrator role on Prisma Cloud to view or edit the Trusted IP Addresses page. See Prisma Cloud Administrator Permissions.
  2. Enter a name or label for the Network.
  3. Enter the CIDR and, optionally, add a Description and then click Done.
     - Enter the CIDR block for IP addresses that are routable through the public internet, you cannot add a private CIDR block. The IP addresses you enter may take up to 15 minutes to take effect.
     - The trusted IP addresses are appropriately classified when you run a network query.

- **Add a Login IP address.**
  1. Select Settings > Trusted Login IP Addresses > + Add New.
     - You must have the System Administrator role on Prisma Cloud to view or edit the Trusted IP Addresses page. See Prisma Cloud Administrator Permissions.
  2. Enter a Name and, optionally a Description.
  3. Enter the CIDR and Create the new login IP address entry.
  4. Verify that the IP address you are logged in with is included in the list.
     - If you are logged in from an IP address that is not listed as a trusted IP address, you will be logged out as soon as you save your changes and can no longer access the Prisma Cloud administrator console and API interface.
  5. Enable the IP address.

- **Add an IP Address to the Anomaly Trusted List.**
  1. Select Settings > Anomaly Settings
You must have the correct role, such as the System Administrator role, on Prisma Cloud to view or edit the Anomaly Settings page. See Prisma Cloud Administrator Permissions for the roles that have access.

2. **Add New > IP Address.**
3. Enter a **Trusted List Name** and, optionally a **Description**.

4. Select the **Anomaly Policies** for which you do not want to generate alerts.

5. Enter the **IP Addresses**.
   You can enter one or more IP addresses in the CIDR format. By default, the IP addresses you add to the trusted list are excluded from generating alerts against any (all) cloud accounts that are onboarded to Prisma Cloud.

6. **(Optional) Toggle Hide Advanced Settings** to select an **Account ID** and **VPC ID**.
   You can select only one Account and VPC ID, or set it to Any to exclude any account that is added to Prisma Cloud.

7. **Save** the list.
   When you save the list, for the selected anomaly policies that detect network issues such as network reconnaissance, network evasion, or resource misuse, Prisma Cloud will not generate alerts for the IP addresses included in this list.

   Only the administrator who created the list can modify the name, description, **Account ID** and **VPC ID**; Other administrators with the correct role can add or delete IP address entries on the trusted list.
Enable Prisma Cloud Alerts

Although Prisma™ Cloud begins monitoring your cloud environments as soon as you onboard a cloud account, you must first enable alerting for each cloud account you onboard before you can receive alerts. Prisma Cloud gives you the flexibility to group your cloud accounts into account groups so that you can restrict access to information about specific cloud accounts to only those administrators who need it. Then you must assign each account group to an alert rule that allows you to select a group of policies and designate where you want to display the Prisma Cloud Alerts and Notifications associated with those policies. This enables you to define different alert rules and notification flows for different cloud environments, such as for both a production and a development cloud environment. In addition, you can set up different alert rules for sending specific alerts to your existing SOC visibility tools. For example, you could send one set of alerts to your security information and event management (SIEM) system and another set to Jira for automated ticketing.

**STEP 1 |** Make sure you have associated all onboarded cloud accounts to an account group.

If you did not associate a cloud account with an account group during the onboarding process, do it now so that you can see alerts associated with the account.

1. Click Settings ( ) and then select Cloud Accounts.
2. For each cloud account, verify that there is a value in the Account Groups column.
3. For any cloud account that isn’t yet assigned to an account group, select the cloud account to edit it and select an Account Group to which to add it.

**STEP 2 |** Create an Alert Rule.

Alert rules define what policy violations trigger alerts for cloud accounts within the selected account group and where to send the alert notifications.

**STEP 3 |** Verify that the alert rule you created is triggering alert notifications.

As soon as you save your alert rule, any violation of a policy for which you enabled alerts results in an alert notification on the Alerts page, as well as in any third-party integrations you designated in the alert rule. Make sure you see the alerts you are expecting on the Alerts page as well as in your third-party tools.
Create an Alert Rule

Alert rules enable you to define the policy violations in a selected set of cloud accounts for which you want to trigger alerts. When you create an alert rule, you select the account groups to which the rule applies and the corresponding set of policies for which you want to trigger alerts. You can add more granularity to the rule by excluding some cloud accounts from the selected account groups, by specifying specific regions for which to send alerts, and even by narrowing down the rule to specific cloud resources identified by resource tags. This provides you with flexibility in how you manage alerts and ensures that you can adhere to the administrative boundaries you defined. You can create a single alert rule that alerts on all policy rules or you can define granular alert rules that send very specific sets of alerts for specific cloud accounts, regions, and even resources to specific destinations.

When you create an alert rule, you can Configure Prisma Cloud to Automatically Remediate Alerts, which enables Prisma Cloud to automatically run the CLI command required to remediate the policy violation directly in your cloud environments. Automated remediation is only available for default policies (Config policies only) that are designated as Remediable (✓) on the Policies page.

In addition, if you Configure External Integrations on Prisma Cloud with third-party tools, defining granular alert rules enables you to send only the alerts you need to enhance your existing operational, ticketing, notification, and escalation workflows with the addition of Prisma Cloud alerts on policy violations in all your cloud environments. To see any existing integrations, click Settings ( ) and then select Integrations.

STEP 1 | Select Alerts > Alert Rules and +Add New alert.

STEP 2 | Enter an Alert Rule Name and, optionally, a Description to communicate the purpose of the rule and then click Next.

STEP 3 | Select the Account Groups to which you want this alert rule to apply and then click Next.

1. Toggle View Advanced Settings to see advanced settings for setting a target.
2. Exclude Cloud Accounts from your selected Account Group.
3. Choose your Region.
4. Add Tags to easily manage or identify the type of your resources.
   Tags apply only to Config and Network policies.
5. Click Next.

STEP 4 | (Optional) If you want to add more granularity for which cloud resources trigger alerts for this alert rule, View Advanced Settings and then provide more criteria as needed:

- Exclude Cloud Accounts—If there are some cloud accounts in the selected account groups for which you do not want to trigger alerts, select the accounts from the list.
- Regions—To trigger alerts only for specific regions for the cloud accounts in the selected account group, select one or more regions from the list.
- Resource Tags—To trigger alerts only for specific resources in the selected cloud accounts, enter the Key and Value of the resource tag you created for the resource in your cloud environment.
Tags apply only to **Config** and **Network** policies. When you add multiple resource tags, it uses the boolean logical OR operator.

When you finish defining the target cloud resources, click **Next**.

**STEP 5** | **Select the policies for which you want this alert rule to trigger alerts and, optionally, Configure Prisma Cloud to Automatically Remediate Alerts.**
1. Either **Select All Policies** or select the specific policies for which you want to trigger alerts on this alert rule.

   *If you enable Automated Remediation, the list of policies shows only Remediable ( Remediable ) policies.*

To help you find the specific group of policies for which you want this rule to alert:

- **Filter Results**—Enter a search term to filter the list of policies to those with specific keywords.
- **Column Picker**—Click **Edit** to modify columns display.
- **Sort**—lick the corresponding **Sort** icon.
- **Column Filter**—Click the corresponding column **Filter** to filter on a specific value in a column. For example, to filter on compliance standards related to NIST, click the filter for the Compliance Standard column, select NIST standards, and then **Set that filter**.
2. Click Next.

STEP 6 | (Optional) Send Prisma Cloud Alert Notifications to Third-Party Tools.

By default, all alerts triggered by the alert rule display on the Alerts page. If you Configure External Integrations on Prisma Cloud, you can also send Prisma Cloud alerts triggered by this alert rule to third-party tools. For example, you can Send Alert Notifications to Amazon SQS or Send Alert Notifications to Jira.

In addition, you can configure the alert rule to Send Alert Notifications Through Email.

STEP 7 | (Optional) If you want to delay the alert notifications for Config alerts, you can configure the Prisma Cloud to Trigger notification for Config Alert only after the Alert is Open for a specific number of minutes.

STEP 8 | Save the alert rule.

STEP 9 | To verify that the alert rule triggers the expected alerts, select Alerts > Overview and ensure that you see the alerts that you expect to see there.

If you configured the rule to Send Prisma Cloud Alert Notifications to Third-Party Tools, make sure you also see the alert notifications in those tools.
Configure Prisma Cloud to Automatically Remediate Alerts

If you want Prisma™ Cloud to automatically resolve policy violations, such as misconfigured security groups, you can configure Prisma Cloud for automated remediation. To automatically resolve a policy violation, Prisma Cloud runs the CLI command associated with the policy in the cloud environments where it discovered the violation. On Prisma Cloud, you can enable automated remediation for default policies (Config policies only) that are designated as remediable (indicated by ☑️ in the Remediable column) and for any cloned or custom policies that you add.

To enable automated remediation, identify the set of policies that you want to remediate automatically and verify that Prisma Cloud has the required permissions in the associated cloud environments. Then Create an Alert Rule that enables automated remediation for the set of policies you identified.

Use caution when you enable automated remediation because it requires Prisma Cloud to make changes in your cloud environments that can adversely affect your applications.

STEP 1 | Verify that Prisma Cloud has the required privileges to remediate the policies you plan to configure for automated remediation.

1. To view remediable policies, select Policies and set the filter to Remediable > True.

   If the Remediable column is not displayed on the Policies page, use the Column Picker (□) to display it.
2. Select a policy for which you want to enable remediation and go to the Remediation page.

Review the required privileges in the CLI Command Description to identify which permissions Prisma Cloud requires in the associated cloud environments to be able to remediate violations of the policy.

You can define up to 5 CLI commands in a sequence for a multi-step automatic remediation workflow. Add the commands in the sequence you want them to execute and separate the commands with a semi colon. If any CLI command included in the sequence fails, the execution stops at that point. See list of supported CLI variables.
STEP 2 | Create an Alert Rule or modify an existing alert rule.

STEP 3 | On the Select Policies page, enable Automated Remediation and then Continue to acknowledge the impact of automated remediation on your application.

The list of available policies updates to show only those policies that are remediable (as indicated by in the Remediable column).

*If you are modifying an existing alert rule that includes non-remediable policies, those policies will no longer be included in the rule. When you modify the rule, Prisma Cloud notifies all account administrators who have access to that rule.*
STEP 4 | Finish configuring and Save the new alert rule or Confirm your changes to an existing alert rule.
Send Prisma Cloud Alert Notifications to Third-Party Tools

Alert rules define which policy violations trigger an alert in a selected set of cloud accounts. When you Create an Alert Rule, you can also configure the rule to send the Alert Payload that the rule triggers to one or more third-party tools. For all channels except email, to enable notification of policy violations in your cloud environments in your existing operational workflows, you must Configure External Integrations on Prisma Cloud. You can either set up an integration before you create the alert rule or use the inline link in the alert rule creation process to set up the integration when you need it.

On some integrations, such as Google CSCC, AWS Security Hub, PagerDuty, and ServiceNow, Prisma Cloud can send a state-change notification to resolve an incident when the issue that generated the alert is resolved manually or if the resource was updated in the cloud environment and the service learns that the violation is fixed.

Refer to the following topics to enable an alert notification channel with third-party tools:

- Amazon SQS
- Azure Service Bus Queue
- Email
- Slack
- Splunk
- Jira
- Google SCC
- ServiceNow
- Webhooks
- PagerDuty
- AWS Security Hub
- Microsoft Teams
- Cortex XSOAR

SendAlertNotifications to Amazon SQS

You can send Prisma Cloud alert notifications to Amazon Simple Queue Service (SQS).

**STEP 1 | Integrate Prisma Cloud with Amazon SQS.**

**STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.**

**STEP 3 | On the Set Alert Notification page for the alert rule, select SQS.**

**STEP 4 | Select the SQS Queues to which you want to send alerts triggered by this alert rule.**
STEP 5 | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to Azure Service Bus Queue

You can send Prisma Cloud alert notifications to an Azure Service Bus queue.

STEP 1 | Integrate Prisma Cloud with Azure Service Bus Queue.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

STEP 3 | On the Set Alert Notification page for the alert rule, select Azure Service Bus Queue.

STEP 4 | Select the Azure Service Bus Queue to which you want to send alerts triggered by this alert rule.

STEP 5 | Save the new alert rule or Confirm your changes to an existing alert rule.
Send Alert Notifications Through Email

To send email notifications for alerts triggered by an alert rule, Prisma Cloud provides a default email notification template. You can customize the message in the template using the in-app rich text editor and attach the template to an alert rule. In the alert notification, you can configure Prisma Cloud to send the alert details as an uncompressed CSV file or as a compressed zip file, of 9 MB maximum attachment size.

All email notifications from Prisma Cloud include the domain name to support Domain-based Message Authentication, Reporting & Conformance (DMARC), and the email address used is noreply@prismacloud.paloaltonetworks.com.

STEP 1 | (Optional) Set up a custom message for your email notification template.

Prisma Cloud provides a default email template for your convenience, and you can customize the lead-in message within the body of the email using the rich-text editor.

1. Select Alerts > Notification Templates.
2. Add New notification template, and choose Email template.
3. Enter a Template Name.
4. Enter a Custom Note.

The preview on the right gives you an idea of how your content will look.

5. Save the email notification template.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

STEP 3 | On the Set Alert Notification page for the alert rule, select Email.

STEP 4 | Enter or select the Emails for which to send the alert notifications.

You can include multiple email addresses and can send email notifications to email addresses in your domain and to guests external to your organization.

STEP 5 | (Optional) Select your custom email Template, if you have one.

STEP 6 | Set the Frequency at which to send email notifications.

- Instantly—Sends an email to the recipient list each time the alert rule triggers an alert.
- **Recurring**—You can select the time interval as Daily, Weekly or Monthly. Prisma Cloud sends a single email to the recipient list that lists all alerts triggered by the alert rule on that day, during that week, or the month.

**STEP 7** | Specify whether to include an attachment to the email.

Including an attachment provides a way for you to include information on the alerts generated and the remediation steps required to fix the violating resource. When you select **Attach detailed report**, you can choose whether to **Include remediation instructions** to fix the root cause for the policy that triggered each alert, and opt to send it as a zip file (**Compress attachment(s)**).

Each email can include up to 10 attachments. An attachment in the zip file format can have 60000 rows, while a CSV file can have 900 rows. If the number of alerts exceed the maximum number of attachments, the alerts with the older timestamps are omitted.

**STEP 8** | **Save** the new alert rule or **Confirm** your changes to an existing alert rule.

**STEP 9** | Verify the alert notification emails.

The email alert notification specifies the alert rule, account name, cloud type, policies that were violated, the number of alerts each policy violated, and the affected resources. Click the `<number>` of alerts view the Prisma Cloud **Alerts > Overview** page.
Send Alert Notifications to a Slack Channel

You can send alert notifications associated with an alert rule to a Slack channel.

**STEP 1** | Integrate Prisma Cloud with Slack.

**STEP 2** | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

**STEP 3** | On the Set Alert Notification page for the alert rule, select Slack.

**STEP 4** | Select the Slack Channels to which you want to send alerts triggered by this alert rule.

**STEP 5** | Set the Frequency at which to send email notifications.

- **As it Happens**—Sends a notification to the selected slack channels each time the alert rule triggers an alert.
- **Daily**—Sends a single notification to the selected Slack channels once each day that lists all alerts triggered by the alert rule on that day.
- **Weekly**—Sends a single notification to the selected Slack channels once each week that lists all alerts triggered by the alert rule during that weekly interval.
- **Monthly**—Sends a single notification to the selected Slack channels once each month that lists all alerts triggered by the alert rule monthly interval.

**STEP 6** | Save the new alert rule or Confirm your changes to an existing alert rule.
Send Alert Notifications to Splunk

You can send alert notifications associated with an alert rule to a Splunk event collector.

**STEP 1** | Integrate Prisma Cloud with Splunk.

**STEP 2** | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

**STEP 3** | On the Set Alert Notification page for the alert rule, select Splunk.

**STEP 4** | Select the Splunk Event Collectors to which you want to send alerts triggered by this alert rule.

**STEP 5** | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to Jira

You can configure alert notifications triggered by an alert rule to create Jira tickets.

**STEP 1** | Integrate Prisma Cloud with Jira.

**STEP 2** | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

**STEP 3** | On the Set Alert Notification page for the alert rule, select Jira.
STEP 4 | Select the Jira **Templates** to use for creating tickets based on the alert payload data for alerts that are triggered by this alert rule.

STEP 5 | **Save** the new alert rule or **Confirm** your changes to an existing alert rule.

Send Alert Notifications to Google Cloud SCC

You can send alert notifications to Google Cloud Security Command Center (SCC).

**STEP 1** | Integrate Prisma Cloud with Google Cloud Security Command Center (SCC).

**STEP 2** | Select **Alerts > Alert Rules** and either **Create an Alert Rule** or select an existing rule to edit.

**STEP 3** | On the **Set Alert Notification** page for the alert rule, select **CSCC**.

**STEP 4** | Select the **Google CSCC Integrations** that you want to use to send notifications of alerts triggered by this alert rule.
STEP 5 | **Save** the new alert rule or **Confirm** your changes to an existing alert rule.

**Send Alert Notifications to ServiceNow**

You can send alert notifications to ServiceNow.

**STEP 1** | Integrate Prisma Cloud with ServiceNow.

**STEP 2** | Select **Alerts > Alert Rules** and either **Create an Alert Rule** or select an existing rule to edit.

**STEP 3** | On the **Set Alert Notification** page for the alert rule, select **now**.
STEP 4 | Select the ServiceNow Templates that you want to use to send notifications of alerts triggered by this alert rule.

STEP 5 | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to Webhooks

You can send alert notifications to webhooks.

STEP 1 | Integrate Prisma Cloud with Webhooks.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

STEP 3 | On the Set Alert Notification page for the alert rule, select webhooks.

STEP 4 | Select the webhook Channels that you want to use to send notifications of alerts triggered by this alert rule.

A webhook notification is delivered as soon as the alert is generated.

STEP 5 | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to PagerDuty

You can send alert notifications to PagerDuty.

STEP 1 | Integrate Prisma Cloud with PagerDuty.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.
STEP 3 | On the Set Alert Notification page for the alert rule, select pagerduty.

STEP 4 | Select the Integration Key.

STEP 5 | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to AWS Security Hub

You can send alert notifications to AWS Security Hub.

STEP 1 | Integrate Prisma Cloud with AWS Security Hub.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

STEP 3 | Select your AWS account from AWS Security Hub.

STEP 4 | Save the new alert rule or Confirm your changes to an existing alert rule.

Send Alert Notifications to Microsoft Teams

You can send alert notifications to Microsoft Teams.

STEP 1 | Integrate Prisma Cloud with Microsoft Teams.

STEP 2 | Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

STEP 3 | On the Set Alert Notification page for the alert rule, select Microsoft Teams.
STEP 4 | Select the **Teams** channels that you want to use to send notifications for alerts triggered by this alert rule.

STEP 5 | Set the **Frequency** at which to send POST notifications.

- **As it Happens**—Sends a notification to the selected channels each time the alert rule triggers an alert.
- **Daily**—Sends a single notification to the selected channels once each day that lists all alerts triggered by the alert rule on that day.
- **Weekly**—Sends a single notification to the selected channels once each week that lists all alerts triggered by the alert rule during that weekly interval.
- **Monthly**—Sends a single notification to the selected channels once each month that lists all alerts triggered by the alert rule monthly interval.

STEP 6 | **Save** the new alert rule or **Confirm** your changes to an existing alert rule.

When a policy rule is violated, a message card displays on the Microsoft teams conversation. The message card is formatted with a red (high), yellow (medium), or gray (low) line to indicate the severity of the alert. For example, the following screenshot is a daily notification summary.
Send Alert Notifications to Cortex XSOAR

You can send alert notifications associated with an alert rule to a Demisto instance.

**STEP 1 |** Integrate Prisma Cloud with Cortex XSOAR.

**STEP 2 |** Select Alerts > Alert Rules and either Create an Alert Rule or select an existing rule to edit.

**STEP 3 |** On the Set Alert Notification page for the alert rule, select Demisto.

**STEP 4 |** Select the Demisto instance to which you want to send alerts triggered by this alert rule.

**STEP 5 |** Save the new alert rule or Confirm your changes to an existing alert rule.
View and Respond to Prisma Cloud Alerts

As soon as you Enable Prisma™ Cloud Alerts, Prisma Cloud generates an alert when it detects a violation in a policy that is included in an active alert rule. To secure your cloud environments, you must monitor alerts. You can either monitor alerts from Prisma Cloud or you can Send Prisma Cloud Alert Notifications to Third-Party Tools to ensure that policy violations in your cloud environments are resolved. The status of an alert can be one of the following:

- **Open**—Prisma Cloud identified a policy violation that triggered the alert and the violation is not yet resolved.
- **Resolved**—Alerts automatically transition to Resolved state when the issue that caused the policy violation is resolved. An alert can also change to Resolved state due to a change in the policy or alert rule that triggered the alert. A resolved alert can also transition back to the open state if the issue resurfaces or there is a policy or alert rule change that causes the alert to trigger again.
- **Snoozed**—A Prisma Cloud administrator temporarily dismissed an alert for a specified time period. When the timer expires, the alert automatically changes to an open or Resolved state depending on whether the issue is fixed.
- **Dismissed**—A Prisma Cloud administrator manually dismissed the alert even though the underlying issue was not resolved. You can manually reopen a dismissed alert if needed.

*If you manually dismiss an alert for a Network policy rule violation, Prisma Cloud automatically reopens the alert when it detects the same violation again.*

- **View alerts from within Prisma Cloud.**

Prisma Cloud displays all alerts for which your role gives you permission to see. Click Alerts to sort and filter the alerts as follows:

- To modify **Edit** and add or remove columns.
- To sort on a specific column, click the corresponding **Sort** icon.
- To filter on specific alert criteria, click the corresponding column **Filter** icon to filter on a specific value in a column. You can also clear filters or save a filter for future use.
- To modify **Add** and then either enter search term to **Filter Results** or add additional filters. You can use the following filters—Account Group, Alert ID, Alert Rule Name, Alert Status, Cloud Account, Cloud Region, Cloud Service, Cloud Type, Compliance Requirement, Compliance Section, Compliance Standard, Policy Label, Policy Name, Remediable, Resource ID, Resource Name, Resource Type, and Risk Grade.

*The filters act as a union operator to combine the results from multiple selections.*
- Download the filtered list of alert details to a CSV file.

- Resolve alerts.

Prisma Cloud generates an alert each time that it finds policy violations in one or more of the account groups that are associated with an alert rule. You can monitor alerts in the cloud accounts for which you are responsible to see any security risks you have and to ensure that any critical issues get resolved. You can Send Prisma Cloud Alert Notifications to Third-Party Tools and remediate using those tools, you can Configure Prisma Cloud to Automatically Remediate Alerts, or you can manually resolve the issues. By reviewing these alerts, you can also decide whether you need to make a change to a policy or alert rule. Depending on the policy type that triggered the alert, you can go directly from the alert to the cloud resource where the violation occurred or you can resolve the issue from the Prisma Cloud Alerts page:

1. Filter the alerts to show only Open alerts that are Remediable.

2. Select the policy for which you want to resolve alerts.
   
   Review the recommendations for resolving the policy rule violation. You can also click the policy name to go directly to the policy.

3. Select the individual alerts you want Prisma Cloud to resolve and then click Resolve.
Pivot from an alert into the cloud resource that triggered the alert to manually resolve the issue.

Prisma Cloud allows you to pivot directly from an alert to view the violating cloud resource and resolve the issue manually.

1. Filter the alert list to show alerts with Alert Status Open and select the Policy Type. For example, Network or Config.
2. Select the policy for which you want to resolve alerts.
   Review the recommendations for resolving the policy violation.
3. Click Resource (🔗) to pivot to the cloud resource containing the violation you want to resolve and follow the recommended steps.

When you click Resource, Prisma Cloud redirects the request to the cloud platform. To view the resource details in the cloud platform, you must be logged in to the same account on the cloud platform where you want to further investigate.
Internet exposed instances port 22 traffic from Russia, China and Albania

### Violating Resources

<table>
<thead>
<tr>
<th>ALERT ID</th>
<th>RESOURCE NAME</th>
<th>ACCOUNT</th>
<th>REGION</th>
<th>ALERT STATUS</th>
<th>RATING</th>
<th>ALERT TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-60520</td>
<td>chly2-vm691</td>
<td>Azure-Redlock-public-demo</td>
<td>Azure West US 2</td>
<td>Open</td>
<td>F</td>
<td>A month ago</td>
</tr>
</tbody>
</table>

Alert Rules - Management Alert Rule, Parth CKadar DO NOT DELETE | Resource Configuration - View Details

https://portal.azure.com/#/resource/subscriptions/
Generate Reports on Prisma Cloud Alerts

You can generate two types of reports on alerts—Cloud Security Assessment report and Business Unit report. These reports enable you to inform your stakeholders on the status of the cloud assets and how they are doing against Prisma Cloud security and compliance policy checks. Sharing the reports on a regular basis enables them to monitor progress without requiring access to the Prisma Cloud administrator console.

The Cloud Security Assessment report is a PDF report that summarizes the risks from open alerts in the monitored cloud accounts for a specific cloud type. The report includes an executive summary and a list of policy violations, including a page with details for each policy that includes the description and the compliance standards that are associated with it, the number of resources that passed and failed the check within the specified time period.

The Business Unit report is a .csv file that includes the total number of resources that have open alerts against policies for any compliance standard, and you can generate the report on-demand or on a recurring schedule. You can opt to create an overview report which shows you how you're doing across all your business units, or get a little more granular about each of the cloud accounts you want to monitor. You can also generate the Business Unit report to review policy violations that are associated with specific compliance standards.

The overview report lists cloud resources by account group and aggregates information about the number of resources failing and the failure percentage against each policy. Whereas, the detailed Business Unit report lists cloud resources by account group, account name, and account ID, and it includes information about the number of resources failing against each policy and the status of cloud resources that have been scanned against that policy. The status can be pass or fail, and the status is reported as pass it means that the count of resources that failed the policy check is zero.

**STEP 1** | Select **Alerts > Reports > +Add New.**

**STEP 2** | Enter a **Name** and select a **Report Type**.

You can select Cloud Security Assessment report and Business Unit report.

- To generate a Business Unit Report:

1. Select the **Account Groups** to include in the file.
2. Select Detailed Report, if you want to include a breakdown of the policy details for each cloud account being monitored. Read above for more details on the difference between the overview and the detailed Business Unit report.

   For the detailed report, select Compress Attachments to get a .zip file for download or email attachment.

3. Filter alerts associated with specific compliance standards.
   1. Select Filter by Compliance Standard.
   2. Select the Compliance Standards from the drop-down.

      This option is useful if you want to view the alerts that are associated with policies that are tied to the selected compliance standards. The report then includes information on alerts that pertain to the selected compliance standards only.

4. Enter the Email address(es) for the file recipient(s).

5. Select the Notification template, if you want to use a custom email template.

      A custom email template allows you to tailor your message and add a URL in the message body. See Send Alert Notifications Through Email to set up a custom template.

6. Set the Frequency at which to send email notifications.

      One-Time—Sends an email to the recipient list only this once.
      Recurring—You can select the time interval as Daily, Weekly or Monthly. Prisma Cloud sends a single email to the recipient list on that day, during that week, or the month.

    You can only edit recurring Alerts Reports to modify some inputs such as the time interval, whether or not to compress attachments.

7. Save your changes.

   • To generate a Cloud Security Assessment PDF Report:

   1. Select the Cloud Type to include in the file.
   2. Select the Account Groups and the Cloud Accounts to include in the file.
   3. Select the cloud Regions and the Time Range for the report.

      If you have a large number of open alerts in the account(s) selected an error message displays. You must remove some accounts from this report or reduce the time range, and create a separate report for the details you need.
   4. Save your changes.
Alert Payload

A Prisma™ Cloud alert payload is a JSON data object that contains detailed information about an alert, such as the cloud account, resource, compliance standard, and policy.

![JSON](image)

<table>
<thead>
<tr>
<th>Alert Payload Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account ID</td>
<td>The ID of the cloud account where the violation that triggered the alert occurred.</td>
</tr>
<tr>
<td>Account Name</td>
<td>Name of the cloud account where Prisma Cloud detected the policy violation.</td>
</tr>
<tr>
<td>Alert ID</td>
<td>Identification number of the alert.</td>
</tr>
<tr>
<td>Alert Rule Name</td>
<td>Name of the alert rule that triggered this alert.</td>
</tr>
<tr>
<td>Callback URL</td>
<td>The URL for the alert in Prisma Cloud.</td>
</tr>
<tr>
<td>Cloud Type</td>
<td>Type of cloud account: AWS, Azure, or GCP.</td>
</tr>
<tr>
<td>Policy Description</td>
<td>Description of the policy as shown within Prisma Cloud.</td>
</tr>
<tr>
<td>Policy ID</td>
<td>Universally unique identification (UUID) number of the policy.</td>
</tr>
<tr>
<td>Policy Labels</td>
<td>Labels associated with the policy.</td>
</tr>
<tr>
<td>Policy Name</td>
<td>Name of the policy.</td>
</tr>
<tr>
<td>Alert Payload Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Policy Recommendation</td>
<td>Remediation recommendations for the policy.</td>
</tr>
<tr>
<td>Saved Search UUID</td>
<td>Universally unique identification (UUID) number of the saved search.</td>
</tr>
<tr>
<td>Remediation CLI</td>
<td>The CLI commands that you can use to resolve the policy violation.</td>
</tr>
<tr>
<td>Compliance Standard name</td>
<td>Name of the compliance standard.</td>
</tr>
<tr>
<td>Compliance Standard description</td>
<td>Description of the compliance standard.</td>
</tr>
<tr>
<td>Requirement ID</td>
<td>Identification number of the requirement in the compliance standard.</td>
</tr>
<tr>
<td>Requirement Name</td>
<td>Name of the requirement in the compliance standard.</td>
</tr>
<tr>
<td>Section ID</td>
<td>Identification number of the section in the compliance standard.</td>
</tr>
<tr>
<td>Section Description</td>
<td>Description of the section in the compliance standard.</td>
</tr>
<tr>
<td>Compliance ID</td>
<td>ID number of the compliance standard.</td>
</tr>
<tr>
<td>System Default</td>
<td>Indicates whether the compliance standard is Prisma Cloud System Default.</td>
</tr>
<tr>
<td>Custom assigned</td>
<td>Indicates if the compliance standard is assigned to a policy.</td>
</tr>
<tr>
<td>Resource Cloud Service</td>
<td>Cloud service provider of the resource that triggered the alert.</td>
</tr>
<tr>
<td>Resource Data</td>
<td>The JSON data of the resource.</td>
</tr>
<tr>
<td>Resource ID</td>
<td>ID of the resource that triggered the alert.</td>
</tr>
<tr>
<td>Resource Name</td>
<td>Name of the resource that triggered the alert.</td>
</tr>
<tr>
<td>Resource Region</td>
<td>Name of the cloud region to which the resource belongs.</td>
</tr>
<tr>
<td>Resource Region ID</td>
<td>ID of the region to which the cloud resource belongs.</td>
</tr>
<tr>
<td>Resource Type</td>
<td>Type of resource that triggered the alert (for example, EC2 instance or S3 bucket).</td>
</tr>
<tr>
<td>Risk Rating</td>
<td>Risk score defined for the resource: A, B, C, or F.</td>
</tr>
<tr>
<td>Alert Payload Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Severity</strong></td>
<td>Severity of the alert: High, Medium, or Low.</td>
</tr>
<tr>
<td><strong>User Attribution data</strong></td>
<td>Data about the user who created or modified the resource and caused the alert.</td>
</tr>
</tbody>
</table>

For alert notifications to include user attribution data, you must Populate User Attribution In Alerts Notifications (Settings > Enterprise Settings). Including user attribution data may delay alert notifications because the information may not be available from the cloud provider when Prisma Cloud is ready to generate the alert.
Prisma Cloud Risk Ratings

A Prisma™ Cloud risk rating is a letter grade from A through F that indicates the severity of the alerts that are associated with a resource. The rating is the collective score of all the policy violations for a resource. Each policy is assigned a severity rating of Low, Medium, or High and, when a resource violates the policy, the corresponding alert accrues the score that matches the severity rating on the policy.

Classification of severity ratings is as follows:

- **N/A**—No risk score is associated with the resource. The resource may not have a risk rating either because you have not configured policies to scan the resource or because the policies that the resource is scanned against do not apply to the resource.
- **A**—No alerts.
- **B**—1 low-severity alert or more.
- **C**—2 medium-severity alerts or more.
- **F**—Multiple alerts; need to assess.

You must immediately review resources that are rated as an **F** because these resources are violating policies with high severity. Because the rating is a cumulative score, an **F** rating may also include violations of medium and low severity policy rules.
Prisma Cloud Dashboards

The interactive Asset Inventory and SecOps dashboards give you visibility into the health and security posture of your cloud infrastructure. The dashboards provide a summarized and graphical view of all your Prisma Cloud cloud accounts and resources, and you can use the predefined or custom time range to view current trends or historical data.

> Assets, Policies, and Compliance on Prisma Cloud
> Prisma Cloud Asset Inventory
> SecOps Dashboard
> Customize the SecOps Dashboard
Assets, Policies, and Compliance on Prisma Cloud

To know the state of your cloud infrastructure, you need visibility into all the assets and infrastructure that make up your cloud environment and a pulse on your security posture.

Whether you want to detect a misconfiguration or you want to continually assess your security posture and adherence to specific compliance standards Prisma Cloud provides out-of-the-box policies (auditable controls) for ongoing reporting and measurement.

Policies are for risk assessment and they help to reduce the risk of business disruptions. Prisma Cloud provides policies that map to compliance standards, and a larger set of policies that enable prevention or detection of security risks to which your cloud assets are exposed. Anomaly policies are an example of policies that are typically not a part of compliance standards, and these policies inform you of actions performed on your cloud assets by entities that are users, services, or IAM roles that have authorization to access and modify your cloud assets, but the entities are not cloud assets.
Prisma Cloud supports the need to keep track of potential risks and threats to your cloud infrastructure with dashboards for your Asset Inventory, Compliance Dashboard, and out-of-the-box policies which generate alerts for cloud assets that are in violation. When a policy is violated, an alert is triggered in real time.

**Ingestion and Alert Cycle**

1. Data is ingested from the cloud
2. Assets are evaluated against policies
3. Alerts are generated for assets that violate policies

While alerts help you detect policy violations in real time and enable you to investigate what happened, the asset inventory and compliance dashboard are hourly snapshots of your assets and compliance posture for the last full hour.
Hourly Snapshots

Asset Inventory

Compliance Dashboard

Asset Explorer

10am

11am

12pm

1pm

Live Data

Alerts

Investigate
From the asset inventory and the compliance dashboard, you can directly access all open alerts by severity, and view asset details from the asset explorer as of the last hour.
Click icon to view alerts

Click number to view assets

Click row to view asset details
Prisma Cloud Asset Inventory

The Asset Inventory dashboard (on the Inventory tab) provides a snapshot of the current state of all cloud resources or assets that you are monitoring and securing using Prisma Cloud. From the dashboard, you gain operational insight over all our cloud infrastructure, including assets and services such as Compute Engine instances, Virtual machines, Cloud Storage buckets, Accounts, Subnets, Gateways, and Load Balancers.

Assets are displayed by default for all account groups, which the service monitors, for the most recent time range (last full hour). The interactive dashboard provides filters to change the scope of data displayed, so that you can analyze information you want to view in greater detail.

At a glance the Asset Inventory dashboard four sections:

- **Resource Summary** - Shows the count of the Total Unique Resources monitored by Prisma Cloud. Click the link to view all the assets on the Asset Explorer. For all these assets, you can toggle to view the following details as numeric value or a percentage:

- **Pass**—Displays the resources without any open alerts. Click the link for the passed resources and you will be redirected to the Asset Explorer that is filtered to display all the resources that have Scan Status set to Pass.
• **Low/Medium/High**—Displays the resources that have generated low, medium, or high severity alerts. On the asset inventory, when a resource triggers multiple alerts, the asset severity assigned to it matches the highest risk to which it is exposed. When you click the link, you will be redirected to the Asset Explorer that is filtered to display all the resources that match the corresponding Asset Severity level.

The View Alerts link enables you to view a list of all resources that have open alerts sorted by severity. Click each link to view the Alerts Overview sorted for low, medium or high severity alerts. You can review the policies that triggered the alerts along with a count of the total number of alerts for each policy.

• **Fail**—Displays the total number of resources that have generated at least one open alert when the hourly snapshot was generated. Click the link and you will be redirected to the Asset Explorer that is filtered to display all resources that have Scan Status set to Failed.

• **Asset Trend**—Trend line to help you monitor the overall health of your cloud resources starting when you added the first cloud account on Prisma Cloud through the time when the hourly snapshot was generated. The green, blue and red trend lines are overlaid to visually display the pass and failed resources against the total resource count. The trends depict the overall security posture of your resources and how they are performing over time so you can identify sudden surges with failed policy checks or sustained improvements with passing policy checks.

• **Asset Classification**—Bar graph for each cloud type (default), region name, or account name that depicts the ratio of passed to failed resources. This interactive graph allows you to drill into the passed and failed resources for details on the corresponding services that passed or failed policy checks; you can click and drag a section of the chart to zoom in further.

• **Tabular data**—The table enables you to group the results by account name, cloud region, or service name (default) and then drill down to view granular information on the resource types within your cloud accounts. All global resources for each cloud are grouped under AWS Global, Alibaba Cloud Global, Azure Global, and GCP Global.

Each row displays the service name with details on the cloud type (which you can filter on), and the percentage of resources that pass policy checks to which you want to adhere. The links is each column help you explore and gain the additional context you may need to take action.

> You may see more failed resources on the Compliance Dashboard compared to the Asset Inventory. This is because the Asset Inventory only counts assets that belong to your cloud account, and the Compliance Dashboard includes foreign entities such as SSO or Federated Users that are not resources ingested directly from the monitored cloud accounts.
SecOps Dashboard

The Dashboard > SecOps provides a graphical view of the performance of resources that are connected to the internet, the risk rating for all accounts that Prisma Cloud is monitoring, the policy violations over time and a list of the policies that have generated the maximum number of alerts across your cloud resources. It makes the security challenges visible to you as a quick summary, so you can dig in.

Monitored Accounts

This graph shows the number of accounts Prisma Cloud is monitoring.

Monitored Resources

Prisma Cloud considers any cloud entity that you work with as a resource. Examples of resources include AWS Elastic Compute Cloud, Relational Databases, AWS RedShift, Load Balancers, Security Groups, NAT Gateways. The Resources graph shows the total number of resources that you currently manage. It gives you a view into the potential growth in the number of resources in your enterprise over a period of time. Hover over the graph to see data as per the timeline.

Open Alerts

Whenever a resource violates a policy, Prisma Cloud generates alerts flagging these policy violations. The Open Alerts graph shows the number of alerts that were generated. The purpose of this graph is to demonstrate risk trends over a timeline. Click on the alert number to go to the ‘Alerts’ section and get the detailed view of the alerts.

Risk Rating by Scanned Accounts

Each resource in Prisma Cloud belongs to a specific account. The intent of this graph is to show all the accounts and their associated resources with their risk scoring. This graph is very useful in telling which accounts are the most vulnerable at any given time (in other words, have the most number of resources with an F or C score), and need to be fixed first. For details on how risk rating is calculated see Risk Rating of a Resource in Prisma Cloud.
Top Instances by Role

This graph summarizes top open ports in your cloud environments and the percentage of the traffic directed at each type of port. The purpose of this graph is to show what types of applications (web server, database) the top workloads are running.

Alerts by Severity

Alerts are graphically displayed and classified based on their severity into High, Medium, and Low. By clicking on the graph, you can directly reach the alerts section.

Policy Violations by Type over Time

This graph displays the type of policy violations (network, config, audit event) over a period of time.

Top Policy Violations

This graph displays the alerts generated by each type of policy over a period of time.

Top Internet Connected Resources

This graph displays top internet connected workloads by role, so you know which workloads are connecting to the Internet most of the time and are prone to malicious attacks. For this report, ELB & NAT Gateway data are filtered out, but includes data from other roles. The data in this chart is based on the account and the time filter.

Connections from the Internet

On a world map, you can see the inbound and outbound connections to different workloads across the globe, so that you visualize where the connections are originating from and see whether the traffic is regular internet traffic, suspicious traffic and all accepted traffic from suspicious IP addresses.
By default, the map shows aggregated numbers by specific regions in the map but you can zoom in on any of the regions in the map to get more granular detail on the specific location.

You can use the multi-select filter option available on the map to only present information for the type of workload(s) you are interested in viewing traffic for. By default, traffic to destination resources that are allowed to accept inbound connections such as NAT Gateways, ELB, Web Servers, and HTTP traffic is filtered out.

To see the network graph representing connections, click on any of the connections from a specific region and get redirected to the Investigate page to see the network graph. The network query will have the IP address, destination resources and the time filters carried forward so you can pinpoint to a specific incident.
Customize the SecOps Dashboard

You can customize the screen space used for each widget on the SecOps dashboard. For the Top Instances by Role widget, you can view the data as a table or click the graph to view the segment details in a table.

- Select Dashboard > SecOps and click Customize.
- Toggle Show or Hide to view or hide a widget.
- Select the icons on the screen to choose whether you want to maximize the screen space for a widget or fit two or three widgets in a row.
- Select a widget and click View as table to view the data in a tabular format.
Prisma Cloud Policies

In Prisma Cloud, a policy is a set of one or more constraints or conditions that must be adhered to. Prisma Cloud provides predefined policies for configurations and access controls that adhere to established security best practices such as PCI, GDPR, ISO 27001:2013, and NIST, and a larger set of policies that enable you to validate security best practices with an impact beyond regulatory compliance. These Prisma Cloud default policies cannot be modified.

In addition to these predefined policies, you can create custom policies to monitor for violations and enforce your own organizational standards. You can use the Default policies as templates to create custom policy. After you set up the policies, any new or existing resources that violate these policies are automatically detected.

> Create a Policy on Prisma Cloud
> Manage Prisma Cloud Policies
> Anomaly Policies
Create a Policy on Prisma Cloud

Create a custom policy with remediation rules that are tailored to meet the requirements of your organization. When creating a new policy, you can either build the query using RQL or you use a saved search to automatically populate the query you need to match on your cloud resources. For Prisma Cloud DevOps Security, you can also create configuration policies to scan your Infrastructure as Code (IaC) templates that are used to deploy cloud resources. The policies used for scanning IaC templates use a JSON query instead of RQL.

If you want to enable auto-remediation, Prisma Cloud requires write access to the cloud platform to successfully execute the remediation commands.

