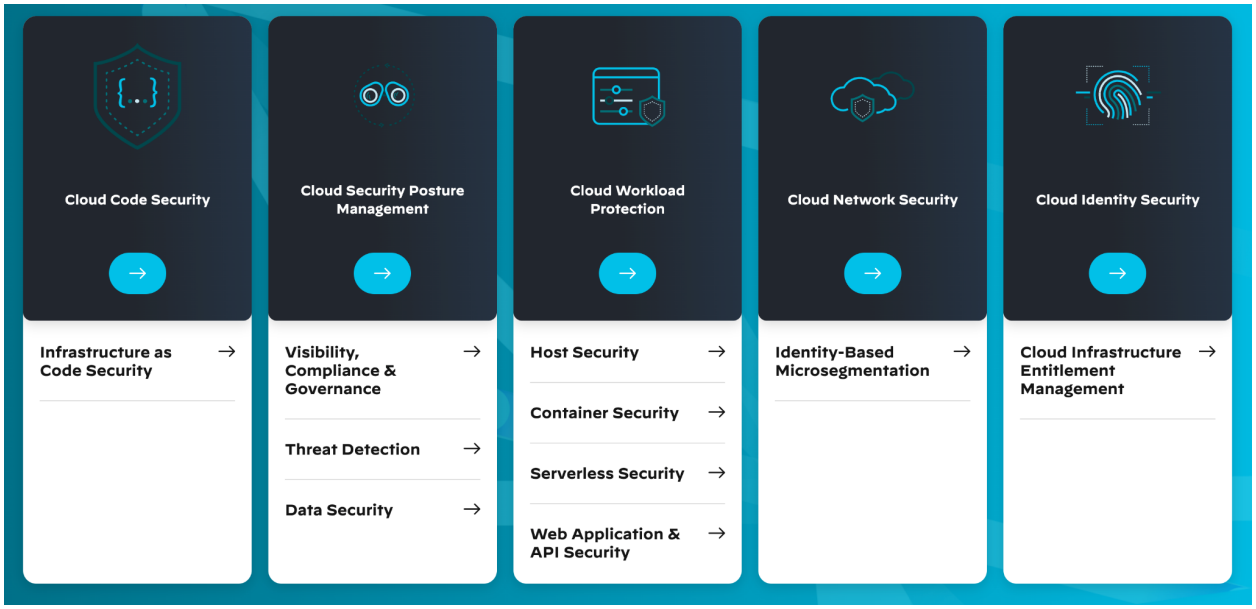
PRISMA®
BY PALO ALTO NETWORKS

Prisma Cloud Field Guide

PRISMA®
BY PALO ALTO NETWORKS

Table of Contents



CSPM

Overview

This document is intended to provide customers and partners with the best practices implementations within Prisma Cloud.

Document Structure

The basic structure of this document follows the sequence of these phases:

1. Setup
2. Configure
3. Integrate
4. Optimize

Setup

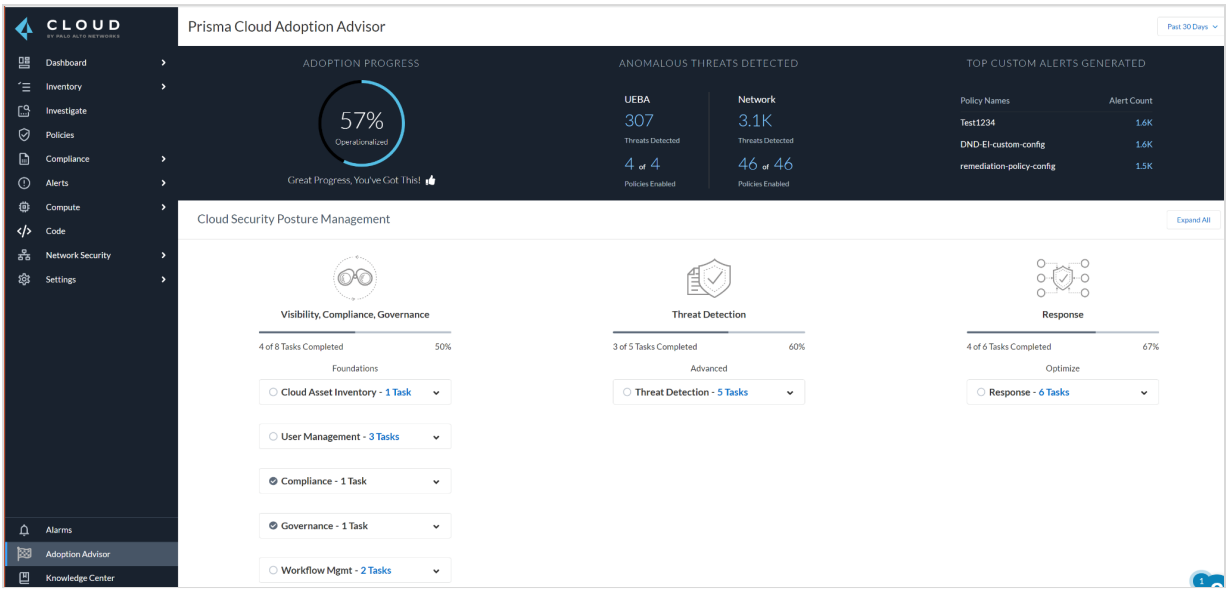
First steps are to inventory cloud accounts to start ingesting platform logs to begin the process of analysis by Prisma Cloud. Administrators need access to the platform based on their role and responsibilities. Enabling the Adoption Advisor is highly recommended, as it allows customers to view progress in configuring initial critical configuration tasks, to get the

most out of the platform and its various modules. The Prisma Cloud FAQs page will answer common questions and provide insight to different areas in the tool.