You can create three types of policies:

- **Config**—Configuration policies monitor your resource configurations for potential policy violations. Configuration policies on Prisma Cloud can be of two sub-types—Build and Run—to enable a layered approach. Build policies enable you to check for security misconfigurations in the IaC templates and ensure that these issues do not make their way into production. The Run policies monitor resources and check for potential issues once these cloud resources are deployed. See Create a Configuration Policy.
- **Network**—Network policies monitor network activities in your environment. See Create a Network or Audit Event Policy.
- **Audit Event**—Event policies monitor audit events in your environment for potential policy violations. You create audit policies to flag sensitive events such as root activities or configuration changes that may potentially put your cloud environment at risk. See Create a Network or Audit Event Policy.

Create a Configuration Policy

Use these instructions to add a custom configuration policy, for checking resources in the build or run phase of your application lifecycle. Because building the rules takes practice, before you start, take a look at a few Prisma Cloud default policies for directly on the administrative console, and review the query format within the rules.

**STEP 1** | Select **Policies** and click **New Policy > Config**.

**STEP 2** | Enter a **Policy Name**.

You can optionally add a **Description** and **Labels**.
STEP 3 | Select the policy subtype and click **Next**.

You can choose one or both the policy subtypes options:

- **Run** subtype enables you to scan cloud resources that are already deployed on a supported cloud platform.
- **Build** subtype enables you to scan IaC templates—Terraform, CloudFormation, Kubernetes manifest—that are used to deploy cloud resources.

![Image](image1.png)

STEP 4 | Select the **Severity** for the policy and click **Next**.

For a Run policy, an alert will be generated on a policy violation.

STEP 5 | Build the query to define the match criteria for your policy.

1. Add a rule for the **Run** phase.

   The Configuration—Run policies use RQL. If you are using a **Saved Search**, you can select from predefined options to auto-populate the query. For building a **New Search**, enter `config where` and use the auto-suggestion to select the available attributes and complete the query.

![Image](image2.png)
Config queries require some mandatory attributes. It should at a minimum have `api.name` in conjunction with `json.rule` or it can have `hostfinding.type` or it can have two `api.name` attributes with a `filter` attribute.

<table>
<thead>
<tr>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>config where cloud.type = 'azure' AND api.name = 'azure-network-usage' AND json.rule = StaticPublicIPAddresses.currentValue greater than 1</code></td>
</tr>
<tr>
<td><code>config where hostfinding.type = 'Host Vulnerability'</code></td>
</tr>
<tr>
<td><code>config where api.name = 'aws-ec2-describe-internet-gateways' as X; config where api.name = 'aws-ec2-describe-vpcs' as Y; filter '$.X.attachments[*].vpcId == $.Y.vpcId and $.Y.tags[*].key contains IsConnected and $.Y.tags[*].value contains true'; show Y;</code></td>
</tr>
</tbody>
</table>

*When creating a custom policy, as a best practice do not include cloud.account, cloud.account.group or cloud.region attributes in the RQL query. If you have a saved search that includes these attributes, make sure to edit the RQL before you create a custom policy. While these attributes are useful to filter the results you see on the Investigate tab, they are ignored when used in a custom policy.*

2. Add a rule for the **Build** phase.

If your policy will include both Run and Build checks, and you have added the RQL query, your cloud type for the build rule is automatically selected. It is based on the cloud type referenced in the RQL query.

1. Select the **Template Type** you want to scan—CloudFormation, Kubernetes, or Terraform. You can add one or more types.

   For scanning Terraform templates, you must select the Cloud Type and the Terraform version. Terraform versions 0.11 and 0.12 are supported.

2. Add the JSON query that specifies the properties or objects for which you want to apply policy checks. For more information see Add a JSON Query for Build Policy Subtype and Prisma Cloud IAC Scan Policy Operators.
If you choose to upload a template in the next step, the query you entered above is validated against the template. Each time you modify the query or upload a new template, the JSON query is re-validated.

3. (Optional) Upload a file to validate the JSON query.

The JSON Template Validation is optional. You can upload a single file or a .zip file. The supported file formats are HCL, YAML, JSON. The uploaded file is converted to JSON and displayed on-screen.

In addition, you can include a variable name and value to pass to the sample file and verify that the build rule works before you save the policy. For example, if you want to check whether EC2 instances include tags to identify the owner, the variables enable you to quickly validate against the sample template you attached.

STEP 6 | Add the compliance standards to your policy.

1. Choose the compliance Standard, Requirement, and Section.
2. Click + to add more standards as required and click Next.

STEP 7 | Enter details in the remediation section, if you want to automatically remediate alerts on a policy violation.

1. Select Run or Build

   Build phase policies do not support remediation CLI. You can however add the instructions for manually fixing the issue in the Recommendation for Remediation.

2. (Configuration—Run policies only) Enter Command Line remediation commands in CLI Remediation.
CLI remediation is available for `config where` queries only. You can add up to 5 CLI commands, and use a semi-colon to separate the commands in the sequence. The sequence is executed in the order defined in policy, and if a CLI command fails, the execution stops at that command. The parameters that you can use to create remediation commands are displayed on the interface as CLI variables, and a syntax example is:

```bash
gcloud -q compute --project=${account}
firewall-rules delete ${resourceName}; gsutil versioning set off gs://
${resourceName};
```

- **$account**—Account is the Account ID of your account in Prisma Cloud.
- **$azurescope**—(Azure only) Allows you to specify the node in the Azure resource hierarchy where the resource is deployed.
- **$gcpzoneid**—(GCP only) Allows you to specify the zone in the GCP project, folder, or organization where the resource is deployed.
- **$region**—Region is the name of the cloud region to which the resource belongs.
- **resourcegroup**—(Azure only) Allows you to specify the name of the Azure Resource Group that triggered the alert.
- **$resourceid**—Resource ID is the identification of the resource that triggered the alert.
- **$resourcename**—Resource name is the name of the resource that triggered the alert.

3. Click **Validate syntax** to validate the syntax of your code.

If you would like to see an example of the CLI syntax in the default remediable policies on Prisma Cloud, clone any existing policy and edit it.

*The default policies include additional variables that are restricted for use in default policies only, and are not supported in custom policies. Syntax validation displays an error if you use the restricted variables.*

4. Click **Save**.

All your System Administrators and Account Administrators are notified when there is a change to the CLI commands.

Create a Network or Audit Event Policy

Use the following instructions to add a custom Network or Audit Event policy on Prisma Cloud.

**STEP 1** | Select **Policies** and click **New Policy**.

**STEP 2** | Select **Audit Event** or **Network**.
### STEP 3 | Enter a Policy Name and Severity.

### STEP 4 | Add an optional Description and Labels before you click Next.

### STEP 5 | Build the query to define the match criteria for your policy by using a New Search or a Saved Search and click Next.

If you are using a Saved Search, you can select from the list of predefined options to auto-populate the query. The Select Saved Search drop-down displays the RQL for saved searches that match the policy type you selected in Step 2 above.

For a building a New Search, the RQL query must begin with `event where` for an Audit Event policy or `network where` for a Network policy. You can then use the auto-suggestion to select the available attributes and complete the query.
STEP 6 | Select the compliance standards for your policy.
   1. Choose the compliance **Standard**, **Requirement**, and **Section**.
   2. Click + to add more standards as required and click **Next**.

STEP 7 | *(Optional)* Provide a **Recommendation for Remediation**.
   CLI commands to enable automatic remediation are not supported on Audit Event or Network policy.

STEP 8 | **Save** the policy.

**Add a JSON Query for Build Policy Subtype**

Policy rules to **Secure Your Infrastructure Automation** are written in JSON. So, you need to first convert your template file—CFT, Kubernetes or Terraform templates—to JSON and then parse the JSON structure to write the JSON query that correctly identifies the parameter and criteria on which you want to be alerted. See **Build Policy Query Examples** below.

STEP 1 | **Convert your template file to JSON.**
   You can also use any online tool to convert your CFT, Kubernetes or Terraform templates to JSON. In this workflow, you use the following sample Terraform template and upload the template file to Prisma Cloud.
      For example:

```
Resources:
  myTrail:
    DependsOn:
      - BucketPolicy
      - TopicPolicy
    Type: AWS::CloudTrail::Trail
    Properties:
      S3BucketName:
        Ref: S3Bucket
      SnsTopicName:
        Fn::GetAtt:
          - Topic
          - TopicName
      IsLogging: true
      IsMultiRegionTrail: true
  myTrail2:
```
2. Select **Policies** and click **New Policy > Config**.
3. Enter a **Policy Name**.
   You can optionally add a **Description** and **Labels**.
4. Select the **Build** policy subtype and click **Next**.
   The build subtype enables you to scan IaC templates that are used to deploy cloud resources. For Run policy subtype, see **Create a Configuration Policy**.

5. Select the **Severity** for the policy and click **Next**.
6. Build the query to define the match criteria for your policy.

   If your policy is for both **Run** and **Build** checks and you have added the RQL query, your cloud type for the build rule is automatically selected. It is based on the cloud type referenced in the RQL query. Otherwise, you must select the template type and the cloud type.

   1. Select the **Template Type** you want to scan—CloudFormation, Kubernetes, or Terraform. The supported files types are HCL, YAML, JSON.

      For scanning Terraform templates, you must select the Cloud Type and the Terraform version. Terraform versions 0.11 and 0.12 are supported. For the other templates, you do not need to select the cloud type.

      2. Upload the template file for conversion to JSON in the JSON Template Validation section.
In order to upload a file you must add a JSON query string. You can enter `object exists` to enable the ability to attach a file.

Then, upload a single file or a .zip file.

The sample template above when converted to JSON looks like this:

```json
{
    "Resources": {
        "myTrail": {
            "Type": "AWS::CloudTrail::Trail",
            "Properties": {
                "S3BucketName": {
                    "Ref": "S3Bucket"
                },
                "IsLogging": true,
                "IsMultiRegionTrail": true
            }
        },
        "myTrail2": {
            "Type": "AWS::CloudTrail::Trail",
            "Properties": {
                "S3BucketName": {
                    "Ref": "S3Bucket"
                },
                "IsLogging": true,
                "IsMultiRegionTrail": true
            }
        }
    }
}
```

**STEP 2** Use JSON path validators to write your JSON query in the policy.

*Use the following guidelines to parse the file structure and write the JSON query that specifies the properties or objects for which you want to apply policy checks:*

- `$` - symbol refers to the root object or element.
- `@` - symbol refers to the current object or element.
- `.` - operator is the dot-child operator, which you use to denote a child element of the current element.
- `[]` - is the subscript operator, which you use to denote a child element of the current element (by name or index).
- `*` - operator is a wildcard, that returns all objects or elements without regard of the name.
- `? ( )` - to query all items that meet a certain criteria.

*For validating the JSON path, you can use validators such as https://jsonpath.com/ or others available on the internet. Each time you modify the query or upload a new template, Prisma Cloud revalidates your JSON query.*

*Refer to the list of Prisma Cloud IAC Scan Policy Operators to define the operators for the match.*

If for example, you require that all AWS accounts across your organization have enabled AWS CloudTrail, you can check for violations where AWS CloudTrail is not enabled.
• $.Resources.*[?(@.Type=='AWS::CloudTrail::Trail')], enables filtering the Resources whose type is AWS::CloudTrail::Trail
• and $.Resources.*[?(@.Type=='AWS::CloudTrail::Trail').Properties.IsMultiRegionTrail 'any null' or $.Resources.*[?(@.Type=='AWS::CloudTrail::Trail').Properties.IsMultiRegionTrail anyFalse enables you to further filter for the value specified for the MultiRegionalTrail parameter.

In this case, you are looking to identify a security issue when the value is missing or set to false. So, the match works as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsMultiRegionalTrail</td>
<td>missing</td>
<td>Because the default is false, the rule will match. It means there a security issue detected.</td>
</tr>
<tr>
<td>IsMultiRegionalTrail</td>
<td>false</td>
<td>The rule will match. It means there is a security issue detected.</td>
</tr>
<tr>
<td>IsMultiRegionalTrail</td>
<td>true</td>
<td>The rule will not match. It means there is no security issue detected.</td>
</tr>
</tbody>
</table>

**STEP 3 | Choose your next steps:**

Continue to Step 6 if you want to add compliance standards to the policy rule or to 5.b. Otherwise, Save the policy rule.

**Build Policy Query Examples**

The following section shows you an example of a Terraform template and an AWS CloudFormation Template (CFT).

• Sample JSON file (after being converted from Terraform)
  1. View the file contents.

```json
{
  "data": [
    {
      "azurerm_client_config": [
        {
          "current": [
            {}
          ]
        }
      ],
      "provider": [
        {
```
"azurerm": [
   
   "features": [
   
   "key_vault": [
   
   "purge_soft_delete_on_destroy": true
   
   ]
   
   ]
   
   ]
   
],
"resource": [

   "azurerm_resource_group": [
   
   "example": [
   
   "location": "West US",
   "name": "resourceGroup1"

   ]
   
   ]
   
   ],

   "azurerm_key_vault": [
   
   "example": [
   
   "access_policy": [
   
   "key_permissions": ["list"],
   "object_id": "1111111-2222-3333-4444-555555555555",
   "secret_permissions": ["list"],
   "storage_permissions": ["get"],
   "tenant_id": "2111-3333-4445-555"

   ],

   "enabled_for_disk_encryption": true,
   "location": "azurerm_resource_group.example.location",
   "name": "testvault",
   "network_acls": [
   
   "bypass": "AzureServices",
   "default_action": "Deny"

   ],

   "purge_protection_enabled": false,
   "resource_group_name": "def",
   "sku_name": "standard",
   "soft_delete_enabled": true,
   "tags": [
2. Define the match criteria for the policy.

This following query checks that the template for an Object ID match. It checks whether the object ID of the user or service principal in Azure Active Directory that is granted permissions matches your organizational policy.

```
$.resource[*].azurerm_key_vault.*[*].*.access_policy.exists
and $.resource[*].azurerm_key_vault.*[*].*.access_policy[*].object_id
== "1111111-2222-3333-4444-555555555555"
```

- Original CFT file that defines the security groups and the ports that allow ingress traffic.

1.

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  testDescription:
    Description: Tests for blocked ports negative case in AWS Security Groups
    Type: String
Resources:
  myELB:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
      AvailabilityZones:
        - eu-west-1a
      Listeners:
        - LoadBalancerPort: '80'
          InstancePort: '80'
          Protocol: HTTP
  myELBIngressGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: ELB ingress group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        SourceSecurityGroupId:
          Fn::GetAtt:
            - myELB
            - SourceSecurityGroup.OwnerAlias
        SourceSecurityGroupName:
          Fn::GetAtt:
            - myELB
            - SourceSecurityGroupName
```
2. Sample JSON file (after being converted from AWS CFT)

```json
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Parameters": {
        "testDescription": {
            "Description": "Tests for blocked ports negative case in AWS Security Groups",
            "Type": "String"
        }
    },
    "Resources": {
        "myELB": {
            "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
            "Properties": {
                "AvailabilityZones": [
                    "eu-west-1a"
                ],
                "Listeners": [
                    {
                        "LoadBalancerPort": "80",
                        "InstancePort": "80",
                        "Protocol": "HTTP"
                    }
                ]
            }
        },
        "myELBIngressGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "ELB ingress group",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 22,
                        "ToPort": 22,
                        "CidrIp": "0.0.0.0/0",
                        "SourceSecurityGroupOwnerId": {
                            "Fn::GetAtt": [
                                "myELB",
                                "SourceSecurityGroup.OwnerAlias"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```
3. Define the policy match

Check for any IPv4 or IPv6 CIDR range that allows unrestricted access for ingress traffic on port 22.

$.Resources.*[?(@.Type == 'AWS::EC2::SecurityGroup')].Properties.SecurityGroupIngress[?(@.IpProtocol == 'tcp' && @.FromPort == '22' && @.ToPort == '22' && @.CidrIp == '0.0.0.0/0')] size greater than 0 or $.Resources.*[?(@.Type == 'AWS::EC2::SecurityGroup')].Properties.SecurityGroupIngress[?(@.IpProtocol == 'tcp' && @.FromPort == '22' && @.ToPort == '22' && @.CidrIp6 == '::/0')] size greater than 0

This rule will match as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpProtocol</td>
<td>tcp</td>
<td>The rule will match.</td>
</tr>
<tr>
<td>FromPort</td>
<td>22</td>
<td>Security issue found.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Outcome</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>ToPort</td>
<td>22</td>
<td>The rule will match.</td>
</tr>
<tr>
<td>CidrIp</td>
<td>0.0.0.0/0</td>
<td>Security issue found.</td>
</tr>
<tr>
<td>IpProtocol</td>
<td>tcp</td>
<td>The rule will match.</td>
</tr>
<tr>
<td>FromPort</td>
<td>22</td>
<td>Security issue found.</td>
</tr>
<tr>
<td>ToPort</td>
<td>22</td>
<td>No security issue found.</td>
</tr>
<tr>
<td>CidrIp6</td>
<td>::/0</td>
<td>No security issue found.</td>
</tr>
<tr>
<td>IpProtocol</td>
<td>ftp</td>
<td>The rule will not match.</td>
</tr>
<tr>
<td>FromPort</td>
<td>23</td>
<td>No security issue found.</td>
</tr>
<tr>
<td>ToPort</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>CidrIp6</td>
<td>0.0.0.0/0 or ::/0</td>
<td></td>
</tr>
<tr>
<td>IpProtocol</td>
<td>tcp</td>
<td>The rule will not match because an IP address restriction is in place.</td>
</tr>
<tr>
<td>FromPort</td>
<td>22</td>
<td>No security issue found.</td>
</tr>
<tr>
<td>ToPort</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>CidrIp6</td>
<td>4.0.0.0/0 or ::/16</td>
<td></td>
</tr>
</tbody>
</table>

- The JSON query on Prisma Cloud does not support the following cases:

  Embedded policies or similar structures that are represented as a JSON string instead of JSON elements.

Example: If your resource template is like this:

```yaml
resource "aws_iam_role" "nat" {
  name = "${local.infra}-nat"
  path = "/"

  assume_role_policy =<<EOF
  {
    "Version": "2008-10-17",
    "Statement": [
      {
        "Action": "sts:AssumeRole",
        "Principal": {
          "Service": "ec2.amazonaws.com"
        },
        "Effect": "Allow",
        "Sid": ""
      }
    ]
  }
EOF
}
```

When converted to JSON, the `assume_role_policy` is a json string within json. While Prisma Cloud can fetch the complete policy, it does not support the ability to filter the parameters inside it.
• Filtering with multiple criteria or with literal * matching.

Example: An IAM policy that allows full administrative permissions, that is access to all AWS actions and resources, is a policy that contains a statement with


The following JSON equivalent for the policy above is not supported on Prisma Cloud:


Prisma Cloud IAC Scan Policy Operators

For Prisma Cloud DevOps Security, you can create configuration policies to scan your Infrastructure as Code (IaC) templates that are used to deploy cloud resources. The policies used for scanning IaC templates use a JSON query instead of RQL. The following list of operators are available for use in a JSON query, when you Add a JSON Query for Build Policy Subtype and specify the properties or objects for which you want to apply policy checks.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Usage Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>'greater than'</td>
<td><code>.securityContext.runAsUser &lt; 9999</code></td>
</tr>
<tr>
<td>'less than'</td>
<td><code>.password_reuse_prevention == 0</code></td>
</tr>
<tr>
<td>'equals'</td>
<td><code>$Resources.*[?(@.Type == 'AWS::EC2::SecurityGroup')].Properties.SecurityGroupIngress[?(@.IpProtocol == 'tcp' &amp;&amp; @.FromPort == '22' &amp;&amp; @.ToPort == '22' &amp;&amp; @.CidrIp == '0.0.0.0/0')] size greater than 0</code></td>
</tr>
<tr>
<td>Operator</td>
<td>Usage Examples</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>'does not equal'</td>
<td>.aws_vpc_peering_connection[<em>].[</em>].peer_vpc_id does not equal $resource[<em>].aws_vpc_peering_connection[</em>].[*].vpc_id</td>
</tr>
<tr>
<td>'starts with'</td>
<td>'startsWith'</td>
</tr>
<tr>
<td>'does not start with'</td>
<td>'!startsWith'</td>
</tr>
<tr>
<td>'ends with'</td>
<td>'endsWith'</td>
</tr>
<tr>
<td>'does not end with'</td>
<td>'!endsWith'</td>
</tr>
<tr>
<td>'contains'</td>
<td>.Properties.KmsMasterKeyId contains alias/aws/sqs</td>
</tr>
<tr>
<td>'includes one'</td>
<td></td>
</tr>
<tr>
<td>'does not contain'</td>
<td>'contains'</td>
</tr>
<tr>
<td>'is empty'</td>
<td>'isEmpty'</td>
</tr>
<tr>
<td>'is not empty'</td>
<td>'isNotEmpty'</td>
</tr>
<tr>
<td>'any empty'</td>
<td>'anyEmpty'</td>
</tr>
<tr>
<td>'none empty'</td>
<td>'noneEmpty'</td>
</tr>
<tr>
<td>'all empty'</td>
<td>'allEmpty'</td>
</tr>
<tr>
<td>'any null'</td>
<td>'anyNull'</td>
</tr>
</tbody>
</table>
## Operator Example Table

<table>
<thead>
<tr>
<th>Operator</th>
<th>Usage Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>'exists'</td>
<td><code>.Properties.VPCOptions exists</code></td>
</tr>
<tr>
<td>'does not exist'</td>
<td>'exists'</td>
</tr>
<tr>
<td>'any start with'</td>
<td>'anyStartWith'</td>
</tr>
<tr>
<td>'none start with'</td>
<td>'noneStartWith'</td>
</tr>
<tr>
<td>'all start with'</td>
<td>'allStartWith'</td>
</tr>
<tr>
<td>'any end with'</td>
<td>'anyEndWith'</td>
</tr>
<tr>
<td>'none end with'</td>
<td>'noneEndWith'</td>
</tr>
<tr>
<td>'all end with'</td>
<td>'allEndWith'</td>
</tr>
<tr>
<td>'any equal'</td>
<td>'anyEqual'</td>
</tr>
<tr>
<td>'none equal'</td>
<td>'noneEqual'</td>
</tr>
<tr>
<td>'all equal'</td>
<td>'allEqual'</td>
</tr>
<tr>
<td>'size equals'</td>
<td>'size =='</td>
</tr>
<tr>
<td>'size does not equal'</td>
<td>'size !='</td>
</tr>
<tr>
<td>'size greater than'</td>
<td>'size &gt;'</td>
</tr>
<tr>
<td>'size less than'</td>
<td>'size &lt;'</td>
</tr>
<tr>
<td>'length equals'</td>
<td>'length =='</td>
</tr>
<tr>
<td>'length does not equal'</td>
<td>'length !='</td>
</tr>
<tr>
<td>'length greater than'</td>
<td>'length &gt;'</td>
</tr>
<tr>
<td>'length less than'</td>
<td>'length &lt;'</td>
</tr>
<tr>
<td>Operator</td>
<td>Usage Examples</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>'any true'</td>
<td>'anyTrue'</td>
</tr>
<tr>
<td>'none true'</td>
<td>'noneTrue'</td>
</tr>
<tr>
<td>'all true'</td>
<td>'allTrue'</td>
</tr>
<tr>
<td>none false'</td>
<td>'noneFalse'</td>
</tr>
<tr>
<td>'all false'</td>
<td>'allFalse'</td>
</tr>
<tr>
<td>'is true'</td>
<td>'isTrue'</td>
</tr>
<tr>
<td>'is false'</td>
<td>'isFalse'</td>
</tr>
<tr>
<td>'is type'</td>
<td>'isType'</td>
</tr>
<tr>
<td>'is not type'</td>
<td>'!isType'</td>
</tr>
<tr>
<td>'is member of'</td>
<td>'isMemberOf'</td>
</tr>
<tr>
<td>'is not member of'</td>
<td>'!isMemberOf'</td>
</tr>
<tr>
<td>IDENTIFIER '[]'</td>
<td>resource[<em>].google_compute_subnetwork[</em>] S.Resources.*[? (@.Type=='AWS::S3::Bucket')].Properties</td>
</tr>
<tr>
<td>IDENTIFIER '[*]'</td>
<td></td>
</tr>
<tr>
<td>IDENTIFIER 'INT'</td>
<td></td>
</tr>
<tr>
<td>IDENTIFIER '?'</td>
<td></td>
</tr>
</tbody>
</table>
Manage Prisma Cloud Policies

To help you find the relevant policies based on your role, Prisma Cloud policies are grouped based on a hierarchy of Category, Class, Type, and Subtype. All of these groupings are available as filters on the Policies page.

The main categories are incidents and risks. An incident is likely a policy that identifies a potential security issue, while a risk is one that checks for risky configurations. The policy type indicates whether the check is performed against the network logs, audit logs, configuration logs, or user activity logs. Each policy type has subtypes for more granularity, for example, Anomaly policies are split into two subtypes—Network and UEBA. Class is another way to logically group policies into buckets such as Misconfiguration or Privileged Activity Monitoring.

<table>
<thead>
<tr>
<th>Category</th>
<th>Class</th>
<th>Type</th>
<th>Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident</td>
<td>Behavioral</td>
<td>Anomaly</td>
<td>UEBA</td>
</tr>
<tr>
<td></td>
<td>Behavioral</td>
<td>Anomaly</td>
<td>Network</td>
</tr>
<tr>
<td></td>
<td>Privileged Activity</td>
<td>Audit Event</td>
<td>Audit</td>
</tr>
<tr>
<td></td>
<td>Monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Network Protection</td>
<td>Network</td>
<td>Network Event</td>
</tr>
<tr>
<td>Risk</td>
<td>Misconfiguration</td>
<td>Config</td>
<td>Run</td>
</tr>
<tr>
<td></td>
<td>Misconfiguration</td>
<td>Config</td>
<td>Build</td>
</tr>
</tbody>
</table>

Use the following workflows to manage your Prisma Cloud policies. You can download policy data, clone, enable, delete, or disable policies from the Policies page.

• To enable global settings for Prisma Cloud default policies click Settings and select Enterprise Settings.

While some high severity policies are enabled to provide the best security outcomes, by default, policies of medium or low severity are in a disabled state. To enable policies based on severity, select Auto enable new default policies of the type—High, Medium, or Low. Based on what you enable, Prisma Cloud will scan your resources in the onboarded cloud accounts against policies that match the severity and generate alerts.

For Anomaly policies, you have more customizable settings, see Set Up Anomaly Policy Thresholds.
When you **Save** your changes, you can choose one of the following options:

- **Enable and Save**—With Enable and Save, you are enabling all existing policies that match your selection criteria and new Prisma Cloud default policies that are periodically added to the service. This option allows you to enable and scan your resources against all existing and new policies to help you stay ahead of threats and misconfigurations.

  You chose to enable all policies with medium priority. 121 policies match your selection. "Enable & Save" to turn on all current and future policies. "Save" to only turn on future policies.

- **Save**—With Save, you are saving your selection criteria and enabling new Prisma Cloud default policies only as they are periodically added to the service. New policies that match your selection, are automatically enabled and your resources are scanned against them after you made the change.

  If you enable policies of a specific severity, when you then clear the checkbox, the policies that were enabled previously are not disabled; going forward, policies that match the severity you cleared are no longer enabled to scan your cloud resources and generate alerts.

  You chose not to enable future policies. For future policies to be applied to your cloud accounts, manually enable them.

- **If you want to disable the policies that are currently active, you must disable the status of each policy on the Policies page.**

  The audit logs include a record of all activities performed in Prisma Cloud. To view the audit logs click Settings and select Audit Logs.

- **To view policies, select Policies.**
To filter Policies enter a keyword in the Filter Results search box or click Add Filters and select the filtering criteria.

The filters enable you to narrow the search results on the page. The values you select within a filter use the AND operator to display results. Across different filters, the selected values work as OR operators.

To find all Prisma Cloud policies of a specific Policy Subtype, when you select the values Build and Run, you can view all policies that are classified as Build policies OR Run policies. To find all policies that are classified as Build and Run, you must select the filter value Build, Run.

To download the details of your policies (or a filtered set of policies) in CSV format so that you can have an offline copy, click Download.
To enable or disable any policy toggle the **Status**.

To edit a custom policy, click the policy and you can edit the details.

> You cannot edit a Prisma Cloud Default policy.

To delete a policy, select the policy and click **Delete**.
To clone a policy, select the policy and click **Clone**.

Cloning a policy is creating a copy of an existing policy. Cloning serves as a quick method of creating a new policy if you choose to change few details of the source policy.

Prisma Cloud comes with default policies. If you want to modify any details, you can clone a policy and then modify details.

To view Alerts associated with a policy click **View Alerts**.
Anomaly Policies

Anomaly policies use audit logs and network flow logs to help you identify unusual network and user activity for all users, and is especially critical for privileged users and assumed roles where detecting unusual activity may indicate the first steps in a potential misuse or account compromise. These policies rely on threat feeds to resolve IP addresses to geo-locations and perform user entity behavior analysis (UEBA). When Prisma Cloud identifies a suspicious IP address, the threat feed enables you to classify and view more information on the malicious IP addresses with which the suspicious IP address is communicating, so you can quickly figure out which alerts to pay attention to and act on.

Before the service can detect unusual activity for your enterprise, you must Define Prisma Cloud Enterprise and Anomaly Settings to specify a training threshold and set the baseline for what are normal trends in your network. To set this baseline, Prisma Cloud gathers information about the user or identities used to access the monitored cloud accounts, the devices used for access, the IP addresses and locations they come from, the ports and protocols typically used, the cloud services they use and the frequency, the hours within which they access these applications, and the activities they perform within the cloud services.

The anomaly policies that are predefined and marked as Prisma Cloud Default policies alert you to these issues:

**Account hijacking attempts**—Detect potential account hijacking attempts discovered by identifying unusual login activities. These can happen if there are concurrent login attempts made in short duration from two different geographic locations, which is *impossible time travel*, or login from a previously unknown browser, operating system, or location.

**Excessive login failures**—Detect potential account hijacking attempts discovered by identifying brute force login attempts. Excessive login failure attempts are evaluated dynamically based on the models observed with continuous learning.

**Unusual user activity**—Discover insider threat and an account compromise using advanced data science. The Prisma Cloud machine learning algorithm profiles a user’s activities on the console, as well as the usage of access keys based on the location and the type of cloud resources.

**Network evasion and resource misuse**—Detects unusual server port activity or unusual protocol activity from a client within or outside your cloud environment to an server host within or outside your network using a server port or an IP protocol that is not typical to your network traffic flows. To identify potential resource misuse, the anomaly policy monitors when a host inside your cloud environment that has no prior mail-related network activity, starts generating outbound SMTP traffic.

**Network reconnaissance**—Detect port scan or port sweep activities that probe a server or host for open ports. The port scanning policies identify when an attacker is performing a vertical scan to find any ports on a target, and the port sweep detects a horizontal scan where an attacker is scanning for a specific port on many targets hosts. The policies identify whether the source of the attack internal that is the port scan or sweep originates from an instance within your cloud environment, or external where the source of the port scan or sweep originates from the internet and targets the cloud environment that is monitored by Prisma Cloud. The policies that detect internal port scan and port sweep activity are enabled by default.

To find all anomaly policies on Prisma Cloud, use the Policy type and Policy Subtypes filters on Policies.
Alerts generated for anomaly policies are grouped by policy and then by user. Because the same IP address can resolve to different locations at different points in time, if there is an unusual user activity from a previously unseen location for an IP address that has been seen before, Prisma Cloud does not generate an anomaly alert (and reduces false positives).

*If you want to add one or more IP addresses as trusted sources, see [Trusted IP Addresses](#) on Prisma Cloud. IP addresses included in the trusted list do not generate alerts for network based anomaly policies such as network reconnaissance, evasion and resource misuse policies.*

To view alerts generated for an anomaly policy, see [Alerts > Overview](#), and filter for alerts generated against anomaly policies and get the details on what was identified as unusual or suspicious activity. Note that multiple alerts of the same type (when a user accesses a resource that is flagged as an anomaly), are logged as a single alert, while a distinct alert is generated if the same user accesses another type of resource.

Alerts generated against the anomaly policies also include additional context based on threat feed information from Autofocus and Facebook Threat Exchange. Use the tooltip to review the threat details. If you have an AutoFocus license, you can click the IP address link to launch the AutoFocus portal and search for a Suspicious IP address directly from the [Investigate](#) page, see [Use Prisma Cloud to Investigate Network Incidents](#).
From the alert details use the [\(\square\)], to pivot to the **Investigate** page. For UEBA anomaly policies, you can also see a **Trending View** of all anomalous activities performed by the entity or user.
Prisma Cloud helps you visualize your entire cloud infrastructure and provides insights into security and compliance risks.

Prisma Cloud helps you connect the dots between configuration, user activity, and network traffic data, so that you have the context necessary to define appropriate policies and create alert rules.

To conduct such investigations, Prisma Cloud provides you with a proprietary query language called RQL that is similar to SQL.

- Investigate Config Incidents on Prisma Cloud
- Investigate Audit Incidents on Prisma Cloud
- Use Prisma Cloud to Investigate Network Incidents
Investigate Config Incidents on Prisma Cloud

Prisma Cloud ingests various services and associated configuration data from AWS, Azure, Alibaba, and GCP cloud services. You can retrieve resource information to identify resource misconfigurations, and detect policy violations that expose your business to undue risk and non-compliance to industry benchmarks.

To investigate configuration issues, you can use Config queries. You can enter your query in the Search bar and if the search expression is valid and complete, a green check mark displays along with your query results.

You can choose to save the searches that you have created for investigating incidents in My Saved Searches. A saved search enables you to use the same query at a later time, instead of typing the query again, and it enables you to use the saved search to create a policy.

Saved Searches has list of search queries saved by any Prisma Cloud administrator.

Select a record to view the Audit Trail or Host Findings. The alerts are displayed when you select the red exclamation mark.
Hover over the configuration record to see the option to view the details of the resource configuration. You can also search directly within the JSON Resource configuration to easily find something that is part of the metadata ingested on Prisma Cloud, and speed up your investigation.

To analyze your configuration events offline, you can download the event search details in a CSV format, click Download on the right hand corner.
Investigate Audit Incidents on Prisma Cloud

Prisma Cloud ingests various services and associated user and event data from AWS, Azure, and GCP cloud services. You can investigate console and API access, monitor privileged activities and detect account compromise and unusual user behavior in your cloud environment.

To investigate audit data you can use Event queries. To build Event RQL queries, enter your query in the Search; use the auto-suggest for the attribute `json.rule` with the operators `=` and `IN`, (auto suggestion is not available for array objects). If the search expression is valid and complete, you can see a green check mark and results of your query. You can choose to save the searches that you have created for investigating incidents in My Saved Searches. Use these queries for future reuse, instead of typing the queries all over again. You can also use the Saved Searches to create a policy. Saved Searches has list of search queries saved by any user in the system.

After you run event search queries, you can view the results in Table View, Trending View, or in Map View. By default you can see the details in the Table view. To pick the columns in the Table view, use the Column Picker on the Right hand corner.

From the table view, select View Event Details to see the resource configuration details.

To analyze your Audit events offline, you can download the event search details in a CSV format, click Download on the right hand corner.
Select **Trending View** to see the results in a timeline. Single click the bubble to view the results for a given timeline. Double click the bubble to drill down further.

Select **Map View** to see a World map with pinpoints to the locations where there are activities and anomalies. You can view usual activities and anomalous activities to their specific locations. Single click on the bubble in the map view to view results for the given location. Double click on the bubble in the map view to drill down further.
Use Prisma Cloud to Investigate Network Incidents

Prisma Cloud ingests and monitors network traffic from cloud services and allows customers to query network events in their cloud environments. You can detect when services, applications or databases are exposed to the internet and if there are potential data exfiltration attempts. Network queries are currently supported for AWS, Azure and GCP.

To view network traffic data, you can use Network queries. Enter your queries in the Search. If the search expression is valid and complete, you can see a green check mark and results of your query. You can choose to save the searches that you have created for investigating incidents in **My Saved Searches**. Use these queries for future reuse, instead of typing the queries all over again. You can also use the Saved Searches to create a policy. **Saved Searches** has list of search queries saved by any user in the system.

Network queries enable you to search for network resources or network flows. By using packets, bytes, source or destination resource, source or destination IP address, and source or destination port information, these queries enable you to monitor traffic and the interconnectivity of the resources that belong to your cloud accounts and regions.

To download network traffic details for your entire network, a node or an instance, or for a specific connection between a source and a destination node in a CSV format, click **Download** on the top right hand corner. This report groups all connection details by port and includes details such as source and destination IP addresses and names, inbound and outbound bytes, inbound and outbound packets, and whether the node accepted the traffic connection.

To see the details of a network resource, click the resource and view **Instance Summary, Network Summary, or Alert Summary**.

To see the accepted and rejected traffic, use the **Traffic Summary** link. Note that the attempted bytes count displays traffic that is either denied by the security group or firewall rules or traffic that was reset by a host or virtual machine that received the packet and responded with a RST packet.
To view details of a connection, click the connection and click **View Details**. If the traffic is from a suspicious IP address as characterized by a threat feed, you get more details on the threat feed source, when it was classified and reason for classification.

And if you have an AutoFocus license, you can click the IP address link to launch the AutoFocus portal and search for a Suspicious IP address directly from the **Investigate** page.
162.247.74.201

VERDICT: Malware

SOURCE: WILDFIRE (IF 6 RECENT SAMPLES REACHED A MALWARE VERDICT)

WILDFIRE: FIRST SEEN 05/04/2020 4:07:09AM  LAST SEEN 05/05/2020 5:32:08PM

PAN-DB Categorization

<table>
<thead>
<tr>
<th>URL</th>
<th>CATEGORY</th>
<th>RISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>162.247.74.201</td>
<td>Malware</td>
<td>N/A</td>
</tr>
</tbody>
</table>

DNS Security Results

<table>
<thead>
<tr>
<th>FQDN</th>
<th>VERDICT</th>
<th>GLOBAL THREAT ID</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>162.247.74.201</td>
<td>Benign</td>
<td>0</td>
<td>3600</td>
</tr>
</tbody>
</table>
Prisma Cloud enables you to view, assess, report, monitor and review your cloud infrastructure health and compliance posture. You can also create reports that contain summary and detailed findings of security and compliance risks in your cloud environment.

- Compliance Dashboard
- Create a Custom Compliance Standard
- Add a New Compliance Report
Compliance Dashboard

The Compliance Overview is a dashboard that provides a snapshot of your overall compliance posture across various compliance standards. Use the Compliance Dashboard as a tool for risk oversight across all the supported cloud platforms and gauge the effectiveness of the security processes and controls you have implemented to keep your enterprise secure. You can also create compliance reports and run them immediately, or schedule them on a recurring basis to measure your compliance over time.

The built-in regulatory compliance standards that Prisma Cloud supports are:

<table>
<thead>
<tr>
<th>Cloud Type</th>
<th>Compliance Standards Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCP</td>
<td>CIS v1.0, CSA CCM v3.0.1, CCPA, GDPR, HITRUST v9.3, HIPAA, ISO 27001:2013, MITRE ATT&amp;CK, NIST 800.53 R4, NIST CSF v1.1, PCI DSS v3.2, PIPEDA, SOC 2</td>
</tr>
</tbody>
</table>

To help you easily identify the gaps and measure how you’re doing against the benchmarks defined in the governance and compliance frameworks, the Compliance Dashboard [Compliance > Overview] combines rich visuals with an interactive design. The dashboard results include data for the last full hour. The timestamp on the bottom right corner of the screen indicates when the data was aggregated for the results displayed.

The compliance dashboard is grouped into three main sections that enable you to continuously monitor progress.
• **Filters**—The left pane provides filters that help sharpen the focus on your compliance posture across different cloud types, accounts, regions, and specific compliance mandates—compliance standards and the requirements and sections within each standard. The compliance time selector allows you to specify the time range for which you want to see your compliance posture. By default, the dashboard shows your compliance state as of today. Because the Prisma Cloud service ingests data on all assets in the connected cloud accounts, you can use this data to audit usage/deployment of resources on each cloud and measure improvement over time. For example, you can see how you were doing three months ago and analyze trends in adherence to compliance guidelines today.

• **Compliance Score and Charts**—The colorful and interactive main section presents the overall health of the cloud resources in your organization. The rich visual display helps you focus your attention on the gaps in compliance for a standard or regulation that is important to you.

  - The compliance score presents data on the total unique resources that are passing or failing the policy checks that match compliance standards. Use this score to audit how many unique resources are failing compliance checks and get a quick count on the severity of these failures. The links allow you to view the list of all resources on the Asset Explorer, and the View Alerts link enables you to view all the open alerts of Low, Medium, or High severity.

  - The compliance trendline is a line chart that shows you how the compliance posture of your monitored resources have changed over time (on the horizontal X axis). You can view the total number of resources monitored (in blue), and the number of resources that passed (in green) and failed (in red) over that time period.

  - The Compliance coverage sunburst chart highlights the passed and failed resource count across all compliance standards and enables easy comparison. When you click on the inner circle, you can drill-down to the summary for a specific compliance standard that needs your attention; click the center of the donut to toggle and view all the compliance standards. When you click on the outer circle, you can view the alerts that map to the failed resources associated with a standard. To review all the details, click the link for the description of the compliance standard.

• **Compliance Standards Table**—The last section is a list of all the built-in and custom standards that you may have defined to monitor and audit your organization’s performance. Each row in the table includes a description of a standard and the total number of policies that map to the standard. It also includes the total number of unique resources monitored for that standard, the pass and fail count, along with a percentage of the resources that passed the compliance checks. For each failed check, the severity of the issue affects where it is counted. For example, if a resource fails a high severity policy, it is not counted towards a medium or low failure even if it fails a medium or low severity policy rule.
To learn about each compliance standard, the requirements/sections that it comprises and the policies that map to each requirement, use the links in each row. You can also click the description in the table to open a new tab that automatically filters the data to display information about the selected compliance standard and then generate a report on demand. To generate compliance reports, see Add a New Compliance Report.

Unlike the Asset Inventory that aggregates all your resources and displays the pass and fail count for all monitored resources, the Compliance Dashboard only displays the results for monitored resources that match the policies included within a compliance standard. For example, even if you have 30 AWS Redshift instances, if none of the compliance standards include policies that check the configuration or compliance and security standards for Redshift instances, the 30 Redshift instances are not included in the resource count on the Compliance Dashboard. The results on the Compliance Dashboard therefore, help you focus your attention on the gaps in compliance for a standard or regulation that is important to you. See Assets, Policies, and Compliance on Prisma Cloud for additional context.
Create a Custom Compliance Standard

You can create your own custom compliance standards that are tailored to your own business needs, standards, and organizational policies. When defining a custom compliance standard, you can add requirements and sections. A custom compliance standard that has a minimum of one requirement and one section can be associated with policies that check for adherence to your standards.

You can create an all new standard or clone an existing compliance standard and edit it.

- Clone an existing compliance standard to customize.
  1. On Prisma Cloud, select **Compliance > Standards**.
  2. Hover over the standard you want to clone, and click **Clone**.

When you clone, it creates a new standard with the same name with Copy in the prefix. You can then edit the cloned compliance standard to include the requirements, sections, and policies you need.

- Create a compliance standard from scratch.
  1. On Prisma Cloud, select **Compliance > Standards > + Add New**.
2. Enter a name and description for the new standard and click Save.

3. Add requirements to your custom compliance standard.
   1. Select the custom compliance standard you just added and click + Add New.
   2. Enter a requirement, name and a description and click Save.

4. Add sections to your custom compliance standard after adding the requirement.
   1. Select the requirement for which you are adding the section and click + Add New.
2. Enter a name for the **Section** a **Description** and click **Save**.

Although you have added the custom standard to Prisma Cloud, it is not listed on the Compliance Standards table on **Compliance > Overview** until you add at least one policy to it.

5. **Add policies to your custom compliance standard.**

You must associate Prisma Cloud Default policies or your custom policies to the compliance standard to monitor your cloud resources for adherence to the internal guidelines or benchmarks that matter to you. The RQL in the policy specifies the check for the resource configuration, anomaly or event.

1. **Select Policies.**

Filter the policies you want to associate with the standard. You can filter by cloud type, policy type and policy severity, to find the rules you want to attach.

2. **Select the policy rule to edit, on 3 Compliance Standards click + and associate the policy with the custom compliance standard.**

3. **Confirm your changes.**
Add a New Compliance Report

Creating compliance reports is the best way to monitor your cloud accounts across all cloud types—AWS, Azure, and GCP—and ensure that you are adhering to all compliance standards. You can create compliance reports based on a cloud compliance standard for immediate online viewing or download, or schedule recurring reports so you can monitor compliance to the standard over time. From a single report, you have a consolidated view how well all of your cloud accounts are adhering to the selected standard. Each report details how many resources and accounts are being monitored against the standard, and, of those, how many of the resources passed or failed the compliance check. In addition, the report provides detailed findings for each section of the standard including a description of the requirements in each section, what resources failed the compliance check, and recommendations for fixing the issues, so that you can prioritize what you need to do to become compliant. From the Compliance Reports dashboard, you can also view or download historic reports so that you can see your compliance trend.

**STEP 1 |** Log in to Prisma Cloud.

**STEP 2 |** Create a new report.

1. Select **Compliance > Overview** and select the standard for which you want to create a new compliance report.
2. On the page for the compliance standard you selected, click **Create Report**.

3. Enter the following information and **Save** the report.
   - Enter a descriptive **Name** for the report.
   - Enter the **Email** address to which to send report when scheduled.
   - Select whether you want to run the report **One Time** or **Recurring**.
     
     If you select **Recurring** you must also specify how often you want to run the report, the interval, day of the week, and time when you want to schedule the recurring report to run.

**STEP 3 | View your compliance reports.**

After you create a compliance report, it will automatically run at the time you specified. You can then view and manage your reports as follows:

- To the list of all compliance reports that have run, select **Compliance > Reports**. You can use the filters to narrow the list of compliance reports shown, or search for the report.

- To view a compliance report, click the report name.

A graphical view of the report displays showing the number of unique cloud resources and how many of them passed and the number and severity of those that failed (you can also toggle this to show percentages instead) and a graphical representation of how well your cloud accounts are doing against all sections of the standard. If this report has run before, you can also see the compliance trend over time. Finally the report shows summarizes compliance against each requirement of the
standard. To drill down into details on a particular requirement of the standard, click the requirement name.

• If you want to refine the report so that it only shows the details you are interested, clone it. You can then use the Compliance filters to customize the report to show only the information you are interested. You can use the Compliance filters set the report timeframe and narrow the report to only show compliance information for specific cloud accounts, cloud regions, or cloud types. As you add or remove filters, the report updates so that you can see your changes reflected in the report. When the cloned report shows the information you want it to, click Create Report to save it as a new report instance.
• You can [Download Report](#) for a PDF of the entire option (unless the report has already been scheduled for download in which case this option is grayed out). You can also download the details about compliance with each requirement of the standard to a CSV file by clicking the download icon.

• You can also download the compliance reports from the Compliance > Reports page by clicking the Download icon that corresponds the specific report you want to download. Note that for recurring reports, this downloads the most recent report generated.

• For recurring reports, you can view the report history by clicking the corresponding History icon. You can then view individual instances of the compliance report, or download them.
• To edit the recurrence settings of a report you added, or to add or remove email addresses of report recipients, click the corresponding Edit icon.

• For recurring reports, you can indicate whether you want to automatically include a PDF of the report to the recipients you defined, or whether you want administrators to be able to download the report on demand rather than emailing it by toggling Enable Scheduling. With this setting enabled, the report will automatically be emailed according to the recurrence schedule you defined. With it disabled, the report will not be emailed, but can be downloaded on demand.
Configure External Integrations on Prisma Cloud

You can integrate Prisma Cloud with third-party services such as Jira, Slack, Splunk, Google CSCC, Qradar, and ServiceNow to enable you to receive, view and receive notification of Prisma Cloud alerts in these external systems. By integrating Prisma Cloud with third-party services you can have an aggregated view of your cloud infrastructure.

Similarly, Prisma Cloud integration with external systems such as Amazon GuardDuty, AWS Inspector, Qualys, and Tenable allow you to import vulnerabilities and provide additional context on risks in the cloud.

- Prisma Cloud Integrations
- Integrate Prisma Cloud with AWS Inspector
- Integrate Prisma Cloud with Amazon SQS
- Integrate Prisma Cloud with Amazon GuardDuty
- Integrate Prisma Cloud with AWS Security Hub
- Integrate Prisma Cloud with Azure Service Bus Queue
- Integrate Prisma Cloud with Google Cloud Security Command Center (SCC)
- Integrate Prisma Cloud with Jira
- Integrate Prisma Cloud with Qualys
- Integrate Prisma Cloud with Slack
- Integrate Prisma Cloud with Splunk
- Integrate Prisma Cloud with Tenable
- Integrate Prisma Cloud with ServiceNow
- Integrate Prisma Cloud with Webhooks
- Integrate Prisma Cloud with PagerDuty
- Integrate Prisma Cloud with Microsoft Teams
- Integrate Prisma Cloud with Cortex XSOAR
- Prisma Cloud Integrations—Supported Capabilities
Prisma Cloud Integrations

Prisma™ Cloud provides multiple out-of-the-box integration options that you can use to integrate Prisma Cloud into your existing security workflows and with the technologies you already use. The Amazon GuardDuty, AWS Inspector, Qualys, and Tenable integrations are inbound or pull-based integrations where Prisma Cloud periodically polls for the data and retrieves it from the external integration system; all other integrations are outbound or push-based integrations where Prisma Cloud sends data about an alert or error to the external integration system.

Alibaba Cloud in the Mainland China regions does not support all the integrations listed below. The supported Integrations are Email, Splunk and Webhooks.

- **Amazon GuardDuty**—Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. Prisma Cloud integrates with Amazon GuardDuty and ingests vulnerability data to provide you with additional context on risks in the cloud.
- **AWS Inspector**—AWS Inspector assesses applications for exposure, vulnerabilities, and deviations from best practices. It also produces a detailed list of security findings prioritized by level of severity. Prisma Cloud integrates with AWS Inspector and ingests vulnerability data and Security best practices deviations to provide you with additional context about risks in the cloud.
- **AWS Security Hub**—AWS Security Hub is a central console where you can view and monitor the security posture of your cloud assets directly from the Amazon console. As the Prisma Cloud application monitors your assets on the AWS cloud and sends alerts on resource misconfigurations, compliance violations, network security risks, and anomalous user activities, you have a comprehensive view of all your cloud assets across all your AWS accounts directly from the Security Hub console.
- **Amazon SQS**—Amazon Simple Queue Service (SQS) helps you send, receive, and store messages that pass between software components at any volume without losing messages and without requiring other services to be always available. Prisma Cloud can send alerts to Amazon SQS, and you can set up the AWS CloudFormation service to enable custom workflows.
- **Azure Service Bus Queue**—Azure Service Bus is a managed messaging infrastructure designed to transfer data between applications as messages. With the Prisma Cloud and Azure Service Bus queue integration, you can send alerts to the queue and set up custom workflows to process the alert payload.
- **Cortex XSOAR**—Cortex XSOAR (formerly Demisto) is a Security Orchestration, Automation and Response (SOAR) platform that enables you to streamline your incident management workflows. With the Prisma Cloud and Cortex XSOAR integration you can automate the process of managing Prisma Cloud alerts and the incident lifecycle with playbook-driven response actions.
- **Email**—Configure Prisma Cloud to send alerts as emails to your email account.
- **Google Cloud SCC**—Google Cloud Security Command Center (SCC) is the security and data risk database for Google Cloud Platform. Google Cloud SCC enables you to understand your security and data attack surface by providing inventory, discovery, search, and management of your assets. Prisma Cloud integrates with Google Cloud SCC and sends alerts to the Google Cloud SCC console to provide centralized visibility in to security and compliance risks of your cloud assets.
- **Jira**—Jira is an issue tracking, ticketing, and project management tool. Prisma Cloud integrates with Jira and sends notifications of Prisma Cloud alerts to your Jira accounts.
- **Microsoft Teams**—Microsoft Teams is cloud-based team collaboration software that is part of the Office 365 suite of applications and is used for workplace chat, video meetings, file storage, and application integration. The Prisma Cloud integration with Microsoft Teams enables you to monitor your assets and send alerts on resource misconfigurations, compliance violations, network security risks, and anomalous user activities—either as they happen or as consolidated summary cards.
- **PagerDuty**—PagerDuty enables alerting, on-call scheduling, escalation policies, and incident tracking to increase the uptime of your apps, servers, websites, and databases. The PagerDuty integration enables...
you to send Prisma Cloud alert information to PagerDuty service. The incident response teams can
investigate and remediate the security incidents.

- **Qualys**—Qualys specializes in vulnerability management security software that scans hosts for potential
vulnerabilities. Prisma Cloud integrates with the Qualys platform and ingests vulnerability data to
provide you with additional context about risks in the cloud.

- **ServiceNow**—ServiceNow is an incident, asset, and ticket management tool. Prisma Cloud integrates
with ServiceNow and sends notifications of Prisma Cloud alerts as ServiceNow tickets.

- **Slack**—Slack is an online instant messaging and collaboration system that enables you to centralize all
your notifications. You can configure Prisma Cloud to send notifications of Prisma Cloud alerts through
your slack channels.

- **Splunk**—Splunk is a software platform that searches, analyzes, and visualizes machine-generated data
gathered from websites, applications, sensors, and devices. Prisma Cloud integrates with cloud-based
Splunk deployments and enables you to view Prisma Cloud alerts through the Splunk event collector.
Prisma Cloud can integrate with on-premises Splunk instances through the AWS SQS integration.

- **Tenable**—Tenable.io is a cloud-hosted vulnerability management solution that provides visibility and
insight in to dynamic assets and vulnerabilities. Prisma Cloud integrates with Tenable and ingests
vulnerability data to provide you with additional context about risks in the cloud.

- **Webhooks**—The webhooks integration enables you to pass information in JSON format to any third-
party integrations that are not natively supported on Prisma Cloud. With a webhook integration, you
can configure Prisma Cloud to send alerts to the webhook URL as an HTTP POST request so that any
services or applications that subscribe to the webhook URL receive alert notifications as soon as Prisma
Cloud detects an issue.

For the outbound integrations—with the exception of PagerDuty and email, Prisma Cloud performs periodic
checks and background validation to identify exceptions or failures in processing notifications. The status
checks are displayed on the Prisma Cloud administrator console: red if the integration fails validation checks
for accessibility or credentials; yellow if one or more templates associated with the integration are invalid;
or green when the integration is working and all templates are valid. Any state transitions are also displayed
on the Prisma Cloud administrator console to help you find and fix potential issues.
Integrate Prisma Cloud with Slack

Integrate Prisma™ Cloud with Slack to get instant messages on your Slack channels. This will help you to collaborate and centralize all your notifications.

**STEP 1 | Set up Slack to get the webhook for your application.**

1. Log in to your web Slack using your company URL. For example https://<company-name-or-abbreviation>.slack.com/apps/manage.
2. Select **Manage > Apps**.

3. Select **Incoming WebHooks**.

4. **Add Configuration**.
5. Select a channel and **Add Incoming WebHooks Integration**.

   ![Add Incoming WebHooks integration](image)

6. **Save Settings**.

   Copy the **Webhook URL** from this page so you can specify this URL in Prisma Cloud.

   ![Setup Instructions](image)

   **STEP 2** | **Configure Slack in Prisma Cloud and complete set up of the integration channel.**

   1. Log in to Prisma Cloud.
   2. Select **Settings > Integrations**.
   3. Create a **+New Integration**.
   4. Set the **Integration Type** to **Slack**.
   5. Enter a name and a description for this integration.
   6. Enter the **WebHook URL**.
7. Click **Next** and then **Test**.

The status check for Slack displays in red text if Prisma Cloud receives any of the following errors: user not found or channel not found, channel is archived, action prohibited, posting to general channel denied, no service or no service ID, or no team or team disabled.

**STEP 3 | Create an Alert Rule** or modify an existing rule to send alerts to your Slack channels. See **Send Prisma Cloud Alert Notifications to Third-Party Tools**.
Integrate Prisma Cloud with Splunk

Splunk is a software platform to search, analyze, and visualize machine-generated data gathered from websites, applications, sensors, and devices.

Prisma™ Cloud integrates with Splunk and monitors your assets and sends alerts for resource misconfigurations, compliance violations, network security risks, and anomalous user activities to Splunk.

**STEP 1** Set up Splunk HTTP Event Collector (HEC) to view alert notifications from Prisma Cloud in Splunk.

Splunk HTTP Event Collector (HEC) lets you send data and application events to a Splunk deployment over the HTTP and Secure HTTP (HTTPS) protocols. This helps consolidate alert notifications from Prisma Cloud into Splunk so that your operations team can review and take action on the alerts.

1. To set up HEC, use instructions in Splunk documentation.
   - For source type, `_json` is the default; if you specify a custom string on Prisma Cloud, that value will overwrite anything you set here.
2. Select **Settings** > **Data inputs** > **HTTP Event Collector** and make sure you see HEC added in the list and that the status shows that it is **Enabled**.

**STEP 2** Set up the Splunk integration in Prisma Cloud.

1. Log in to Prisma Cloud.
2. Select **Settings** > **Integrations**.
3. Create a **+New Integration**.
4. Set **Splunk** as the **Integration Type**.
5. Enter an **Integration Name** and, optionally, a **Description**.
6. Enter the **Splunk HTTP Event Collector URL** that you set up earlier.
   - The Splunk HTTP Event Collector URL is a Splunk endpoint for sending event notifications to your Splunk deployment. You can either use HTTP or HTTPS for this purpose.
7. Enter **Auth Token**.
   - The integration uses token-based authentication between Prisma Cloud and Splunk to authenticate connections to Splunk HTTP Event Collector. A token is a 32-bit number that is presented in Splunk.
8. **(Optional)** Specify the **Source Type** if you want all Prisma Cloud alerts to include this custom name in the alert payload.

9. Click **Test** and then **Save** your changes.

   The integration status check for Splunk displays as red if the event collector URL is not reachable or times out or if the authentication token is invalid or receives an HTTP 403 response.

**STEP 3 | Create an Alert Rule** or modify an existing rule to receive alerts in Splunk. (See **Send Prisma Cloud Alert Notifications to Third-Party Tools**.)
Integrate Prisma Cloud with Amazon SQS

Prisma™ Cloud supports Amazon Simple Queue Service (SQS) to send alerts, and you can use the AWS CloudFormation service to enable custom workflows, as needed.

As soon as an alert is generated, the entire alert payload is sent to Amazon SQS. The alert payload also includes the command-line interface (CLI) remediation (the CLI commands and any instructions for those commands).

**STEP 1 | Configure Amazon SQS to receive Prisma Cloud alerts.**

1. Log in to the Amazon console with the necessary credentials to create and configure the SQS.
2. Click Simple Queue Services (under Application Integration).
3. Create New Queue or use an existing queue.

4. Enter a Queue Name and choose a Queue Type—**Standard** or **FIFO**.
5. Click **Configure Queue**.

   For the attributes specific to the Queue, use either the AWS default selection or set them per your company policies. Use SSE to keep all messages in the Queue encrypted, and select the default AWS KMS Customer Master Key (CMK) or enter your CMK ARN.

6. Create Queue.
This creates and displays your SQS Queue
7. Click the Queue that you created and view the Details and copy the URL for this queue.
You provide this value in Prisma Cloud to integrate Prisma Cloud notifications in to this Queue.

STEP 2 | If you are using a Customer Managed Key to encrypt queues in Amazon SQS, you must configure the Prisma Cloud Role with explicit permission to read the key.
1. On the Amazon console, select KMS > Customer Managed Keys and Create Key.
Refer to the AWS documentation for details on creating keys.
2. Enter an Alias and Description, and add any required Tags and click Next.
3. Select the IAM users and roles who can use this key through the KMS API and click Next.
4. Select the IAM users and roles who can use this key to encrypt and decrypt the data.
5. Review the key policy and click Finish.

STEP 3 | Enable read-access permissions to Amazon SQS on the IAM Role policy.
The Prisma Cloud IAM Role policy you use to onboard your AWS setup needs these permissions:

"sqs:GetQueueAttributes", "sqs:ListQueues", "sqs:SendMessage",
"sqs:SendMessageBatch", "tag:GetResources"

If you used the CFT templates to onboard your AWS account and the SQS queue belongs to the same cloud account, Prisma Cloud IAM Role policy already has the permissions required for Amazon SQS. If the SQS belongs to a different cloud account, you must provide the relevant IAM credentials (Access Key and Secret Key) when you enable the SQS integration in the next step.

STEP 4 | Set up Amazon SQS integration in Prisma Cloud.
1. Log in to Prisma Cloud.
2. Select Settings > Integrations.
3. Set the Integration Type to Amazon SQS.
4. Enter a Name and Description for the integration.
5. Enter the Queue URL that you copied when you configured Prisma Cloud in Amazon SQS.
6. Select More Options to provide the credentials if you want to use a different IAM role to access SQS.

By default, Prisma Cloud uses the same credentials with which you onboarded the AWS account to Prisma Cloud to access the SQS queue. If you want to include a different IAM role, you must enter the IAM Security Credentials—Access Key and Secret Key.
This IAM user must have `sqs:SendMessage` and `sqs:SendMessageBatch` permissions.

7. Click **Next** and then **Test**.
   
   You should receive a success message.

8. Click **Save**.
   
   After you set up the integration successfully, if the SQS URL is unresponsive, the status red ([Settings > Integrations](settings-integrations)) and green when the issue is resolved.

**STEP 5** | Create an Alert Rule or modify an existing rule to enable the Amazon SQS Integration.
Integrate Prisma Cloud with Amazon GuardDuty

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes Virtual Private Cloud (VPC) Flow Logs and AWS CloudTrail event logs. GuardDuty uses security logic and AWS usage statistics techniques to identify unexpected and potentially unauthorized and malicious activity.

Prisma™ Cloud integrates with GuardDuty and extends its threat visualization capabilities. Prisma Cloud starts ingesting GuardDuty data, correlates it with the other information that Prisma Cloud already collects, and presents contextualized and actionable information through the Prisma Cloud app.

**STEP 1** | Enable Amazon GuardDuty on your AWS instances (see Amazon Documentation).

**STEP 2** | Enable read-access permissions to Amazon GuardDuty on the IAM Role policy.

The Prisma Cloud IAM Role policy you use to onboard your AWS setup needs to include these permissions:

- `guardduty:List*`
- `guardduty:Get*`

If you used the CFT templates to onboard your AWS account, the Prisma Cloud IAM Role policy already has the permissions required for Amazon GuardDuty.

**STEP 3** | After Prisma Cloud has access to the Amazon GuardDuty findings, use the following RQL queries for visibility into the information collected from Amazon GuardDuty.

**Config Query:**

```
config where hostfinding.type = 'AWS GuardDuty Host'
```

**Network Query:**

```
network where dest.resource IN (resource where hostfinding.type = 'AWS GuardDuty Host')
```

Click on the resource to see the **Audit Trail**.
Click Host Findings for information related to vulnerabilities. Select AWS GuardDuty Host or AWS GuardDuty IAM in the filter to view vulnerabilities detected by AWS GuardDuty.
Integrate Prisma Cloud with AWS Inspector

Prisma™ Cloud ingests vulnerability data and security best practices deviations from AWS Inspector to provide organizations with additional context about risks in the cloud. You can identify suspicious traffic to sensitive workloads, such as databases with known vulnerabilities.

**STEP 1 |** Enable AWS Inspector on your EC2 instances. To set up AWS Inspector, see Amazon documentation.

**STEP 2 |** Enable read-access permissions to AWS Inspector on the IAM Role policy.

The Prisma Cloud IAM Role policy that you use to onboard your AWS setup needs these permissions:

```plaintext
inspector:Describe*
inspector:List*
```

If you used the CFT templates to onboard your AWS account, the Prisma Cloud IAM Role policy already has the permissions required for AWS Inspector.

**STEP 3 |** After the Prisma Cloud service begins ingesting AWS Inspector data, you can use the following RQL queries for visibility into the host vulnerability information collected from AWS Inspector.

- **Config queries:**

  ```plaintext
  config where hostfinding.type = 'AWS Inspector Runtime Behavior Analysis'
  config where hostfinding.type = 'AWS Inspector Security Best Practices'
  ```

**AWS Inspector Runtime Behavior Analysis**—Fetches all resources which are in violation of one or more rules reported by the AWS Runtime Behavior Analysis package.
**AWS Inspector Security Best Practices**—Fetches all resources which are in violation of one or more rules reported by the AWS Inspector security best practices package.

- Network queries:

  ```sql
  network where dest.resource IN ( resource where hostfinding.type = 'AWS Inspector Runtime Behavior Analysis' )
  ```

  ```sql
  network where dest.resource IN ( resource where hostfinding.type = 'AWS Inspector Security Best Practices' )
  ```

  Click on the resource to see an Audit trail.

  Click **Host Findings** for information related to vulnerabilities.
<table>
<thead>
<tr>
<th>Finding Type</th>
<th>Finding Name</th>
<th>Title</th>
<th>Status</th>
<th>Severity</th>
<th>CVSS Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Vulnerability</td>
<td>CVE-2014-5355</td>
<td>Instance i-0bc894efc1ae18153 is vulnerable to CVE-2014-5355</td>
<td>Open</td>
<td>High</td>
<td>9</td>
</tr>
<tr>
<td>Host Vulnerability</td>
<td>CVE-2014-8484</td>
<td>Instance i-0bc894efc1ae18153 is vulnerable to CVE-2014-8484</td>
<td>Open</td>
<td>High</td>
<td>9</td>
</tr>
<tr>
<td>Host Vulnerability</td>
<td>CVE-2014-8485</td>
<td>Instance i-0bc894efc1ae18153 is vulnerable to CVE-2014-8485</td>
<td>Open</td>
<td>High</td>
<td>9</td>
</tr>
</tbody>
</table>

Total 349 result(s)
Integrate Prisma Cloud with AWS Security Hub

You can use AWS Security Hub as a central console to view and monitor the security posture of your cloud assets on the Amazon AWS Security Hub console.

Integrate Prisma™ Cloud with AWS Security Hub for centralized visibility into security and compliance risks associated with your cloud assets on the AWS Security Hub console.

As part of the integration, Prisma Cloud monitors your assets on your AWS cloud and sends alerts about resource misconfigurations, compliance violations, network security risks, and anomalous user activities directly to the Security Hub console so that you have a comprehensive view of the cloud assets deployed on your AWS accounts.

**STEP 1** | Attach a AWS Security Hub read-only policy to your AWS administrator user role to enable this integration on the Amazon console.

1. Log in to the AWS console and select IAM.
2. Select **Users** and select the AWS administrator who is creating the integration.
3. Add permissions.

4. Attach existing policies Directly.

5. Select **AWSSecurityHubReadOnlyAccess** and then **Next: Review**.
6. **Add Permissions.**

**STEP 2** | Sign up for Prisma Cloud on AWS Security Hub.

1. Log in to the AWS console and select **Security Hub**.

2. Select **Settings > Integrations** and enter **Palo Alto Networks** as the search term.
3. Find **Palo Alto Networks: Prisma Cloud** and Enable Integration.

**STEP 3 |** Set up the AWS Security Hub Integration on Prisma Cloud.

Set up the AWS Security Hub as an integration channel on Prisma Cloud so that you can view security alerts and compliance status for all your AWS services from the AWS console.

1. Log in to Prisma Cloud.
2. Select Settings > Integrations.
3. Select +New Integration.
4. Select AWS Security Hub as the Integration Type.
5. Set the Integration Name to the AWS account to which you assigned AWS Security Hub read-only access.
6. Enter a Description and select a Region.
   You can select regions only if you enabled Prisma Cloud on AWS Security Hub for your cloud account.

7. Click Next and then Test.

   After you set up the integration successfully, if there is a permission exception for the enabled regions, the status turns red (Settings > Integrations) and turns green when the issue is resolved.

**STEP 4** Modify an existing alert rule or create a new alert rule to specify when to send alert notifications. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)
**STEP 5 | View Prisma Cloud alerts on AWS Security Hub.**

1. Log in to the AWS console and select **Security Hub**.
2. Click **Findings** to view the alerts.
3. Select the **Title** to view details about the alert description.
Integrate Prisma Cloud with Azure Service Bus Queue

Prisma™ Cloud can send alerts to a queue on the Azure Service Bus messaging service. To authorize access, you can either use a Shared Access Signature for limiting access permissions to the Service Bus namespace or queue, or use the service principal credentials associated with the Azure Cloud account you have onboarded to Prisma Cloud. If you plan to use the service principal that uses Azure Active Directory to authorize requests, you must include the additional role—Azure Service Bus Data Sender— and enable send access to the Service Bus namespace and queues.

When configured, as soon as an alert is generated, the entire alert payload is sent to the queue.

**STEP 1 |** Configure the Azure Service Bus to receive Prisma Cloud alerts.

1. Log in to the Azure portal, to create a Service Bus namespace and add a queue.

   Copy the queue URL.

   ![Azure Portal Screenshot](image1)

2. Choose your authentication method.

   To authenticate and authorize access to Azure Service Bus resources, you can either use Azure Activity Directory (Azure AD) or Shared Access Signatures (SAS).

   - **If you want to use Azure AD** Add the Azure Service Bus Data Sender role to the service principal associated with the Prisma Cloud App registered on your Azure AD tenant.

     Refer to the Azure documentation on assigning roles.

   ![Azure Role Assignments Screenshot](image2)

   - **If you want to use a SAS** Get the connection string to enable Prisma Cloud to authenticate to the Azure Service Bus namespace or queue.

     You can define the scope for the connection string to be the namespace or a specific queue. Refer to the Azure documentation for getting the connection string.

     You can either use the RootManageSharedAccessKey policy that enables access to the Service Bus namespace, and is created by default. This policy includes a Shared Access Signature (SAS) rule with an associated pair of primary and secondary keys that you can use on Prisma Cloud.
Or, you can limit access to a specific queue, and create a policy with the minimum permissions for send access to the Azure Service Bus queue.

**STEP 2** Add the Azure Service Bus Queue on Prisma Cloud.

1. Log in to Prisma Cloud.
2. Select **Settings > Integrations**.
3. Set the **Integration Type** to **Azure Service Bus Queue**.
4. Enter a **Name** and **Description** for the integration.
5. Enter the **Queue URL** that you copied earlier.
6. Select the method to authorize access to the queue.
   - Select **Azure Account** if you want to access the queue with the Prisma Cloud credentials which you used to onboard your Azure subscription. If you missed adding the **Azure Service Bus Data Sender** role to the service principal, an error message will display when you save the integration. Select the Azure account from the drop-down.
   - Select **Shared Access Signature**, if you want to use a role with limited permissions, and paste the connection string value for the scope selection.
7. Click **Next** and then **Test**.

You should receive a success message.

And can verify that the message count increments on the queue on the Azure portal.
8. Click **Save**.

When the communication is successful, the status of the integration is green (**Settings > Integrations**). If the Queue URL is unreachable or if permissions are insufficient, the status turns red.

**STEP 3 | Create an Alert Rule** or modify an existing rule to enable the Azure Service Bus Queue integration.
Integrate Prisma Cloud with Jira

Integrate Prisma™ Cloud with Jira and receive Prisma Cloud alert notifications in your Jira accounts. With this integration, you can automate the process of generating Jira tickets with your existing security workflow.

To set up this integration, you need to coordinate with your Jira administrator and gather the inputs needed to enable communication between Prisma Cloud and Jira.

This integration supports Jira Cloud and Jira On-Premises versions, and is qualified with the most recent GA versions of Jira.

**STEP 1 | Configure Prisma Cloud in your Jira account.**

1. Login to Jira as a Jira Administrator.
2. Locate Application Links.
   - For Jira Cloud, select Jira Settings > Products > Application Links.
   - For Jira On-Premises, select Settings > Applications > Application Links.
3. Enter the URL for your instance of Prisma Cloud in Configure Application Links and Create new link. See Access Prisma Cloud for details on the URL.

4. Disregard the message in Configure Application URL and Continue.

5. Enter the Application Name and set the Application Type to Generic Application.

6. Create incoming Link and Continue.
7. On Link Applications, specify a Consumer Key and a Consumer Name. Save the Consumer Key because you will need this value when you enter the information in Prisma Cloud.

8. Copy the Public Key shown below and Continue.

MIIBi JANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnYoXB+BZ555jUlfyN+0b3g7haTchseyWwdUtrTcebbN1jy5zjZCvp31://L9HzAOWCfmgj5hahFcm1lbCFY93oibsiWmMLgDYBghpManIQ73TEHDIAsV49z2TLtx0iRNSW65Cef+SMj/1hB59LPVIn0bf415ME1PcCJ3yow258a0T7TAJ0e1yyhC3iM +nVQXP+1V0ztqnso0UypA7UKvdI0Qf1ZsvlyHNNw1Ng7xgYc +H64cBmAgfcfDNzxyPmJ2kM7cGC2y4ukQIDAQAB
Prisma Cloud is listed in your Jira account after successful creation.

**STEP 2 | Setup Jira as one of the integration channels in Prisma Cloud.**

1. Login to Prisma Cloud.
2. Select **Settings > Integrations**.
3. + **Add New** integration.
4. Set **Integration** to **JIRA**.
5. Specify a meaningful **Integration Name** and, optionally, add a **Description**.
6. Enter the **JIRA Login URL**.
   - Make sure the URL starts with https and does not have a trailing slash ('/') at the end.
7. Enter the Consumer Key that you created when you created the Prisma Cloud application in Jira and **Generate Token**.
8. After you see the **Token Generated** message, click **Next**.
9. Click the secret key URL link to retrieve your secret key.
   The URL with the verification code is valid for only 10 minutes.
10. When redirected to the **Welcome to JIRA** page, **Allow** Prisma Cloud read and write access to data in your Jira account.

11. Copy the verification code displayed on the page, paste it as the **Secret Key**, and **Generate Token**, **Test** and **Save**.
The integration will be listed on the Integrations page.

**STEP 3** | Create Jira notification templates to configure and customize Prisma Cloud alerts.

The Jira fields that are defined as mandatory in your project are displayed in the template. The types of fields in Jira (such as text, list, single select check boxes, and option type fields) are supported in Prisma Cloud. If you add any other type of fields as mandatory in Jira (such as date fields), it will fail. Do not configure any Date fields as Mandatory and define any text fields in Jira as free-form text so that alert data is displayed correctly and completely.

1. Log in to Prisma Cloud.
2. From **Alerts**, select **Notification Templates** and **+Add New** template.
3. Select the Jira Notification template and enter a **Template Name**.
4. Select an **Integration**.
5. Select your **Project**.

  
  Select the project where you want to see the Prisma Cloud alerts. Because every alert translates to a Jira ticket, as a best practice, create and use a dedicated project for Prisma Cloud ticketing and issue management.

6. Select your **Issue Type** and click **Next**.
7. Select the **Jira Fields** that you would like to populate.

   The Jira fields that are defined as mandatory in your project are already selected and included in the alert.

8. Select information that goes in to **Summary** and **Description** from the alert payload.

9. Select the **Reporter** for your alert from users listed in your Jira project.

   This option is available only if the administrator who set up this integration has the appropriate privileges to modify the reporter settings on Jira.
10. **Click Next** to go to the review dialog and review your selection.

11. **Save** your changes.

You can delete or edit the Jira notification from the Notification Template dialog.
After you set up the integration, Prisma Cloud performs periodic status checks with Jira red if the Jira Login URL is not reachable or if any request to Jira results in an HTTP 400 or 403 response.

STEP 4 | Create an Alert Rule or modify an existing rule to send alerts to Jira.
Integrate Prisma Cloud with Qualys

Prisma™ Cloud integrates with the Qualys platform to ingest and visualize vulnerability data for your resources that are deployed on the AWS and Azure cloud platforms.

**STEP 1** | Gather the information that you need to set up the Qualys integration on Prisma Cloud.

- You must obtain the Qualys Security Operations Center (SOC) server API URL (also known as or associated with a POD—the point of delivery to which you are assigned and connected for access to Qualys).

  Get the API URL from your Qualys account (Help > About). The Qualys API URL is listed under **Qualys Scanner Appliances**. When you enter this URL in as the **Qualys API Server URL**, do not include :443.

- You must provide Qualys users with the privileges required to enable the integration using the Manager role, the Unit Manager role, or both. You can modify the Manager role to enable read-only access permission if needed. (Refer to the Qualys documentation for details about User Roles Comparison (Vulnerability Management).)

- You must enable Vulnerability Management (VM), Cloud Agent (CA), and Asset View (AV) for Qualys users.

- You must enable Qualys API and Qualys EC2 API access for Qualys users.

  **(AWS only)** You must configure Qualys Sensors for AWS cloud, such as Virtual Scanner Appliances, Cloud Agents, AWS Cloud Connectors, and Internet Scanners.

  The cloud agents or cloud connectors enable Prisma Cloud to retrieve vulnerability data so that you can correlate this data with your AWS asset inventory. (Refer to the Qualys documentation for more information.)
☐ **(Azure only)** For Azure accounts, deploy the Qualys Virtual Scanner Appliance using Microsoft Azure Resource Manager (ARM) (see the Qualys documentation).

You can use Qualys Cloud Agents (Windows and Linux) for Azure instances from the Azure Security Center console to view vulnerability assessment findings within Azure Security Center and your Qualys subscription (see Qualys Documentation.)

☐ **(Azure only)** Make sure that Azure VM Information is visible in Qualys.

---

### STEP 2 | Set up Qualys Integration on Prisma Cloud.

1. Select **Settings > Integrations**.
2. Create a **+New Integration**.
3. Set the **Integration Type** to Qualys.
4. Enter an **Integration Name** and **Description**.
5. Enter the **Qualys API Server URL (without http[s])**.
   
   This is the **API URL** for your Qualys account. When you enter this URL, do not include the protocol (http(s)) or the port (:443).
6. Enter your Qualys **User Login** and **Password**.
7. **Save** your changes.

   The integration will be listed on the Integrations dialog, where you can enable, disable, or delete integrations as needed.

**STEP 3** | View Qualys host vulnerability data in Prisma Cloud.

After you configure Prisma Cloud with access to the Qualys findings, you can use RQL queries for visibility into the host vulnerability information collected by Qualys.

1. **Use Config Query** for visibility for host vulnerabilities.

   ```
   config where hostfinding.type = 'Host Vulnerability'
   ```
Click a resource to get information about vulnerabilities. View the **Audit Trail** to see the CVE numbers.

Click **Host Findings** for information related to vulnerabilities. The Source column in Host Findings displays the Qualys icon to help you easily identify the source for the vulnerability findings.
Network Query

network where dest.resource IN ( resource where hostfinding.type = 'Host Vulnerability' )

STEP 4 | Use the Qualys APIs on the CLI to confirm if API access is enabled for your account.

If you have trouble connecting with Qualys API, enter your username, password, and the URL for the Qualys service in the following Curl examples:

```bash
curl -H "X-Requested-With: Curl Sample" -u "Username:Password" "https://qualysapi.qg1.apps.qualys.in/api/2.0/fo/scan/?action=list&echo_request=1"

curl -k "https://qualysapi.qg1.apps.qualys.in/msp/asset_group_list.php" -u "Username:Password"
```
curl -k -H "X-Requested-With:curl" "https://qualysapi.qg1.apps.qualys.in/api/2.0/fo/scan/stats/?action=list" -u "Username:Password"
Integrate Prisma Cloud with Google Cloud Security Command Center (SCC)

Integrate Prisma™ Cloud with Google Cloud Security Command Center (SCC) for centralized visibility in to security and compliance risks associated with your cloud assets on the Google Cloud Platform (GCP).

You can set up this integration for a GCP Organization that you are monitoring with Prisma Cloud. The alerts generated by Prisma Cloud for GCP accounts based on your alert rule are posted to Google Cloud SCC. To show Prisma Cloud alerts in Google Could SCC for cloud accounts of other cloud types (such as AWS and Azure), contact Prisma Cloud support on the Palo Alto Networks LIVE Community.

**STEP 1 |** The service account you use to onboard the GCP Organization into Prisma Cloud should include Viewer, Organization Viewer, and Security Center Findings Editor roles.

**STEP 2 |** To view assets and findings on the Cloud SCC console, enable the **Cloud Security Command Center API**.

1. Go to the GCP Console API Library and select your GCP project.
   
   Make sure to enable the **Cloud Security Command Center API** in the project owns the Service Account that you will use to onboard the GCP Organization into Prisma Cloud.

2. **Enable APIs and Services.**
3. Enable the **Cloud Security Command Center API**.

**STEP 3 |** Sign up for the Prisma Cloud SCC solution on the Google console.

A security center administrator can set up this integration on the Google console.

1. Go to the Google Console and search for **Prisma Cloud CSCC**.
2. **Visit Palo Alto Networks site to Signup.**
3. Select the organization that you onboarded in to Prisma Cloud.
4. Select the **Service account** you used to onboard the GCP Organization.
5. Copy the Source ID. You need the Source ID when you set up this integration in Prisma Cloud.

6. Click Done.

**STEP 4** | Set up Google Cloud SCC as one of the integration channels in Prisma Cloud.

1. Log in to Prisma Cloud.
2. Select Settings > Integrations.
3. Create a +New Integration.
4. Select CSCC as the Integration Type.
5. Specify a meaningful Integration Name and Description.
6. Enter the Source ID that you copied Prisma Cloud
7. Select the **GCP Organization**.
8. Click **Next** and then **Test**.

   For a successful integration, you must configure adequate permissions for the service account (as listed above). After you successfully set up the integration, the status **Settings > Integrations** turns red when there are any issues and green when there are no issues or all issues are resolved.

**STEP 5 | Create an Alert Rule** or modify an existing rule to send alerts to Google Cloud SCC. See **Send Prisma Cloud Alert Notifications to Third-Party Tools**.

**STEP 6 | View alerts in Cloud SCC.**

1. Go to the **Google Console** and select **Security > Security Command Center**.

2. Click **Findings** to view the alerts.
3. Select the rule to see the details about the alerts.

Finding Details

Summary

Finding type
GCP Firewall rule allows internet traffic to HTTP port (80)

First discovered
Jan 4, 2019 7:38 AM (6 days ago)

Most recently seen
Jan 4, 2019 7:38 AM (6 days ago)

Source
RedLock CSICC

Security marks
No marks

Attributes

category
GCP Firewall rule allows internet traffic to HTTP port (80)

eventTime
January 4, 2019 at 7:38:48 AM UTC+3:00

externalUrl
https://app2.dev.redlock.io/alerts/filters#/alerts?P-1176187?timeType=now&timeUnit=epoch

firstDiscovered
January 4, 2019 at 7:38:48 AM UTC+3:00

id
organizations/24920024936/sources/13004436760025870642/findings/C2U5Lnd3d8y6gqcwi5oPi1176187

parent
organizations/24920024936/sources/13004436760025870642

resourceName
custom-network1 allow-http

sourceDisplayName
RedLock CSICC

Source Properties

Account Id
RedLock-dev-playground

Account Name
RedLock GCP Dev Playground

Close
Integrate Prisma Cloud with Tenable

Prisma™ Cloud ingests vulnerability data from Tenable to provide you with additional context about risks in the cloud. This integration enables you to, for example, identify suspicious traffic to sensitive workloads, such as databases with known vulnerabilities.

AWS, Azure, and GCP clouds support the Prisma Cloud integration with Tenable.

**STEP 1** | Tenable.IO provides API access to assets and their vulnerability information. Configure the Tenable account to use the Tenable AWS, Azure, and GCP connectors. Without connectors, you cannot identify the cloud resource.

The Tenable API requires an access key and secret key in the header. Generate an access key and secret key per user on the Tenable.io app. (See Tenable documentation for information.) Also, make sure that the Tenable role that you use to enable this integration has administrator permissions that include vulns-request-export and assets-request-export API access.

**STEP 2** | Set up Tenable integration on Prisma Cloud.

1. Select **Settings > Integrations**.
2. Set the **Integration Type** to **Tenable**.
3. Enter an **Integration Name** and **Description**.
4. Enter the **Access Key** and the **Secret Key** that are generated in Tenable.io. See Tenable documentation for information.

5. Click **Next** and Test the integration.

**STEP 3** | View vulnerabilities detected by Tenable in Prisma Cloud.

1. After Prisma Cloud has access to the Tenable findings, you can use the following RQL queries for visibility into the host vulnerability information collected from Tenable.

   **Config Query**
config where hostfinding.type = 'Host Vulnerability' AND hostfinding.source = 'Tenable' AND hostfinding.severity = 'high'

Select a resource to get information about vulnerabilities. Select Audit Trail to view the CVE numbers.

Network Query
network where dest.resource IN ( resource where hostfinding.type = 'Host Vulnerability' )

Click Host Findings to see details.
Integrate Prisma Cloud with ServiceNow

Integrate Prisma™ Cloud with ServiceNow and get automatically notified about Prisma Cloud alerts through ServiceNow tickets to prioritize incidents and vulnerabilities that impact your business. Prisma Cloud integrates with the ITSM module (incident table), the Security Incident Response module (sn_si_incident table), and the Event Management modules (em_event table) on ServiceNow to generate alerts in the form of ITSM Incident, Security Incident, and Event tickets. After you enable the integration, when Prisma Cloud scans your cloud resources and detects a policy violation, it generates an alert and pushes it to ServiceNow as a ticket. When you dismiss an alert on Prisma Cloud, Prisma Cloud sends a state change notification to update the ticket status on ServiceNow. This integration seamlessly fits in to the existing workflows for incident management (ITSM), security operations management (Security Incident Response) or event management for your organization.

The Prisma Cloud integration with ServiceNow is qualified with the most recent GA versions of ServiceNow.

If you are using a ServiceNow developer instance, make sure that it is not hibernating.

1. Set Up Permissions on ServiceNow
2. Enable the ServiceNow Integration on Prisma Cloud
3. Set up Notification Templates
4. View Alerts

If you see errors, review how to Interpret Error Messages.