Detailed Initial Configuration for CSPM

Adoption Advisor

Prisma Cloud's Adoption Advisor (AA) is a tool that helps you see how far you've explored the tool's capabilities. It allows you to view the various tasks to perform in order to adopt Prisma Cloud – providing you visibility into security areas that you have not discovered yet. Adoption Advisor is tied to the CSPM side (CWPP still in development) and is grouped into three categories – Foundations, Advanced, and Optimize. There's a percentage that's associated with how much you've adopted Prisma Cloud so far, and completing tasks under the three categories will further increase the percentage shown. You will see AA in the bottom left of your screen with the percentage showing. Your team should utilize AA to get the most of the Prisma Cloud tool and discover/learn capabilities that you may have not configured yet but are beneficial to have within your organization. Complete the different actions where you'll see a task, description, summary, and then clicking into it will have the tool walk you through how to complete that specific task.



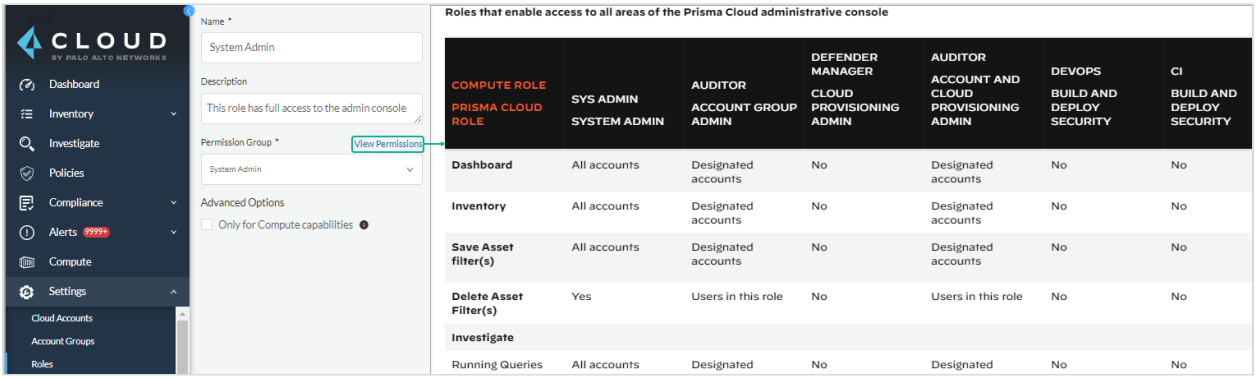
Configuring and spending time in the initial onboarding process will help in the long run when it comes to other Prisma Cloud configuration tasks, such as Alert Rules, RBAC, alert monitoring, and more.

Note: Adoption Advisor is currently available for CSPM tasks. CWPP tasks are in development and will be available shortly.

Managing Prisma Cloud Administrators

Review Prisma Cloud role permissions, create organization-specific Roles tied to the appropriate Permission Groups (System Admin, Account Group Read-Only, Cloud Provisioning Admin, etc.). It is best practice to not make every user a System Administrator and to tie the least amount of access needed to each user/team. A common role used across organizations is an Account Group Administrator or Account and Cloud Provisioning Administrator. These two roles allow a user to utilize the Prisma Cloud tool but only access the Account Group(s) they're responsible for.

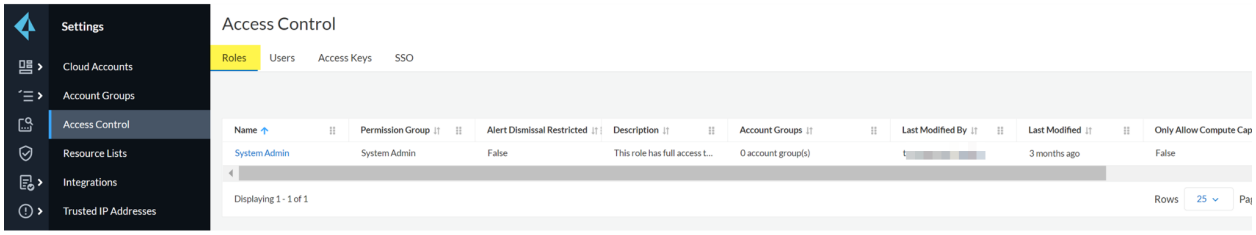
Select Settings → Roles → Add Role.



Here are some key concepts to consider:

- What is the function of the user?
- Does the user need to access the Prisma Cloud Portal, or will automation/integration provide what they need?
- If the user does not require access to the Prisma Cloud Portal, will an emailed report be sufficient enough?
- Does the user need to do more than just consume asset/security data?
- Is there asset/security data the user should NOT have access to?
- Is there a capability the user needs to access that cannot be done with a read-only role?

See the below screen capture showing Access Control settings. Here, you can access "Roles", "Users", "Access Keys", and "SSO" configurations.



Single Sign On

In setting up authentication management, SSO integration is recommended as best practice. You can enable SSO using the Identity Provider (IdP) your organization utilizes, as long as it supports SAML.

Examples include Okta, Microsoft Active Directory Federation Services (AD FS), Azure Active Directory, Google, or OneLogin. Note, you can only configure a single IdP across the cloud accounts that Prisma monitors. Organizations can add administrative users on Prisma Cloud to create their local account when SSO is enabled or utilize Just-In-Time (JIT) provisioning on the SSO configuration if you'd like to have the accounts created locally.

Note: It is important to provide at least two admin users who can bypass the third-party SSO - a setting under the SSO settings/configuration. This is needed in the event there are SSO issues, such as a SSL certificate expiration or an IdP problem.