Set Up Permissions on ServiceNow

To integrate Prisma Cloud and ServiceNow, you must have the privileges on ServiceNow to configure users, roles, fields on ServiceNow, which then allow you to set up the data mapping for the Notification Templates on Prisma Cloud.

If you do not have the privileges required listed below, you must work with your ServiceNow administrator.

- Prerequisites for the Prisma Cloud and ServiceNow Integration
  1. You must have permissions to create a local user account on ServiceNow.

     Create a Username and password that are local on the instance itself. A local user account is a requirement because the ServiceNow web services cannot authenticate against an LDAP or SSO Identity provider and it is unlike the authentication flow that ServiceNow supports for typical administrative users who access the service using a web browser. Refer to the ServiceNow documentation for more information.
2. Review the ServiceNow roles required.

PrismaCloud has verified that the following roles provide the required permissions. If your implementation has different roles and RBAC mechanisms, work with your ServiceNow administrator.

- **London**
  1. `usage_admin` for checking the `v_plugin` table for whether Security Incident Response is enabled.
  2. *(Optional)* `personalize` for accessing tables.
     
     Personalize role is recommended to support type-ahead fields in notification templates for ServiceNow on Prisma Cloud. With this permission, when you enter a minimum of three characters in a type-ahead field, this role enables you to view the list of available options. If you do not enable personalize permissions, you must give table specific read-access permissions for type-ahead inputs.
  3. `sn_si.basic` role for all operations related to Security Incident Response

- **Madrid, New York, and Orlando**
  1. *(Optional)* `personalize` for accessing tables.
     
     Personalize role is recommended to support type-ahead fields in notification templates for ServiceNow on Prisma Cloud. With this permission, when you enter a minimum of three characters in a type-ahead field, this role enables you to view the list of available options. If you do not enable personalize permissions, you must give table specific read-access permissions for type-ahead inputs.
  2. `evt_mgmt_integration` basic role has create access to the Event `[em_event]` and Registered Nodes `[em_registered_nodes]` tables to integrate with external event sources.

- For the user you added earlier, create a custom role with the permissions listed above.

   These permissions are required to create tickets and access the data in the respective ITSM, Events, and Security Incident Response tables and fields on ServiceNow.

   Prisma Cloud needs access to the Plugins (`V_plugin`), Dictionary (`sys_dictionary`), and Choice Lists (`sys_choices`) tables to fetch data from the ServiceNow fields. You can view this information in the ServiceNow notification templates that enable you to customize Prisma Cloud alerts in ServiceNow.

   1. Select **User Administration > Roles** to create a new role and assign it to the local administrative user you created earlier.
2. Pick a table, such as the **Plugins** table, and select the menu (“hamburger”) icon next to a table column header to **Configure > Table**.

3. **Elevate the role** to security_admin to enable modification of the access control list (ACL).

4. Select **Access Controls > New**.

5. Set **Operation** to **Read** and assign this permission to the role.

6. Enable permissions for the remaining tables and assign them to the same role.

   Verify that all three tables—Plugins (**V_plugin**), Dictionary (**sys_dictionary**), and Choice Lists (**sys_choices**) have the role and the required permission especially if you have defined field-level ACL rules to restrict access to objects in your ServiceNow implementation.

4. You must be familiar with the fields and field-types in your ServiceNow implementation to set up the Notification templates on Prisma Cloud. Because this knowledge is essential for setting up the mapping of the Prisma Cloud alert payload to the corresponding fields on ServiceNow, you must work with your ServiceNow administrator to successfully enable this integration.

- **Prerequisites for the Security Incident Module**
  
  The Security Incident Response plugin is optional but is required if you want to generate Security Incident tickets. To create Security Incident tickets, you must also have the Security Incident Response plugin installed on your ServiceNow instance.

  Verify that the Security Incident Response plugin is **activated**. To activate a plugin you must be ServiceNow administrator; if you do not see the plugin in the list, verify that you have purchased the subscription.

- **Prerequisites for the Event Management Module**
  
  The Event Management plugin is optional but is required if you want to generate Event tickets on ServiceNow. To create Event tickets, you must have the Event Management subscription and the plugin installed on your ServiceNow instance.

  Verify that the Event Management plugin is **activated**. To activate a plugin you must be ServiceNow administrator; if you do not see the plugin in the list, verify that you have purchased the subscription.
Enable the ServiceNow Integration on Prisma Cloud

Set up ServiceNow as an external integration on Prisma Cloud.

STEP 1 | Log in to Prisma Cloud and select **Settings > Integrations > +Add New.**

STEP 2 | Set the **Integration Type** to **ServiceNow**.

STEP 3 | Enter a meaningful **Integration Name** and a **Description**.

STEP 4 | Enter your **FQDN** for accessing ServiceNow.

   Make sure to provide the FQDN for ServiceNow—not the SSO redirect URL or a URL that enables you to bypass the SSO provider (such as sidedoor or login.do) for local authentication on ServiceNow. For example, enter `<yourservicenowinstance>.com` and not any of the following:

   - https://www.<yourservicenowinstance>.com
   - <yourservicenowinstance>.com/
   - <yourservicenowinstance>.com/sidedoor.do
   - <yourservicenowinstance>.com/login.do

   *If you switch the FQDN from one ServiceNow instance to another, state change notifications for existing alerts will fail.*

STEP 5 | Enter the **Username** and **Password** for the ServiceNow administrative user account.

   The ServiceNow web services use the SOAP API that supports basic authentication, whereby the administrative credentials are checked against the instance itself and not against any LDAP or SSO Identity provider. Therefore, you must create a local administrative user account and enter the credentials for that local user account here instead of the SSO credentials of the administrator. This method is standard for SOAP APIs that pass a basic authentication header with the SOAP request.

STEP 6 | Select the **Service Type** for which you want to generate tickets—**Security**, **Incidents**, and/or **Event**.

   You must have the plugin installed to create **Security** incident tickets or **Event** tickets; make sure to work with your ServiceNow administrator to install and configure the Security Incident Response module or Event Management module. If you select **Security** only, Prisma Cloud generates all tickets as Security Incident Response (SIR) on ServiceNow.

STEP 7 | Click **Next** and then **Test**.

   If you have omitted any of the permissions listed in Set Up Permissions on ServiceNow, an HTTP 403 error displays.
STEP 8 | **Test** and **Save** the integration.

Continue with setting up the notification template, and then verify the status of the integration on **Settings > Integrations**.

---

**Set up Notification Templates**

Notification templates allow you to map the Prisma Cloud alert payload to the incident fields (referred to as *ServiceNow fields* on the Prisma Cloud interface in the screenshot) on your ServiceNow instance. Because the incident, security, and event tables are independent on ServiceNow, to view alerts in the corresponding table, you must set up the notification template for each service type — *Incidents*, *Events* or *Security Incidents* on Prisma Cloud.

**STEP 1** | Log in to Prisma Cloud and select **Alerts > Notification Templates**.

**STEP 2** | **Add New** notification template, and choose the template for ServiceNow.

**STEP 3** | Enter a **Template Name** and select your **Integration**.

Use descriptive names to easily identify the notification templates.

**STEP 4** | Set the **Service Type** to **Security**, **Incident** or **Event**.

The options in this drop-down match what you selected when you enabled the ServiceNow integration on Prisma Cloud.
STEP 5 | Click **Next** and select the alert status for which you want to set up the ServiceNow fields. You can choose different fields for the Open, Dismissed, or Resolved states. The fields for the Snoozed state are the same as that for the Dismissed state.

STEP 6 | Select the **ServiceNow Fields** that you want to include in the alert.

Prisma Cloud retrieves the list of fields from your ServiceNow instance dynamically, and it does not store any data. Depending on how your IT administrator has set up your ServiceNow instance, the configurable fields may support a drop-down list, long-text field, or type-ahead. For a type-ahead field, you must enter a minimum of three characters to view a list of available options. When selecting the configurable fields in the notification template, at a minimum, you must include the fields that are defined as mandatory in your ServiceNow implementation.

In this example, **Description** is a long-text field, hence you can select and include the Prisma Cloud Alert Payload fields that you want in your ServiceNow Alerts. You must include a value for each field you select to make sure that it is included in the alert notification. See Alert Payload for details on the context you can include in alerts.

To generate a ServiceNow Event, **Message Key** and **Severity** are required. The **Message Key** determines whether to create a new alert or update an existing one, and you can map the **Message Key** to Account Name or to Alert ID based on your preference for logging Prisma Cloud alerts as a single alert or multiple alerts on ServiceNow. **Severity** is required to ensure that the event is created on ServiceNow and can be processed without error; without severity, the event is in an Error state on ServiceNow.

For **Number**, use AlertID from the Prisma Cloud alert payload for ease of scanning and readability of incidents on ServiceNow.
STEP 7 | Click Next to go to the review pane and review your selection.

STEP 8 | Test and Save your changes.

After you set up the integration and configure the notification template, Prisma Cloud uses this template to send alerts to your ServiceNow instance. When the communication is successful, the status of the integration is green on Settings > Integrations. If the ServiceNow instance URL is unreachable or if your credentials are invalid, the status turns red. When a failure occurs, Prisma Cloud performs periodic checks to verify the connection status.

The status, however, does not transition to red if:

- Prisma Cloud cannot resolve the alert or update an existing alert field for a deleted record or missing record on your ServiceNow instance.

- Prisma Cloud is unable to send a test message to ServiceNow because of an HTTP 404 error.
Interpret Error Messages

The following table displays the most common errors when you enable the ServiceNow integration on Prisma Cloud.

<table>
<thead>
<tr>
<th>What is Wrong?</th>
<th>Error Message that Displays</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ServiceNow URL you entered is incorrect.</td>
<td>You must provide an IP address or an FQDN without the protocol http or https</td>
</tr>
<tr>
<td>The ServiceNow URL you entered is invalid.</td>
<td>The FQDN is invalid it should be a valid host name or IP address.</td>
</tr>
<tr>
<td>The ServiceNow URL you entered is not reachable.</td>
<td>The FQDN provided is either not reachable or is an invalid ServiceNow instance.</td>
</tr>
<tr>
<td>A required field is missing in the ServiceNow configuration.</td>
<td>Missing Required Field - {{param}}</td>
</tr>
<tr>
<td>Your ServiceNow username or password is not valid or is inaccurate.</td>
<td>Invalid Credentials</td>
</tr>
<tr>
<td>The ServiceNow permissions you have enabled are not adequate.</td>
<td>Required roles or Plugins is/are missing for {{table}}</td>
</tr>
<tr>
<td>The Notification template for this integration does not have adequate permissions.</td>
<td>Insufficient permission to read the field from {{table}} table</td>
</tr>
<tr>
<td></td>
<td>Error Fetching Suggestions For {{table}}</td>
</tr>
<tr>
<td>The ServiceNow integration is not successfully configured.</td>
<td>Failed Service Now Test - {{reason}}</td>
</tr>
</tbody>
</table>

View Alerts

Verify that the integration is working as expected. On the incidents view in ServiceNow, add the Created timestamp in addition to the same columns you enabled in the Prisma Cloud notification template to easily correlate alerts across both administrative consoles.

**STEP 1** | Modify an existing Alert Rule or create a new Alert Rule to send alert notifications to ServiceNow. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)

**STEP 2** | Login to ServiceNow to view Prisma Cloud alerts.
When alert states are updated in Prisma Cloud, they are automatically updated in the corresponding ServiceNow tickets.

1. To view incidents (incident table), select **Incidents**.

   In ServiceNow, all the Open Prisma Cloud have an incident state of **New** and all the Resolved or Dismissed alerts have an incident state of **Resolved**.

2. To view security incidents (sn_si_incident table), select **Security Incidents**.

   In ServiceNow, all the Open Prisma Cloud alerts have a state of **Draft** and all the Resolved or Dismissed alerts have a state of **Review**.

3. To view event incidents (events table), select **Event Management > All Events**.
<table>
<thead>
<tr>
<th>Time of event</th>
<th>Source</th>
<th>Description</th>
<th>Node</th>
<th>Type</th>
<th>Resource</th>
<th>Message key</th>
<th>State</th>
<th>Severity</th>
<th>Alert</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-03-30 17:20:45</td>
<td>Alert P-0276</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0276</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0280</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0280</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0278</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0278</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0283</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0283</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0279</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0279</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0282</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0282</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0277</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0277</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0281</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0281</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:47</td>
<td>Alert P-0236</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0236</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:46</td>
<td>Alert P-0289</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0289</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:46</td>
<td>Alert P-0290</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0290</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:46</td>
<td>Alert P-0231</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0231</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:46</td>
<td>Alert P-0287</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0287</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020-03-30 17:20:46</td>
<td>Alert P-0288</td>
<td>AlertRuleName event_modul...</td>
<td>Alert P-0288</td>
<td>Processed</td>
<td>Critical</td>
<td>Alert011999</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Integrate Prisma Cloud with Webhooks

Integrate Prisma™ Cloud with webhooks to send Prisma Cloud alerts to webhooks and pass information to any third-party integrations that are not natively supported on Prisma Cloud. Your incident response teams can monitor the webhook channels to investigate and remediate security incidents. With a webhook integration, you can configure Prisma Cloud to send information to the webhook as an HTTP POST request as soon as an alert is generated. And if you have internal services or applications that subscribe to the webhook, these subscribing clients can get data immediately in JSON format.

STEP 1 | Obtain your Webhook URL.

If you have additional details that you want to include in the payload to enable additional security or to verify the authenticity of the request, you can include these as key-value pairs in a custom header.

STEP 2 | Set up webhooks as an integration channel on Prisma Cloud.

1. Log in to Prisma Cloud and select Settings > Integrations.
2. +Add New integration.
3. Set the Integration Type to webhooks.
4. Enter your Webhook URL.
5. Add any custom HTTP Headers as key-value pairs.
   You can, for example, include an authentication token in the custom header. The integration includes Content-Type as a default header and you cannot edit it.
6. Test and Save the integration.
   After you set up the integration successfully, the status (Settings > Integrations) turns red when the webhook URL is unreachable or when Prisma Cloud cannot authenticate to it successfully and turns green when there aren't any issues or the issues are resolved.

STEP 3 | Modify an existing alert rule or create a new alert rule to send alert notifications to webhook. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)

STEP 4 | View the alert POST requests on Webhook.

   In one POST request, alerts are sent in a batch of 30.
Configure External Integrations on Prisma Cloud
Integrate Prisma Cloud with PagerDuty

Integrate Prisma™ Cloud with PagerDuty to aid alerting, on-call scheduling, escalation policies, and incident tracking to increase uptime of your apps, servers, websites, and databases. When integrated, Prisma Cloud sends alerts to the PagerDuty service so that your incident response teams are notified to investigate and remediate the security incidents.

**STEP 1** | Add a new service in PagerDuty and get the integration key. You have to provide this integration key in Prisma Cloud.
1. Log in to PagerDuty.
2. Click **Configuration > Services** and add a **+New Service**.

3. Complete the **Add a Service** form.
4. In the **Integration Settings**, set the **Integration Type** to **Use our API Directly** and select **Events API V2**.

5. After you fill out all the details, **Add Service**.
6. Copy and save the **Integration Key**.

You will need to enter this integration key in Prisma Cloud when you add this integration.
STEP 2 | Set up PagerDuty as an integration channel on Prisma Cloud.

1. Log in to Prisma Cloud and select Settings > Integrations.
2. +Add New integration.
3. Set the Integration Type to pagerduty.
4. Enter the Integration Key of your service on PagerDuty.
5. Click Next and then Test.

6. Save the integration.

Prisma Cloud creates a test incident and sends it to your service in PagerDuty. To make sure that this integration is successful, look for the test integration in your PagerDuty Service.
STEP 3 | Modify an existing alert rule or create a new alert rule to send alert notifications to PagerDuty. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)

STEP 4 | View Prisma Cloud in PagerDuty.

In PagerDuty, all the open alerts display the Incident State as **Triggered** and all the resolved alerts display the Incident State as **Resolved**.
Integrate Prisma Cloud with Microsoft Teams

Microsoft Teams is a cloud-based team collaboration software that is part of the Office 365 suite of applications and is used for workplace chat, video meetings, file storage, and application integration.

Prisma™ Cloud integrates with Microsoft Teams and monitors your assets and sends alerts on resource misconfigurations, compliance violations, network security risks, and anomalous user activities either as they happen or as a consolidated summary card—determined by how you configure alert notifications. Each alert message is a webhook notification that includes details such as the cloud type, policy name, and the resource that triggered the alert and the message card indicates the severity with a red (high), yellow (medium) or gray (low) line for easy scanning.

**STEP 1 |** Set up Microsoft Teams to view alert notifications from Prisma Cloud.

You must create an incoming webhook connector on a new channel or on a pre-existing channel on Microsoft Teams to enable the integration with Prisma Cloud. This webhook channel helps consolidate alert notifications from Prisma Cloud in Microsoft Teams so that your operations team can review and take action on the alerts. To enable this integration, you must have administrative privileges or contact the Microsoft 365 administrator who manages Team settings for your organization.

1. In Microsoft Teams, click More options (…) next to the channel name and select Connectors.
2. Scroll to Incoming Webhook, Add a webhook, and Install it.
3. Enter a name for the webhook and Create it.
4. Copy the webhook URL to the clipboard and save it before you click Done.

**STEP 2 |** Set up Microsoft Teams on Prisma Cloud.

1. Log in to Prisma Cloud.
2. Select Settings > Integrations.
3. +Add New integration.
4. Set Microsoft Teams as the Integration Type.
5. Enter the Integration Name and, optionally, a Description.
6. Paste the Webhook URL that you previously copied from Microsoft Teams.
7. Test and then Save.

*After successful integration in the Microsoft Teams conversation, you will receive a test message card with a green line.*

**STEP 3 |** Create an Alert Rule or modify an existing rule to receive alerts in Microsoft Teams. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)

The message card includes information on the policy rules and the resource names that have violated the policy and it includes a direct link with the relevant filters to access Prisma Cloud and view the alert or scheduled notification summary directly in the app.
Integrate Prisma Cloud with Cortex XSOAR

With the Prisma™ Cloud and Cortex XSOAR (formerly Demisto) outbound or push-based integration, you can send a Prisma Cloud alert generated by a policy violation to Cortex XSOAR. This integration enables your Security operations team to define custom playbooks or use the out-of-box playbooks on Cortex XSOAR to create multi-step workflows for incident management of your cloud resources; this is an alternative to the pull-based integration that you can configure from Cortex XSOAR.

Using the policy ID in the alert, Cortex XSOAR categorizes the alert as a specific incident type. For an incident type, the Prisma Cloud alert payload is mapped to a Cortex XSOAR layout that specifies the incident fields for data classification and mapping on Cortex XSOAR. The current list of incident types are: AWS CloudTrail Misconfiguration, AWS EC2 Instance Misconfiguration, AWS IAM Policy Misconfiguration, and Prisma Cloud. If the policy ID is not categorized to a specific incident type, it is automatically mapped to the generic Prisma Cloud incident type. Every incident type is mapped to a Cortex XSOAR layout and associated with a playbook to enable autoremediation of the violating resource, except for the generic Prisma Cloud incident type.

On autoremediation, Prisma Cloud performs a scan that detects that the issue is resolved and marks the alert as resolved.

Currently, this integration does not support the use of notification templates and Prisma Cloud does not receive state change notifications from Cortex XSOAR after it resolves an open alert.

- Enable the Cortex XSOAR Integration on Prisma Cloud
- Set Up the Integration on Cortex XSOAR

Enable the Cortex XSOAR Integration on Prisma Cloud

Set up Cortex XSOAR as an external integration on Prisma Cloud. If you have a firewall or cloud Network Security Group between the internet and Cortex XSOAR, you must add the NAT Gateway IP Addresses for Prisma Cloud to the allow list and enable the connection to Prisma Cloud.

For the push-based integration, you must use Cortex XSOAR version 5.0.0 and content release version 19.10.2 or later.

**STEP 1** | Log in to Prisma Cloud and select Settings > Integrations > +Add New.

**STEP 2** | Set the Integration Type to Cortex XSOAR.

**STEP 3** | Enter a meaningful Integration Name and a Description.

**STEP 4** | Enter your Cortex XSOAR Instance FQDN/IP address.

If you are adding a Cortex XSOAR instance that is part of a multi-tenant deployment, enter the tenant URL without the protocol (http or https).

**STEP 5** | Enter the API Key associated with the Cortex XSOAR administrative user account.

The API key you provide must belong to a Cortex XSOAR administrative user who has read-write permissions, which are required to enable this push-based integration. Within Cortex XSOAR, navigate to Settings > Integrations > API Keys and Get Your Key.
STEP 6 | Click Next and then Test.

STEP 7 | Save the integration.

After you set up the integration, the status indicates whether Prisma Cloud is connected to Cortex XSOAR.

STEP 8 | Modify an existing Alert Rule or create a new Alert Rule to send alert notifications to Cortex XSOAR. (See Send Prisma Cloud Alert Notifications to Third-Party Tools.)

STEP 9 | Get your Prisma Cloud Access Key.

If you do not have an access key, see Create and Manage Access Keys. You need the Access Key ID and Secret Key ID to complete the integration on Cortex XSOAR.

STEP 10 | Set Up the Integration on Cortex XSOAR.

Set Up the Integration on Cortex XSOAR

Before you can view Prisma Cloud alerts as incidents on Cortex XSOAR, you need content release 19.10.2 or a later version. The content release includes the incident fields required for this push-based integration. When you have the content release, the Classifier, incident types, and layouts are available automatically.

Cortex XSOAR maps Prisma Cloud alerts to out-of-the-box incident types such as AWS CloudTrail Misconfiguration, AWS EC2 Instance Misconfiguration, AWS IAM Policy Misconfiguration, GCP Compute Engine Misconfiguration, and Prisma Cloud. The out-of-box, Incident Layouts map the Prisma Cloud alert data to the classification rules. These layouts provide the Incident Classifier & Mapping that is required for classifying incidents to the correct incident type and mapping the fields in the Prisma Cloud alert payload to the Cortex XSOAR incident fields. When an incident is created, the playbook attached to the incident type automatically executes.

Find all the Cortex XSOAR playbooks that are available to support remediation on Prisma Cloud for example Prisma Cloud Remediation - AWS EC2 Instance Misconfiguration and Prisma Cloud Remediation - GCP VPC Network Misconfiguration; search for playbook-PrismaCloudRemediation_.

If you want to use the pull-based integration from Cortex XSOAR, see Cortex documentation. In a pull-based integration, you must enable the instance to Fetches incidents.
STEP 1 | Install Cortex XSOAR content release 19.10.2 or a later version on your Cortex XSOAR 5.0.0 or later instance.

19.10.2 is the minimum content release version that includes the Prisma Cloud incident fields required for this push-based integration. You can see the incident fields on Settings > Advanced > Fields.

STEP 2 | Enable the connection between Cortex XSOAR and Prisma Cloud.

1. Navigate to Settings > Integrations > Servers&Services.
2. Search for Prisma Cloud and Add Instance.
3. Complete the set up.

   Provide a Name for the Prisma Cloud instance you are integrating (the name must be unique from other Integrations within cortex XSOAR), the Server URL that corresponds to the API endpoint for the Prisma Cloud instance, and your access key and secret keys as username and password.

   If you access your Prisma Cloud instance at https://app2.eu.prismacloud.io, the API endpoint is https://api2.eu.prismacloud.io

STEP 3 | Review the classification mapping for incident types.

When Prisma Cloud pushes alerts to the Cortex XSOAR endpoint, the alerts are classified under the Prisma Cloud App (/prismacloud_app) path, and listed in Settings > Integrations > Classification Mapping

and the playbooks associated with each incident type are in Settings > Integrations > Advanced > Incident Types
**STEP 4 |** View incidents on Cortex XSOAR.

Verify that the integration is working as expected and that Prisma Cloud alerts display as incidents and are mapped to specific incident types.

**STEP 5 | (Optional)** Create additional classification and mapping rules and incident layouts to classify Prisma Cloud alerts to distinct incident types on Cortex XSOAR.

Cortex XSOAR includes a few incident types for Prisma Cloud to which you can associate one of the AWS playbooks (listed above) for autoremediation. Refer to the Cortex XSOAR documentation for detailed instructions about customizing your incident types, creating different classifications, mapping and layouts for Prisma Cloud alerts, and to associate different playbooks to take action and enable incident resolution for other cloud platforms. Refer to the Cortex XSOAR GitHub repository for some sample packs.
## Prisma Cloud Integrations—Supported Capabilities

The following table provides the details of features supported on each integration with Prisma™ Cloud.

<table>
<thead>
<tr>
<th>Integration</th>
<th>Integration Method</th>
<th>User Attribution</th>
<th>Notification Template</th>
<th>Alert Notification Delay</th>
<th>Status Check</th>
<th>State Change Notification</th>
<th>Alert Notification Grouping</th>
<th>Frequency of Alert Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Guard Duty</td>
<td>Pull</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>AWS Inspector</td>
<td>Pull</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>AWS Security Hub</td>
<td>Push</td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Amazon SQS</td>
<td>Push</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Aure Service Bus Queue</td>
<td>Push</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Demisto</td>
<td>Push</td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Email</td>
<td>Push</td>
<td>—</td>
<td>✓</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Google Cloud SCC</td>
<td>Push</td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Jira</td>
<td>Push</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Microsoft Teams</td>
<td>Push</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PagerDuty</td>
<td>Push</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ServiceNowPush</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Slack</td>
<td>Push</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Splunk</td>
<td>Push</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Integration Method</td>
<td>User Attribution</td>
<td>Notification Template</td>
<td>Alert Notification Delay</td>
<td>Status Check</td>
<td>State Change Notification</td>
<td>Alert Notification Grouping</td>
<td>Frequency of Alert Notification</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
<td>--------------</td>
<td>--------------------------</td>
<td>----------------------------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>Tenable</td>
<td>Pull</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Qualys</td>
<td>Pull</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Webhook</td>
<td>Push</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ (w/ batch support)</td>
<td>–</td>
<td></td>
</tr>
</tbody>
</table>
The Prisma Cloud devOps security capabilities are geared to meet the common goal of delivering releases faster and preventing security lapses by applying a consistent set of checks through the build-to-release process that keep your applications and infrastructure secure.

- Secure Your Infrastructure Automation
- Prisma Cloud Plugins
- Set Up Your Prisma Cloud Configuration File for IaC Scan
- Use the Prisma Cloud Extension for AWS DevOps
- Use the Prisma Cloud Extension for Azure DevOps
- Use the Prisma Cloud Plugin for CircleCI
- Use the Prisma Cloud App for GitHub
- Use the Prisma Cloud Extension for GitLab
- Use the Prisma Cloud Plugin for IntelliJ IDEA
- Use the Prisma Cloud Extension for Visual Studio Code
- Use the Prisma Cloud IaC Scan REST API
- View the Prisma Cloud IaC scan policies available to perform checks on your IaC templates.
Secure Your Infrastructure Automation

Prisma Cloud DevOps Security enables DevOps and security teams to identify insecure configurations in Infrastructure-as-Code (IaC) templates and vulnerabilities in container images so that security issues are identified before actual resources are deployed in runtime environments.

To identify potential issues you can scan content in your IaC templates such as AWS CloudFormation Templates (JSON or YAML format), HashiCorp Terraform templates (HCL format), and Kubernetes App manifests (JSON or YAML format) against a list of IaC policies.

With a valid Prisma Cloud Enterprise edition license, you can use the IaC scanning and container image scanning functionality in any of the following ways:

- **Plugins/Extensions**—Install and configure the Prisma Cloud Plugins for popular IDEs such as VScode, IntelliJ; Source Control Management systems such as Github; CI/CD tools such as Jenkins, CircleCI, Azure DevOps. These plugins are designed to easily integrate in to your application development and deployment processes so that you can scan and fix issues in your current workflows without additional tools, thereby reducing the friction and boosting the adoption of better security checks.

- **Prisma Cloud IaC API**—Interact with the Prisma Cloud IaC scanning API endpoint using tools such as Curl, shell scripts, or Postman to scan IaC templates. Prisma Cloud recommends that you use the published plugins/extensions to perform IaC scanning, but you can use the IaC APIs directly for integrating with custom tools or specific use cases. See Use the Prisma Cloud IaC Scan REST API.

- **Twistcli**—Install and scan container images using twistcli. Twistcli is a command-line tool supported on Linux, macOS, and Windows, and it requires a Docker Engine to be installed on the machine where you are scanning images for vulnerabilities and malware.
Prisma Cloud Plugins

Prisma Cloud plugins enable you to check your DevOps infrastructure templates for security misconfigurations and scan container images to proactively prevent issues by shifting left.

The plugins or extensions as called on some environments, scan your templates against Prisma Cloud IaC policies to ensure compliance with security best practices before you deploy it into the cloud infrastructure. These plugins enable you to stay secure while being agile because they make it easy to scan your files, review any potential security issues, fix and validate code before you check it in to your source control repository or integrate it in your CI/CD pipeline.

<table>
<thead>
<tr>
<th>Integration</th>
<th>Category</th>
<th>Marketplace</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS DevOps</td>
<td>CI/CD</td>
<td>GitHub repository</td>
<td>Use the Prisma Cloud Extension for AWS DevOps</td>
</tr>
<tr>
<td>Azure DevOps</td>
<td>CI/CD</td>
<td>Azure Visual Studio Marketplace</td>
<td>Use the Prisma Cloud Extension for Azure DevOps</td>
</tr>
<tr>
<td>CircleCI</td>
<td>CI/CD</td>
<td>Circle CI Orb Registry</td>
<td>Use the Prisma Cloud Plugin for CircleCI</td>
</tr>
<tr>
<td>GitHub</td>
<td>SCM</td>
<td>GitHub Marketplace</td>
<td>Use the Prisma Cloud App for GitHub</td>
</tr>
<tr>
<td>GitLab</td>
<td>SCM and CI/CD</td>
<td></td>
<td>Use the Prisma Cloud Extension for GitLab</td>
</tr>
<tr>
<td>IntelliJ IDEA</td>
<td>IDE</td>
<td>IntelliJ Marketplace</td>
<td>Use the Prisma Cloud Plugin for IntelliJ IDEA</td>
</tr>
<tr>
<td>Integration</td>
<td>Category</td>
<td>Marketplace</td>
<td>Documentation</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Jenkins</td>
<td>CI/CD</td>
<td>Get the plugin from the Prisma Cloud administrative console (Compute&gt; Manage &gt; System &gt; Downloads)</td>
<td>Use the Prisma Cloud plugin for Jenkins</td>
</tr>
</tbody>
</table>
Set Up Your Prisma Cloud Configuration File for IaC Scan

Prisma Cloud IaC Scan requires a Prisma Cloud configuration file in the repository where your templates are stored. This configuration file can include information about your IaC module structure, runtime variables, and tags that help refine your IaC Scan use. It enables Prisma Cloud IaC scan to support complex module structures and variable formats.

Create this file as `.prismaCloud/config.yml` in the root directory of your repository branch.

The content of your Prisma Cloud configuration file depends on the IaC Scan support you need. The following show configuration details.

- Configure IaC Scan to Support Terraform
- Configure IaC Scan to Support AWS CloudFormation
- Configure IaC Scan to Support Kubernetes
- Configure Prisma Cloud Tags

Make sure to use a syntax validation tool when you copy and paste content from this page.

Configure IaC Scan to Support Terraform

The following shows the parameters in the Prisma Cloud configuration file that enable you to configure the IaC scan for Terraform 0.11 module with a variable file and/or input variables.

Make sure to use a syntax validation tool when you copy and paste content from this page.

```
# Specify the template type. Valid values are as follows.
# - For Terraform: TF
# - For AWS CloudFormation: CFT
# - For Kubernetes: K8S

template_type: TF

terraform_version: 0.11

# If terraform_version is 0.11, then terraform_011_parameters is required.
# The value for variable_files is an array of custom variable file names. The path of each file is relative to your repository branch root directory.
# The value for variable_values is an array of name/value pairs that identify the input variables your template uses.

terraform_011_parameters:
variable_files:
- scan/rich-value-types/network/variables.tf
```
variable_values:
- name: check
  value: public-read-write

The following shows the parameters in the Prisma Cloud configuration file that enable you to configure the IaC scan for Terraform 0.12.

```
# Specify the template type. Valid values are as follows.
# - For Terraform: TF
# - For AWS CloudFormation: CFT
# - For Kubernetes: K8S

template_type: TF

# Valid values for terraform_version are 0.11 or 0.12.

terraform_version: 0.12

# If terraform_version is 0.12, then terraform_012_parameters is required.
# The value of terraform_012_parameters is an array of root_modules.
# The value for root_module is relative to your repository branch root directory.
# Each root module can have:
# - variable_files, which is an array of variable file names relative to your repository branch root directory
# - variables, which is an array of name/value pairs that identify the input variables for the module

terraform_012_parameters:
- root_module: scan/rich-value-types/
  variables:
    - name: check
      value: public-read-write
    - name: varName2
      value: varValue2
  - root_module: scan/for-expressions/
    variable_files:
    - scan/rich-value-types/expressions/variables.tf
```

Configure IaC Scan to Support AWS CloudFormation

The following shows the parameters in the Prisma Cloud configuration file that enable you to configure the IaC scan for Amazon CloudFormation templates with variables.

```
# Specify the template type. Valid values are as follows.
# For Terraform: TF
# For AWS CloudFormation: CFT
# For Kubernetes: K8S

template_type: CFT
```
# If template_type value is CFT, set cft_parameters (optional)
# variable_values is an array of name/value pairs, which identifies
# the
# template variables

cft_parameters:
  variable_values:
  - name: KeyName
    value: 10
  - name: AMI
    value: ami-45785

Configure IaC Scan to Support Kubernetes

The following shows the parameters in the Prisma Cloud configuration file that enable you to configure the IaC scan for Kubernetes.

```yaml
# Specify the template type. Valid values are as follows.
# For Terraform: TF
# For AWS CloudFormation: CFT
# For Kubernetes: K8S

template_type: K8S
```

Configure Prisma Cloud Tags

The following shows the parameters in the Prisma Cloud configuration file that enable you to identify Prisma Cloud tags in your template. These tags offer a flexible way to identify and organize your resources in Prisma Cloud.

```yaml
# Prisma Cloud Tags
# tags is an array of labels that enable you to organize your resources
# with these key/value pairs in Prisma Cloud

tags:
- Org:Engineering
- Team:Shift_Left
```
Use the Prisma Cloud Extension for AWS DevOps

With a Prisma Cloud Enterprise Edition license, you can integrate compliance and vulnerability checks into your AWS continuous integration/continuous (CI/CD) and build environments. This extension enables you to scan Infrastructure-as-Code (IaC) templates like AWS CFT, Terraform templates, and Kubernetes deployment files against Prisma Cloud security policies. It also enables you to use Prisma Cloud Compute to scan container images for vulnerabilities.

The sections below show how to integrate the Prisma Cloud extension with your AWS CodePipeline pipelines and AWS CodeBuild projects.

- Set Up IaC Scanning with AWS CodePipeline
- Set Up Container Image Scanning with AWS CodeBuild

Set Up IaC Scanning with AWS CodePipeline

You can customize your AWS CodePipeline to check Infrastructure-as-Code (IaC) templates. The following examples show you how to integrate IaC scan into your CodePipeline.

You have two options to scan your IaC templates against Prisma Cloud security policies. You can use an AWS Lambda function with Python scripting, or you can use a custom action with a Bash shell script.

The prerequisites for IaC scan integration regardless of whether you use an AWS Lambda function or a custom action with a Bash shell script are as follows:

- You have a valid Prisma Cloud Enterprise Edition license
- You have a valid AWS CodePipeline service role to give AWS CodePipeline access to other resources in your account.
- You have configured a two-stage pipeline in AWS CodePipeline.
- If your customization uses any AWS commands, then you have installed and configured the AWS command line interface.

Use an AWS Lambda Function with Python Scripting

The following table describes the variables you need to set for your Lambda function, whether you are using the AWS CLI or the AWS console to configure your Lambda function. If you use the AWS console, these variables are environment variables. If you use a script that invokes the AWS CLI, you may specify the variables directly in the script.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Prisma_Cloud_API_URL</code></td>
<td>Your Prisma Cloud API URL (e.g. <a href="https://api.prismacloud.io">https://api.prismacloud.io</a>). The exact URL depends on the Prisma Cloud region and cluster of your tenant</td>
</tr>
<tr>
<td><code>Access_Key</code></td>
<td>Your Prisma Cloud access key for API access. If you do not have an access key, you must Create and Manage Access Keys</td>
</tr>
</tbody>
</table>
## Variable | Value
---|---
**Secret_Key** | The secret key that corresponds to your Prisma Cloud access key
**Asset_Name** | Identifies the repository you want to scan
**Tags** | Organizes the templates that are scanned with this service connection, for visibility on Prisma Cloud

### STEP 1 | Create a Lambda function.

This example shows you how to use the AWS command line interface to create a Lambda function that scans the IaC templates in your AWS CodePipeline for checking against Prisma Cloud security policies.

1. **Set Up Your Prisma Cloud Configuration File for IaC Scan**
   
   Create the `.prismaCloud/config.yml` file and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

2. **Download PrismaCloudIaCScan.zip to an accessible location.**
   
   You can find the file at [https://github.com/PaloAltoNetworks/Prisma-Cloud-DevOps-Security/blob/aws-codepipeline/aws-codepipeline/PrismaCloudIacScan/Lambda/PrismaCloudIaCScan.zip](https://github.com/PaloAltoNetworks/Prisma-Cloud-DevOps-Security/blob/aws-codepipeline/aws-codepipeline/PrismaCloudIacScan/Lambda/PrismaCloudIaCScan.zip)

3. Run the following command to create your Lambda function.

   ```bash
   export AWS_PROFILE=prisma-scan
   export AWS_DEFAULT_REGION=us-west-1
   export AWS_LAMBDA_FUNCTION=AWSCodePipeline-gn
   export AWS_ROLE=iam::/CustomPipelineWithLambda
   aws --profile ${AWS_PROFILE} --region ${AWS_DEFAULT_REGION} \
   lambda create-function \
   --role ${AWS_ROLE} \
   --function-name ${AWS_LAMBDA_FUNCTION} \
   --runtime python3.6 \
   --handler PrismaCloudIaCScan.lambda_handler \
   --environment
   Variables="{Prisma_Cloud_API_URL<hostname>, Access_Key=<accesskey>, \
   Secret_Key=<secretkey>, Asset_Name=<assetname>, Tags=<tags>}"
   --zip-file fileb://PrismaCloudIaCScan.zip
   ```

If you prefer to use the AWS console instead of the AWS CLI to create your Lambda function, you can use the steps below.

1. **Set Up Your Prisma Cloud Configuration File for IaC Scan**

   Create the `.prismaCloud/config.yml` file and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

2. **Download PrismaCloudIaCScan.zip to an accessible location.**

   You can download this file from [https://github.com/PaloAltoNetworks/Prisma-Cloud-DevOps-Security/blob/aws-codepipeline/aws-codepipeline/PrismaCloudIacScan/Lambda/PrismaCloudIaCScan.zip](https://github.com/PaloAltoNetworks/Prisma-Cloud-DevOps-Security/blob/aws-codepipeline/aws-codepipeline/PrismaCloudIacScan/Lambda/PrismaCloudIaCScan.zip)
3. In the AWS console, set the environment variables listed at the beginning of Use an AWS Lambda Function with Python Scripting.

4. Open the AWS Lambda console and navigate to the Create function page.
5. Provide a function name (e.g. LambdaFunctionForAWSCodePipeline).
6. Chose a runtime of either Python 3.6 or Python 3.7.
7. Either create a new execution role or choose an existing role that has the proper permissions.
   The proper permissions are:
   - Write permission for AWS Code Pipeline
   - List, Read, and Write permissions for AWS Cloudwatch Logs
   - Read permission for your S3 bucket if it is your data source
8. Select Create function.
9. Set the handler to PrismaCloudIaCScan.lambda_handler.
   The handler is defined in Basic Settings.
10. Choose the Execution role (optional).
11. Set a timeout.
12. Select Save.
STEP 2 | Add the Lambda function to your pipeline.

1. In the AWS console, navigate to Services > Developer Tools > CodePipeline > Edit Pipeline. Choose your pipeline and select Edit.
2. Between any phase stage, select + Add Stage and provide a stage name of your choice.
3. Select + Add action group. In Edit action, provide the information required to define a custom action.

The table below identifies the fields that have values specific to Prisma Cloud. The value for the User parameters is in JSON format and specifies the conditions under which the pipeline job status will fail. For the example in the table, the job will fail if the extension finds one high-severity violation or two medium-severity violations or five low-severity violations.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action provider</td>
<td>AWS Lambda</td>
</tr>
<tr>
<td>Function name</td>
<td>The function name you used when you created the Lambda function (e.g. PrismaCloudIaCScan)</td>
</tr>
<tr>
<td>User parameters</td>
<td>Example: {&quot;FailureCriteria&quot;: {&quot;High&quot;: 1,&quot;Medium&quot;: 2,&quot;Low&quot;: 5,&quot;Operator&quot;: &quot;or&quot;}}</td>
</tr>
<tr>
<td></td>
<td>Example with Tags: {&quot;FailureCriteria&quot;: {&quot;High&quot;: 1,&quot;Medium&quot;: 2,&quot;Low&quot;: 5,&quot;Operator&quot;: &quot;or&quot;}, &quot;Tags&quot;: [{&quot;team&quot;: devOps&quot;, &quot;env&quot;: test&quot;}}</td>
</tr>
<tr>
<td></td>
<td>Valid values for “Operator” are “or” and “and”</td>
</tr>
</tbody>
</table>

The following example shows the Edit action entries.

4. Select Done.
5. Review the results after you’ve executed your pipeline.

To start a pipeline manually through the console, select **Release change** on the pipeline details page. Select the link to execution details to see the latest CloudWatch logs to view any security violations that Prisma Cloud identified.

### Use a Custom Action with Bash

**STEP 1 | Create a custom action.**

The following example shows how to use an AWS custom action with a Bash shell script to scan your IaC templates and compare them against Prisma Cloud security policies.

> It's assumed that your source is created in a GitHub repository.
1. If it's not already installed, install jq, version 1.6 or higher on the EC2 instance or system where your job worker will run. 
   
   jq is available at
   
   https://stedolan.github.io/jq/.

2. If your job worker runs in an EC2 instance, ensure your EC2 instance user has permission to run CodePipeline.

3. Ensure the AWS CLI is available where your job worker runs.

   The job worker uses the following:
   
   - `aws codebuild`
   - `aws codepipeline`

4. Create a file `CustomAction.json` in a working location, such as your EC2 instance, and copy the following content to that file.