Some Examples of SSO Roles

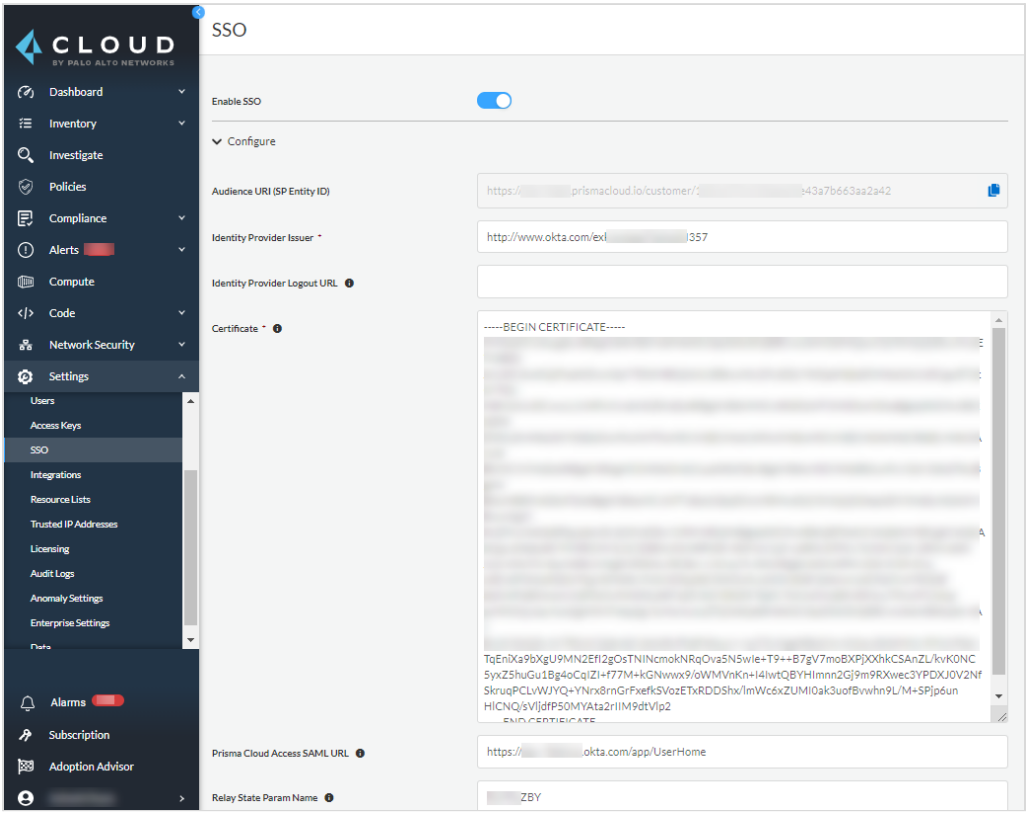
- Create roles based on user personas (DevOps, SecOps, SOC/IR, Compliance, App Developers)
- Attach roles to different account groups based on cloud account ownership

If you are facing issues with user authentication and SSO configuration, you can find a list of the last five SAML login failures by navigating to the bottom of the SSO configuration page.

SAML Troubleshooting common errors

1. Mandatory Fields: Check to ensure that mandatory fields are filled out correctly. IdP Issuer and Certificate are the two required fields. If you're using HIT, the additional fields must also be filled correctly.
2. SSO Not Enabled: Enable SSO under the main Prisma Cloud settings.

3. Authentication Failed Errors: If a user experiences an authentication failure when they try to log in, you can investigate the issue using a SAML browser plugin to capture the assertion that's being sent to the user's browser. SAML Assertion is the XML document that the IdP sends to the service provider that contains the user authorization. It is important to remember that the URL or certificate information in the asset may not match the Prisma Cloud configuration.
4. JIT Authentication Failed Errors: The URL, certificate, or JIT user parameters may not be correct and can be analyzed from the Assertion XML document. There may be missing attribute values in that the Prisma Cloud SSO config may have an incorrect attribute key name.

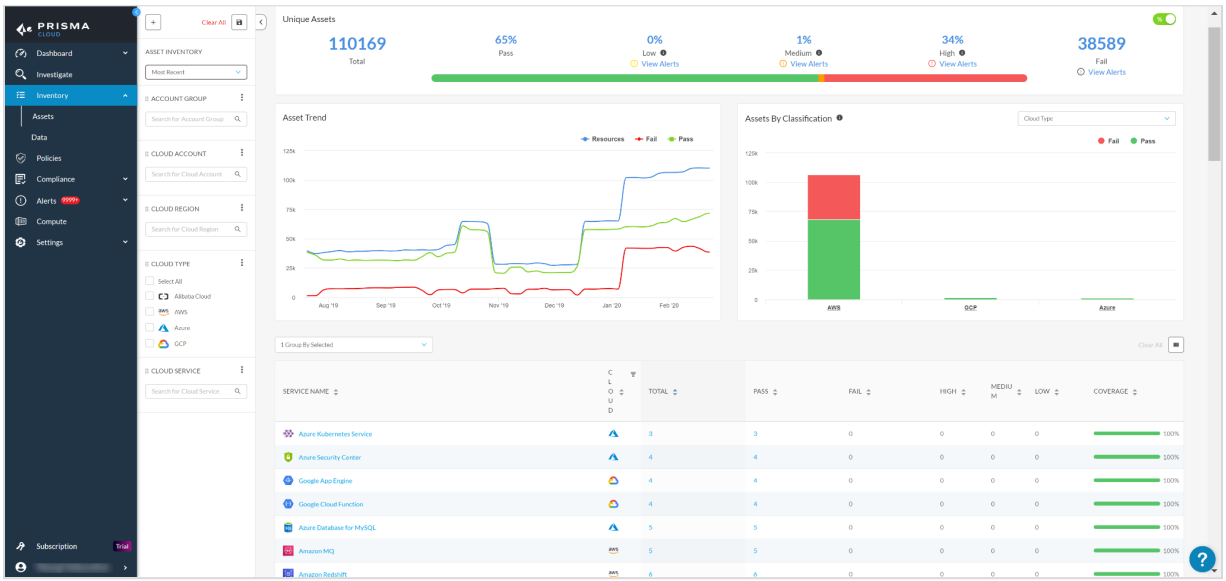


Onboarding Best Practices

Onboarding a cloud account at the top hierarchical level is recommended as it saves you time and helps smoothly onboard cloud accounts within the hierarchy (i.e. AWS Organization, Azure Tenant, GCP Organization). Any individual cloud accounts that were onboarded prior to the hierarchy but are within it, will be automatically structured without needing to specify additional information. You have the ability to onboard individual units under the hierarchy,

such as eight out of ten Organizational Units (OU) in an AWS Organization. If there are OUs that you would like to exclude, it can be done at the time of onboarding.

At the last step of onboarding, there is a status check that will let you know if Configuration, Cloudtrail, VPC Flow Logs, and the Organization are connecting properly. Validating the statuses to ensure there are no issues (validated by a green check mark) will allow you to ingest data properly from your cloud account or hierarchy. After onboarding is complete, you can navigate to Prisma Cloud's Asset Inventory to review the resources that are being ingested and specific details such as the number of specific resources/assets, configuration details, audit trails, network connectivity, and more.



AWS Onboarding

Individual Account Onboarding

An individual AWS cloud account can be set up/added to Prisma Cloud manually as well as in an automated method. There are a total 4 major required steps in adding an AWS account to Prisma Cloud. When you choose the automated method it only completes three of the four required steps. There is one step that has to be completed manually when choosing the automated method. More details about the steps and permissions and cloudFormation template can be found here

It is **recommended** that you **use the automated method to onboard an AWS cloud account** into Prisma Cloud as it is quick and easy to use and it also reduces any chance of misconfigurations in your setup.

When setting up an AWS account, there are 4 required steps that need to be completed:

1. Create custom role for Prisma Cloud service (Automated or Manual)
2. Enable CloudTrail (Automated or Manual)
3. Setup CloudWatch and Enable VPC flow log (Manual)
4. Add AWS account to Prisma Cloud console (Automated or Manual)

The first 3 steps are done in the AWS console and the 4th step is done in the Prisma console.

Step 1: Create a custom Role for AWS Resource Configuration data
The first step is to create a custom role within the AWS account to allow Prisma Cloud to make the required API calls to your AWS cloud account for collecting the metadata for your cloud resources. Here are some of the sample AWS APIs ingested by Prisma Cloud . This is required to ingest the configuration data from your cloud account for the deployed resources into Prisma Cloud. This step is taken in your AWS cloud account. This can be done manually or by using a CloudFormation template from your AWS console.

Step 2: Enable CloudTrail for Event data
CloudTrail is usually enabled by default for all cloud regions within AWS but if you have disabled CloudTrail, you will need to re-enable it for on boarding your AWS account into Prisma Cloud. CloudTrail is needed for ingesting user and event data from your AWS cloud account. This step is taken in your AWS cloud account.

Step 3: Set up cloudwatch and Enable VPC Flow logs
This is a manual step which is required to ingest the network level traffic data from your cloud account onto Prisma Cloud. For this, you must set up a CloudWatch log group and also enable VPC flow logs to get the network traffic data into Prisma Cloud. This step is taken in your AWS cloud account. Please note that this step is required even if you use the CloudFormation template (CFT) for configuring the other settings.

Step 4: Add an AWS account to Prisma Cloud
You must add your AWS account using the Prisma Cloud console. This step is taken within the Prisma console.

Prerequisites for adding an AWS account to Prisma Cloud:

1. The **AWS external ID** that is defined when the role is created. This is used to establish a trust relationship between your Prisma Cloud account and the AWS account to pull your data.
2. The **Amazon Resource Name (ARN)** for the role that was created. This is a role that you create in AWS for Prisma Cloud.

Note: The External ID regenerates a new value each time you repeat the cloud onboarding setup. It is recommended to copy the External ID and save it and use it consistently throughout the onboarding process.

Recommendations for when on-boarding an AWS account Manually:

1. Make sure you choose the appropriate cloud account type
2. Make sure you choose the right mode “Monitor” (Prisma Cloud will only have read access to your AWS cloud account resources) or “Monitor & Protect”(Prisma Cloud will have read as well as access to remediate resource configuration issues when allowed/given access by admin)
3. Make sure that the Role is configured correctly for the Prisma Cloud in your AWS console in the same region as your AWS account in order to establish a proper trust relationship. More details on the permissions and roles can be found here
4. Ensure that you have the VPC flow logs enabled to send logs to cloudwatch and have the proper permissions for it

Note: If you would like Prisma Cloud to ingest data and logs from other integrations like AWS guardDuty, AWS S3 or AWS inspector make sure to enable these from AWS console as these are disables by default.

If you plan to enable “Data security” within Prisma Cloud to scan to prevent data leaks and to protect your cloud storage data then this option is only available with “Monitor” mode as data protection policies on Prisma Cloud do not support auto remediation. If by any chance you choose the “Data Security” option with monitor & protect mode you will have to manually fix the issues to address alerts generated by data policies. More details can be found here.

AWS Organizations

You can on-board your master or root AWS account on Prisma Cloud. When you enable AWS organizations on the AWS management console and add the master account that has the payer role, all member accounts within the master or root account are added in a streamlined manner onto Prisma Cloud. This helps in bulk onboarding of AWS accounts into Prisma Cloud.

The flow to on board an organization or master account is that you first deploy the CloudFormation template in the master account to create the Prisma Cloud role to monitor or monitor and protect your resources deployed in the master account and then you use the CloudFormation stacksets to create the Prisma Cloud role to access each member account within the master account. This automated process will basically on board any new member account that you add to your master account automatically on Prisma Cloud within a few hours.

You can also choose to exclude a few OUs (organizational units) by manually disabling individual member accounts on Prisma Cloud once they are on boarded or you could also on

board a subset of accounts and exclude the OUs you don't want while deploying the stackset so that Prisma Cloud role is only created in the desired OU you want to onboard.

It is recommended that you have a predefined list of OUs you want to have included/excluded while on boarding for better experience and to save time later to modify existing setup.

Note: After you have on boarded an account as an AWS organization, you cannot roll back.

There are 2 different scenarios that you can come across while on boarding and AWS org account:

1. Add a new AWS organization account to Prisma Cloud
2. Update an onboarded AWS organization

To **add a new AWS organization account to Prisma Cloud** here are some of the basic steps required on the Prisma Cloud side:

Step 1: Access your Prisma Cloud console and select Settings> Cloud Accounts> Add New

Step 2: Select AWS as the cloud provider

Step 3: Enter a Cloud Account Name and select onboard as "Organization" . Please note that a cloud account name will be auto populated for you but you can replace it with a cloud account name that uniquely identifies your AWS Organization on Prisma Cloud.