   ```json
   {
     "category": "Test",
     "provider": "Prisma-Cloud-IaC-Scan",
     "version": "1e",
     "settings": {
       "entityUrlTemplate": "https://s3.console.aws.amazon.com/s3/buckets/{Config:S3BucketName}/?
region={Config:S3BucketRegion}&tab=overview",
       "executionUrlTemplate": "https://s3.console.aws.amazon.com/s3/buckets/{Config:S3BucketName}/?
region={Config:S3BucketRegion}&tab=overview"
     },
     "configurationProperties": [
       {
         "name": "S3BucketName",
         "required": true,
         "key": true,
         "secret": false,
         "queryable": false,
         "description": "The S3 bucket name. The results with the vulnerabilities will be stored in this bucket."
       },
       {
         "name": "S3BucketRegion",
         "required": true,
         "key": true,
         "secret": false,
         "queryable": false,
         "description": "The S3 bucket region."
       },
       {
         "name": "Prisma_Cloud_API_URL",
         "required": true,
         "key": true,
         "secret": false,
         "queryable": false,
         "description": "Prisma Cloud server URL"
       }
     ]
   }
   ```
```json
{
  "name": "Access_Key",
  "required": true,
  "key": true,
  "secret": false,
  "queryable": false,
  "description": "Prisma Cloud access key",
  "type": "String"
},
{
  "name": "Secret_Key",
  "required": true,
  "key": true,
  "secret": true,
  "queryable": false,
  "description": "Prisma Cloud secret key",
  "type": "String"
},
{
  "name": "Asset_Name",
  "required": true,
  "key": true,
  "secret": false,
  "queryable": false,
  "description": "Provide the asset name for the pipeline",
  "type": "Number"
},
{
  "name": "Failure_Criteria",
  "required": true,
  "key": true,
  "secret": false,
  "queryable": false,
  "description": "Provide failure threshold for high, medium and low severity security issues along with the operator. Ex. high:5, medium:0, low:2, op:or",
  "type": "String"
},
{
  "name": "Tags",
  "required": false,
  "key": true,
  "secret": false,
  "queryable": false,
  "description": "Provide the tags for the IaC Scan task",
  "type": "String"
}
}

```

Optionaly, you can edit the provider and version fields, but do not modify the configurationProperties field.
5. Execute the following AWS command to create the custom action.

```
aws codepipeline create-custom-action-type --cli-input-json \
file://CustomAction.json
```

6. Create the required IAM policies.
   - Navigate to IAM > Policies, and create a policy to transfer files to and from the S3 bucket.
     This policy enables the worker to pull build artifacts from the S3 bucket for scanning and publish the logs to the bucket.

   ```
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Action": [
   "s3:ListAllMyBuckets"
   ],
   "Effect": "Allow",
   "Resource": "***"
   },
   {
   "Action": [
   "s3:PutObject",
   "s3:PutObjectAcl",
   "s3:GetObject",
   "s3:GetObjectAcl",
   "s3:DeleteObject",
   "s3:ListObject"
   ],
   "Effect": "Allow",
   "Resource": [
   "arn:aws:s3:::codepipeline-<S3 Region name>-<Random number generated by AWS>/*",
   "arn:aws:s3:::codepipeline-<S3 Region name>-<Random number generated by AWS>/*"
   ]
   }
   ]
   }
   ```

   - Create the scan job worker for custom actions.
     - Execute `aws configure` and set the default output format to JSON.
     - Copy the job worker shell script `poll.sh` to your local machine or the EC2 instance, depending on the job worker’s running location.
       Make sure your EC2 instance user has permission to run your pipeline and the job worker has permission to access CodePipeline.
     - Execute the job worker shell script with the following command:
       ```
       ./poll.sh "category=Test, owner=Custom, version=1, provider=Prisma-Cloud-IaC-Scan"
       ```

       The job worker is now configured to listen to requests from CodePipeline.

**STEP 2 | Add the custom action to your pipeline.**

1. In the AWS console, navigate to Services > Developer Tools > CodePipeline > Create/Edit Pipeline to add your Scan custom action to your pipeline.
   To add your custom action to your pipeline as a test step, navigate to Test > Test provider, and select Custom Action.

2. Configure the values for the pipeline.
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input artifacts</strong></td>
<td>The output artifact from the previous step</td>
</tr>
<tr>
<td><strong>Prisma_Cloud_API_URL</strong></td>
<td>Your Prisma Cloud API URL (e.g. <a href="https://api.prismacloud.io">https://api.prismacloud.io</a>). The exact URL depends on the region and cluster of your Prisma Cloud tenant</td>
</tr>
<tr>
<td><strong>Access_Key</strong></td>
<td>Your Prisma Cloud access key for API access</td>
</tr>
<tr>
<td><strong>Secret_Key</strong></td>
<td>Your Prisma Cloud secret key</td>
</tr>
<tr>
<td><strong>Asset_Name</strong></td>
<td>identifies the repository you want to scan</td>
</tr>
<tr>
<td><strong>Tags</strong></td>
<td>Organizes templates that are scanned, for visibility on Prisma Cloud. Example: env: test, team:devOps</td>
</tr>
<tr>
<td><strong>Failure_Criteria</strong></td>
<td>Failure criteria for high, medium, and low severity issues. Example: high: 0, med: 0, low: 0, operator: or</td>
</tr>
<tr>
<td><strong>S3BucketName</strong></td>
<td>Although this field is not specific to Prisma Cloud, a valid S3BucketName is required for this custom action</td>
</tr>
<tr>
<td><strong>S3BucketRegion</strong></td>
<td>Although this field is not specific to Prisma Cloud, a valid S3BucketRegion is required for this custom action</td>
</tr>
</tbody>
</table>
3. Save the pipeline changes.

**STEP 3 | Test your pipeline.**

You can use the AWS console to release the pipeline manually. After your stage completes, you can view the results of the checks against Prisma Cloud security profile in the log report in S3 by selecting the **Details** link.

---

**Set Up Container Image Scanning with AWS CodeBuild**

You can enable container image scanning with Prisma Cloud Compute. Add the following steps to your normal AWS CodeBuild build project set-up steps to add container scans to your build project.

**STEP 1 | On the Prisma Cloud Compute Console, add a vulnerability scan rule.**

1. Select **Compute > Defender > Vulnerabilities > Images > CI.**

![Screenshot of Prisma Cloud Compute Console](image-url)
2. Add Rule and enter a Rule name.

3. Specify the Alert and Failure thresholds.
   You can set the vulnerability scan to fail on critical, high, medium, or low severity. The failure threshold must be greater than the alert threshold.

4. Specify the Grace period.
   The grace period is the number of days for which you want

For more information about these settings, see the Prisma Cloud Compute Guide.

STEP 2 | Use the following example as your AWS buildspec file, buildspec.yaml, which is in the root directory of your source.

This file runs the twistcli command to scan the specified container image for vulnerabilities.

The following example splits some of the lines of code for documentation formatting. If you choose to copy this example directly, ensure the commands are not split into multiple lines in your code.

```yaml
version: 0.2

# In this example, we're using environment variables
# to store the username and password of our Prisma Cloud Compute CI user account
# and the URL to our console

# PC_COMPUTE_USER: The Prisma Cloud Compute user with the CI User role
# PC_COMPUTE_PASS: The password for this user account
# PC_COMPUTE_CONSOLE_URL: The base URL for the console -- http://console.my_company>.com:8083 -- without a trailing /

phases:
  install:
    runtime-versions:
      docker: 18
  build:
```
commands:
- echo Build started on `date`
- echo Building the Docker image..$IMAGE_TAG
- docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .

post_build:
commands:
- echo Build completed on `date`
- curl -k -u $PC_COMPUTE_USER:$PC_COMPUTE_PASS --output ./twistcli $PC_COMPUTE_CONSOLE_URL/api/v1/util/twistcli
- chmod +x ./twistcli
- echo Scanning with twistcli $PC_COMPUTE_PASS $PC_COMPUTE_USER

# Run the scan with twistcli, providing detailed results in
CodeBuild and
# pushing the results to the Prisma Cloud Compute console.
# --details returns all vulnerabilities and compliance issues rather
than just summaries.
# --address points to our Prisma Cloud Compute console
# -u and -p provide credentials for the console. These creds only
# need the CI User role.
# Finally, we provide the name of the image we built with 'docker
build', above.

- ./twistcli images scan --details --address $PC_COMPUTE_CONSOLE_URL
  -u $PC_COMPUTE_USER -p $PC_COMPUTE_PASS $IMAGE_REPO_NAME:$IMAGE_TAG

# See twistcli documentation for more details.

STEP 3 | In your AWS CodeBuild build project, set the following environment variables, which the
sample buildspec.yml file will use.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_COMPUTE_USER</td>
<td>Prisma Cloud Compute user with the CI User role</td>
</tr>
<tr>
<td>PC_COMPUTE_PASS</td>
<td>Prisma Cloud Compute user password</td>
</tr>
<tr>
<td>PC_COMPUTE_CONSOLE_URL</td>
<td>Base URL for the Prisma Cloud Compute console (e.g. <a href="http://console">http://console</a>.&lt;example&gt;.com:8083)</td>
</tr>
<tr>
<td>IMAGE_REPO_NAME</td>
<td>Docker repository for image to be scanned for vulnerabilities</td>
</tr>
<tr>
<td>IMAGE_TAG</td>
<td>Docker tag for image to be scanned for vulnerabilities</td>
</tr>
</tbody>
</table>
### Environment variables - optional

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_COMPUTE_USER</td>
<td>9c7a45bc-013a-41bb-a433-58b2f229de</td>
<td>Plaintext</td>
<td></td>
</tr>
<tr>
<td>PC_COMPUTE_PASS</td>
<td>L2mU16Vko69z28+aGuYQf5zFf=</td>
<td>Plaintext</td>
<td></td>
</tr>
<tr>
<td>PC_COMPUTE_CONSOLE_URL</td>
<td><a href="https://api.prismacloud.io/">https://api.prismacloud.io/</a></td>
<td>Plaintext</td>
<td></td>
</tr>
<tr>
<td>IMAGE_REPO_NAME</td>
<td>nginx</td>
<td>Plaintext</td>
<td></td>
</tr>
<tr>
<td>IMAGE_TAG</td>
<td>latest</td>
<td>Plaintext</td>
<td></td>
</tr>
</tbody>
</table>

### STEP 4 | View the results of the container image scan.

```bash
#!/bin/bash
set -u
set -e
```

poll.sh

```bash
#!/bin/bash
set -u
set -e
```
```bash
# exec&> >(while read line; do echo "$(date +'%h %d %H:%M:%S') $line" >> cmds.log; done;)

#set -x

if [[ -z "$1" ]]; then
  echo "Usage: ./poll.sh <action type id>" >&2
  echo -e "Example:
  ./poll.sh
""category=Test,owner=Custom,version=1,provider=Prisma-Cloud-IaC-Scan"" >&2
  exit 1
fi

run() {
  local action_type_id="$1"
  echo_ts "actiontypeid: $action_type_id"

  local job_json="$(fetch_job "$action_type_id")"
  if [[ "$job_json" != "null" && "$job_json" != "None" && "$job_json" != "" ]]; then
    local job_id="$(echo "$job_json" | jq -r '.id')"
    mkdir $job_id
    chmod +x $job_id
    cd $job_id || update_job_status "$job_json" "job id not found"
    acknowledge_job "$job_json"
    echo "job_json: $job_json"
    local build_json=$(create_build "$job_json")
  else
    sleep 10
  fi
}

acknowledge_job() {
  local job_json="$1"
  local job_id="$(echo "$job_json" | jq -r '.id')"
  local nonce="$(echo "$job_json" | jq -r '.nonce')"

  echo_ts "Acknowledging CodePipeline job (id: $job_id nonce: $nonce)" >&2
  aws codepipeline acknowledge-job --job-id "$job_id" --nonce "$nonce" > /dev/null 2>&1
}

fetch_job() {
  local action_type_id="$1"

  aws codepipeline poll-for-jobs --max-batch-size 1 \
    --action-type-id "$action_type_id" \
    --query 'jobs[0]'
}
```

action_configuration_value() {
    local job_json="$1"
    local configuration_key="$2"

    echo "$job_json" | jq -r ".data.actionConfiguration.configuration | .["$configuration_key"]"
}

update_job_status() {
    local job_json="$1"
    local build_state="$2"

    local job_id="$(echo "$job_json" | jq -r ".id")"

    echo_ts "Updating CodePipeline job with '$build_state' and job_id '$job_id'\nresult" >&2

    if [[ "$build_state" == *"succeeded"* ]]; then
        aws codepipeline put-job-success-result \
        --job-id "$job_id" \
        --execution-details "summary=$build_state,externalExecutionId=$job_id,percentComplete=100"
    else
        aws codepipeline put-job-failure-result \
        --job-id "$job_id" \
        --failure-details "type=JobFailed,message=Build $build_state,externalExecutionId=$job_id"
        fi
    }

decide_job_status(){
    local job_json="$1"
    local stats="$2"
    local in_high="$3"
    local in_med="$4"
    local in_low="$5"
    local in_oper="$6"

    local resp_high="$(echo "$stats" | jq -r ".high")"
    local resp_med="$(echo "$stats" | jq -r ".medium")"
    local resp_low="$(echo "$stats" | jq -r ".low")"

    if [[ $in_oper == null ]];then
        in_oper="or"
    fi
    if [[ $in_high == null ]];then
        in_high=0
    fi
    if [[ $in_med == null ]];then
        in_med=0
    fi
    if [[ $in_low == null ]];then
        in_low=0
    fi
    if [[ $stats != null ]];then
        fi
    }
}
if [[ "$in_oper" == "or" && ( "$resp_high" -ge "$in_high" || "$resp_med" -ge "$in_med" || "$resp_low" -ge "$in_low" ) ]] ;then
    local failure_message="Prisma Cloud IaC scan failed with
issues as security issues count (high: $resp_high, medium: $resp_med,
Low: $resp_low) meets or exceeds failure criteria (high: $in_high, medium:
in_med, Low: $in_low)"
    echo_ts "$failure_message"
    update_job_status "$job_json" "failed: Prisma Cloud IaC scan
failed with issues."

    elif [[ "$in_oper" == "and" && ( "$resp_high" -ge "$in_high" &&
"$resp_med" -ge "$in_med" && $resp_low" -ge "$in_low" ) ]] ; then
    local failure_message="Prisma Cloud IaC scan failed with
issues as security issues count (high: $resp_high, medium: $resp_med,
Low: $resp_low) meets or exceeds failure criteria (high: $in_high, medium:
in_med, Low: $in_low)"
    echo_ts "$failure_message"
    update_job_status "$job_json" "failed: Prisma Cloud IaC scan
failed with issues."

else
    local partial_success="Prisma Cloud IaC scan succeeded
with issues as security issues count (high: $resp_high, medium: $resp_med,
Low: $resp_low) does not exceed failure criteria (high: $in_high, medium:
in_med, Low: $in_low)"
    echo_ts "$partial_success"
    update_job_status "$job_json" "succeeded: Prisma Cloud IaC
scan succeeded with issues as security issues."
fi
else
    update_job_status "$job_json" "success"
fi

create_build() {
    ls
    local job_json="$1"
    local job_id="$(echo "$job_json" | jq -r '.id')"
    local pipelineName="$(echo "$job_json" | jq -r
".data.pipelineContext.pipelineName")"
    local s3_bucket=$(action_configuration_value "$job_json" "S3BucketName")
    local bucketName="$(echo "$job_json" | jq -r
."data.inputArtifacts[0].location.s3Location | .["bucketName"]")"
    local object_key="$(echo "$job_json" | jq -r
."data.inputArtifacts[0].location.s3Location | .["objectKey"]")"
    local output_object="$(echo "$job_json" | jq -r
."data.outputArtifacts[0].location.s3Location | .["objectKey"]")"
    local console_url="$(echo "$job_json" | jq -r
."data.actionConfiguration.configuration.Prisma_Cloud_API_URL")"
    local access_key="$(echo "$job_json" | jq -r
."data.actionConfiguration.configuration.Access_Key")"
    local secret_key="$(echo "$job_json" | jq -r
."data.actionConfiguration.configuration.Secret_Key")"

    aws codepipeline get-pipeline --name "$pipelineName" >
    pipelineDetails.json
    jq '.pipeline.stages[] | select(.name == "Source")' pipelineDetails.json >
    source.json
#cat source.json
local user_id="$(cat source.json | jq -r ".actions[].configuration.Owner")"
local project_name="$(cat source.json | jq -r ".actions[].configuration.Repo")"

if [ -z "$console_url" ]; then
  echo_ts "Please enter valid Prisma Cloud API URL in plugin in Input param. For details refer to :plugin link"
  update_job_status "$job_json" "Please enter valid Prisma Cloud API URL in plugin in Input param. For details refer to plugin link"
  exit 1;
fi

echo "executing login api"

local login_url="${console_url}/login"
local req_cmd=$(curl -k -i -o -X POST $login_url -H "Content-Type:application/json" --user-agent "AWS-CodePipeline-CustomAction/2.0.0" -d "\"username\":\"${access_key}\",\"password\":\"${secret_key}\"\"" -x http://127.0.0.1:8080  ) || update_job_status "$job_json" "$err_500"

local err_400="Invalid credentials please verify that API URL, Access Key and Secret Key in Prisma Cloud plugin settings are valid For details refer to Extension link https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-extension-for-aws-codepipeline.html"
local err_500="Oops! Something went wrong, please try again or refer to documentation on https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-extension-for-aws-codepipeline.html"

http_status=$(echo "$req_cmd" | grep HTTP | awk '{print $2}')
echo "http status: $http_status"
if [[ -z "$http_status" ]]; then
  echo_ts "$err_500" >&2
  update_job_status "$job_json" "error"
  exit 1;
fi

if [[ "$http_status" == 400 || "$http_status" == 401 ]]; then
  echo_ts "$err_400" >&2
  update_job_status "$job_json" "error"
  exit 1
fi

if [[ "$http_status" == 500 || "$http_status" > 500 ]]; then
  echo "http_status: $http_status"
  echo_ts "$err_500" >&2
  update_job_status "$job_json" "error"
  exit 1
fi

echo "req cmd: $req_cmd"
output_response=$(echo "$req_cmd" | grep token)
local token="$(echo "$output_response" | jq .token | tr -d '"')"
#echo "token: $token"

local scan_location="$(echo $bucketName/$object_key)"
aws s3 cp s3://$scan_location . || update_job_status "$job_json" "Copy Object from S3 bucket failed"
local file=( *.zip )
# echo "file: $file"
mv $file artifact.zip

file_size="$(wc -c artifact.zip | awk '{print $1}')"
# echo "$file_size"
file_size_limit=1000000
if [[ "$file_size" -gt "$file_size_limit" ]]
then
  printf "\nDirectory size $project_name more than 2 MB is not supported."
  exit 1;
fi

mkdir .prismaCloud
unzip -p artifact.zip .prismaCloud/config.yml >.prismaCloud/config.yml
if [[ ! -f .prismaCloud/config.yml ]]
then
  echo "File not present"
fi

iacAPI=${console_url}/iac_scan
echo "executing scan api: $iacAPI"

####################################################
# Generate the url and the headers
# Generate the url and the headers
####################################################
# echo "m here"

if [[ ! -f .prismaCloud/config.yml ]]
then
  echo "Can not find config.yml under .prismaCloud folder in repo $project_name. Please make sure the file is present in correct format (refer: https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-extension-for-aws-devops.html) at the root of your repo under .prismaCloud folder."
  exit 1;
fi

headers=""
url=""
fileContents=$(yq read -j .prismaCloud/config.yml)
# echo "file contents are: "$fileContents
templateType="$(echo "$fileContents" | jq -r '.template_type')"
# echo "template type: "$templateType

if [[ ! -z "$templateType" && ( "$templateType" == "TF" || "$templateType" == "tf" ) ]]
then
  url="$console_url/iac/tf/v1/scan"
terraformVersion="$(echo "$fileContents" | jq -r '.terraform_version')"
  if [[ ! -z "$terraformVersion" && ( "$terraformVersion" == 0.12 || "$terraformVersion" > 0.12 ) ]]
  then
    headers=" -H terraform-version:0.12"
    isTerraform12ParamsPresent="$(echo "$fileContents" | jq -r '.terraform_012_parameters')"
    if [[ "$isTerraform12ParamsPresent" != null ]]
    then
      terraformContents="$(echo "$fileContents" | jq -r '.terraform_012_parameters[] |= with_entries(.key |= gsub("root_module";
"root-module") } | jq -r '.terraform_012_parameters[] |=
with_entries( .key |= gsub("variable_files"; "variable-files") )'
"terraform012Parameters=$(echo "$terraformContents" | jq -r
'.terraform_012_parameters' | tr -d '\n\t' | tr -d '[:blank:]')"
if [[ "$terraform012Parameters" != null ]]
then
  headers+=" -H terraform-012-parameters:$terraform012Parameters"
fi
else
headers+=" -H terraform-version:0.11"
#read terraform 0.11 parameters
variableFiles=$(echo "$fileContents" | jq -r
'.terraform_011_parameters.variable_files' | tr -d '\n\t' | tr -d '[:blank:]')"
variableValues=$(echo "$fileContents" | jq -r
'.terraform_011_parameters.variable_values' | tr -d '\n\t' | tr -d '[:blank:]')"
if [[ "$variableFiles" != null ]]
then
  headers+=" -H rl-variable-file-names:$variableFiles"
fi
if [[ "$variableValues" != null ]]
then
  headers+=" -H rl-parameters:$variableValues"
fi
elif [[ ! -z "templateType" && ( "$templateType" == "CFT" ||
"$templateType" == "cft" ) ]]
then
  url="$console_url/iac/cft/v1/scan"
  variableValues=$(echo "$fileContents" | jq -r
'.cft_parameters.variable_values' | tr -d '\n\t' | tr -d '[:blank:]')"
if [[ "$variableValues" != null ]]
then
  headers+=" -H rl-parameters:$variableValues"
fi
elif [[ ! -z "templateType" && ( "$templateType" == "K8S" ||
"$templateType" == "k8s" || "$templateType" == "K8s" ) ]]
then
  url="$console_url/iac/k8s/v1/scan"
else
  echo "No valid template-type found in config.yml file in repo
$project_name. Please specify either of these values: TF, CFT or K8s as
template-type variable in the config.yml"
exit 1;
fi
#echo url: "$url"
#echo header: "$headers"

# Metadata Structure
# Tags
task_tags="$(echo "$job_json" | jq -r
".data.actionConfiguration.configuration.Tags")"
repo_tags="$(echo "$fileContents" | jq -r ".tags" | tr -d '\n\t' | tr -d '[:blank:]')"
prisma_tags=""
if [[ ! -z "\$repo_tags" ]]
then
  prisma_tags+="\"repo_tags\":\$repo_tags"
fi
if [[ ! -z "\$task_tags" ]]
then
  temp="\$(sed 's/,/",/g' \<<< "\$task_tags")"
  if [[ "\$prisma_tags" == "" ]]
    then
      prisma_tags+="\"task_tags\":\$temp"
    else
      prisma_tags+=", \"task_tags\":\$temp"
    fi
  else
    prisma_tags+="\"task_tags\":\$temp"
fi

aws codepipeline get-pipeline --name "$pipelineName" > pipelineDetails.json
jq '.pipeline.stages[] | select(.name == "Source")' pipelineDetails.json > source.json
local user_id="$(cat source.json | jq -r ".actions[].configuration.Owner")"
local project_name="$(cat source.json | jq -r ".actions[].configuration.Repo")"
local stage_name="$(echo "$job_json" | jq -r ".data.pipelineContext.stage.name")"
local action_name="$(echo "$job_json" | jq -r ".data.pipelineContext.action.name")"
local asset_name="$(echo "$job_json" | jq -r ".data.actionConfiguration.configuration.Asset_Name")"
local failure_criteria="$(echo "$job_json" | jq -r ".data.actionConfiguration.configuration.Failure_Criteria")"
if [[ -z "\$failure_criteria" ]];then
  failure_criteria_high_severity=0
  failure_criteria_medium_severity=0
  failure_criteria_low_severity=0
  failure_criteria_operator="or"
else
  echo "failure criteria: "$failure_criteria
  failure_criteria_removed_spaces=${printf '%s' \$failure_criteria} delimiter=
  s=${failure_criteria_removed_spaces}$delimiter array=()
  while [[ ! \$s ]]; do
    array+=( "${s%%"$delimiter"*}" );
    s=${s#*"$delimiter"};
  done;
  declare -p array
  failure_criteria_high_severity=$(awk -F ':' '{print $2}' <<<"${array[0]}"")
  failure_criteria_medium_severity=$(awk -F ':' '{print $2}' <<<"${array[1]}"")
  failure_criteria_low_severity=$(awk -F ':' '{print $2}' <<<"${array[2]}"")
  failure_criteria_operator=$(awk -F ':' '{print $2}' <<<"${array[3]}")

#echo "Failure Criteria:" $failure_criteria_high_severity $failure_criteria_medium_severity $failure_criteria_low_severity $failure_criteria_operator

# Metadata

metadata_json={"asset-name":"$asset_name","asset-type":"AWS-CodePipeline","user-id":"${user_id}","prisma_tags":{"$prisma_tags"},"scan-attributes":{"project-name":"${project_name}","pipeline-details": {"pipeline-name":"$pipelineName","job-id":"$job_id","stage-name":"$stage_name","action-name":"$action_name"},"failure-criteria": {"high":"$failure_criteria_high_severity","medium":"$failure_criteria_medium_severity","low":"$failure_criteria_low_severity","operator":"$failure_criteria_operator"}}

echo metadata "$metadata_json"

#################################################################################################

# IaC Scan Execution

#################################################################################################

echo "Executing the scan api"
local response=$(curl -k -X POST $url -H "x-redlock-auth:${token}" --user-agent "AWS-CodePipeline-CustomAction/2.0.0" $headers -H "x-redlock-iac-metadata:${metadata_json}" -F templateFile=@artifact.zip -x http://127.0.0.1:8080) || update_job_status "$job_json" "Call from API failed"
#echo "response: $response"
local result=$(echo "$response" | jq -r '.result.is_successful')
#echo "result: $result"
if ![[ $result ]]
then
    local partial_failure=$(echo "$response" | jq -r '.result.partial_failure')
    local matched=$(echo "$response" | jq -r '.result.rules_matched')
    if ![[ $matched != null ]];then
        local stats=$(echo "$response" | jq -r '.result.severity_stats')
        decide_job_status "$job_json" "$stats"

        "$failure_criteria_high_severity" "$failure_criteria_medium_severity"
        "$failure_criteria_low_severity" "$failure_criteria_operator"
        display=$(echo "$matched" | jq -r 'sort_by(.severity) | (["SEVERITY", "NAME", "FILES"] | ., map(length*-"")) , ([] | [.severity , .name , .files[0]] ) | join(",", ) | column -t -s ",")' ||
        update_job_status "$job_json" "Unknown Error"
        if ![-z "$partial_failure"]
        then
            display+="\n$partial_failure"
        fi
    else
        echo_ts "Good job! Prisma Cloud did not detect any issues."
        fi
    else
        local error_message=$(echo "$response" | jq -r '.result.error_details')
        echo_ts "$error_message"
        update_job_status "$job_json" "$error_message"
        exit 1
    fi

echo_ts "$display" >&2
aws s3 cp Prisma_Cloud_IaC_Scan.log s3://$s3_bucket/Prisma_Cloud_IaC_Scan_$job_id.log || update_job_status "$job_json" "upload results to S3 bucket failed"
rm -fr $job_id
}
run "$1"
Use the Prisma Cloud Extension for Azure DevOps

Use the Prisma Cloud extension to scan IaC templates, container images, and serverless functions in the build or release phase of the Azure DevOps pipeline. After you install this extension from the Azure Visual Studio Marketplace, you can set up the service connections for Prisma Cloud IaC Scan and Prisma Cloud Compute Scan, and then use custom tasks in the build or release pipeline for scanning IaC templates—AWS CloudFormation Templates, Terraform templates (version 0.11 and 0.12), Kubernetes app deployment YAML files—container images, or serverless zip files. When you create a custom task, you can specify the build or pipeline failure criteria based on severity of the security issues that the extension identifies.

When you set up the Prisma Cloud extension to scan, you can specify the tags at different stages. Prisma Cloud tags enable visibility on the Prisma Cloud administrator console, and are different from Azure DevOps tags or cloud tags that you may have included within your IaC templates. You can include these tags as `key:value` pairs in a comma separated list when you set up the service connection, and within the .prismaCloud/config.yml at the repository-level, or where you define the failure criteria for a Prisma Cloud IaC scan at the task level, and use it as a filter on Prisma Cloud (coming soon).

- Install and Configure the Prisma Cloud Extensions
- Set up a Custom Task for IaC Scanning
- Set Up Container Image Scanning
- (Required with nested virtualization only) Set Up RASP Defender
- Sample YAML File

Install and Configure the Prisma Cloud Extensions

You need to add the prisma-cloud-config.yml in the root directory of your repository branch, and get the Prisma Cloud extension from the Visual Studio Marketplace, set up service connections to authenticate with Prisma Cloud and start scanning IaC templates, container images, and serverless functions.

**STEP 1 | Set up your Azure DevOps organization and pipeline.**

If you are just getting started with Azure Pipeline, refer to the Azure documentation.

1. Create a project.
2. Create a new pipeline.
3. Select your code repository, configure, and save the pipeline.

**STEP 2 | Set Up Your Prisma Cloud Configuration File for IaC Scan file.**

Create the .prismaCloud/config.yml and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

**STEP 3 | Install the extension.**

2. **Install** the extension in your Azure DevOps organization.

Select **Organization settings > Extensions** to verify that the extensions displays in the list of **Installed** extensions.
STEP 4 | Add service connections to authenticate to Prisma Cloud.

You must create a new service connection for each type of scan— one for IaC scanning and one for scanning container image or serverless functions.

1. Select Project Settings > Service Connections > New Service Connection > Prisma Cloud IaC Console.

2. Enter the following information for the Prisma Cloud for IaC scanning and save your changes.

   • Enter the Prisma Cloud API URL as Server URL.
     The URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. The tenant provisioned for you is, for example, https://app2.prismacloud.io or https://app.eu.prismacloud.io. Replace app in the URL with api and enter it here. Refer to the Prisma Cloud REST API Reference for more details.
   
   • Enter your Prisma Cloud Access Key.
     The access key enables programmatic access. If you do not have a key, you must Create and Manage Access Keys.
   
   • Enter your Prisma Cloud Secret Key.
     You should have saved this key when you generated it. You cannot view it on the Prisma Cloud web interface.

   • Enter an Asset Name to identify the repository you want to scan.
   
   • Enter the Tags to organize the templates that are scanned with this service connection, for visibility on Prisma Cloud.
   
   • Provide a Service connection name.
   
   • Verify that Grant access permission to all pipelines is selected and Save your changes.

3. Continue to the next step if you want to set up another service connection for container image scanning. If not, go to Set up a Custom Task for IaC Scanning.


5. Enter the following information for Prisma Cloud Compute Console and save your changes.
• Server URL.

You need to copy the server URL from the Prisma Cloud interface, **Compute > Manage > System > Downloads > Path to Console**. For Prisma Cloud Compute Edition, get the URL from **Manage > System > Downloads > Path to Console**

• Username and password.

These credentials are required for the service connection to authenticate with Prisma Cloud. If you are using Prisma Cloud Compute Edition (self-hosted), create a role and enter your username and password.

If you are using Prisma Cloud Compute, you must first **Create Prisma Cloud Roles** with the Build and Deploy Security permission group and assign this role to the administrative user so that they can **create an access key**. The access key is the username and the secret key is your password.

If your password has special characters, make sure to escape any special characters when you enter your password.

• Optional CA certificate if you are using certificate-based authentication.

• Add a **Name** for the service connection.

• Verify that **Grant access permission to all pipelines** is selected.

**STEP 5 | Continue with Set up a Custom Task for IaC Scanning.**

Set up a Custom Task for IaC Scanning

Use the following instructions to add a custom task for IaC scanning and container image and serverless functions scanning in your azure-pipelines.yml. In each task, you can define the pipeline failure criteria based on the severity of the issues that are detected during the scan.

**STEP 1 | Under Pipelines, select your pipeline and Edit to add custom task.**

**STEP 2 | Add a custom task for IaC scanning.**
1. Under Task, search for Prisma Cloud IaC Scan you created earlier.

2. Enter the path for the directory you want to scan.
   If you want to scan the entire repository, use `. or $(System.DefaultWorkingDirectory).

3. Select the Service Endpoint, which is the service connection you created in the previous task.

4. Enter the Tags you want to apply to the templates that are being scanned.
   The tags format is name:value, and you can add multiple tags that are separated using commas.

5. Select the Failure Criteria for the scan.
   You can set the count for High, Medium, Low severity issues and decide whether you want to use
   the AND or OR operator to specify your criteria. For example, if you have a very strict threshold and
   set the failure criteria to 0,0,0 with the OR operator your build will fail if the policy checks detect any
   issues.

6. Add to yml file, and Save the task.

7. Enable system diagnostics and Run.
STEP 3 | Run the task.

1. In Azure DevOps, click **Queue** to execute your task on the next available build agent. If your task configuration is incomplete, a red status message displays **Some settings need attention** just below **Run your build**.
   
2. Check the results.
   - If the IaC Scan finds no issues the pipeline task result is successful.
   - If the IaC Scan finds issues but the failure criteria threshold you defined is not met, the job is successful but it displays the list of issues that were detected.
If the failure criteria you defined is more stringent than the default scan threshold, the job will fail and you can review results in the log file.

Set Up Container Image Scanning

On Windows and Linux OS, you can scan container images and serverless functions when you enable twistcli, add a vulnerability scan rule where you define the criteria to fail the build, and set up a task to scan the image or function in the pipeline.
STEP 1 | Add a vulnerability scan rule on the Prisma Cloud Compute Console.

1. Select **Compute > Defender > Vulnerabilities > Images > CI**.

2. **Add Rule** and enter a **Rule name**.

3. Specify the **Alert** and **Failure** thresholds.
   
   You can set the vulnerability scan to fail on critical, high, medium, low severity. The failure threshold must be equal to or greater than the alert threshold.

4. **(Optional)** Specify the **Images** to scan.
   
   The image or function zip file name is required later when you add the scan task to the pipeline in Step 3.

5. **(Optional)** Select **Apply rule when vendor fixes are available**, if you want to scan only for vulnerabilities that have fixes available.

6. Specify the **Grace period**.
   
   The grace period is the number of days for which you want to ignore a vulnerability. The timeframe is measured in days starting at the date from the first vendor publish. For more details on the advanced settings, see the **Prisma Cloud Compute guide**.

STEP 2 | Add a pipeline task to scan container images using twistcli.
1. Select Pipelines, and Edit your pipeline and to add custom task.
2. Search for Prisma in the task list and select Prisma Cloud Compute twistcli scan.
3. Select the Scan type—Images or Serverless.
4. Select the Prisma Cloud Compute Console service connection name that you created earlier, from the drop-down.
5. Specify the Image name or serverless Function zip file name. The image name you enter here must match the name of the image you are building in the pipeline, if it doesn’t the scan will fail.

**STEP 3 | View the results of the scan.**

See the results in

To see results on Prisma Cloud, select Compute > Monitor > Vulnerabilities > Twistcli Scans
Set Up RASP Defender

If you are using Docker-in-Docker, where you have a Docker container that itself has Docker installed, and from within the container you use Docker to pull images, build images, run containers, you have to set up RASP Defenders to secure containers at runtime.

Update the Dockerfile and embed the RASP defender as part of the Azure DevOps build.

1. Select Pipelines, and Edit your pipeline and to add custom task
2. Search for Prisma in the task list and select Prisma Cloud Compute embed RASP.
3. Select the Scan type—Images or Serverless.
4. Select the Service connection you created earlier for Prisma Cloud Compute Console.
5. Provide a unique Application ID for the RASP defender.
   For example, <your company>-<app>
6. Enter the **Console Host**, which is the DNS name or IP address of your Prisma Cloud Compute Console.
7. Specify the **Data Folder**, which is the read-write directory in the container file system.
   
   For example, `/twistlock/`.
8. Enter the **Dockerfile path** of the container image to which you want to add the RASP defender.

Sample YAML File

The following is a sample azure-pipeline.yml when you enable both the Prisma Cloud IaC scan and Prisma Cloud Compute scan. This file autogenerates is referenced below as an example.

```yaml
# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more: # https://aka.ms/yaml
trigger:
  branches:
    include:
    - master
pool:
  vmImage: 'ubuntu-latest'
steps:
- task: Palo-Alto-Networks.build-release-task.custom-build-release-task.prisma-cloud-compute-scan@1
  displayName: 'Prisma Cloud Compute Scan'
  inputs:
    twistlockService: 'NewEnv Connection'
    artifact: 'nginx:latest'
- task: Prisma Cloud IaC Scan@1
  displayName: 'Prisma Cloud IaC Scan'
  inputs:
    Path: 'repo'
    prismaCloudService: 'Prisma Cloud Scan'
    High: '0'
    Medium: '0'
    Low: '0'
    Operator: 'or'
- script: |
  echo Add other tasks to build, test, and deploy your project.
  echo See https://aka.ms/yaml
  displayName: 'Run a multi-line script'
```
Use the Prisma Cloud Plugin for CircleCI

Use the Prisma Cloud orb for CircleCI to scan IaC templates and container images during CircleCI pipelines. In order to use Prisma Cloud IaC scan functionality, you need to have a connection to a Prisma Cloud API server and have details for that connection specified as environment variables. Similarly, in order to perform container image vulnerability scans, you need a connection to the Prisma Cloud Compute console. When you create a custom task to embed this functionality in your CircleCI pipeline, you can specify the build or pipeline failure criteria based on the severity of the security issues that are identified.

**STEP 1 | Verify the prerequisites.**

- Verify the `.circleci/config.yml` is in your project root directory.

CircleCI uses this file each time it runs a build.

- Set CircleCI org permissions to allow orbs that are not part of the Certified and Partner list. Your CircleCI org admin can opt in to use uncertified third-party orbs by navigating to **Settings > Security** and selecting the opt-in setting.

- For IaC scan, get the details for enabling authentication to Prisma Cloud.

  - **Prisma Cloud API URL.**
The URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. The tenant provisioned for you is, for example, https://app2.prismacloud.io or https://app.eu.prismacloud.io. Replace app in the URL with api and enter it here. Refer to the Prisma Cloud REST API Reference, which is accessible from the Help Center within the Prisma Cloud web interface for more details.

- **Access Key.**
  The access key enables programmatic access to Prisma Cloud. If you do not have a key, you must [Create and Manage Access Keys](#).

- **Secret Key.**
  You should have saved this secret key when you generated it. You cannot view it on the Prisma Cloud web interface.

- **For image scan, get the details for authenticating to the Prisma Cloud Compute.**
  - Prisma Cloud Compute URL.
    You need to copy the URL from the Prisma Cloud interface, [Manage > Defenders > Deploy](#).
  - Prisma Cloud Compute username and password.

**STEP 2 | Add the environment variables for enabling authentication to Prisma Cloud.**

On CircleCI, you must add the environment variables as name value pairs. The following table lists the environment variables, and the figure below shows an example of environment variable settings for IaC scans.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>prisma_cloud_asset_name</td>
<td>CircleCI server. Examples: creditapp_server,</td>
<td>Used to track specific results in Prisma Cloud. For IaC scan only.</td>
</tr>
<tr>
<td></td>
<td>ConsumerBU_server</td>
<td>Required.</td>
</tr>
<tr>
<td>prisma_cloud_secret_key</td>
<td>Prisma Cloud secret key</td>
<td>See <a href="#">Create and Manage Access Keys</a> for details about the secret</td>
</tr>
<tr>
<td></td>
<td></td>
<td>key. For IaC scan only. Required.</td>
</tr>
<tr>
<td>prisma_cloud_compute_pass</td>
<td>Prisma Cloud Compute password</td>
<td>The Prisma Cloud Compute users's password.</td>
</tr>
<tr>
<td>prisma_cloud_tags</td>
<td>Comma-separated list of key/value pairs.</td>
<td>Used for visibility in Prisma Cloud UI. For IaC scan only. Optional.</td>
</tr>
<tr>
<td></td>
<td>Examples: project:x, owner:mr.y, compliance:pci</td>
<td></td>
</tr>
</tbody>
</table>
1. For IaC scan, enter the Name and Value for the Prisma Cloud secret key. The default name is `prisma_cloud_secret_key`. If you enter a different name, make sure to match this name in the config.yml file in the next step.

2. For IaC scan, enter the Name and Value for your asset name. Prisma Cloud uses `prisma_cloud_asset_name` to track specific scan results.

3. For IaC scan, enter the Name and Value for all the `prisma_cloud_tags` you want to define. The Value is a comma-separated list of key/value pairs for tags that you want to define. Use an equals sign to assign tag values to tag keys. Tags are optional but enable visibility in the Prisma Cloud UI.

4. For image scan, enter the Name and Value for the Prisma Cloud Compute password. The default name is `prisma_cloud_compute_pass`. If you enter a different name, make sure to match this name in the config.yml file in the next step.

**STEP 3 | Add the Prisma Cloud configuration file.**

The Prisma Cloud configuration file supports various IaC scan features. To add this file, create a subdirectory and file `.prismacloud/config.yml` in the root folder of your project or repository branch. See Set Up Your Prisma Cloud Configuration File for IaC Scan for details. Note that this file is different from `.circleci/config.yml`, and subsequent references to config.yml in these steps indicate the `.circleci/config.yml` file.

**STEP 4 | Add the Prisma Cloud orb to the config.yml.**

1. Modify the config.yml to include the orb named prisma_cloud/devops_security for IaC and image scanning.

   Note that, through the orb, you can customize the IaC scan failure criteria and the vulnerability thresholds for image scanning based on your security needs.