Step 4: Select mode (Monitor or Monitor & protect) based on your requirements whether you want to enable read only or read-write access for the resources in your cloud account. Your selection will determine which cloud formation template to be used to automate the process of creating the custom role required for Prisma Cloud.

Step 5: Set up Prisma Cloud role on the AWS master account. This step can be automated using a cloud formation template. The cloud formation template (CFT) enabled the ingestion of configuration data and AWS cloudtrail logs (audit events) only. It does not support the ability to enable VPC flow logs for your AWS account or any other integration such as Amazon GuardDuty or AWS inspector.

Step 6: Log in to your AWS account and click on "create stack" from Prisma Cloud console from your onboarding flow.. For creating a stack you will need stack name, external ID (Prisma Cloud ID), Prisma Cloud role name (role that will be used by Prisma Cloud to authenticate and access resources in your AWS account). These values will be autofilled for you.

Once the stack is created, you will need the PrismaCloudARN value which you can find from the "outputs" tab from your stack screen (from AWS console) which you can paste in the Prisma Cloud console screen under "Configure Master Account" section.

Step 7: Create **stacksets** to create a Prisma Cloud role within each of your **AWS Member Accounts.**

a. You can download the template file from here based on (Monitor or Monitor and protect mode)

Monitor Mode: https://s3.amazonaws.com/redlock-public/cft/rl-read-only-member.template
Monitor & Protect Mode:
https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write-member.template

b. In AWS console , select Services> CloudFormation> StackSets> Create stackset (**verify that you are logged into your AWS Master account)**
Upload the template file which you downloaded (step 7.a) and give a name to the StackSet.

c. Enter **ExternalID** (create one, e.g Test-Number-1234) and **PrismaCloudRoleName** (this value is auto-populated, but you can also modify it as long as it contains ord within the string value)

d. Select service managed permissions, and then select Deploy to organization under deployment targets in the next screen. In this step, you have the choice to select which member accounts you would like to on board if you don't want to on board all the member accounts. You can select Deploy to organization unit OUs and deploy stackset to only the selected/desired OUs you want.

e. Set automatic deployment "enabled" and account removal behavior "Delete stacks"

f. Specify one region and make sure that the region you choose is enabled on all accounts within your AWS organization otherwise if the region is disabled then the template will not be able to deploy resources within that region and will fail with errors.

g. Select percentage and set it to 100 for max concurrent accounts and failure tolerance options under deployment options and keep the default setting for sequential under region concurrency.

Step 8: Review configuration, acknowledge and submit. Once submitted, the stackSet creating process will be initiated. Wait for the "SUCCEEDED" status to show up. Once the process completes successfully, you will see each member account where the role was created would be listed under "stack instances" in the AWS console.

Step 9: Copy the values for **PrismaCloudRoleName** and **ExternalID** and configure the details on the Prisma Cloud. Acknowledge that the stackset has been created in member accounts and then click next.

Step 10: You must ensure to select an account group. You must assign all the member accounts twitch the AWS org to an account group for better ease of use and also so that you can create an alert rule for checks to associate with that account group so that alerts are generated when a policy violation occurs within those accounts.

Note: you can also modify the cloud account settings if you would like to selectively assign AWS member accounts to different account groups on Prisma Cloud.

Step 11: Verify the status of your configuration. Note, if you do not acknowledge the "i confirm the stackset has created Prisma roles in member accounts successfully'' checkbox, then you will not see the number of member accounts in the status screen.

To **Update an onboarded AWS organization** , the steps are similar to how we on board an AWS org onto Prisma Cloud but some of the differences are as listed below:

In addition to updating the CFT stack for enabling permissions for new services, you can also use this workflow to update the account groups that are secured with Prisma Cloud, change the protection mode from monitor to monitor and protect or vice versa, redeploy Prisma Cloud role in member accounts etc.

Step1: The steps are similar to **adding a new AWS organization account to Prisma Cloud** but instead of creating a new stack you can just go to your AWS management console and update the cloudformation stack. Select the **PrismaCloudApp stack** and click **"Update stack"**

Step 2: Upload the template that you can download from :
Monitor Mode: https://s3.amazonaws.com/redlock-public/cft/rl-read-only.template
Monitor & Protect Mode:
https://s3.amazonaws.com/redlock-public/cft/rl-read-and-write.template

Step 3: Review changes , acknowledge and submit.

To configure the member accounts, the steps are similar to how we saw in **add a new AWS organization account to Prisma Cloud** the only difference is that you

Step 4: navigate to Services> cloudformation>stacksets and use **"Edit StackSets details"**

Step 5: Upload the new template and follow the steps mentioned in the add a new AWS organization account to Prisma Cloud section above

Azure Onboarding

Overview

Prisma Cloud CSPM Onboarding allows customers to add their Cloud Account by Subscription or by Tenant when it relates to Azure.

Before You Begin

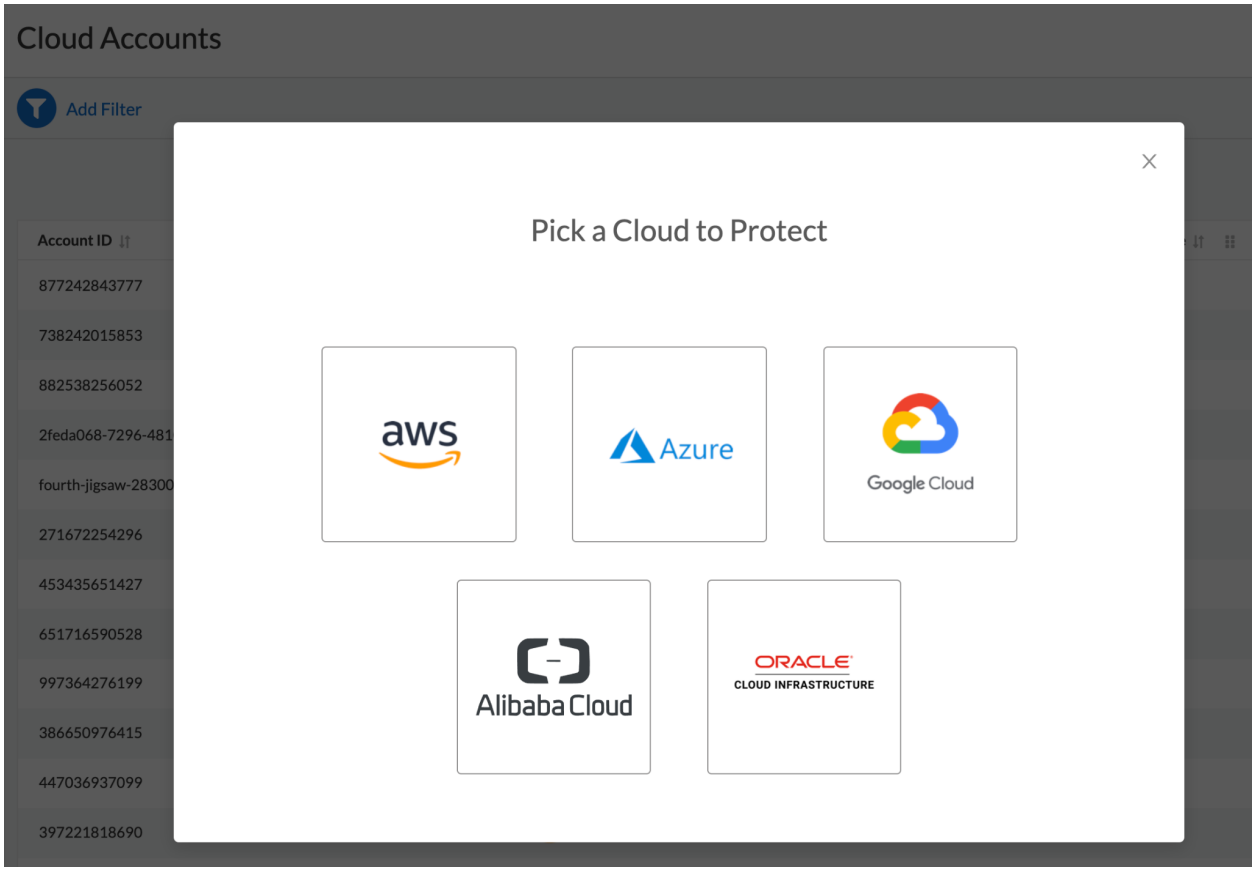
To accomplish this, you will:

- Need to have access to Management Groups
- Need to have access to the Tenant Root Group