   More details about the Prisma Cloud orb are available at Prisma Cloud Orb Quick Start Guide. The following table lists the parameters you can specify to customize the Prisma Cloud IaC scan job in your orb.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Req</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>access_key</td>
<td>Prisma Cloud access key</td>
<td>no</td>
<td>$prisma_cloud_access_key</td>
<td>String</td>
</tr>
<tr>
<td>secret_key</td>
<td>Prisma Cloud secret key</td>
<td>no</td>
<td>prisma_cloud_secret_key</td>
<td>Environment variable</td>
</tr>
<tr>
<td>prisma_cloud_api_url</td>
<td>Prisma Cloud server URL</td>
<td>no</td>
<td>$prisma_cloud_console_url</td>
<td>String</td>
</tr>
<tr>
<td>terraform_variable_filenames</td>
<td>Comma-separated list of file names containing Terraform variables</td>
<td>no</td>
<td>&quot;&quot;</td>
<td>String</td>
</tr>
<tr>
<td>templates_directory_path</td>
<td>Directory path where IaC templates are stored. Note: The total size of the IaC templates in this directory cannot exceed 9 MB.</td>
<td>no</td>
<td>.</td>
<td>String</td>
</tr>
<tr>
<td>failure_criteria_high_severity</td>
<td>Provides failure threshold for high severity security issues</td>
<td>no</td>
<td>0</td>
<td>Integer</td>
</tr>
<tr>
<td>failure_criteria_medium_severity</td>
<td>Provides failure threshold for medium severity security issues</td>
<td>no</td>
<td>0</td>
<td>Integer</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Required</td>
<td>Default</td>
<td>Type</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>failure_criteria_low_severity</td>
<td>Provides failure threshold for low severity security issues</td>
<td>no</td>
<td>0</td>
<td>Integer</td>
</tr>
<tr>
<td>failure_criteria_operator</td>
<td>Provides operator for high, medium, low severity failure thresholds</td>
<td>no</td>
<td>or</td>
<td>String</td>
</tr>
<tr>
<td>tags</td>
<td>Comma-separated list of tags for your task. Used for visibility in Prisma Cloud UI. For IaC scan only. Optional.</td>
<td>no</td>
<td>''</td>
<td>String</td>
</tr>
</tbody>
</table>

The following table lists the parameters you can specify to customize the Prisma Cloud Compute container image scanning job in your orb.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
<th>Default</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>prisma_cloud_compute_user</td>
<td>The Prisma Cloud Compute user with the CI User role</td>
<td>no</td>
<td>$prisma_cloud_compute_user</td>
<td>String</td>
</tr>
<tr>
<td>prisma_cloud_compute_pass</td>
<td>The Prisma Cloud Compute user's password</td>
<td>no</td>
<td>prisma_cloud_compute_pass</td>
<td>Environment variable</td>
</tr>
<tr>
<td>prisma_cloud_compute_url</td>
<td>The base URL for the console -- e.g. <a href="http://console">http://console</a>.&lt;abc&gt;.com:8083</td>
<td>no</td>
<td>$prisma_cloud_compute_url</td>
<td>String</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Required</td>
<td>Default</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>workspace_name</td>
<td>Name of workspace to &quot;docker save&quot; the image-tar into so it can be scanned by orb</td>
<td>no</td>
<td>workspace</td>
<td>String</td>
</tr>
<tr>
<td>image_tar</td>
<td>The name of the image tar file stored in the workspace</td>
<td>no</td>
<td>image.tar</td>
<td>String</td>
</tr>
<tr>
<td>image</td>
<td>The name of the image to scan -- myimage or myorg/myimage or myorg/myimage:latest</td>
<td>yes</td>
<td></td>
<td>String</td>
</tr>
</tbody>
</table>

The following script is an example that shows you how to add the details required to set up both IaC and container image scanning.

```
version: 2.1
orbs:
  scan: prisma_cloud/devops_security@2.0.0
jobs:
  scan_iac: scan/prisma_cloud
docker_build_and_save:
    executor: scan/compute
    steps:
      - checkout
      - run: docker pull nginx
      - run: mkdir -p workspace
      - run: docker image
      - run: 'docker save nginx:latest -o workspace/image.tar'
      - persist_to_workspace:
          root: workspace
          paths:
            - image.tar
workflows:
```

Make sure that you have defined the secret key (prisma_cloud_secret_key for IaC scan) and the Prisma Cloud compute password (prisma_cloud_compute_pass for container image scanning), each as an environment variable.
scan:
  jobs:
    - scan_iac:
      # Default env var for below: prisma_cloud_console_url
      prisma_cloud_api_url: <prisma cloud api url>
      # Default env var for below: prisma_cloud_access_key
      access_key: <prisma cloud access key>
      # Default env var for below: prisma_cloud_secret_key
      secret_key: prisma_cloud_secret_key
      failure_criteria_high_severity: 1
      failure_criteria_medium_severity: 2
      failure_criteria_low_severity: 3
      failure_criteria_operator: and
      tags: env:development, team:devOps
    - docker_build_and_save
    - scan/scan_image:
      requires:
        - docker_build_and_save
      # Default env var for below: prisma_cloud_compute_url
      prisma_cloud_compute_url: <prisma cloud compute console url>
      # Default env var for below: prisma_cloud_compute_user
      prisma_cloud_compute_user: <prisma cloud compute username>
      # Default env var for below: prisma_cloud_compute_pass
      prisma_cloud_compute_pass: prisma_cloud_compute_pass
      image: 'myrepo/myimage:tag'
      image_tar: image.tar
      vulnerability_threshold: critical
      compliance_threshold: ''
      only_fixed: true

2. Check the scan results.

   After you update the config.yml, whenever a PR is created, the Prisma Cloud orb checks for any potential issues. The build is a Success or Failure depending on whether the number of the of issues detected is lower than or more than the specified threshold.

   When the scan starts, you can view the status:

   ![Status Image]

   In the following image you can view the status of the checks. The IaC scan reports as successful, while the image scan has completed the prerequisite check and is pending completion.

   ![Status Image]

   The ability to merge code is enabled only when the result is successful.
Add more commits by pushing to the auth branch on ngap-panw/terraform-scan-api.

All checks have passed
1 successful check

ci/circleci: iac_scan_job — Your tests passed on CircleCI!

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Click Details to view more information. When any of the checks are unsuccessful, the results are uploaded in a file named scan.csv in the Artifacts tab.

The following is an example of IaC scan results when the scan is successful and you have no detected security issues.

The following is an example of IaC scan results when the result was successful but with issues.

---

The following is an example of IaC scan results when the scan is successful and you have no detected security issues.

The following is an example of IaC scan results when the result was successful but with issues.

---
The following is an example of IaC scan results that returned a failure because of the number and type of security issues it found.

<table>
<thead>
<tr>
<th>Index</th>
<th>Severity</th>
<th>Name</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>medium</td>
<td>medium Access logging not enabled on S3 buckets</td>
<td>cloudtrail.tf.json;cloudtrail.tf</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>medium CloudTrail logs are not encrypted using Custom Master Keys</td>
<td>cloudtrail.tf.json;cloudtrail.tf</td>
</tr>
<tr>
<td>4</td>
<td>medium</td>
<td>medium IAM password policy allows password reuse</td>
<td>iampassword.tf</td>
</tr>
<tr>
<td>5</td>
<td>medium</td>
<td>medium IAM password policy does not have a minimum of 14 characters</td>
<td>iampassword.tf</td>
</tr>
<tr>
<td>6</td>
<td>medium</td>
<td>medium S3 object versioning is disabled</td>
<td>cloudtrail.tf.json;cloudtrail.tf</td>
</tr>
<tr>
<td>7</td>
<td>medium</td>
<td>medium VPC allows unauthorized peering</td>
<td>vpcpeering.tf</td>
</tr>
<tr>
<td>8</td>
<td>medium</td>
<td>medium VPC NACL allows traffic from blocked ports</td>
<td>$0.tf</td>
</tr>
<tr>
<td>9</td>
<td>medium</td>
<td>medium VPC NACL allows traffic from blocked ports</td>
<td>$0.tf</td>
</tr>
<tr>
<td>10</td>
<td>low</td>
<td>low AMS IAM policy attached to users</td>
<td>main.tf</td>
</tr>
</tbody>
</table>

Prisma Cloud IaC scan has failed as security issues count (high: ‘8’, medium: ‘8’, low: ‘7’) meets or exceeds the failure criteria (high:1, medium:2, low:3, operator:or).

Exited with code exit status 1

The following shows an example of container image scan results that failed because the IaC scan found security issues in the image.

<table>
<thead>
<tr>
<th>Image</th>
<th>ID</th>
<th>CVE</th>
<th>Package</th>
<th>Version</th>
<th>Severity</th>
<th>Status</th>
<th>CVSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-2281</td>
<td>libjpeg-turbo</td>
<td>1:1.5.2-2</td>
<td>critical</td>
<td>open</td>
<td>9.3</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-18224</td>
<td>libidn2</td>
<td>2.0.5-1</td>
<td>high</td>
<td>open</td>
<td>7.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-12886</td>
<td>goc-8</td>
<td>8.3.0-6</td>
<td>high</td>
<td>open</td>
<td>8.1</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-17546</td>
<td>tiff</td>
<td>4.0.10-4</td>
<td>medium</td>
<td>open</td>
<td>6.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-12920</td>
<td>libidn2</td>
<td>2.0.5-1</td>
<td>medium</td>
<td>open</td>
<td>5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-19126</td>
<td>glibc</td>
<td>2.29-10</td>
<td>low</td>
<td>open</td>
<td>2.1</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2017-16932</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>7.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-14484</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>7.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2017-18258</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-14567</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-19956</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2017-17942</td>
<td>tiff</td>
<td>4.0.10-4</td>
<td>low</td>
<td>open</td>
<td>8.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-14973</td>
<td>tiff</td>
<td>4.0.10-4</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2016-9318</td>
<td>libxml2</td>
<td>2.9.4+dfsg-1-7</td>
<td>low</td>
<td>open</td>
<td>7.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-14855</td>
<td>gnupg2</td>
<td>2.2.12-1+deb10u1</td>
<td>low</td>
<td>open</td>
<td>5.9</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2015-10228</td>
<td>glibc</td>
<td>2.28-10</td>
<td>low</td>
<td>open</td>
<td>5.9</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-18276</td>
<td>bash</td>
<td>5.0-4</td>
<td>low</td>
<td>open</td>
<td>10</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-17543</td>
<td>lz4</td>
<td>1.8.3-1</td>
<td>low</td>
<td>open</td>
<td>6.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-1551</td>
<td>openssl1</td>
<td>1.1.1d-8+deb10u2</td>
<td>low</td>
<td>open</td>
<td>5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-15847</td>
<td>goc-8</td>
<td>8.3.0-6</td>
<td>low</td>
<td>open</td>
<td>7.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2016-2781</td>
<td>coreutils</td>
<td>8.30-3</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-17371</td>
<td>libpng1.6</td>
<td>1.6.36-6</td>
<td>low</td>
<td>open</td>
<td>4.3</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-13627</td>
<td>libcrypto20</td>
<td>1.8.4-5</td>
<td>low</td>
<td>open</td>
<td>6.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-12984</td>
<td>libcrypto20</td>
<td>1.8.4-5</td>
<td>low</td>
<td>open</td>
<td>5.9</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-2644</td>
<td>systemd</td>
<td>241-7-deb10u2</td>
<td>low</td>
<td>open</td>
<td>7.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2019-3843</td>
<td>systemd</td>
<td>241-7-deb10u2</td>
<td>low</td>
<td>open</td>
<td>7.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-28899</td>
<td>systemd</td>
<td>241-7-deb10u2</td>
<td>low</td>
<td>open</td>
<td>9.8</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-21929</td>
<td>systemd</td>
<td>241-7-deb10u2</td>
<td>low</td>
<td>open</td>
<td>7.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-7169</td>
<td>shadow</td>
<td>1.4.5-1.1</td>
<td>low</td>
<td>open</td>
<td>5.3</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-1152</td>
<td>libjpeg-turbo</td>
<td>1:1.5.2-2</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2018-14498</td>
<td>libjpeg-turbo</td>
<td>1:1.5.2-2</td>
<td>low</td>
<td>open</td>
<td>6.5</td>
</tr>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>CVE-2013-0337</td>
<td>nginx</td>
<td>1.17.6-1-buster</td>
<td>low</td>
<td>open</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Vulnerability threshold check results: FAIL

Compliance

<table>
<thead>
<tr>
<th>Image</th>
<th>ID</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nginx:latest</td>
<td>f7bb5781a33c0e67</td>
<td>high</td>
<td>(CIS_Docker_CE_v1.1.0 - 4.1) Image should be created with a non-root user</td>
</tr>
</tbody>
</table>

Exited with code exit status 1
Use the Prisma Cloud Plugin for IntelliJ IDEA

With the Prisma Cloud Enterprise edition license, you can install the IntelliJ IDEA plugin that enables you to check Infrastructure-as-Code (IaC) templates and deployment files against Prisma Cloud IaC policies, within your integrated development environment (IDE). The following steps show how simple it is to install and check your IaC templates and files for potential security misconfigurations.

If you were using version 1.2 or earlier of the Prisma Cloud plugin for IntelliJ IDEA, you must update the plugin to version 1.3 or later. Use the instructions in this section to set up the plugin with the updated Prisma Cloud API URL and enter the credentials that are required to authenticate to Prisma Cloud.

1. Install the Prisma Cloud Plugin for IntelliJ
2. Configure the Prisma Cloud Plugin for IntelliJ
3. Scan Using the Prisma Cloud Plugin for IntelliJ

Install the Prisma Cloud Plugin for IntelliJ

The Prisma Cloud plugin supports IntelliJ IDEA version 2016.2 and above.

STEP 1 | In IntelliJ IDEA, select File > Settings > Plugins (on macOS, select Preferences > Plugins).

STEP 2 | On the Plugins page, select Marketplace and search for Prisma Cloud.

STEP 3 | Install the plugin.

Restart the IDE and verify that the Prisma Cloud plugin displays in the list of Installed plugins.
Configure the Prisma Cloud Plugin for IntelliJ

After you install the plugin, you must provide the Prisma Cloud API URL and Prisma Cloud access key information to authenticate and start scanning your IaC templates. If your access key changes, you'll need to update the access key information in this configuration.

**STEP 1** | In IntelliJ IDEA, select Settings > Tools > Prisma Cloud Plugin (on macOS, select Preferences > Tools > Prisma Cloud Plugin).

**STEP 2** | Enter the following information to set up the plugin.

![Prisma Cloud Plugin Configuration](image)

- **Prisma Cloud API URL.**
  
  The URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. The tenant provisioned for you is, for example, https://app2.prismacloud.io or https://app.eu.prismacloud.io. Replace `app` in the URL with `api` and enter it here. Refer to the Prisma Cloud REST API Reference, which is accessible from the Help Center within the Prisma Cloud web interface for more details.

- **Access Key.**
  
  The access key enables programmatic access to Prisma Cloud. If you do not have a key, you must [Create and Manage Access Keys](#).

- **Secret Key.**
  
  You should have saved this secret key when you generated it. You cannot view it on the Prisma Cloud web interface.

- **Asset Name**
  
  Enter an asset name to identify the repository you want to scan.

- **Tags.**
  
  Define tags to organize the templates that are scanned with this service connection, for visibility on Prisma Cloud.
STEP 3 | Add the Prisma Cloud configuration file.

The Prisma Cloud configuration file supports IaC scanning of complex module structures and variable formats. To add this file, create a subdirectory and file .prismaCloud/config.yml in the root folder of your project or repository branch. See Set Up Your Prisma Cloud Configuration File for IaC Scan for details.

Scan Using the Prisma Cloud Plugin for IntelliJ

Now, you are ready to scan your templates and view the results before you check it in to the repository or pipeline.

You must have a Prisma Cloud Enterprise edition license and valid credentials to scan IaC templates.

STEP 1 | Scan the files for insecure configurations.

Right-click to scan your template file or folder in the IDEA Project window and select Prisma Scan.
STEP 2 | View the results of the scan in the Scan Result tool window.

The title of the Scan Result window includes the date and time of the scan. For each scan, a new scan result window is added. The tab situated farthest to the right displays the results of the latest scan.

If the scan detects no potential issues, the message displays as follows:
If the scan detects any policy violations, the scan result displays the following details for each violation:

- Name of the violated policy
- Description of the violated policy
- Severity of the violation
- Name of the file with the issue

By default, the results are sorted by severity. You can sort the Scan Result using the policy name also.

The following examples show scan results for various template types. The first example shows the result of scanning a Kubernetes deployment file with content that violates policies. You will need to change content of your Prisma Cloud configuration file, .prismaCloud/config.yml, depending on the template types and variables in your project.
The following example shows the result of scanning a folder with CloudFormation templates that have policy violations.
The example below shows the result of scanning a folder with Terraform 0.12 templates that contain a policy violation.
Use the Prisma Cloud App for GitHub

This Prisma Cloud app for GitHub enables you to scan IaC templates to check them against security policies when you open a pull request. For each pull request, you can define the pass criteria and view the scan results directly on GitHub. When the defined criteria are not met, the pull request fails and you can view all the checks that failed. In addition, the Prisma Cloud app creates an issue and adds the scan results as comments, so that you can fix all the issues reported before the changes are merged to the repository.

Use this app for scanning files in a private GitHub repository that has enabled restricted access. Be sure to create a read-only role on Prisma Cloud and to generate a secret key and access key for a user. You will need to provide these credentials to authenticate to Prisma Cloud for API access for the scanning capabilities.

1. Set up the Prisma Cloud App Files for GitHub
2. Install the Prisma Cloud App for GitHub

Recent versions of the app capture the Prisma Cloud credentials as part of the installation process and no longer require the credentials to be hard-coded in configuration file .github/prisma-cloud-config.yml. Existing customers should remove the credentials from this file after the app upgrade.

Set up the Prisma Cloud App Files for GitHub

To set up for IaC scans for a repository, you need to create IaC scan configuration files. These files enable you to control the behavior of your scans to meet your needs for that repository. For example, depending on the thresholds you defined in these files, the Prisma Cloud app will perform checks that allow or fail requests to merge or commit changes.

Creating these files before you install the Prisma Cloud app for GitHub enables the app installation itself to run a full IaC scan of selected repositories as part of the installation.

The three new files are:
- The Prisma Cloud configuration file .prismaCloud/config.yml
  This file identifies the templates types you wish to scan.
- .github/prisma-cloud-config.yml
  This file includes the criteria that defines whether or not you allow the commit for the pull request.
- .github/prisma-template-for-scan-results.yml
  This file specifies how the scan results are made available to the person who created the pull request.

STEP 1 | Set Up Your Prisma Cloud Configuration File for IaC Scan.

Create the .prismaCloud/config.yml in the root directory of your repository branch. This file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

STEP 2 | Create the prisma-cloud-config.yml file to support the ability to scan IaC templates.

1. Select Create new file.
   Add a new folder called .github, and name the file prisma-cloud-config.yml. The path should be <your repository name>/github/prisma-cloud-config.yml.
2. Copy the template for this new file.

Copy and paste the following contents into `.github/prisma-cloud-config.yml`.

```yaml
# Please update with the respective environment values and commit
# to master branch under the .github folder before performing scans

# Define the failure criteria for creating checks. If the criteria
# matches a check will be created. The template for the checks can
# be customized in the "/.github/prisma-template-for-scan-results" file.
failure_criteria_for_creating_checks:
  high: 1
  medium: 1
  low: 1
  operator: or

# Define the failure criteria for creating issues. If the criteria
# matches an issue will be created. The template for issues can be
# customized in the "/.github/prisma-template-for-scan-results" file.
failure_criteria_for_creating_issues:
  high: 1
  medium: 1
  low: 1
  operator: or

# Define github asset name
github_asset_name: "Github Asset Dev"

tags:
- phase:testing
- env:QA
```
3. Define the parameter values in prisma-cloud-config.yml.

The parameters in prisma-cloud-config.yml define the failure criteria for pull requests. You can set `failure_criteria_for_creating_checks` to define the number and severity of security policy check failures that need to occur to trigger a merge request failure. The syntax for the `failure_criteria_for_creating_checks` value is as follows.

```yaml
high: x
medium: y
low: z
operator: op
```

In the syntax above, x is a count of high-severity policy check failure, y is a count of medium-severity policy check failures, and z is a count of low-severity policy check failures. The `operator` value determines what combination of High/Medium/Low counts should result in a merge request failure. The default for each count is 0. The value for operator, op, can be either OR or AND. The default is OR. Some examples of settings for `failure_criteria_for_creating_checks` are as follows.

- The setting below would result in a failed merge request security check for any detected policy check failure

```yaml
high: 0
medium: 0
low: 0
operator: OR
```

- The setting below would result in merge requests never failing a security check.

```yaml
high: 1000
medium: 1000
low: 1000
operator: AND
```

You can also use `failure_criteria_for_creating_issues` to define the number and severity of security policy check failures that need to occur to trigger creation of a GitHub issue, during a pull request. The syntax of the variable value is the same as that for `failure_criteria_for_creating_checks`. The value includes `high`, `medium`, and `low` counts and includes an `operator` whose possible values are `AND` and `OR`.

Prisma Cloud uses the asset name to track results. Some example names are creditapp_server and ConsumerBU_server.

Prisma Cloud tags enable visibility on the Prisma Cloud administrator console.

**STEP 3** Create the `.github/prisma-template-for-scan-results.yml` file to support how the scan results are displayed.

Create the file `.github/prisma-template-for-scan-results.yml` with the same steps you used to create `.github/prisma-cloud-config.yml`.

1. Select **Create new file** and add file `.github/prisma-template-for-scan-results.yml` just as you created `.github/prisma-cloud-config.yml` earlier.
2. Copy the template for the newly created `prisma-template-for-scan-results.yml` file.
Install the Prisma Cloud App for GitHub

You must set up the app to authenticate to Prisma Cloud, and you can optionally customize the scan settings.

**STEP 1** | Search for Prisma Cloud on the GitHub marketplace.

**STEP 2** | Select **Settings > Integrations & services > Add service** and add **Prisma Cloud**.

This app requires the following permissions:

- Read access to code, to perform scan on template files.
- Read/write access to check for issues and open pull requests.
- Read access to metadata.
STEP 3 | Specify where you want to install the app.

You can choose to install the Prisma Cloud app for GitHub on all repositories or only on selected repositories. You can change this setting later to include more repositories for scanning.

Repository access

- **All repositories**
  - This applies to all current and future repositories.

- **Only select repositories**
  - Select repositories
  - Selected 1 repository:
    - ANW/TestPrismaScanApp

STEP 4 | Specify the Prisma Cloud API URL, Prisma Cloud access key ID, and corresponding secret key to use for the integration.

The Prisma Cloud API URL you specify depends on the region and cluster of your Prisma Cloud tenant. For example, if your Prisma Cloud admin console URL is https://app.prismacloud.io, then your Prisma
Cloud API URL is https://api.prismacloud.io. See the Prisma Cloud REST API Reference for a list of Prisma Cloud API URLs.

See Create and Manage Access Keys for details about Prisma Cloud access keys.

Once you’ve entered your settings, select Validate. If the settings are valid, a Save button appears, which enables you to save your settings.

**STEP 5** To add other repositories or to modify the configuration, you can select Settings > Integrations & services > Prisma Cloud to Configure the app.

Whenever you use this option to add repositories, the addition will result in an IaC scan of the repository if all the configuration files for the Prisma Cloud app are set up.
Installed GitHub Apps

GitHub Apps augment and extend your workflows on GitHub with commercial, open source, and homegrown tools.

Prisma Cloud

Configure

Services

Services are pre-built integrations that perform certain actions when events occur on GitHub.
Use the Prisma Cloud Extension for GitLab

Use the Prisma Cloud extension to scan IaC templates in the build or release phase of the GitLab CI/CD pipeline or SCM when you create or merge a request. Container image or serverless function scanning is not available with this plugin, currently.

- Use the Prisma Cloud Extension for the GitLab CI/CD Pipeline
- Use the Prisma Cloud Extension for GitLab SCM

Use the Prisma Cloud Extension for the GitLab CI/CD Pipeline

You can use the Prisma Cloud extension for GitLab CI/CD to scan IaC templates to check against Prisma Cloud policies or to scan container images to check for vulnerabilities.

To scan IaC templates in the build or release phase of the GitLab CI/CD pipeline, you need to configure the Prisma Cloud extension. The first step is to set up a connection to the Prisma Cloud API server and configure the details—as environment variables—for that connection. Then, the IaC scanning capability becomes available as a script that you can embed as a custom task in your GitLab pipelines. You can trigger the custom task to scan on every commit (pull request) or on a schedule, and specify the build or release pipeline failure criteria based on the severity of the security issues that it identifies. The scan uses the failure thresholds you specify to pass or fail the check. When the scan is successful, the code can be merged. If the scan is unsuccessful, the security issues must be fixed in order to merge code changes.

The list of inputs that are required for scanning IaC templates in the build or release phase of the GitLab CI/CD pipeline are:

- Connection settings as environment variables to enable communication between the Prisma Cloud API server and your GitLab repository.
- The .gitlab-ci.yml at the root level in your repository. For any commit or push to your repository, this file start jobs on GitLab runners according to the contents of the file. You must have a shared runner or a project-specific/custom runner for the job to run successfully.
- iac_scan.sh script. This script uses the values that you provided in the environment variables to call the Prisma Cloud API endpoint. You have the flexibility either to provide the path where this iac_scan.sh script resides within your repository in the .gitlab-ci.yml file or to copy the script itself into the gitlab-ci.yml file.
- When the script runs, if you have any missing or incorrect environment variables, an error message displays on the pipeline console.
- config.yml file at the root-level within the project under the .prismaCloud directory. The path for this file must be .prismaCloud/config.yml. Prisma Cloud requires this configuration file to learn about your IaC module structure, runtime variables, and tags so that it can scan the IaC templates in your repository.

The Prisma Cloud extension also enables you to scan container images for vulnerabilities. The steps to configure the extension for container image scans are similar to those for IaC scans in that you need to connect to Prisma Cloud Compute through environment variables and that container image scanning is available through a script that you invoke through a job.

The list of resources that are required for scanning container images are:
• Connection settings as environment variables to enable communication between the Prisma Cloud Compute console and your GitLab repository.
• The .gitlab-ci.yml at the root level in your repository.
• container_scan.sh

This script uses the values that you provide in the environment variables to call the Prisma Cloud Compute endpoint. You have the flexibility either to provide the path where this container_scan.sh script resides within your repository in the .gitlab-ci.yml file or to copy the script itself into the gitlab-ci.yml file.
• config.yml file at the root-level within the project under the .prismaCloud directory.

To set up the Prisma Cloud GitLab extension for CI/CD:
• Configure the Prisma Cloud Extension for GitLab CI/CD
• Set Up a Custom Job for IaC Scan
• Set Up a Custom Job for Container Image Scan
• Prisma Cloud Custom Script—iac_scan.sh

Configure the Prisma Cloud Extension for GitLab CI/CD

The GitLab extension is made up of scripts, and configuration includes defining environment variables and creating scripts.

Much of the configuration involves defining environment variables in your GitLab project settings. The table below summarizes the environment variables you will set to configure your project for both IaC scans and container image scans.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prisma_cloud_api_url</td>
<td>Prisma Cloud base API URL (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_access_key</td>
<td>Prisma Cloud access key for API access (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_secret_key</td>
<td>Secret key that corresponds to Prisma Cloud access key (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_cicd_asset_name</td>
<td>GitLab server (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_cicd_failure_criteria</td>
<td>String that defines criteria that should cause a pipeline failure (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_cicd_tags</td>
<td>Prisma Cloud tags for future use (for IaC scan)</td>
</tr>
<tr>
<td>prisma_cloud_compute_url</td>
<td>Base URL for the Prisma Cloud Compute console (for container image scan)</td>
</tr>
<tr>
<td>prisma_cloud_compute_username</td>
<td>Prisma Cloud Compute user with the CI User role (for container image scan)</td>
</tr>
<tr>
<td>prisma_cloud_compute_password</td>
<td>Prisma Cloud Compute user password (for container image scan)</td>
</tr>
</tbody>
</table>
STEP 1 | Set up the connection to the Prisma Cloud API.

1. Add the connection settings as environment variables to Project > Settings > CICD > Variables.

   - **Variables**
     - Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. Additionally, they can be masked so they are hidden in job logs, though they must match certain regexp requirements to do so. You can use environment variables for passwords, secret keys, or whatever you want. You may also add variables that are made available to the running application by prepending the variable key with `KBS_SECRET_`. More information

   ![](variables-table.png)

   - **Type** | **Key** | **Value** | **Protected** | **Masked** | **Environments**
   - Var | access_key | dfgdfdgdgfdgfdgfdg | ✗ | ✗ | All (default)
   - Var | cicd_asset_name | gitlab cicd | ✗ | ✗ | All (default)
   - Var | cicd_failure_criteria | High : 0, Medium : 0, Low : 0,... | ✗ | ✗ | All (default)
   - Var | cicd_tags | Org:Engineering,Team:red | ✗ | ✗ | All (default)
   - Var | prisma_cloud_api_url | https://api.prismacloud.io | ✗ | ✗ | All (default)
   - Var | secret_key | testtesttestesttesttest | ✗ | ✗ | All (default)

2. Set the Prisma Cloud API URL as the value for the `prisma_cloud_api_url` environment variable.

   The API URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. If the tenant provisioned for you is, for example, `https://app2.prismacloud.io` or `https://app.eu.prismacloud.io`, replace `app` in the URL with `api` and enter it here. Refer to the Prisma Cloud REST API Reference, for more details.

3. Add your Prisma Cloud access key as the value for the `prisma_cloud_access_key` environment variable.

   The access key enables programmatic access. If you do not have a key, see Create and Manage Access Keys.

4. Add your GitLab server name as the value for the `prisma_cloud_cicd_asset_name` environment variable.

   On Prisma Cloud, the asset name is used to track results. Some examples names are - `creditapp_server`, `ConsumerBU_server`, etc

STEP 2 | Set up environment variables for container image scans.

Set up the following environment variables only if you want to run container image scans. As with the environment variables that support IaC scans, you navigate to Project > Settings > CICD > Variables to add new environment variables.
1. Add `prisma_cloud_compute_url`, whose value is the base URL for your Prisma Cloud Compute console. An example value is `http://console<example>.com:8083`.

2. Add `prisma_cloud_compute_username`, whose value is the Prisma Cloud Compute user with a CI user role.

3. Add `prisma_cloud_compute_password`, whose value is the password for the Prisma Cloud Compute user.

**STEP 3 | Set Up Your Prisma Cloud Configuration File for IaC Scan.**

Create the `.prismaCloud/config.yml` file and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

**Set Up a Custom Job for IaC Scan**

**STEP 1 | Create the iac_scan.sh custom script.**

Use the sample Prisma Cloud Custom Script—iac_scan.sh to create the file. Then, add the file to a folder from where it can be accessed in your GitLab pipeline. This file enables you to view the scan results.

**STEP 2 | Add Prisma Cloud IaC scan job to the GitLab CI configuration.**

The GitLab CI configuration is stored in the `.gitlab-ci.yml` file.

1. Make the script executable.
   
   ```bash
   chmod +x ./prismacloud-scripts/iac_scan.sh
   ```

2. Add the path to your IaC templates in the gitlab-ci.yml file.

   Add the command `./prismacloud-scripts/iac_scan.sh $CI_BUILDS_DIR/prisma_scan` to your gitlab-ci.yml file. `$CI_BUILDS_DIR/prisma_scan` is the path to your IaC templates location.

   ```bash
   $CI_BUILDS_DIR/prisma_scan
   ```

   *The IaC templates in this directory must not exceed the 1 MB size limit.*

Refer to the GitLab documentation to learn about the `gitlab-ci.yml` file. A sample file is included here.

```yaml
#variables are specific to your environment, please change accordingly.
```
variables:
  GIT_STRATEGY: fetch
  GIT_CHECKOUT: "true"
  GIT_CLONE_PATH: $CI_BUILDS_DIR/prisma_scan

prisma-cloud-scan:
  stage: build
  before_script:
    - apt-get update -qy
    - apt-get install -y jq
    - wget https://github.com/mikefarah/yq/releases/download/3.2.1/yq_linux_386
    - mv yq_linux_386 /usr/local/bin/yq
    - chmod +x /usr/local/bin/yq
    - apt-get install bsdmainutils # needed for displaying file output in column format on console
    - apt-get -y install zip unzip
  script:
    # If you wish to pull code of your project using git clone before next steps if wish to clone at different stage than build.
    
    # Here ./prismacloud-scripts/iaa_scan.sh is the location of the script in the gitlab repo or wherever it is stored. $CI_BUILDS_DIR/prisma_scan is the argument to file which is 'full cloned repository path'.
    # prisma_scan is the project/repository name.
    # The chmod +x ./prismacloud-scripts/iaa_scan.sh is required to make the script executable.
    - chmod +x ./prismacloud-scripts/iaa_scan.sh
    - ./prismacloud-scripts/iaa_scan.sh $CI_BUILDS_DIR/prisma_scan
    
    # You can also pass the absolute repo path to the script as shown below
    # ./prismacloud-scripts/iaa_scan.sh /build/prisma_scan
  artifacts:
    when : always
    paths:
      - report/scan_results.csv

STEP 3 | Set up the failure criteria for the Prisma Cloud IaC scan.

Define the number of issues by severity in the `prisma_cloud_cicd_failure_criteria` environment variable. Set the High : x, Medium : y, Low : z, Operator: O, where, x,y,z is the number of issues of each severity, and the operator is OR, AND.

For example:
- To fail the pipeline for any security issue detected, `prisma_cloud_cicd_failure_criteria = High : 0, Medium : 0, Low : 0, Operator: OR`
- To never fail the pipeline, `prisma_cloud_cicd_failure_criteria = High : 1000, Medium : 1000, Low : 1000, Operator: AND`

STEP 4 | Set up the Prisma Cloud tags.

Prisma Cloud tags are different from GitLab tags or cloud tags that you may have included within your IaC templates. Prisma Cloud tags enable visibility on the Prisma Cloud administrator console.
Provide the values as a comma separated list of tags in the `prisma_cloud_cicd_tags` environment variable. For example, `prisma_cloud_cicd_tag:project:x,owner:mr.y,compliance:pci`.

**STEP 5 | View IaC scan results.**

The Prisma Cloud IaC scan uses the failure criteria you defined in the `prisma_cloud_cicd_failure_criteria` environment variable to pass or fail a scan. When it detects a security issue, it generates an artifact.

- To download the artifact, select **Project > CI/CD > Pipeline**, select the job and **Download** artifacts for the job.

![IaC Scan result when the scan is successful and you have no security issues.](image1)

- IaC Scan result when the scan fails.

![IaC Scan result when the scan fails.](image2)

- The Prisma Cloud artifact is a `.csv` file that lists the security issues detected. Download the artifact and open `report/scan_results.csv` to view the list of issues.

![Scan results](image3)

- View the Prisma Cloud IaC scan results on the console of CI/CD pipeline log output (**Project > Setting > Pipeline**, select the job and view **Log console output**).

<table>
<thead>
<tr>
<th>Index Severity</th>
<th>Name</th>
<th>Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 medium</td>
<td>AWS S3 Object Versioning is disabled</td>
<td>This policy identifies the S3 buckets which have Object Versioning disabled.</td>
<td>for-expressions</td>
</tr>
</tbody>
</table>

**Set Up a Custom Job for Container Image Scan**

You can add container image scans to the extension you have already configured for IaC scans, or you can configure the extension to perform just container image scans.

**STEP 1 | Create the custom script to scan container images.**
Use the sample Prisma Cloud custom script below, container_scan.sh, to create your custom script. Add your script to a folder where your GitLab pipeline can access it. This file enables you to invoke the container image scan and to view the results of the scan.

```bash
#!/bin/bash
#######container_scan.sh - Container scan###########
if [[ "$1" != "" ]];then
    image=$1
    echo "image name:"$image
else
    echo "Please enter the image name to be scanned as an argument to bash script."
    exit 1;
fi
mkdir -p securethecloud
docker save $image -o securethecloud/image.tar
docker load -i securethecloud/image.tar
curl -u $prisma_cloud_compute_username:$prisma_cloud_compute_password --output ./twistcli $prisma_cloud_compute_url/api/v1/util/twistcli
chmod +x ./twistcli
./twistcli --version
IMAGEID=`docker images $image --format {{.ID}}`
./twistcli images scan --details --address $prisma_cloud_compute_url -u $prisma_cloud_compute_username -p $prisma_cloud_compute_password $IMAGEID if [ "$?$" == "1" ]; then
    exit 1;
fi
```

**STEP 2 |** Add a job to your GitLab CI configuration, that invokes your custom script.

The sample below shows a .gitlab-ci.yaml file that includes the details for both IaC and container image scanning. Refer to the GitLab documentation to learn about the gitlab-ci.yml file. In this sample, the job prisma-cloud-container-scan invokes the container image scan.

```yaml
variables:
  GIT_STRATEGY: fetch
  GIT_CHECKOUT: "true"
  GIT_CLONE_PATH: $CI_BUILDS_DIR/prisma_scan

prisma-cloud-iac-scan:
  stage: build
  before_script:
    - apt-get update -qy
    - apt-get install -y jq
    - wget https://github.com/mikefarah/yq/releases/download/3.2.1/yq_linux_386
    - mv yq_linux_386 /usr/local/bin/yq
    - chmod +x /usr/local/bin/yq
    - apt-get install bsdmainutils #needed for displaying file output in column format on console
    - apt-get -y install zip unzip
  script:
    # If necessary, you can use git clone in a different step to pull your project code
```
STEP 3 | Add a vulnerability scan rule on the Prisma Cloud Compute Console.

1. In the Prisma Cloud UI, select **Compute** > **Defender** > **Vulnerabilities** > **CI**.

2. Select **Add Rule** and enter a **Rule name**.
3. Specify the Alert and Failure thresholds.
4. (Optional) Specify the image to scan.
   The alternative is to specify the image to scan in your GitLab job, as in step 2 above.
5. (Optional) Select Advanced Settings, for options to refine your rule further.

   For more details on the advanced settings, see the Prisma Cloud Compute guide

STEP 4 | Run your pipeline and view the job results.