Procedure 1: Onboard Azure Tenant

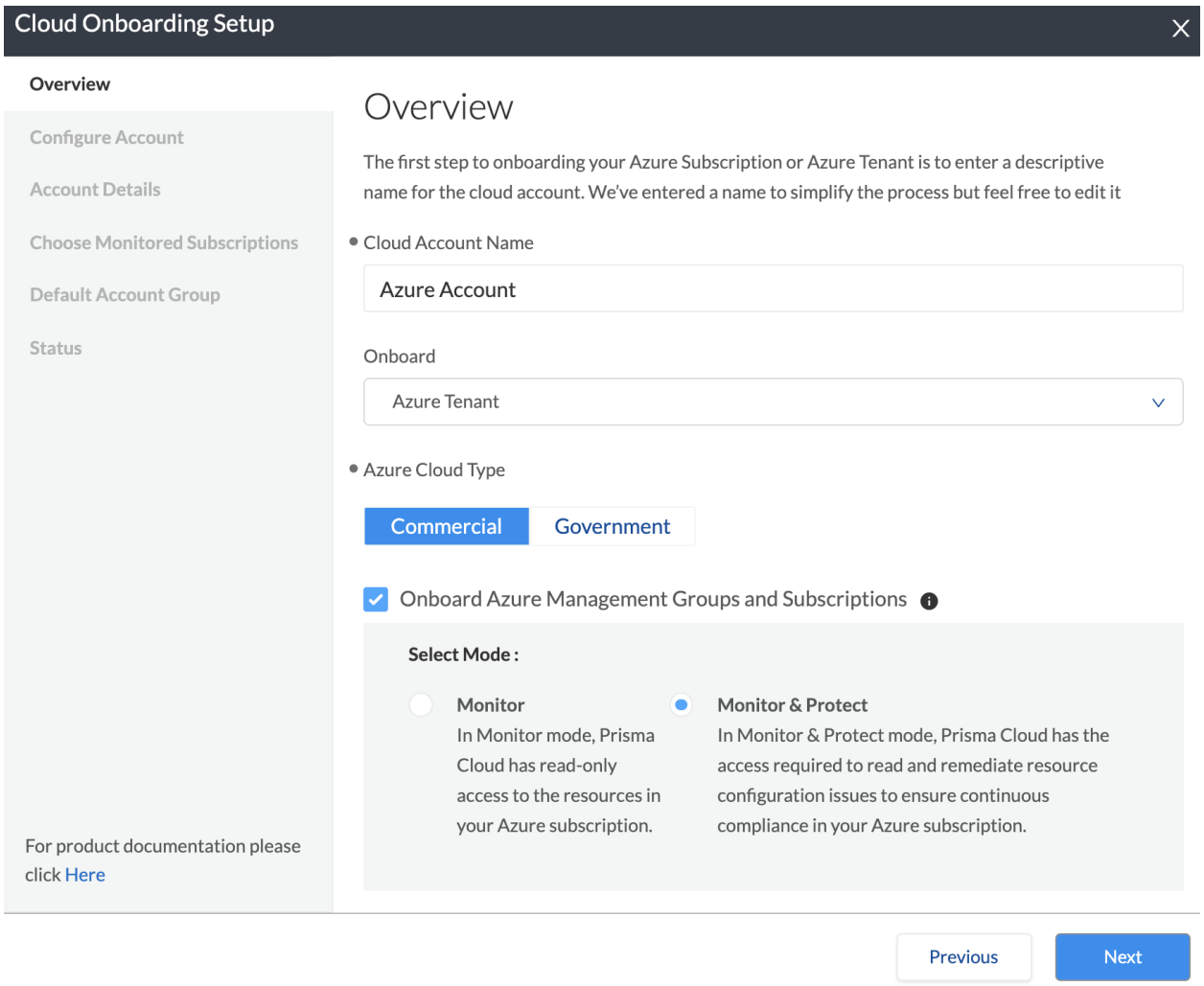
Step 1 Login to Prisma Cloud

Go to Settings > Cloud Account > Add Account > Select Azure

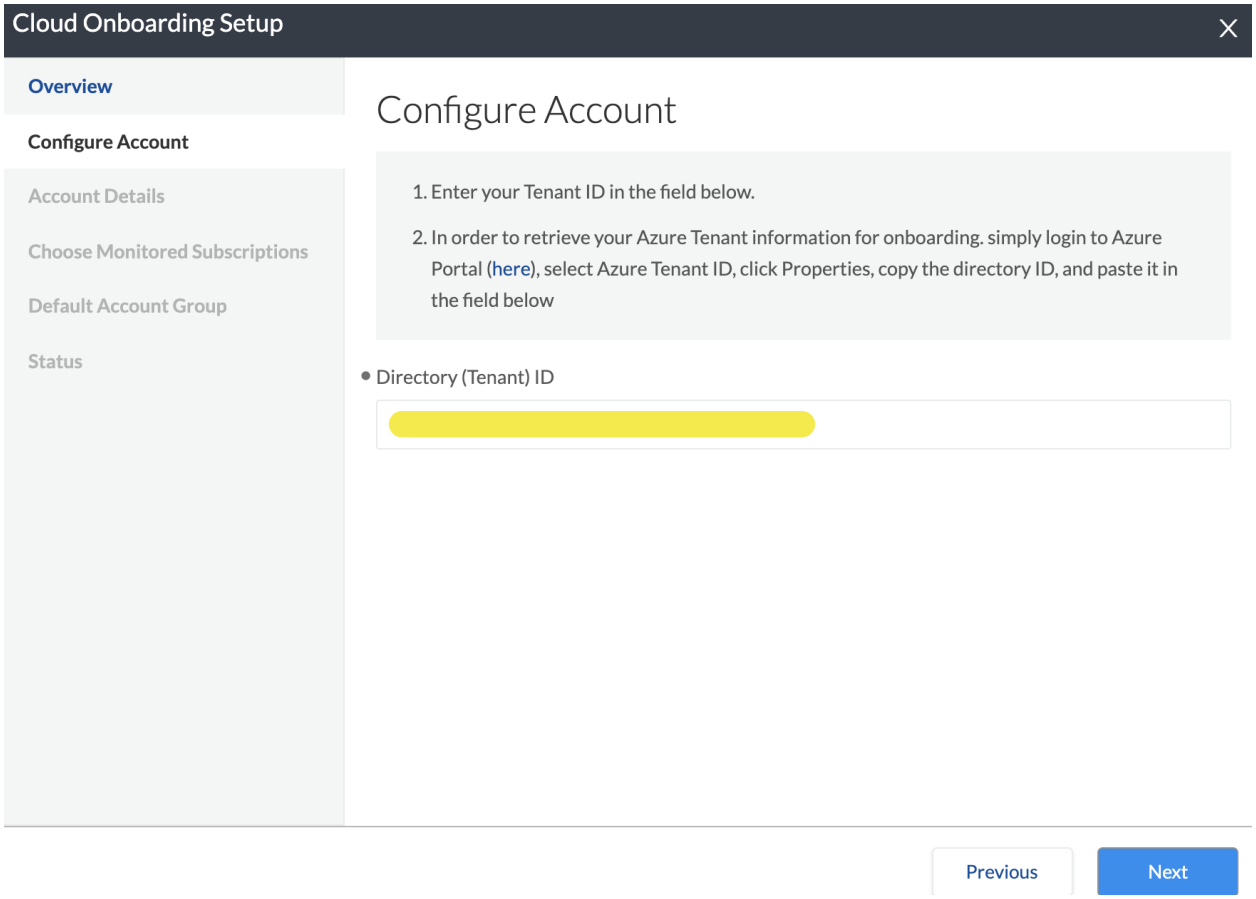


Step 2 Select Onboard as Azure Tenant

Select > Onboard Azure Management Groups and Subscriptions



Step 3 Add your Tenant ID



Step 4 Configure Account Details

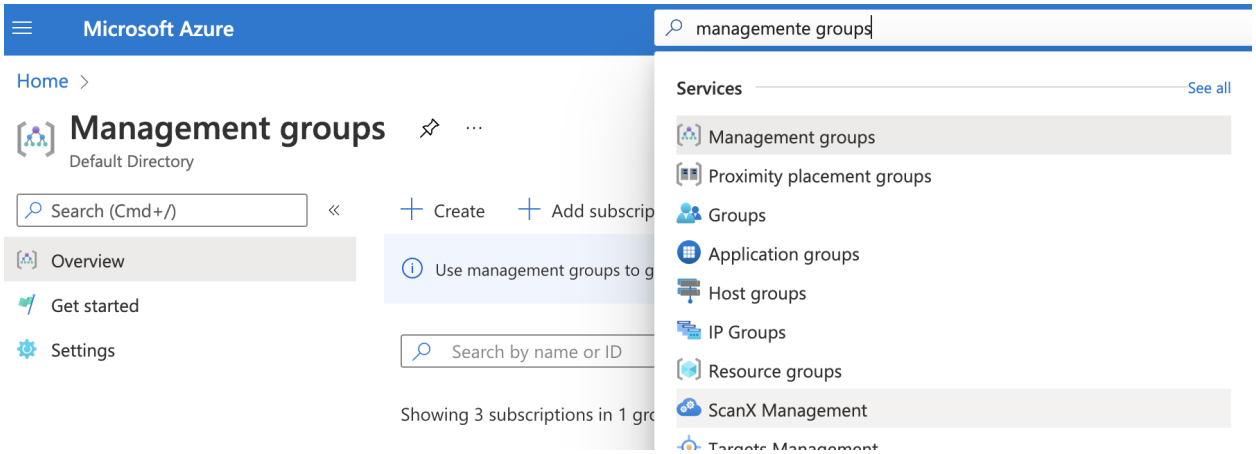
Get the following details from your account running the Terraform Script or by doing manually using the following link
Register an App on Azure Active Directory
https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/connect-your-cloud-platform-to-prisma-cloud/onboard-your-azure-account/register-an-app-on-azure-active-directory.html#ida0e4567f-7cf3-455b-b755-b2e2072ae0a0