   You can view the results of the container image scan on the CI/CD pipeline log console output. In GitLab, navigate to **Project > Settings > Pipeline**, select your job, and view **Log console output**.
Prisma Cloud Custom Script—iac_scan.sh

iac_scan.sh script for Prisma Cloud IaC scan GitLab CI/CD extension.

```bash
#!/bin/bash

# Perform IaC scan

#echo "Entered full cloned repo path:" $1

if [[ "$1" != "" ]]; then
  repo_path=$1
else
  echo "Please enter the full cloned repository path on build server/runner. For details refer to https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
  exit 1;
fi

#echo "repo_path:" $repo_path

ls -al $repo_path

#read ENV variables

echo $prisma_cloud_api_url $prisma_cloud_access_key # $prisma_cloud_secret_key

if [[ -z "$prisma_cloud_api_url" ]]; then
  exit 1;
```

---

438  PRISMA™ CLOUD ADMINISTRATOR’S GUIDE  |  Prisma Cloud DevOps Security

© 2020 Palo Alto Networks, Inc.
```bash
echo "Please enter a valid URL. For details refer to https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
exit 1;
fi
if [[ -z "$prisma_cloud_access_key" || -z "$prisma_cloud_secret_key" ]]; then
  echo "Invalid credentials, verify that access key and secret key in environment variables are valid. For details refer to https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
  exit 1;
fi
if [[ ! -f $repo_path/.prismaCloud/config.yml ]]; then
  echo "Can not find config.yml under .prismaCloud folder in repo $CI_PROJECT_TITLE. Please make sure the file is present in correct format https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html at the root of your repo under .prismaCloud folder."
  exit 1;
fi
if [[ -z "$prisma_cloud_cicd_asset_name" ]]; then
  echo "Please enter a valid cicd asset name. For details refer to https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
  exit 1;
fi
#####Compress the repo and check if compressed zip file size>5MB#################
#echo "current path:"
#pwd
cd $repo_path
#ls -al .
zip -r $repo_path/iacscan.zip . -x ".git\" #here cd inside repo_path and '.' as source is mandatory else while zipping copies else it will zip from root instead of files inside repo
#echo "after zip content of repo_path"
#ls -al $repo_path
file_size="$(wc -c $repo_path/iacscan.zip | awk '{print $1}')"
#echo "file_size:"
file_size_limit=5242880
if [[ "$file_size" -gt "$file_size_limit" ]]; then
  echo "Directory size $repo_path more than 8MB is not supported"
  exit 1;
fi
#$CI_PROJECT_DIR is default inbuilt dir used to upload the artifacts but you can change to any one the job has access to.
#echo "view content of $CI_PROJECT_DIR"
#ls -al $CI_PROJECT_DIR

#Check failure criteria exists, if not default 0,0,0,or######
if [[ -z "$prisma_cloud_cicd_failure_criteria" ]]; then
  failure_criteria_high_severity=0
  failure_criteria_medium_severity=0
  failure_criteria_low_severity=0
  failure_criteria_operator="or"
```
else
  echo "failure criteria:" $prisma_cloud_cicd_failure_criteria
  cicd_failure_criteria_removed_spaces=$(printf '%%s' $prisma_cloud_cicd_failure_criteria)
  #-- echo $cicd_failure_criteria_removed_spaces
delimiter=,
s=$cicd_failure_criteria_removed_spaces$delimiter
array=();
while [[ $s ]]; do
  array+=( "${s%%"$delimiter"*}" );
s=${s#*"$delimiter"};
done;
#-- declare -p array
  failure_criteria_high_severity=$(awk -F':' '{print $2}' <<< "$array[0]"
  failure_criteria_medium_severity=$(awk -F':' '{print $2}' <<< "$array[1]"
  failure_criteria_low_severity=$(awk -F':' '{print $2}' <<< "$array[2]"
  failure_criteria_operator=$(awk -F':' '{print $2}' <<< "$array[3]"
# echo "Failure Criteria:" $failure_criteria_high_severity
  $failure_criteria_medium_severity $failure_criteria_low_severity
  $failure_criteria_operator
fi

#################################################
# Read .prismaCloud/config.yml and form headers for scan
#################################################

fileContents=$(yq read -j $repo_path/.prismaCloud/config.yml)
# echo "file contents are:" $fileContents
t_Type="$(echo "$fileContents" | jq -r '.template_type')"
# echo "template type:" $t_Type
headers=""
url=""
if [[ ! -z "$t_Type" ]]; then
templateType=${t_Type^^}
# echo $templateType
else
  echo "No valid template-type found in config.yml file in repo
  $CI_PROJECT_TITLE. Please specify either of these values: TF, CFT or K8s as
  template-type variable in the config.yml"
  exit 1;
fi

if [[ "$templateType" == "TF" ]]; then
  url="$prisma_cloud_api_url/iac/tf/v1/scan"
  terraformVersion="$(echo "$fileContents" | jq -r '.terraform_version')"
  if [[ ! -z "$terraformVersion" && "$terraformVersion" == "0.12" ]]; then
    headers="-H terraform-version:$terraformVersion"
  fi
  read terraform 0.12 parameters
  isterraform12ParamsPresent="$(echo "$fileContents" | jq -r '.terraform_012_params')"
  if [[ "$isterraform12ParamsPresent" != null ]]; then
    terraformContents="$(echo "$fileContents" | jq -r '.terraform_012_params[] |= with_entries { .key |= gsub("root_module"); "root-module") }' | jq -r '.terraform_012_params[] |= with_entries { .key |= gsub("variable_files"); "variable-files") }')"
    terraform012Parameters="$(echo "$terraformContents" | jq -r '.terraform_012_params[] |= tr -d '\n' | tr -d '[[:blank:]]')"
    if [[ "$terraform012Parameters" != null ]]; then
      headers="-H terraform-012-parameters:$terraform012Parameters"
fi
    fi
else
  #headers+=" -H terraform-version:0.11" no version header needed for 0.11
  #read terraform 0.11 parameters
  variableFiles="$(echo "$fileContents" | jq -r 
'.terraform_011_parameters.variable_files')"
  variableValues="$(echo "$fileContents" | jq -r 
'.terraform_011_parameters.variable_values')"
  if [[ "$variableFiles" != null ]]; then
    headers+=" -H rl-variable-file-names:$variableFiles"
  fi
  if [[ "$variableValues" != null ]]; then
    headers+=" -H rl-parameters:$variableValues"
  fi
fi
elif [[ "$templateType" == "CFT" ]]; then
  url="$prisma_cloud_api_url/iac/cft/v1/scan"
  variableValues="$(echo "$fileContents" | jq -r 
'.cft_parameters.variable_values' | tr -d '
	' | tr -d '"[:blank:]')"
  if [[ "$variableValues" != null ]]; then
    headers+=" -H 'rl-parameters:$variableValues'"
  fi
elif [[ "$templateType" == "K8S" ]]; then
  url="$prisma_cloud_api_url/iac/k8s/v1/scan"
else
  echo "No valid template-type found in config.yml file in repo $CI_PROJECT_TITLE. Please specify either of these values: TF, CFT or K8s as template-type variable in the config.yml"
  exit 1;
fi

# LOGIN TO GET TOKEN
#echo "Get token using login api"

result=$(curl -k -i -X POST $prisma_cloud_api_url/login --user-agent "GitLab PrismaCloud/DevOpsSecurity-1.0.0" -H 'Content-Type:application/json' -d "\"username\":"\"${prisma_cloud_access_key}\"","\"password\":\"${prisma_cloud_secret_key}\""")
#echo $result
code=$(echo "$result" | grep HTTP | awk '{print $2}')
echo $code

if [[ "$code" -eq 400 || "$code" -eq 401 || "$code" -eq 403 ]]; then
  echo "Invalid credentials, verify that access key and secret key in environment variables are valid. For details refer to https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
  exit 1;
elif [[ "$code" -eq 500 || "$code" -eq 501 || "$code" -eq 503 ]];then
  echo "Oops! Something went wrong, please try again or refer to documentation https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html"
  exit 1;
elif [[ "$code" -ne 200 ]];then
  echo "Oops! Something went wrong, please try again or refer to documentation https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-
```bash
cloud-admin/prisma-cloud-devops-security/use-the-prisma-cloud-app-for-gitlab.html

exit 1;

fi

output_response=$(echo "$result" | grep token)

token=$(echo "$output_response" | jq .token | tr -d '"

# Start PROCESSING PRISM CLOUD IAC SCAN

#echo url:$url
#echo header:$headers

#form prisma-tags
prisma_tags=""

if [[ ! -z "$prisma_cloud_cicd_tags" ]]; then
  temp_str=$(printf '%s' $prisma_cloud_cicd_tags)
  if [[ ! -z "$temp_str" ]]; then
    settings_tags="$(sed 's/,/"","/g' <<< "$temp_str")"
    prisma_tags="""settings-tags":"$settings_tags"
  fi
fi

#tags from config.yml
repo_tags=$(echo "$fileContents" | jq -r '.tags' | tr -d '\n\t' | tr -d '

if [[ $repo_tags != null ]]; then
  prisma_tags+=","repo-tags":$repo_tags"
fi

# creating metadata structure
metadata_json={"asset-name": "$prisma_cloud_cicd_asset_name" , "asset-type": "Gitlab" , "user-id": "$GITLAB_USER_LOGIN" , "prisma-tags": {"prisma_tags"}, "scan-attributes": {"build-number": "$CI_JOB_ID" , "project-name": "$CI_PROJECT_TITLE" } , "failure-criteria": {"high": "$failure_criteria_high_severity" , "medium": "$failure_criteria_medium_severity" , "low": "$failure_criteria_low_severity" } , "operator": "$failure_criteria_operator" }

#cd $CI_BUILDS_DIR
#ls
cp $repo_path/iacscan.zip .
response=$(curl -k -X POST $url -H "x-redlock-auth:${token}" --user-agent "GitlabCI PrismaCloud/DevOpsSecurity-1.0.0" $headers -H "x-redlock-iac-metadata:${metadata_json}" -F templateFile=iacscan.zip)
#echo $response

result=$(echo "$response" | jq '.result.is_successful')
mkdir results
if [[ "$result" == true ]]; then
  matched=$(echo "$response" | jq '.result.rules_matched')
  if [[ $matched != null ]]; then
    stats=$(echo "$response" | jq '.result.severity_stats')
    echo $matched | jq '["Severity", "Name", "Description", "Files"], (map({severity, name, description, files}))' | @csv' | tr -d '
"'> results/stats.csv
    awk -F'"\t"' -v OFS='"\t"'
```

---

© 2020 Palo Alto Networks, Inc.
NR == 1 {print "Index", $0; next}
{print (NR-1), $0}
'results/scan.csv > results/scan_results.csv

#format console output file to display
echo $matched | jq '(["Severity","Name","Files"], (map({severity, name, files} ) | .[] | [.severity, .name, (.files[]join(";"))]) | @csv) | column -t -s "," | tr -d "\\n" | results/formatted.csv
awk -F'\\t' -v OFS='\\t' '
NR == 1 {print "Index", $0; print

next}
{print (NR-1), $0}
'results/formatted.csv > results/console_output.csv
#show result on console
cat results/console_output.csv

#echo $CI_PROJECT_DIR
mkdir $CI_PROJECT_DIR/report
cp -r results/scan_results.csv $CI_PROJECT_DIR/report
#ls -la $CI_PROJECT_DIR/report

high="$(echo "$stats" | jq -r '.high')"
med="$(echo "$stats" | jq -r '.medium')"
low="$(echo "$stats" | jq -r '.low')"
if [[ ( ( $failure_criteria_operator == "or" ) &&
( "$high" -ge $failure_criteria_high_severity ) ||
( "$medium" -ge $failure_criteria_medium_severity ) ||
( "$low" -ge $failure_criteria_low_severity ) ) ]
then
echo "Prisma Cloud IaC scan failed with issues as security issues count (high:$high, medium:$med, low:$low) meets or exceeds the failure criteria (high:$failure_criteria_high_severity, medium:$failure_criteria_medium_severity, low:$failure_criteria_low_severity, operator:$failure_criteria_operator)"
exit 1;
else
echo "Prisma Cloud IaC Scan has been successful as security issues count (high:$high, medium:$med, low:$low) does not exceed the failure criteria (high:$failure_criteria_high_severity, medium:$failure_criteria_medium_severity, low:$failure_criteria_low_severity, operator:$failure_criteria_operator)"
exit 0;
fi
else
error_message="$(echo "$response" | jq -r '.result.error_details')"
echo "$error_message"
exit 1;
fi

Use the Prisma Cloud Extension for GitLab SCM

Use the Prisma Cloud extension to scan IaC templates when you create or update a merge request. You can define failure criteria for each GitLab project and view the scan results directly in the GitLab user interface.
In addition, the Prisma Cloud extension can create GitLab issues that report details from IaC scans for checks against security policies. This ability enables you to fix all the reported issues before your changes are merged into the repository.

The sections below describe how to set up the Prisma Cloud extension and how to use it.

- Configure the Prisma Cloud Extension for GitLab SCM
- Run an IaC Scan in a Merge Request

Configure the Prisma Cloud Extension for GitLab SCM

The Prisma Cloud Extension for GitLab SCM does not require a separate software installation, but does require the following configuration steps.

much of the configuration involves setting environment variables in your GitLab project settings. The image below summarizes the environment variables you will set to configure your project for IaC scans.

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Value</th>
<th>Protected</th>
<th>Masked</th>
<th>Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>prisma_cloud_access_key</td>
<td>**************</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_api_url</td>
<td><a href="https://api.prismacloud">https://api.prismacloud</a>....</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_scm_asset_name</td>
<td>Org Asset Value</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_scm_failure_iss...</td>
<td>High : 0, Medium : 0, Low: 0, ...</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_scm_failure_mr...</td>
<td>High : 5, Medium : 5, Low: 5, ...</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_scm_tags</td>
<td>Org:Engineering,Team:Shift_L...</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
<tr>
<td>Variable</td>
<td>prisma_cloud_secret_key</td>
<td>**************</td>
<td>X</td>
<td>X</td>
<td>All (default)</td>
</tr>
</tbody>
</table>

If you want to run IaC scans for both GitLab SCM and GitLab CICD in a single project, you can set environment variables for both in your project settings.

**STEP 1** | Set environment variables to support a connection to the Prisma Cloud API.

1. In GitLab, navigate to Project > Settings > CICD > Variables, and add the connection settings as environment variables.
2. Set the Prisma Cloud API URL as the value for the `prisma_cloud_api_url` environment variable.

The API URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. If the tenant provisioned for you is, for example, `https://app2.prismacloud.io` or `https://app.eu.prismacloud.io`, replace app in the URL with api and enter it here. Refer to the Prisma Cloud REST API Reference for more details.

3. Add your Prisma Cloud access key as the value for the `prisma_cloud_access_key` environment variable.

The access key enables programmatic access. If you do not have a key, see Create and Manage Access Keys.

4. Add your GitLab server name as the value for the `prisma_cloud_scm_asset_name` environment variable.

Prisma Cloud uses the asset name to track results. Some example names are `creditapp_server` and `ConsumerBU_server`.

5. Create a GitLab access token by navigating to User settings > access tokens and creating a new GitLab access token with the following permissions: api, read_user, and read_repository. Use a bot account or service account that you’d generally use for your webhooks or pipeline integration to create this access token. The account should have a project Maintainer or Owner role or an Administrator role to ensure the Prisma Cloud extension can read the environment variables.

These permissions are necessary to enable the webhook to send necessary data to the Prisma Cloud IaC service to perform the checks against security policies.

**STEP 2 | Set up the failure criteria for merge request checks.**

You can set the environment variable `prisma_cloud_scm_failure_mr_criteria` to define the number and severity of security policy check failures that need to occur to trigger a merge request failure. The syntax for the `prisma_cloud_scm_failure_mr_criteria` value is as follows.

```
High: x, Medium: y, Low: z, Operator: op
```

In the syntax above, x is a count of high-severity policy check failure, y is a count of medium-severity policy check failures, and z is a count of low-severity policy check failures. The Operator value determines what combination of High/Medium/Low counts should result in a merge request failure. The default for each count is 0. The value for Operator, op, can be either OR or AND. The default is OR. Some examples of settings for `prisma_cloud_scm_failure_mr_criteria` are as follows.

- The setting below would result in a failed merge request security check for any detected policy check failure

  ```
  High: 0, Medium: 0, Low: 0, Operator: OR
  ```

- The setting below would result in merge requests never failing a security check.

  ```
  High: 1000, Medium: 1000, Low: 1000, Operator: AND
  ```

**STEP 3 | Set up the failure criteria for GitLab issue creation.**

You can set the environment variable `prismacloud.scmscmm_failure_issue_criteria` to define the number and severity of security policy check failures that need to occur to trigger creation of a GitLab issue, during a merge request. The syntax of the variable value is the same as that for `prismacloud_scm_failure_mr_criteria`. The value includes High, Medium, and Low counts and includes an Operator whose possible values are AND and OR.
STEP 4 | Set up the Prisma Cloud tags.

Prisma Cloud tags are different from GitLab tags or cloud tags that you might have included within your IaC templates. Prisma Cloud tags enable visibility on the Prisma Cloud administrator console.

Provide the value for this environment variable as a comma-separated list of tags that you define. An example is: `prisma_cloud_scm_tags=project x, owner=mr.y, compliance=pci`.

STEP 5 | Set up a webhook to perform the IaC scan during merge request operations.

1. Navigate to Project > Settings > Webhooks

   ![Webhooks Settings](image)

   - **URL**: `http://example.com/trigger-ci.json`
   - **Secret Token**: Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.
   - **Trigger**: Select **Merge request events** as the trigger.


   This URL is the Prisma Cloud SaaS API that supports IaC scanning for GitLab.
3. Provide the GitLab access token that you generated earlier in the Secret Token field.
4. Select **Merge request events** as the trigger.
5. Select **Enable SSL verification**.

---

446 PRISMA™ CLOUD ADMINISTRATOR’S GUIDE | Prisma Cloud DevOps Security

© 2020 Palo Alto Networks, Inc.
6. Select the Add webhook button to add the webhook you just configured.

Your newly added webhook should appear under the Project Hooks list on the same page.

STEP 6 | Set Up Your Prisma Cloud Configuration File for IaC Scan file.

Create the `.prismaCloud/config.yml` and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.

Run an IaC Scan in a Merge Request

When you create, update, or reopen a merge request with added or modified files, this set up will trigger a merge request event to invoke a Prisma Cloud IaC scan for all files in the merge request. The scan does not include deleted files.

You can see the results of the IaC scan through a comment on the merge request. If the scan results meets or exceeds the failure criteria set in the environment variable `prisma_cloud_scm_failure_mr_criteria`, then the results will show that the security check failed.

The following shows the result of an IaC scan for a merge request. In this example, the IaC scan resulted in some security policy check failures. Since the number and severity of the failures did not meet the failure criteria set in the environment variable `prisma_cloud_scm_failure_mr_criteria`, the security check passed, and the merge request succeeded.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Name</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium</td>
<td>AWS S3 Object Versioning is disabled</td>
<td><code>test.tf</code></td>
<td>This policy identifies the S3 buckets which have Object Versioning disabled. S3 Object Versioning is an important capability in protecting your data within a bucket. Once you enable Object Versioning, you cannot remove it; you can suspend Object Versioning at any time on a bucket if you do not wish for it to persist. It is recommended to enable Object Versioning on S3.</td>
</tr>
<tr>
<td>low</td>
<td>AWS S3 CloudTrail buckets for which access logging is disabled</td>
<td><code>test.tf</code></td>
<td>This policy identifies S3 CloudTrail buckets for which access is disabled. S3 Bucket access logging generates access records for each request made to your S3 bucket. An access log record contains information such as the request type, the resources specified in the request worked, and the time and date the request was processed. It is recommended that bucket access logging be enabled on the CloudTrail S3 bucket.</td>
</tr>
</tbody>
</table>

The following is an example of output that occurs when the failure criteria in the environment variable `prisma_cloud_scm_failure_issue_criteria` is met or exceeded.
Prisma Cloud Security Issues (1) for Merge Request: !12

There are some policies matched

**Related merge requests**

- demo #1

---

**Discussion**

- **Prisma Scan** label 16 hours ago

Prisma Cloud IaC scan has found 1 security issues with High: 0, Medium: 1, Low: 0 severity. Your merge request has failed because it meets or exceeds the failure criteria High: 0, Medium: 0, Low: 0, Operator: or

<table>
<thead>
<tr>
<th>Severity</th>
<th>Name</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium</td>
<td>AWS S3 Object Versioning is disabled</td>
<td>&quot;for-</td>
<td>This policy identifies the S3 buckets which have Object Versioning disabled. S3 Object Versioning is an important capability in protecting your data within a bucket. Once you enable Object Versioning, you cannot remove it; you can suspend Object Versioning at any time on a bucket if you do not wish for it to persist. It is recommended to enable Object Versioning on S3.</td>
</tr>
</tbody>
</table>
Use the Prisma Cloud Extension for Visual Studio Code

With the Prisma Cloud Enterprise edition license, you can install the Prisma Cloud extension for Visual Studio (VS) Code to detect issues in your Infrastructure-as-Code (IaC) templates and deployment files against Prisma Cloud security policies early in the software development process, directly within your VS Code editor. The following steps show how simple it is to install and check your templates and files for potential security misconfigurations.

1. Install Prisma Cloud Extension for Visual Studio Code
2. Configure the Prisma Cloud Extension for VS Code
3. Scan Using the Prisma Cloud VS Code Extension

Install Prisma Cloud Extension for Visual Studio Code

The Prisma Cloud extension supports VS Code version 1.36.0 and later.

STEP 1 | In VS Code, navigate to Extensions.

STEP 2 | Enter Prisma Cloud in search.

STEP 3 | Install the extension.

Configure the Prisma Cloud Extension for VS Code

Before you can use the Prisma Cloud extension for VS Code, you'll need to configure the extension to include your API access key, secret key, and Prisma Cloud API URL. If your access keys change, you must update the details in the extension settings.
STEP 1 | In VS Code, navigate to Settings > Extensions > Prisma Cloud.

STEP 2 | Enter the following information for the Prisma Cloud extension:

- **Prisma Cloud API URL.**
  
  The URL for Prisma Cloud varies depending on the region and cluster on which your tenant is deployed. The tenant provisioned for you is, for example, https://app2.prismacloud.io or https://app.eu.prismacloud.io. Replace `app` in the URL with `api` and enter it here. Refer to the Prisma Cloud REST API Reference, which is accessible from the Help Center within the Prisma Cloud web interface for more details.

- **Access Key.**
  
  The Prisma Cloud access key enables programmatic access. If you do not have a key, you must Create and Manage Access Keys.

- **Secret Key.**
  
  You should have saved this key when you generated your Prisma Cloud access key and corresponding secret key. You cannot view the secret key on the Prisma Cloud web interface.

- **Asset Name**
  
  Give your VSCode instance an asset name. You can choose an arbitrary name. Prisma Cloud uses the asset name to track results. Some examples of names are appteam_vscode or johndoe_vscode.

- **Prisma Cloud Tags**
  
  Prisma Cloud tags are different from cloud tags that you may have included within your IaC templates. Prisma Cloud tags enable visibility in the Prisma Cloud administrator console.

  Provide the values as a comma-separated list of tags in the Prisma Cloud Tags field. An example list is: `owner:johndoe, team:creditapp, env:dev`.

STEP 3 | Set Up Your Prisma Cloud Configuration File for IaC Scan

Create the `.prismaCloud/config.yml` file and add it to the root directory of your repository branch. The file is required, and it must include the template type, version, and the template specific parameters and tags you use in your environment.
Scan Using the Prisma Cloud VS Code Extension

Now, you are ready to scan your templates and view the results within the VS Code editor.

**STEP 1 | Scan a file.**

Right-click on your template file in the VS Code Explorer and select **Prisma Scan** to check your template against Prisma Cloud IaC policies.

**STEP 2 | View the scan results.**

Select the **Prisma Cloud** icon on the Activity Bar.
The results of the check will appear in the **Prisma Cloud Result** window. If the extension discovers any policy violations, the **Prisma Cloud Result** window sorts the results by severity and displays the following details for each violation:

- Name of the violated policy
- Severity of the violation
- Names of the module or files that have issues
- Timestamp of the scan

When you scan a different template, the result window refreshes to display the latest scan results.
Use the Prisma Cloud IaC Scan REST API

Prisma Cloud makes the IaC scanning functionality available as a SaaS solution through a REST API. The Prisma Cloud IaC scan service supports Terraform templates, CloudFormation templates, and Kubernetes app manifests. While it is recommended that you take advantage of the IaC scan plugins that are available, there are situations where you might want to use the IaC scan REST API directly.

The following support exists.

- IaC Scan API Version 2
- REST API Request to Scan Terraform Files (Deprecated)
  - Terraform 0.12
  - Terraform 0.11
- REST API Request to Scan AWS CloudFormation Templates (Deprecated)
- REST API Request to Scan Kubernetes Templates (Deprecated)
- Legacy Prisma Cloud IaC Scan REST API (Deprecated)

REST API Authentication

The Prisma Cloud IaC scan REST API uses authentication based on JSON web tokens (JWT). To gain secure access to the API, you need to start with a Prisma Cloud access key, which your Prisma Cloud administrator normally assigns. You can use the access key to obtain a JWT through a Prisma Cloud REST API request for authentication. You will, in turn, enter the JWT in the header of your IaC scan REST API request. See Access the Prisma Cloud REST API for details.

REST API Base URL

All IaC scan REST API request paths are relative to a base URL. Your IaC scan REST API base URL depends on the region and cluster of your Prisma Cloud tenant. For example, if your Prisma Cloud admin console URL is https://app.prismacloud.io, then your Prisma Cloud API base URL is https://api.prismacloud.io. See the Prisma Cloud REST API Reference for a list of Prisma Cloud API URLs.

IaC Scan API Version 2

With version 2 of the IaC scan API, you can initiate IaC scans asynchronously and integrate your IaC scan results with Prisma Cloud.

The following steps enable you to:

- Create an IaC scan asset in Prisma Cloud.
- Use a presigned URL from the scan asset creation to upload the templates to be scanned.
- Start an asynchronous job to perform a scan of your uploaded templates.
- Query your job status.
- Request the results of the scan once the job is completed.

The following table shows the required request-header fields for all the version 2 IaC scan APIs.

<table>
<thead>
<tr>
<th>Request-Header Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
</tr>
</tbody>
</table>
Version 2 of the IaC scan API is JSON API compliant.

As with other Prisma Cloud public API requests, the version 2 IaC scan API requests return standard HTTP response codes. In addition, in an error response, the version 2 IaC scan API requests return details about the error in an array of objects in the response object. The following table lists the properties of a single object in the array.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>errors</td>
<td>array of objects</td>
<td>List of error details</td>
</tr>
<tr>
<td>errors.status</td>
<td>string</td>
<td>HTTP status code</td>
</tr>
<tr>
<td>errors.code</td>
<td>string</td>
<td>Application specific error code</td>
</tr>
<tr>
<td>errors.detail</td>
<td>string</td>
<td>Detailed description of the error</td>
</tr>
<tr>
<td>errors.source</td>
<td>string</td>
<td>Indicates the part of the request document that caused the error</td>
</tr>
</tbody>
</table>

The following is an example of an error response object.

```json
{
  "errors": [
    {
      "status": "405",
      "code": "405",
      "detail": "Invalid template file provided",
      "source": "Scan-worker-service"
    }
  ]
}
```

The steps below illustrate the series of API requests needed to run and manage an asynchronous IaC scan job in Prisma Cloud.

**STEP 1 |** Create an IaC scan asset in Prisma Cloud.

The following API enables you to create an IaC scan asset. This asset provides a mechanism to scan your templates and retrieve the scan results asynchronously.

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/v2/scans</td>
</tr>
</tbody>
</table>
The request body parameters both define failure criteria for an IaC scan job and provide metadata about your scan. The details about these parameters are shown in the table below and in the Prisma Cloud REST API Reference.

<table>
<thead>
<tr>
<th>Request Body Parameter Key</th>
<th>Type</th>
<th>Description</th>
<th>Default (* = required)</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>object</td>
<td>Top-level data key in compliance with JSON:API</td>
<td>*</td>
</tr>
<tr>
<td>data.type</td>
<td>string</td>
<td>Any preferred value. For reference only</td>
<td>**</td>
</tr>
<tr>
<td>data.attributes.assetName</td>
<td>string</td>
<td>Asset name that will appear in Prisma Cloud asset inventory</td>
<td>*</td>
</tr>
<tr>
<td>data.attributes.assetType</td>
<td>string</td>
<td>Asset type. Supported types:</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• AzureDevOps for Azure DevOps Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• AWSCodePipeline for AWS CodePipeline</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CircleCI for CircleCI project</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GitHub for GitHub repo</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GitLab for GitLab repo</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GitLab-SCM for GitLab pipeline</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IntelliJ for IntelliJ IDE plugin-managed files</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VSCode for VSCode plugin-managed files</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• twistcli for Twistlock CLI attachment</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• iacAPI for direct IAC API attachment</td>
<td></td>
</tr>
<tr>
<td>data.attributes.tags</td>
<td>object</td>
<td>key/value pairs that define tags for the asset or scan job</td>
<td>empty</td>
</tr>
<tr>
<td>data.attributes.scanAttributes</td>
<td>object</td>
<td>key/value pairs that define additional attributes for the scan job, See Prisma Cloud REST API Reference for suggested keys</td>
<td>empty</td>
</tr>
<tr>
<td>data.attributes.failureCriteria</td>
<td>object</td>
<td>key/value pairs that define thresholds for failure asset result after evaluation.</td>
<td>Defaults for high, medium, and low are shown below</td>
</tr>
<tr>
<td>Request Body Parameter Key</td>
<td>Type</td>
<td>Description</td>
<td>Default (* = required)</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>data.attributes.failureCriteria.high</td>
<td>integer</td>
<td>Threshold for the number of high severity violations that define an asset failure</td>
<td>1</td>
</tr>
<tr>
<td>data.attributes.failureCriteria.medium</td>
<td>integer</td>
<td>Threshold for the number of medium severity violations that define an asset failure</td>
<td>1</td>
</tr>
<tr>
<td>data.attributes.failureCriteria.low</td>
<td>integer</td>
<td>Threshold for the number of low severity violations that define an asset failure</td>
<td>1</td>
</tr>
</tbody>
</table>

The following is an example of a cURL request to create an IaC scan asset.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/v2/scans' 
--header 'x-redlock-auth: <JWT Token> 
--header 'Content-Type: application/vnd.api+json' 
--data-raw '{
  "data": {
    "type": "my-scan1",
    "attributes": {
      "assetName": "my-asset1",
      "assetType": "AzureDevOps",
      "tags": {
        "env": "dev",
        "region": "us-west-1"
      },
      "scanAttributes": {
        "buildNumber": 999,
        "projectName": "my-project",
        "prName": "SL-1234",
        "pipelineName": "PrismaCloudScan-Lamb",
        "pipelineLambda": "AWSCodePipeline-dm",
        "pipelineStageName": "SourceArtifact",
        "pipelineActionName": "SourceArtifact"
      },
      "failureCriteria": {
        "high": 1,
        "medium": 10,
        "low": 30,
        "operator": "or"
      }
    }
  }
}'
```

The following example shows the response of a successful request. In this example, the data.id identifies a scan id that you can use in subsequent requests to manage your scan job. The data.url value is a presigned URL you can use to upload the files you want scanned to Prisma Cloud.

```json
{
}
```
STEP 2 | Use the presigned URL from the scan asset creation to upload the templates to be scanned.

Prisma Cloud uses a presigned URL gives you temporary access to upload an object. There are multiple ways to upload your files, but the following is an example of a cURL request to upload a file to the presigned URL.

```
curl -X PUT '<presigned URL> --form 'file=@<path and file name of file to be uploaded>'
```

STEP 3 | Start a job to perform a scan of your uploaded templates.

The following API enables you start an asynchronous job to perform a scan of your uploaded file. The path parameter is the scan ID from the response of your earlier request to create a scan asset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/v2/scans/{scanID}</td>
</tr>
</tbody>
</table>

The request body parameters provide metadata necessary for the scan itself. The details about these parameters in the table below and in the Prisma Cloud REST API Reference.

<table>
<thead>
<tr>
<th>Request Body Parameter Key</th>
<th>Type</th>
<th>Description</th>
<th>Default (* = required)</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>object</td>
<td>Top-level data key in compliance with JSON:API</td>
<td>*</td>
</tr>
<tr>
<td>data.id</td>
<td>string</td>
<td>Scan UUID</td>
<td>The scan ID from the API URL</td>
</tr>
<tr>
<td>data.attributes.templateType</td>
<td>string</td>
<td>IaC template type. Valid values are: cft, k8s, tf</td>
<td>*</td>
</tr>
<tr>
<td>data.attributes.templateVersion</td>
<td>string</td>
<td>Template version</td>
<td>For Terraform, IaC scan auto detects version from</td>
</tr>
</tbody>
</table>
The following is an example of a cURL request to start a scan job.

curl -X POST 'https://<Prisma Cloud API URL>/iac/v2/scans/[scanID]' \
--header 'x-redlock-auth: <JWT Token>' \
--header 'Content-Type: application/vnd.api+json' \
--data-raw '{
  "data": {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "attributes": {
      "templateType": "cft",
      "templateVersion": "0.13",
      "templateParameters": {
        "variables": {
          "region": "us-east-1",
          "image_id": "amzn2-ami-hvm-2.0"
        },
        "variableFiles": [
          ".dev.tfvars",
          ".us/qa.tfvars"
        ]
      }
    }
  }
}'}
"policyIdFilters": [
  "123e4567-e89b-12d3-a456-426614174000"
],
"files": [
  ".//dev/auto_scale.tf",
  ".//dev/app.tf"
],
"folders": [
  ".//dev",
  ".//under_develop",
  ".//modules/dev"
]
}
}
}
}

A successful request will return an HTTP code of 200.

**STEP 4 | Query your job status.**

The following API enables you to query the status of your asynchronous IaC scan job.

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/v2/{scanID}/status</td>
</tr>
</tbody>
</table>

The `scanID` path parameter is the id you received in the response object from the request to create a scan asset and which you also used as a path parameter to start the scan job. The following is an example cURL request to request the job status.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/v2/scans/[scanID]/status' \
   --header 'x-redlock-auth: <JWT Token> \
   --header 'Content-Type: application/vnd.api+json'
```

The following example shows the response of a successful request.

```
{
  "data": {
    "id": 12345678-5717-4562-b3fc-2c963f66afa6",
    "attributes": {
      "status": "processing"
    }
  }
}
```

**STEP 5 | Request IaC scan results.**

The following API enables you to request your IaC scan results after the scan job is done.

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/v2/scans/[scanID]/results</td>
</tr>
</tbody>
</table>
The \textit{scanID} path parameter is the id you received in the response object from the request to create a scan asset and which you also used as a path parameter to start the scan job. The following is an example cURL request to request the job status.

\begin{verbatim}
curl -X POST 'https://<Prisma Cloud API URL>/iac/v2/scans/\{scanID\}/results' \
--header 'x-redlock-auth: <JWT Token> \
--header 'Content-Type: application/vnd.api+json'
\end{verbatim}

The following example shows the response of a successful request.

\begin{verbatim}
{
  "meta": {
    "matchedPoliciesSummary": {
      "high": 0,
      "medium": 0,
      "low": 0
    },
    "errorDetails": [
      {
        "status": "string",
        "code": "405",
        "detail": "Invalid template file provided",
        "source": "Scan-worker-service"
      }
    ]
  },
  "data": [
    {
      "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "attributes": {
        "severity": "high",
        "name": "string",
        "rule": "string",
        "desc": "string",
        "files": [
          "string"
        ],
        "policyId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
      }
    }
  ]
}
\end{verbatim}

**REST API Request to Scan Terraform Files (Deprecated)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/tf/v1/scan</td>
</tr>
</tbody>
</table>

This REST API request scans a Terraform file or a zip archive that contains multiple Terraform files for comparison against Prisma Cloud security policies. The body of the API request contains the file or zip...
The module you scan can have either Terraform 0.12 or prior version templates. The request-header fields differ, depending on the type of Terraform module you want to scan.

*The IaC scan service cannot scan any files that are not valid Terraform files, either in the incorrect or invalid .tf format.*

**Terraform 0.12**

The following table shows request-header fields required to request a scan of Terraform 0.12 modules.

<table>
<thead>
<tr>
<th>Request-Header Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
<td>Required</td>
</tr>
<tr>
<td>content-type</td>
<td>To scan single files: text/plain. To scan zip archives: multipart/form-data</td>
<td>Required</td>
</tr>
<tr>
<td>terraform-version</td>
<td>0.12</td>
<td>Required</td>
</tr>
<tr>
<td>terraform-012-parameters</td>
<td>An array of key/value pairs that describe the variables in your module. See details below.</td>
<td>Required</td>
</tr>
</tbody>
</table>

The value of `terraform-012-parameters` differs, depending on whether your Terraform 0.12 module has (1) standard variables or (2) custom variable file names and/or external variables.

- If the Terraform module has variable files but no external variables, then the array elements that make up the value of `terraform-012-parameters` is as follows.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>root-module</td>
<td>Terraform 0.12 root module</td>
</tr>
</tbody>
</table>

The following example shows a cURL request to scan a Terraform 0.12 module that has standard variables.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/tf/v1/scan' \
--header 'x-redlock-auth: '<JWT token>' \
--header 'Content-Type: multipart/form-data' \
--header 'terraform-version: '0.12'' \
--header 'terraform-012-parameters: ["root-module":"/scan/rich-value-types"],{"root-module":"/scan/rich-value-types/network/"}']' \
--form 'templateFile=@<path and file name of single Terraform file or zip archive>''
```

The following example shows the response of a successful request.

```json
{
    "result": {
        "is_successful": true,
```
"rules_matched": [
  {
    "severity": "medium",
    "name": "AWS S3 Object Versioning is disabled",
    "rule": " $.resource[*].aws_s3_bucket exists and ($.resource[*].aws_s3_bucket.*[*].*.versioning[*].enabled does not exist or $.resource[*].aws_s3_bucket.*[*].*.versioning[*].enabled anyFalse)",
    "description": "This policy identifies the S3 buckets which have Object Versioning disabled. S3 Object Versioning is an important capability in protecting your data within a bucket. Once you enable Object Versioning, you cannot remove it; you can suspend Object Versioning at any time on a bucket if you do not wish for it to persist. It is recommended to enable Object Versioning on S3."
  },

  "files": [
    "/scan/for-expressions"
  ],

  "id": "89ea62c1-3845-4134-b337-cc82203b8ff9"
}
],

"severity_stats": {
  "high": 0,
  "low": 0,
  "medium": 1
}
}

You can set the value of **terraform-012-parameters** to enable a scan of Terraform variable files with custom names or Terraform external variables. If your Terraform module has either of these variable uses, then the value of **terraform-012-parameters** is an array of key/value pairs where the key/value pairs can be one or more of the following.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>root-module</td>
<td>Terraform 0.12 root module</td>
</tr>
<tr>
<td>variable-files</td>
<td>An array of custom variable file names. The path of each file is relative to your root module.</td>
</tr>
<tr>
<td>variables</td>
<td>An array of key/value pairs. Each array element has a name and value that together identify an input variable (e.g. [{&quot;name&quot;:&quot;varName1&quot;,&quot;value&quot;:&quot;varValue1&quot;},{&quot;name&quot;:&quot;varName2&quot;,&quot;value&quot;:&quot;varValue2&quot;}])</td>
</tr>
</tbody>
</table>

The following example shows a cURL request to scan a Terraform 0.12 zip archive that has custom variable file names.

curl -X POST 'https://<Prisma Cloud API URL>/iac/tf/v1/scan' \
    --header 'x-redlock-auth: <JWT token>' \
    --header 'Content-Type: multipart/form-data' \
    --header 'terraform-version: 0.12' \
    --header 'terraform-012-parameters: ["root-module": "/scan/rich-value-types/"],"root-module": "/scan/rich-value-types/network/", "variable-
The following example shows a cURL request to scan a Terraform 0.12 zip archive that has external variables.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/tf/v1/scan' \
--header 'x-redlock-auth: <JWT token>' \
--header 'Content-Type: multipart/form-data' \
--header 'terraform-version: 0.12' \
--header 'terraform-012-parameters: ["root-module": "/", "variables": 
  [ { "name": "region", "value": "us-west-1" }, { "name": "bucket",  
    "value": "testbucket" } ] ]' \
--form 'templateFile=@<absolute file path of template or zip>'
```

- The following example shows a cURL request to scan a Terraform 0.12 plan file that is in JSON format.

```bash
curl --location --request POST 'https://api.prismacloud.io/iac/tf/v1/scan' \
--header 'x-redlock-auth: '<JWT token>' \
--form 'templateFile=@<absolute file path of plan JSON file>'
```

Note that, as the request above shows, the only required header is `x-redlock-auth`. The following is an example of successful response to this request.

```json
{
  "result": {
    "is_successful": true,
    "rules_matched": [
    {
      "severity": "medium",
      "name": "AWS S3 Object Versioning is disabled",
      "rule": "$.resource[*].aws_s3_bucket exists and
      ($.resource[*].aws_s3_bucket.*[].*.versioning[*].enabled does not exist
      or $.resource[*].aws_s3_bucket.*[].*.versioning[*].enabled anyFalse)",
      "description": "This policy identifies the S3 buckets
      which have Object Versioning disabled. S3 Object Versioning is an
      important capability in protecting your data within a bucket. Once you
      enable Object Versioning, you cannot remove it; you can suspend Object
      Versioning at any time on a bucket if you do not wish for it to persist.
      It is recommended to enable Object Versioning on S3.",
      "id": "89ea62c1-3845-4134-b337-cc82203b8ff9"
    }
    ],
    "severity_stats": {
      "high": 0,
      "low": 0,
      "medium": 1
    }
  },
  "response_id": "35760530-70d3-4652-b4d2-2a06a9eb776e"
}
```
### Terraform 0.11

The following table shows request-header fields required to request a scan of Terraform 0.11 modules that have only standard variables.

<table>
<thead>
<tr>
<th>Request-header Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
<td>Required</td>
</tr>
<tr>
<td>content-type</td>
<td>To scan single files: text/plain. To scan zip archives: multipart/form-data</td>
<td>Required</td>
</tr>
</tbody>
</table>

The following is an example of a cURL request to scan a Terraform 0.11 module that has only standard variable file names.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/tf/v1/scan' \
   --header 'x-redlock-auth: <JWT token>' \
   --header 'Content-Type: multipart/form-data' \
   --form 'templateFile=@<absolute file path of template or zip>'
```

The following table shows request-header fields required to request a scan of Terraform 0.11 modules that have custom variable file names or external variables.

<table>
<thead>
<tr>
<th>Request-header Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
<td>Required</td>
</tr>
<tr>
<td>content-type</td>
<td>To scan single files: text/plain. To scan zip archives: multipart/form-data</td>
<td>Required</td>
</tr>
<tr>
<td>rl-parameters</td>
<td>An array of key/value pairs. Each array element has a name and value that together identify an input variable (e.g. [&quot;name&quot;:&quot;varName1&quot;,&quot;value&quot;:&quot;varValue1&quot;,&quot;name&quot;:&quot;varName2&quot;,&quot;value&quot;:&quot;varValue2&quot;])</td>
<td>Required for input variables</td>
</tr>
<tr>
<td>rl-variable-file-names</td>
<td>An array of variable file names. The path of each file is relative to your repository branch root directory</td>
<td>Required for variable files</td>
</tr>
</tbody>
</table>

The following is an example of a cURL request to scan a Terraform 0.11 module that has custom file names and external variables.

```bash
curl -X POST 'https://<Prisma Cloud API URL>/iac/tf/v1/scan' \
   --header 'x-redlock-auth: <JWT token>' \
   --header 'Content-Type: multipart/form-data' \
```

© 2020 Palo Alto Networks, Inc.
The following is an example of a successful response to the request above.

```json
{
  "result": {
    "is_successful": true,
    "rules_matched": [
      {
        "severity": "high",
        "name": "AWS Security Groups allow internet traffic to SSH port (22)",
        "rule": "$.resource[*].aws_security_group exists and
               ($1).aws_security_group[*].*[*].ingress[?( @.protocol == 'tcp' &&
               @.from_port<23 && @.to_port>21 )].cidr_blocks[*] contains 0.0.0.0/0 or
               ($1).aws_security_group[*].*[*].ingress[?( @.protocol == 'tcp' &&
               @.from_port<23 && @.to_port>21 )].ipv6_cidr_blocks[*] contains ::/0)",
        "description": "This policy identifies AWS Security Groups which do allow inbound traffic on SSH port (22) from public internet. Doing so, may allow a bad actor to brute force their way into the system and potentially get access to the entire network.",
        "files": [
          "demo/securitygroup22.tf"
        ],
        "id": "617b9138-584b-4e8e-ad15-7fbabafbed1a"
      },
      "severity_stats": {
        "high": 0,
        "low": 0,
        "medium": 1
      }
    ],
    "response_id": "bb3ba05a-2e31-4fc3-9a8e-91b31f673500"
  }
}
```

**REST API Request to Scan AWS CloudFormation Templates (Deprecated)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/cft/v1/scan</td>
</tr>
</tbody>
</table>

This REST API request scans AWS CloudFormation template files for comparison against Prisma Cloud security policies. Support exists for both JSON and YAML formats. Prisma Cloud IaC API also supports parameters for CloudFormation templates. You can also scan either a single template or a zip archive of template files with a single API request. Note that scan support does not currently exist for nested references, macros, or intrinsic functions in CloudFormation templates.

The following table shows the request-header fields. The body of the API request contains the file or zip archive to be scanned.
<table>
<thead>
<tr>
<th>Request-header Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
<td>Required</td>
</tr>
<tr>
<td>content-type</td>
<td>To scan single files: <code>text/plain</code>. To scan zip archives: <code>multipart/form-data</code></td>
<td>Required</td>
</tr>
<tr>
<td>rl-parameters</td>
<td>An array of key/value pairs. Each array element has a name and value that together identify a parameter (e.g. <code>[{&quot;name&quot;:&quot;varName1&quot;,&quot;value&quot;:&quot;varValue1&quot;},{&quot;name&quot;:&quot;varName2&quot;,&quot;value&quot;:&quot;varValue2&quot;}]</code>)</td>
<td>Required for parameters</td>
</tr>
</tbody>
</table>

The following example shows a cURL request to scan an AWS CloudFormation template with external variables.

```
curl -X POST 'https://<Prisma Cloud API URL>/iac/cft/v1/scan' \  
  --header 'x-redlock-auth: <JWT token>' \  
  --header 'Content-Type: multipart/form-data' \  
  --header 'rl-parameters: "[{"name":"varName1","value":"varValue1"}, {"name":"varName2","value":"varValue2"}]"' \  
  --form 'templateFile=@<absolute file path of template or zip>'
```

The following is an example of a successful response to this request.

```
{
  "result": {
    "is_successful": true,
    "rules_matched": [
    {
      "severity": "high",
      "name": "AWS Security Groups allow internet traffic to SSH port (22)",
      "rule": "$resource[\*].aws_security_group exists and ($resource[\*].aws_security_group[\*].\*[\*].ingress[?( @protocol == 'tcp' & @from_port<23 & @to_port>21 )].cidr_blocks[\*] contains 0.0.0.0/0 or $resource[\*].aws_security_group[\*].\*[\*].ingress[?( @protocol == 'tcp' & @from_port<23 & @to_port>21 )].ipv6_cidr_blocks[\*] contains ::/0)",
      "description": "This policy identifies AWS Security Groups which do allow inbound traffic on SSH port (22) from public internet. Doing so, may allow a bad actor to brute force their way into the system and potentially get access to the entire network.",
      "files": ["cftdemo/cft_sg.json"],
      "id": "617b9138-584b-4e8e-ad15-7fbabafbed1a"
    },
    "severity_stats": {
      "high": 0,
      "low": 0,
      "medium": 1
    }
  }
}
```
REST API Request to Scan Kubernetes Templates (Deprecation)

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac/k8s/v1/scan</td>
</tr>
</tbody>
</table>

This REST API request scans Kubernetes manifests to compare against Prisma Cloud security policies, including manifests that you generate from Helm charts. You can scan either a single manifest or a zip archive of manifest files with a single API request.

The following table shows the request-header fields. The body of the API request contains the file or zip archive to be scanned.

<table>
<thead>
<tr>
<th>Request-header Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-redlock-auth</td>
<td>Your JWT token</td>
<td>Required</td>
</tr>
<tr>
<td>content-type</td>
<td>To scan single files: text/plain. To scan zip archives: multipart/form-data</td>
<td>Required</td>
</tr>
</tbody>
</table>

The following example shows a cURL request to scan a single Kubernetes manifest file.

```
curl --location --request POST 'https://<Prisma Cloud API URL>/iac/k8s/v1/scan' \
    --header 'x-redlock-auth: <JWT token>' \
    --header 'Content-Type: multipart/form-data' \
    --form 'templateFile=@<absolute file path of template or zip>'
```

The following is an example of a successful response object.

```
{
    "result": {
        "is_successful": true,
        "rules_matched": [
            {
                "severity": "high",
                "name": "All capabilities should be dropped",
                "rule": "$\.spec.template.spec.containers[*].securityContext.capabilities.drop exists and not $\.spec.templates.spec.containers[*].securityContext.capabilities.drop[*] contains ALL",
                "description": "Ensure that all capabilities are dropped.",
                "id": "4682a6f1-2a1b-4f5a-938c-cdd3fa421a63"
            },
            {
                "severity": "medium",
                "name": "All capabilities should be dropped",
                "rule": "\$\.spec.template.spec.containers[*].securityContext.capabilities.drop exists and not $\.spec.templates.spec.containers[*].securityContext.capabilities.drop[*] contains ALL",
                "description": "Ensure that all capabilities are dropped."
            }
        ]
    }
}
```
Legacy Prisma Cloud IaC Scan REST API (Deprecated)

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>https://&lt;Prisma Cloud API base URL&gt;/iac_scan</td>
</tr>
</tbody>
</table>

This REST API offers the ability to scan IaC templates and test them against Prisma Cloud IaC policies.

*This API is deprecated. All the functionality this API offers is available through the other IaC scan REST APIs listed in this chapter.*

You can submit a template for scanning either by (1) copying the template content directly into the REST API body parameter or (2) specifying a template file. You can scan templates that are in either JSON or YAML format. You can also scan a zip archive that contains Terraform templates.

**Specify Template Content in the Body Parameter**

If you want to scan a single template for vulnerabilities, you can specify the template content in the body parameter of your REST API request. The following example shows how to use cURL to scan a single template by submitting the template content (*CFT.json* in this example) in the body parameter of your API request. Note that the request requires a header parameter `Content-Type: text/plain`.

```
curl --data-binary @CFT.json -X POST https://<Prisma Cloud API>/iac_scan \
-H 'Content-Type: text/plain' \
-H 'x-redlock-auth: <JWT>'
```

**Scan a Template by Uploading the Template File**

The following example shows how to scan a single template by uploading the template file through the REST API request.
Note that the request requires a header parameter Content-Type: text/plain.

curl -X POST https://<Prisma Cloud API URL>/iac_scan \  
-H 'Content-Type: text/plain' \ 
-H 'x-redlock-auth: <JWT>' \ 
-F templateFile=@/Users/ab/CFT.json

You can scan multiple templates that are in a zip archive, such as a zip file containing compressed Terraform files. The following example shows how to upload a zip archive to scan all the templates in that archiver for vulnerabilities.

curl -X POST https://<Prisma Cloud API>/iac_scan \ 
-H 'Content-Type: multipart/form-data' \ 
-H 'x-redlock-auth: <JWT>' \ 
-F templateFile=@/Users/ab/Templates.zip

If you scan a zip archive that contains both compressed Terraform files and custom variable files, you can specify the Terraform custom variable files that are in the zip archive to ensure these files are included in the scan for vulnerabilities. The following is an example of a cURL call that identifies a Terraform custom variable file vars.tf.json in the zip archive Terraform.zip. Standard variable files in the zip archive, like variable.tf, will be scanned for vulnerabilities without being listed in rl-variable-file-name.

curl -X POST https://<Prisma Cloud API>/iac_scan \ 
-H 'rl-variable-file-name:["vars.tf.json"]' \ 
-H 'Content-Type: multipart/form-data' \ 
-H 'x-redlock-auth: <JWT>' \ 
-F templateFile=@/Users/ab/Terraform.zip

See the Response Schema

The following example show the format of the response for a successful request that discovered one or more vulnerabilities.

```
{
   "result":{
      "rules_matched": [ 
      {
        "severity": "",
        "name": "",
        "id": ""
      },
      "severity_stats": {
        "high": 0,
        "low": 0,
        "medium": 1
      },
      "is_successful": "true"
    }
}
```
Prisma Cloud Data Security

Prisma Cloud Data Security is Limited GA available to select Prisma Cloud Enterprise customers only. The Data Security capabilities on Prisma Cloud enable you to discover and classify data stored in AWS S3 buckets and protect accidental exposure, misuse, or sharing of sensitive data. To identify and detect confidential and sensitive data, Prisma Cloud Data Security integrates with Palo Alto Networks’ Enterprise DLP service and provides built-in data profiles, which include data patterns that match sensitive information such as PII, health care, financial information and Intellectual Property. In addition to protecting your confidential and sensitive data, your data is also protected against threats—known and unknown (zero-day) malware—using the Palo Alto Networks’ WildFire service.

> What is Included with Prisma Cloud Data Security
> Enable the Prisma Cloud Data Security Module
> Monitor Data Security Scan Results on Prisma Cloud
> Disable Prisma Cloud Data Security and Offboard AWS account
> Guidelines for Optimizing Data Security Cost on Prisma Cloud
What is Included with Prisma Cloud Data Security

- AWS S3 support for USA regions.
- Visibility, exposure, and classification of S3 buckets & objects on the new Data Dashboard, Data Inventory, and Object Explorer.
  - S3 objects in standard storage class only are ingested for scanning.
  - File sizes and scanning:
    - For data classification and malware scanning, the file size must be less than 20 MB.
    - For ML-based classification scanning, the file size must be less than 1MB.
    - For backward scan, each tenant is limited to a daily limit of 300GB or 500,000 files, whichever threshold is met first.
- Malware detection of objects (only Windows executables & Linux binaries).
- Default Data policies to detect public exposure of sensitive information. The data policies, currently five, generate alerts on Prisma Cloud and you can set up notification to external integration channels supported on Prisma Cloud.
- Freemium experience that offers 300GB per tenant, before you are charged for using the Data Security module. When your data exceeds the freemium threshold you use credits from the Prisma Cloud Enterprise Edition license.

In order to maintain predictable performance for all customers of Prisma Cloud Data Security, there is a daily cap of 300GB of scanning per day for each Prisma Cloud tenant who has enabled Data Security. The daily cap will reset at midnight UTC timezone.

- Ability to scan all or selected S3 buckets when you onboard your AWS account(s) on Prisma Cloud. You can choose to enable a forward or backward scan when you add the cloud account.

Prisma Cloud Data Security needs to read objects stored on your AWS S3 buckets for scanning them.

- Integration with Config RQL to show all objects in an S3 bucket, including exposure, Data Profile & malware detection in Resource Explorer.
Enable the Prisma Cloud Data Security Module

Prisma Cloud Data Security requires you to configure an AWS CloudTrail bucket. To save cost, ensure that you follow the instructions to only select **Write** events instead of **Read** and **Write** events.

**STEP 1 |** Log in to the Prisma Cloud administrative console.

**STEP 2 |** Select Subscription to Learn More on Data Security.

**STEP 3 |** Enable Data Security
STEP 4 | Edit An existing account or onboard a new account to get started with scanning the data for a specific AWS account.

- If you select “Onboard a New Account”, click AWS in the new popup screen. Then go to the Configuration > Add a new AWS Account section in this document for instructions on how to onboard a new AWS account and enable Prisma Cloud Data Security for it.

- If you select “Edit An Existing Account”, move to the Configuration > Edit an existing AWS Account section in this document for instructions on how to enable Prisma Cloud Data Security for an already onboarded AWS account.
Add a New AWS Account and Enable Data Security

1. Go to **Settings > Cloud Accounts > Add New.**
2. Select **AWS** and add a **Cloud Account Name.**
3. Select **Data Security.**
   Only Monitor mode is supported.
4. Select **Next.**
5. Create a stack in your AWS account.
6. Select **Create Stack.**

   **NOTE:** Log in to your AWS account in a separate tab. The CloudFormation template defaults to N. Virginia. You must change it to the region where your AWS Account is before you create the stack.
Configure Account

1. Create Stack in your AWS Account.
2. Select Acknowledge that AWS CloudFormation might create IAM resources with custom names and click Create Stack.
3. When status is CREATE_COMPLETE, copy Role ARN from Outputs tab.
4. When status is CREATE_COMPLETE, copy SNS Topic ARN from Outputs tab.

- External ID
  b5f152ca-e9a8-483c-8228-e605a11f02f

- Role ARN
  arn:aws:iam::123456789012:role/bdf

- SNS Topic ARN
  arn:aws:sns:us-east-1:123456789012:DataProtection

Quick create stack

Template

Template URL
https://redlock-public.s3.amazonaws.com/cft/rl-dlp-read-only.template

Stack description
Prisma Cloud IAM Role to set read permissions

Stack name

Stack name
PrismaCloudApp

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.
1. Select **I acknowledge that AWS CloudFormation might create IAM resources with custom names and Create stack**.

   Wait for the CREATE_COMPLETE status.


7. Setup AWS CloudTrail & SNS.
1. Create new CloudTrail or use an existing CloudTrail

2. Select **Write-only** events to save cost.

3. Select **all S3 buckets in your account** or **Add S3 bucket** for only specific buckets. Select **Write** events only.

4. Create New or Use an existing S3 bucket
   Click Advanced
5. Select Send SNS notification for every log file delivery - Yes, and select SNS topic - PrismaCloudSNS. This was created earlier when you created the stack.

6. Click **Create**.
   
   Confirm that the CloudTrail bucket is created.

7. **Create bucket policy** to enable Prisma Cloud to read from your CloudTrail bucket.
8. Click Next.

8. Select the S3 buckets in which you want to scan data.

You can choose to scan All or Custom storage buckets. Scanning all buckets is an expensive operation depending on the number of files in all buckets and scan type.

NOTE: Currently active and old CloudTrail buckets will be skipped because they contain AWS CloudTrail generated logs instead of customer data. Objects containing ELB access logs, VPC flow logs and S3 access logs are also automatically skipped.

- Select whether you want to scan data Forward only or Forward and Backward.

Forward scan is enabled by default and cannot be disabled. Prisma Cloud scans all new files added or edited by a user in the bucket near real time.

When you select Backward scan, Prisma Cloud starts scanning all existing files in the bucket in a batch operation. Depending on number of files in your bucket, backward scan can be expensive.
- Custom buckets will list all buckets in your AWS account.
  Custom option lets you choose individual buckets to scan based on scan type (recommended)
- Choose buckets and select a scan type for each bucket.

9. Click Next and select Account Groups
10. Click Next and wait for a few seconds for the configuration to complete.
Edit an AWS Account Onboarded on Prisma Cloud to Enable Data Security

1. Go to Settings > Cloud Accounts.
2. Select an AWS account that is currently onboarded on Prisma Cloud.
3. Select Data Security and Next.

**NOTE:** In the first release, only Monitor model is offered for Data Security.

4. Update your stack using instructions below
   1. Download CFT [here](#) and login to AWS Console to update stack and go through steps 1 - 9 in the Cloud Onboarding Setup window.
2. Go to AWS Management Console -> Stacks. Select PrismaCloudApp Stack (if you have previously used CFT to deploy PrismaCloudApp Stack) and Click "Update" button. If not, pick the stack you manually created for Prisma Cloud.

3. Click Update

4. Select "Replace current template" and "Upload a template file". Then upload the CFT you downloaded earlier from step a) and click the "Next" button.
5. Copy the "Callback URL" from Prisma Cloud - Cloud Accounts - Configure Account page

6. Paste the "Callback URL" in the SNS Endpoint field in the “Specify stack details” page in the AWS Management Console. Click "Next".

7. Click "Next" to go through the next couple of screens until you get to this page to complete the Update operation. Click "Update stack".
8. The update CFT operation internally will also create a PrismaCloudSNS Topic and will be used to monitor CloudTrail data events.

9. Copy RoleARN from AWS Management Console's Outputs tab in the stack.

1. Paste Role ARN into the Prisma Cloud Cloud - Cloud Accounts - Configure Account page, replacing the previous Role ARN.

1. Copy SNS ARN from AWS Management Console's Outputs tab in the stack.
2. Paste the SNS Topic: ARN into Prisma Cloud - Cloud Accounts - Configure Account page. Click “Next” to continue.

5. Setup AWS CloudTrail & SNS.
   1. Create new CloudTrail or use an existing CloudTrail

   ![CloudTrail Image]

   **The new CloudTrail console experience is now available**

   We’ve redesigned the CloudTrail console to make it easier to use. Try out the new console.

   Trails
   Deliver logs to an Amazon S3 bucket. CloudTrail events can be processed by one trail for free. There is a charge for processing events with additional trails. For more information, see AWS CloudTrail Pricing.

   **Create trail**

<table>
<thead>
<tr>
<th>Trail name</th>
<th>Home region</th>
<th>Multi-region trail</th>
<th>Insights</th>
<th>Organization trail</th>
<th>S3 bucket</th>
<th>Log file prefix</th>
<th>CloudWatch Logs Log group</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org_CloudTrail_01</td>
<td>US West (N. California)</td>
<td>Yes</td>
<td>Disabled</td>
<td>Yes</td>
<td>cloudtrailbucket07aug2019</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   2. Select **Write-only** events to save cost.
3. Select all S3 buckets in your account or Add S3 bucket for only specific buckets. Select Write events only.

4. Create New or Use an existing S3 bucket
   Click Advanced

5. Select Send SNS notification for every log file delivery - Yes, and select SNS topic - PrismaCloudSNS. This was created earlier when you created the stack.
6. Click **Create**.

   Confirm that the CloudTrail bucket is created.

7. **Create bucket policy** to enable Prisma Cloud to read from your CloudTrail bucket.

8. Click **Next**.
6. Select the S3 buckets in which you want to scan data.

You can choose to scan All or Custom storage buckets. Scanning all buckets is an expensive operation depending on the number of files in all buckets and scan type.

**NOTE:** Currently active and old CloudTrail buckets will be skipped because they contain AWS CloudTrail generated logs instead of customer data. Objects containing ELB access logs, VPC flow logs and S3 access logs are also automatically skipped.

- Select whether you want to scan data **Forward** only or **Forward and Backward**.

  Forward scan is enabled by default and cannot be disabled. Prisma Cloud scans all new files added or edited by a user in the bucket near real time.

  When you select Backward scan, Prisma Cloud starts scanning all existing files in the bucket in a batch operation. Depending on number of files in your bucket, backward scan can be expensive.
Custom buckets will list all buckets in your AWS account. Custom option lets you choose individual buckets to scan based on scan type (recommended).

Choose buckets and select a scan type for each bucket.

7. Click Next and select Account Groups.
8. Click Next and wait for a few seconds for the configuration to complete.
Monitor Data Security Scan Results on Prisma Cloud

- Resource Explorer
- Data Inventory
- Data Dashboard
- Object Explorer
- Exposure Evaluation
- Supported File Extensions—Prisma Cloud Data Security
- Supported Data Profiles & Patterns

Use the Data Policies to Scan

Prisma Cloud includes default data policies to help you start scanning.

1. Create a new alert rule or edit an existing rule.
   1. Select the Data policies that you want to scan against S3 buckets.
      - Healthcare information public exposed
      - Intellectual Property public exposed
      - Objects containing Financial Information publicly exposed
      - Objects containing PII data public exposed
      - Objects containing malware

2. Select the notification channels.
   Prisma Cloud Data Security only supports—Amazon SQS, Splunk, and Webhook integration. See Configure External Integrations on Prisma Cloud.

3. Confirm to save the alert rule.

2. Optional Create a custom data policy.
   You must first onboard an account and enable Data Security before you can create a custom data policy.
      Enter a Policy Name, Description, Severity, and Labels for the new policy.
   2. Select a data classification profile.
You can select one of the four data profiles—Financial Information, Healthcare Information, Intellectual Property, PII.

3. Select the File Exposure.

Private, Conditional, or Public. See Exposure Evaluation.

4. Select the file extensions that you want to scan for sensitive information.

For example, txt. If you select Financial Information, Public, and txt, the policy will generate an alert if a publicly exposed .txt file has Financial Information. Do not use a dot before the extension; if you do, an error message displays.

5. Save your changes. If you would like to define a rule to scan for information specific to your requirements, create a custom policy, and add it to an alert rule to generate alerts.


1. Select Alerts > Overview.
2. Filter on Policy Type—Data, to see all alerts related to Data policies.
3. Select an alert to view details on:
   - Click Bucket Name to see bucket information in the Data Inventory.
   - Click Object Name to see object information in Data Inventory, Object Explorer.
   - Click on Alert Rule to see the Alert Rule that generates this particular Alert instance

Data Security settings

1. Click on Settings -> Data
2. See Scan Settings & Data Profiles tabs

**Note:** The Scan Settings tab is not enabled yet in V1. It will be in the future to allow the user to control which file extensions to ignore for scanning and enable/disable Malware scanning.

3. Click on Data Profiles tab to show the list of Data Profiles enabled - Financial Information, Healthcare, Intellectual Property & PII.
4. Click on any of the Profiles to see all the Data Patterns supported.
5. See all the Data Patterns supported for any of the four Data Profiles supported - Financial Information, Healthcare, Intellectual Property & PII.
6. Click on any of the Data Patterns to see additional details.

7. See details of any specific Data Pattern.
Data Dashboard

The new Data Dashboard tab provides complete visibility into your S3 storage. The dashboard widgets below give you insight into how many storage buckets and objects you have, what kind of data is stored in those objects, across which regions, who owns what and what is the exposure of the objects. This tab is available under the Dashboard menu.

1. Total Buckets
   This widget shows the total number of buckets (except empty buckets) discovered in your AWS Account. Buckets are categorized into Private and Public.
   - Private buckets are internal and not publicly accessible.
   - Public Buckets are accessible to everyone. See Exposure Evaluation to learn how exposure is calculated.

   You can click on either the Private or Public circle in the widget to view those buckets in the Data Inventory.

2. Total Objects
   This widget shows the total number of objects discovered in all your S3 storage buckets. Objects are categorized into Public, Sensitive and Malware. Public objects are accessible to everyone. Sensitive objects contain data such as Financial Information, Healthcare, PII and Intellectual Property. Malware objects contain malicious code.

   Click on either the Public, Sensitive of Malware circle in the widget to see those objects in the Data Inventory view.
3. Top Publicly Exposed Objects By Profile

1. This widget shows you top 5 publicly exposed objects with Data Profiles of Financial Information, Healthcare, PII and Intellectual Property. Click on any of the bars in the widget to view those objects in the Data Inventory.

4. Top Object Owners by Exposure

This widget shows you top 5 objects owner with exposure (Public, Private or Conditional). You can click on any of the bars in the widget to view objects in the Data Inventory.
5. Data Alerts by Severity

This widget shows you the breakdown of Data Alerts by Severity (High, Medium, Low). You can click on any particular severity segment in the circle to see those Alerts in the Alerts view. The data for this widget is based on the timestamp of when the alert was generated, while data on other widgets use the objects created/updated timestamp.

6. Top Data Policy Violations

1. This widget shows you the top Data Policy violated by objects in your S3 buckets. The data for this widget is based on the timestamp of when the alert was generated, while data on other widgets use the objects created/updated timestamp. You can click on any bar in the widget to see those Alerts in the Alerts view.
7. Object Profile by Region

1. This chart shows object profiles such as Financial Information, Healthcare, PII and Intellectual Property across AWS Regions. Click on any region in the widget to see those objects in the Data Inventory view.

Data Inventory

The new Data Inventory tab provides detailed insights into your S3 storage objects. The Data Inventory page comprises data cards, objects insights and a detailed inventory view of objects across accounts, account groups, buckets and regions. It also provides a number of filters such as Time Range, Bucket Name, Bucket Exposure, Object Profile, Object Pattern etc. to allow the user to filter the specific buckets/objects they are interested in. This tab is available under the Inventory menu.
1. The Data Inventory page shows 6 data cards on top:
   - **Total Buckets**
     - Total number of buckets discovered in AWS (except empty buckets)
   - **Public Buckets**
     - Total number of buckets identified as public based on exposure
   - **Total Objects**
     - Total number of files discovered in buckets
   - **Public Objects**
     - Total objects with exposure public
   - **Sensitive Objects**
     - Total number of objects containing sensitive data such as Financial Information, Healthcare, PII and Intellectual Property
   - **Malware Object**
     - Total number of objects identified by Wildfire as Malware

1. The Inventory table at the bottom of this page represents a hierarchical view of data grouped by account name, service name and region name. There are 4 views available on this table with the default view under cloud type followed by service name followed by bucket view followed by object view.
   - **View 1 (Cloud View)**
- **View 3 (Bucket View)**

- **View 4 (Object View)**

The Object View above includes the following information

- Object Name
- Name of the file as discovered in the bucket
- Object Exposure
  - Private, public or conditional
- Data Profile
  - Object content is classified under one of the following profiles: Financial Information, Healthcare, PII and Intellectual Property
  - If an object belongs to any of the above categories, it is identified as sensitive data
- Object Patterns
  - Profiles are categorized by one or more patterns; e.g., an object with the PII profile may have patterns like Driver's License #, SSN, etc.
- Malware
  - Yes/No/Unknown
- User
  - Owner of the object
- Bucket Name
  - Name of the bucket that the object belongs to
- Account Name
  - Name of the account that the object belongs to
- Region Name
  - Name of the region that the object belongs to
- Service Name
  - Name of cloud storage service (e.g. S3)
- Last Modified
  - Object creation time or last updated time in S3.

Resource Explorer

The Resource Explorer provides you visibility into each cloud resource monitored by Prisma Cloud. The Resource Explorer for S3 bucket has been extended to show you all objects that belong to a particular bucket. You can now view all objects with object name, object profile, object pattern, malware and exposure for each object. Type the following RQL or config where cloud.service = 'Amazon S3'

's3api-get-bucket-acl'

See Resource Explorer including the new Objects tab.
Object Explorer

The new Object Explorer provides granular visibility into each object. You can view the metadata for each object along with exposure and Data Profile.

1. Select any object in Data Inventory -> Level 4 (Object View) or Alerts -> Overview -> Click on specific Alert -> Expand on an Alert instance to open Object Explorer or click on any object in the Resource Explorer’s Objects tab to access the Object Explorer.

2. Objects that contain malware display a malware icon
Optionally you can download the malware report (currently available only in XML).

3. The top left pane of the object explorer presents object metadata details.
   - Type—The type of file
   - Owner—The owner of the file
   - Created On—Timestamp of when the file was created on AWS
   - Last Updated On—Timestamp of when the file was last updated on AWS
   - Region—The region where the object is stored.
   - Bucket Name—Name of the bucket where the object is stored.
   - Tags—AWS tags that help in identifying the file.
   - URL—AWS URL for accessing the file.

4. The top left pane of the object explorer presents object exposure details.
   - Exposure Type—Exposure derived based on bucket policy, object ACL and bucket ACL. It can be Public, Private or Conditional.
   - Last Matched—Most recent exposure derivation.
   - Object ACL—JSON attribute that defines the object ACL

5. The right side of the page represents Data Profile
   - Object Profile—Object content is classified under one of the following profiles: Financial Information, Healthcare, PII and Intellectual Property.
   - Object Patterns—List the pattern and the profile with which it is associated. For example, a profile can be PII with pattern that matches Driver's License number or SSN.
   - Frequency—Number of times the pattern occurred in the file.
   - Detection Time—Timestamp of when the pattern was detected.

Exposure Evaluation

Prisma Cloud and the Cloud Service Provider both monitor the configuration of a bucket and access to the objects within. Exposure evaluation on Prisma Cloud Data Security determines the access level defined for an object within a bucket, that is who has access to the object, and whether the user(s) can download or exfiltrate content from a bucket. While on AWS, exposure is an attribute of the bucket or object in the Cloud Service Provider's (CSP) Storage service which allows applications and end users to access structured data for their use cases.

Prisma Cloud Data Security categorizes exposure levels as follows:
<table>
<thead>
<tr>
<th>Exposure Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>Is a file/bucket accessible to everyone? If so this may be a potential risk and needs to be reviewed by the Customer.</td>
</tr>
<tr>
<td>Private</td>
<td>The file/bucket is internal and not publicly accessible. This is the safest state to be in for the customer. However there may be legitimate reasons why some files are Public e.g. CDN web templates hosted by a server etc.</td>
</tr>
</tbody>
</table>
| Conditional    | This usually applies to resources that have a Bucket policy attached to it allowing access when some set of conditions are met. These conditions are contextual and cannot be deterministically resolved as they may be specific to the customer's environment. Some examples include:  
  - access to a bucket is only allowed/denied within a time window  
  - access to a bucket is only allowed when there is a match on user principal or a specific set of users access this bucket.  
  - Access to a bucket is allowed/denied if the client request IP matches a certain range mentioned as a whitelist. |

**How is Exposure Evaluated?**

This is again dependent on each service provider but here we will look closer at AWS and S3 service. S3 access evaluation performs a least-privilege based evaluation. This means a default Deny, explicit Deny takes precedence over any Allow.

**Bucket Exposure Evaluation**

- Normalize all controls into a Policy Document (i.e. Bucket ACL, Bucket Policy)
- Evaluate all policy documents normalized above following the steps outlined above in the diagram. The evaluation is checked against a known set of S3 specific API methods/Actions to check for allow and/or deny.

  Supported Bucket Events are:
  - DeleteBucket
  - CreateBucket
• DeleteBucketPolicy
• PutBucketAcl
• PutBucketPolicy

• If the final result comes out to be that the bucket is publicly accessible i.e. either the whitelisted set of actions are allowed for everyone globally then the verdict is presented as Public.
• If the final result is a Deny for the set of known actions against all policy documents for public users - then the verdict is considered Private.
• If any of the policy document contains Conditional tags indicative of access to the resource under specific conditions, the verdict returned is Conditional. Here we expect feedback from the customer to evaluate the risk posture for the bucket.

Object Exposure Evaluation

• The same steps are followed again as bucket exposure influences object exposure. In addition to the normalized bucket ACL and bucket policy we also normalize the object ACL and factor it into the evaluation.

Supported Object Events are:
• DeleteObject (from CLI, limitation of AWS https://forums.aws.amazon.com/thread.jspa?threadID=261594)
• PutObject
• PubObjectAcl
• PutObjectTagging
• DeleteObjectTagging
• All steps for Bucket policy evaluation is followed again to determine the eventual exposure verdict of the file/object.

Supported File Extensions—Prisma Cloud Data Security

Review the file extensions that Prisma Cloud scans on your storage buckets.

<table>
<thead>
<tr>
<th>File extensions supported for Data Profile / Data Patterns</th>
<th>File extensions supported for Malware scanning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.JAR</td>
<td>.exe</td>
</tr>
<tr>
<td>.DOC, .DOCX, .XLS, .XLSX, .PPT, .PPTX</td>
<td>.dll</td>
</tr>
<tr>
<td>.pdf</td>
<td>.so</td>
</tr>
<tr>
<td>.tar, .rar, .zip, .7zip, .gz</td>
<td>.o</td>
</tr>
<tr>
<td>BAT, JS, VBS, PSI, Shell Script, HTA</td>
<td>.a</td>
</tr>
<tr>
<td>.HTML, .XHTML</td>
<td>.elf</td>
</tr>
<tr>
<td>.xml</td>
<td>.pl</td>
</tr>
<tr>
<td>.c, .js, .py, .pyp, .r, .rb, .v, .vhdl, .java, .asm, .ps1, .vb</td>
<td>.ko</td>
</tr>
<tr>
<td>.mbox, .msg, .pst</td>
<td></td>
</tr>
<tr>
<td>.odt, .ods, .odp, .ott, .ots, .otp</td>
<td></td>
</tr>
</tbody>
</table>
### Supported Data Profiles & Patterns

<table>
<thead>
<tr>
<th><strong>Object Profiles</strong></th>
<th><strong>Object Patterns</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PII</td>
<td>Driver License - Austria</td>
</tr>
<tr>
<td></td>
<td>Driver License - Belgium</td>
</tr>
<tr>
<td></td>
<td>Driver License - Bulgaria</td>
</tr>
<tr>
<td></td>
<td>Driver License - Cyprus</td>
</tr>
<tr>
<td></td>
<td>Driver License - CzechRepublic</td>
</tr>
<tr>
<td>Driver License - Germany</td>
<td></td>
</tr>
<tr>
<td>Driver License - Denmark</td>
<td></td>
</tr>
<tr>
<td>Driver License - Estonia</td>
<td></td>
</tr>
<tr>
<td>Driver License - Spain</td>
<td></td>
</tr>
<tr>
<td>Driver License - Finland</td>
<td></td>
</tr>
<tr>
<td>Driver License - France</td>
<td></td>
</tr>
<tr>
<td>Driver License - UK</td>
<td></td>
</tr>
<tr>
<td>Driver License - Greece</td>
<td></td>
</tr>
<tr>
<td>Driver License - Croatia</td>
<td></td>
</tr>
<tr>
<td>Driver License - Hungary</td>
<td></td>
</tr>
<tr>
<td>Driver License - Ireland</td>
<td></td>
</tr>
<tr>
<td>Driver License - Iceland</td>
<td></td>
</tr>
<tr>
<td>Driver License - Italy</td>
<td></td>
</tr>
<tr>
<td>Driver License - Liechtenstein</td>
<td></td>
</tr>
<tr>
<td>Driver License - Lithuania</td>
<td></td>
</tr>
<tr>
<td>Driver License - Luxembourg</td>
<td></td>
</tr>
<tr>
<td>Driver License - Latvia</td>
<td></td>
</tr>
<tr>
<td>Driver License - Malta</td>
<td></td>
</tr>
<tr>
<td>Driver License - Netherlands</td>
<td></td>
</tr>
<tr>
<td>Driver License - Norway</td>
<td></td>
</tr>
<tr>
<td>Driver License - Poland</td>
<td></td>
</tr>
<tr>
<td>Driver License - Portugal</td>
<td></td>
</tr>
<tr>
<td>Driver License - Romania</td>
<td></td>
</tr>
<tr>
<td>Driver License - Sweden</td>
<td></td>
</tr>
<tr>
<td>Driver License - Slovenia</td>
<td></td>
</tr>
<tr>
<td>Driver License - Slovakia</td>
<td></td>
</tr>
<tr>
<td>Driver License - Canada</td>
<td></td>
</tr>
<tr>
<td>Passport - Austria</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Passport - Belgium</td>
<td></td>
</tr>
<tr>
<td>Passport - Bulgaria</td>
<td></td>
</tr>
<tr>
<td>Passport - Cyprus</td>
<td></td>
</tr>
<tr>
<td>Passport - CzechRepublic</td>
<td></td>
</tr>
<tr>
<td>Passport - Germany</td>
<td></td>
</tr>
<tr>
<td>Passport - Denmark</td>
<td></td>
</tr>
<tr>
<td>Passport - Estonia</td>
<td></td>
</tr>
<tr>
<td>Passport - Spain</td>
<td></td>
</tr>
<tr>
<td>Passport - Finland</td>
<td></td>
</tr>
<tr>
<td>Passport - France</td>
<td></td>
</tr>
<tr>
<td>Passport - UK</td>
<td></td>
</tr>
<tr>
<td>Passport - Greece</td>
<td></td>
</tr>
<tr>
<td>Passport - Croatia</td>
<td></td>
</tr>
<tr>
<td>Passport - Hungary</td>
<td></td>
</tr>
<tr>
<td>Passport - Ireland</td>
<td></td>
</tr>
<tr>
<td>Passport - Iceland</td>
<td></td>
</tr>
<tr>
<td>Passport - Italy</td>
<td></td>
</tr>
<tr>
<td>Passport - Liechtenstein</td>
<td></td>
</tr>
<tr>
<td>Passport - Lithuania</td>
<td></td>
</tr>
<tr>
<td>Passport - Luxembourg</td>
<td></td>
</tr>
<tr>
<td>Passport - Latvia</td>
<td></td>
</tr>
<tr>
<td>Passport - Malta</td>
<td></td>
</tr>
<tr>
<td>Passport - Netherlands</td>
<td></td>
</tr>
<tr>
<td>Passport - Norway</td>
<td></td>
</tr>
<tr>
<td>Passport - Poland</td>
<td></td>
</tr>
<tr>
<td>Passport</td>
<td>Country</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Passport - Portugal</td>
<td></td>
</tr>
<tr>
<td>Passport - Romania</td>
<td></td>
</tr>
<tr>
<td>Passport - Sweden</td>
<td></td>
</tr>
<tr>
<td>Passport - Slovenia</td>
<td></td>
</tr>
<tr>
<td>Passport - Slovakia</td>
<td></td>
</tr>
<tr>
<td>Passport - Canada</td>
<td></td>
</tr>
<tr>
<td>Tax Id - US</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Austria</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Belgium</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Bulgaria</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Cyprus</td>
<td></td>
</tr>
<tr>
<td>Tax Id - CzechRepublic</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Germany</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Denmark</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Estonia</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Spain</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Finland</td>
<td></td>
</tr>
<tr>
<td>Tax Id - France</td>
<td></td>
</tr>
<tr>
<td>Tax Id - UK</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Greece</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Croatia</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Hungary</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Ireland</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Iceland</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Italy</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Liechtenstein</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Lithuania</td>
<td></td>
</tr>
<tr>
<td>Tax Id</td>
<td>Location</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>Tax Id - Luxembourg</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Latvia</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Malta</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Netherlands</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Norway</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Poland</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Portugal</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Romania</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Sweden</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Slovenia</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Slovakia</td>
<td></td>
</tr>
<tr>
<td>Tax Id - Australia</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>National Id</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>National Id - Austria - Central Register of Residents</td>
<td></td>
</tr>
<tr>
<td>National Id - Austria Social Security Card (e-card)</td>
<td></td>
</tr>
<tr>
<td>National Id - Belgium - National Registration Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Belgium - Citizen Service Number (BSN)</td>
<td></td>
</tr>
<tr>
<td>National Id - Bulgaria - Uniform Civil Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Cyprus - Identity Card</td>
<td></td>
</tr>
<tr>
<td>National Id - Czech - Birth Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Czech - National eID Card</td>
<td></td>
</tr>
<tr>
<td>National Id - Germany</td>
<td></td>
</tr>
<tr>
<td>National Id - Denmark - CPR Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Estonia - Personal Identification Code</td>
<td></td>
</tr>
<tr>
<td>National Id - Spain - National Identity Document (Documento Nacional de Identidad)</td>
<td></td>
</tr>
<tr>
<td>National Id - Spanish NIE Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Finland - Personal Identity Code</td>
<td></td>
</tr>
<tr>
<td>National Id</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>National Id - UK National Insurance Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Greece</td>
<td></td>
</tr>
<tr>
<td>National Id - Croatia - Personal Identification Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Hungary - Personal Identification Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Ireland - Personal Public Service Number (PPSN)</td>
<td></td>
</tr>
<tr>
<td>National Id - Iceland</td>
<td></td>
</tr>
<tr>
<td>National Id - Italy - Fiscal Code Card (Codice Fiscale)</td>
<td></td>
</tr>
<tr>
<td>National Id - Liechtenstein</td>
<td></td>
</tr>
<tr>
<td>National Id - Lithuania</td>
<td></td>
</tr>
<tr>
<td>National Id - Luxembourg</td>
<td></td>
</tr>
<tr>
<td>National Id - Latvia - Personal Public Service Number (PPSN)</td>
<td></td>
</tr>
<tr>
<td>National Id - Malta</td>
<td></td>
</tr>
<tr>
<td>National Id - Netherlands - Citizen Service Number (BSN)</td>
<td></td>
</tr>
<tr>
<td>National Id - Norway - Identification Number (Fødselsnummer)</td>
<td></td>
</tr>
<tr>
<td>National Id - Poland</td>
<td></td>
</tr>
<tr>
<td>National Id - Portugal</td>
<td></td>
</tr>
<tr>
<td>National Id - Romania - Identity Card (CNP)</td>
<td></td>
</tr>
<tr>
<td>National Id - Sweden - Personal Identity Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Slovenia</td>
<td></td>
</tr>
<tr>
<td>National Id - Slovakia</td>
<td></td>
</tr>
<tr>
<td>National Id - Brazil - CPF</td>
<td></td>
</tr>
<tr>
<td>National Id - Brazil - CNPJ</td>
<td></td>
</tr>
<tr>
<td>National Id - Japan Corporate Number</td>
<td></td>
</tr>
<tr>
<td>National Id - Japan My Number</td>
<td></td>
</tr>
<tr>
<td>National Id - France - INSEE</td>
<td></td>
</tr>
<tr>
<td>National Id - France - Social Security Number (NIR)</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>National Id - Canada Social Insurance Number</td>
<td></td>
</tr>
<tr>
<td>National Id - US Social Security Number</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Malware</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

| Financial info                                  |
| Bank - American Bankers Association Routing Number |
| Bank - Bankruptcy Filings                       |
| Bank - International Bank Account Number         |
| Bank - Statements                                |
| Committee on Uniform Securities Identification Procedures number |
| Credit card number [Supports Strict Check]       |
| Financial - Financial Accounting                 |
| Financial - Personal Finance                     |
| Financial - Invoice                              |
| Financial - Others                               |
| Magnetic Stripe Information                      |

| Healthcare                                       |
| Health - CLIA                                   |
| Health - DEA                                    |
| Health - Document Others                        |

<p>| Intellectual Property                           |
| Secret Info - AWS Access Key ID                 |
| Secret Info - AWS Secret Access Key             |
| Secret Info - RSA Private Key                   |
| Company Confidential                             |
| Source Code - Cfamily                           |
| Source Code - java                              |</p>
<table>
<thead>
<tr>
<th>Source Code - javascript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Code - perl</td>
</tr>
<tr>
<td>Source Code - python</td>
</tr>
<tr>
<td>Source Code - ruby</td>
</tr>
<tr>
<td>Source Code - r</td>
</tr>
<tr>
<td>Source Code - verilog</td>
</tr>
<tr>
<td>Source Code - vhdl</td>
</tr>
<tr>
<td>Source Code - x86_assembly</td>
</tr>
<tr>
<td>Source Code - powershell</td>
</tr>
<tr>
<td>Source Code - vbs</td>
</tr>
<tr>
<td>Source Code - others</td>
</tr>
</tbody>
</table>

Non Sensitive | Doesn't contain any of the above patterns |
Disable Prisma Cloud Data Security and Offboard AWS account

- **Disable Prisma Cloud Data Security**
  - On Settings > Cloud Accounts, and edit the AWS account. Select the AWS account on which you want to disable Prisma Cloud Data Security and clear the **Data Security** option. Prisma Cloud stops ingestion for monitoring data stored in S3 buckets for the account. Data Security information from the earlier scans are not deleted on Prisma Cloud, and you have access to the alerts for the period when Data Security was enabled (using Time Query).
  - If you do not plan to re-enable Prisma Cloud Data Security for this AWS account, you can also delete the SNS topic to avoid the additional cost of that SNS topic sending messages to Prisma Cloud Data Security. You can also stop sending S3 events to the CloudTrail bucket, if you had set it up only for Prisma Cloud.
  - You can easily enable Prisma Cloud Data Security again on Settings > Cloud Accounts. Select an AWS account and check the **Data Security** checkbox. Prisma Cloud will start ingestion again and you will resume being charged for usage. All the data (before you disabled) and new data (after you re-enable Data Security) will be available for you.

- **Offboard and onboard an AWS account within 24 hours**
  - If you had onboarded an AWS account and enabled Prisma Cloud Data Security enabled, all the scanned data results including alerts can no longer using the administrative console or API. Prisma Cloud Data Security stops further ingestion to monitor the data stored within the account.
  - If you add the same AWS account within 24 hours and enable Prisma Cloud Data Security, all the previously scanned data will be available from the administrative console and the API again. You can view all previously generated alerts as reported before you offboarded the account, for example open alerts will remain open, resolved will remain resolved. Prisma Cloud Data Security resumes ingestion for this account.

- **Offboard and onboard an AWS account after 24 hours**
  - Customer offboards a previously onboarded AWS account that has Prisma Cloud Data Security enabled with scanned data results. All the scanned data results, including Alerts will stop being available in the UI or via API. Prisma Cloud Data Security stops doing any more ingestion for Prisma Cloud Data Security for this account.
  - After the account has been offboarded for more than 24 hours, Prisma Cloud Data Security will delete all the previously scanned data related to Prisma Cloud Data Security in the customer tenant.
  - Because Prisma Cloud Data Security has stopped ingestion of new S3 data, Prisma Cloud Data Security will also stop incurring additional charges for Prisma Cloud Data Security.
  - Customer onboards the same AWS account after 24 hours and enable Prisma Cloud Data Security. Prisma Cloud Data Security will start ingestion again for the account from scratch and start charging for usage. Only the new data (after you re-enable Prisma Cloud Data Security) will be there. All previously scanned data will be gone, except for alerts that were generated.

  The original alerts would not have been resolved and new alerts will be created on the same objects, creating duplicate alerts on the same objects.

- **Offboard an AWS account that was previously enabled for Prisma Cloud Data Security**
  - After the AWS account has been offboarded for more than 24 hours, all the previously scanned data and alerts will be deleted.
If you do not plan to enable Prisma Cloud Data Security for this AWS account in the future, you can also delete the SNS topic to avoid the additional cost of that SNS topic sending messages to Prisma Cloud Data Security. You can also stop sending S3 events to the CloudTrail bucket, if you had set it up only for Prisma Cloud.
Guidelines for Optimizing Data Security Cost on Prisma Cloud

Cost Implications and Control

- Prisma Cloud leverages the Data Events published by CloudTrail to keep track of any changes. These are exported as compressed log files to the customer’s S3 bucket. To ensure these log files are not contributing to unnecessary storage cost we would recommend enforcing a Bucket Lifecycle policy on this bucket for 1 month TTL (time to live). This will ensure that the files don’t contribute to the per month pricing model.
- The Cloud Trail should be enabled for Write Data events instead of Read and Write. The Read events volume in general is orders of magnitude more than Write events and AWS provides the flexibility to customers to only enable either or both. For current Prisma Cloud features around Data security, read events are not leveraged.
- Prisma Cloud will automatically skip active CloudTrail buckets and skip inactive CT logs, ELB access logs, VPC flow logs, and S3 access logs on a best-effort basis based on documentation from AWS on how to distinguish these logs from other objects. Prisma Cloud is skipping all these because they usually don’t contain sensitive information and there is no point for Prisma Cloud to handle these and charge customers for scanning them. HOWEVER, there is no guarantee that Prisma Cloud can catch all possible logs because AWS may change their log format.

API Throttling and Egress Implications

- The solution is cloud service provider (CSP) API driven and undergoes the same throttling as any other CSP API.
- The solution performs client side API throttling to ensure we don’t overuse/abuse the API rate limits enforced by the CSP.
- The client side rate limiting feature also ensures the full quota of API limit is not consumed by default to ensure this does not come in the way of customer’s processes’ or applications’ API usage.
- The data that is downloaded from customer’s storage systems are NEVER persisted anywhere on Prisma Cloud infrastructure and is only held for the duration of processing of the content for Data Profile analysis or limited by a maximum time out limit (24 hrs) whichever is hit earlier.
- There will be egress cost implications for the customer as the solution seeks to evaluate all content in customer’s buckets. The customer can choose to optimize on cost by only selecting those buckets requiring scans and filtering out any known good files that would not require any Data Profile analysis or malware analysis. E.g. Database backup files etc.