- Application (Client ID)
- Application Client Secret
- Enterprise Application Object ID

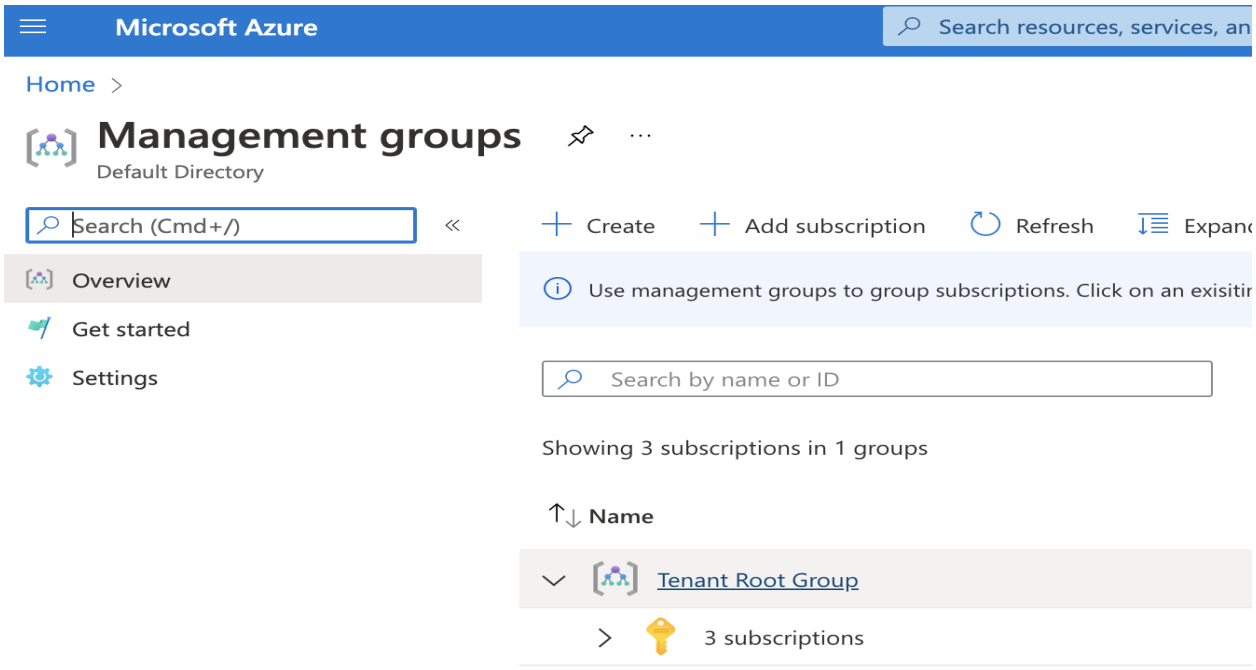
Procedure 2: Add Roles to the Root Group

Step 1 Validate you have access to the Tenant Root Group

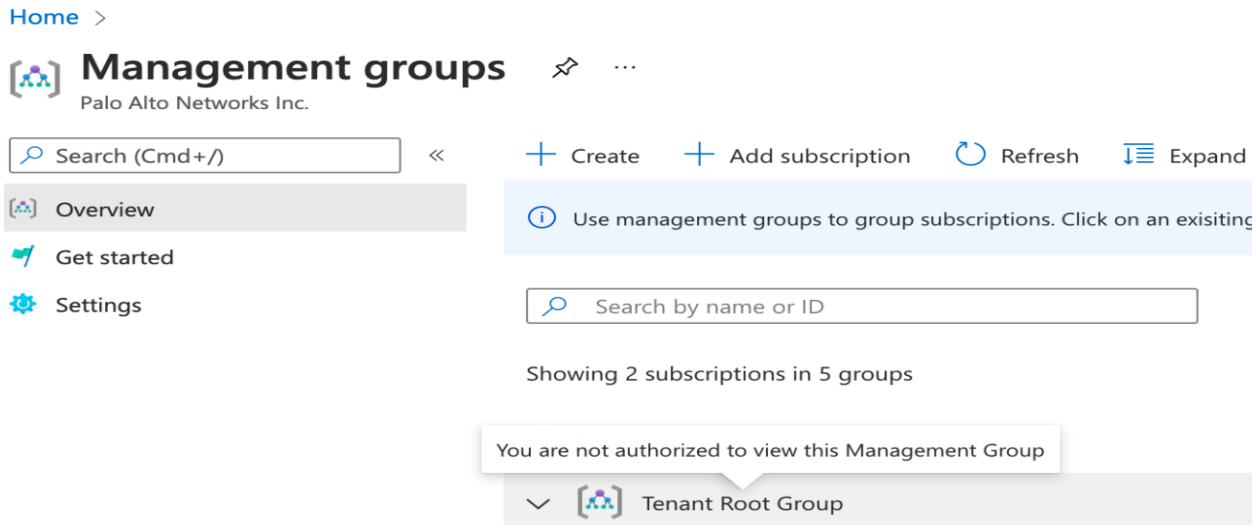
Go to your Azure Portal and search Management Groups



Step 2 Click on top of Tenant Root Group then Go to IAM

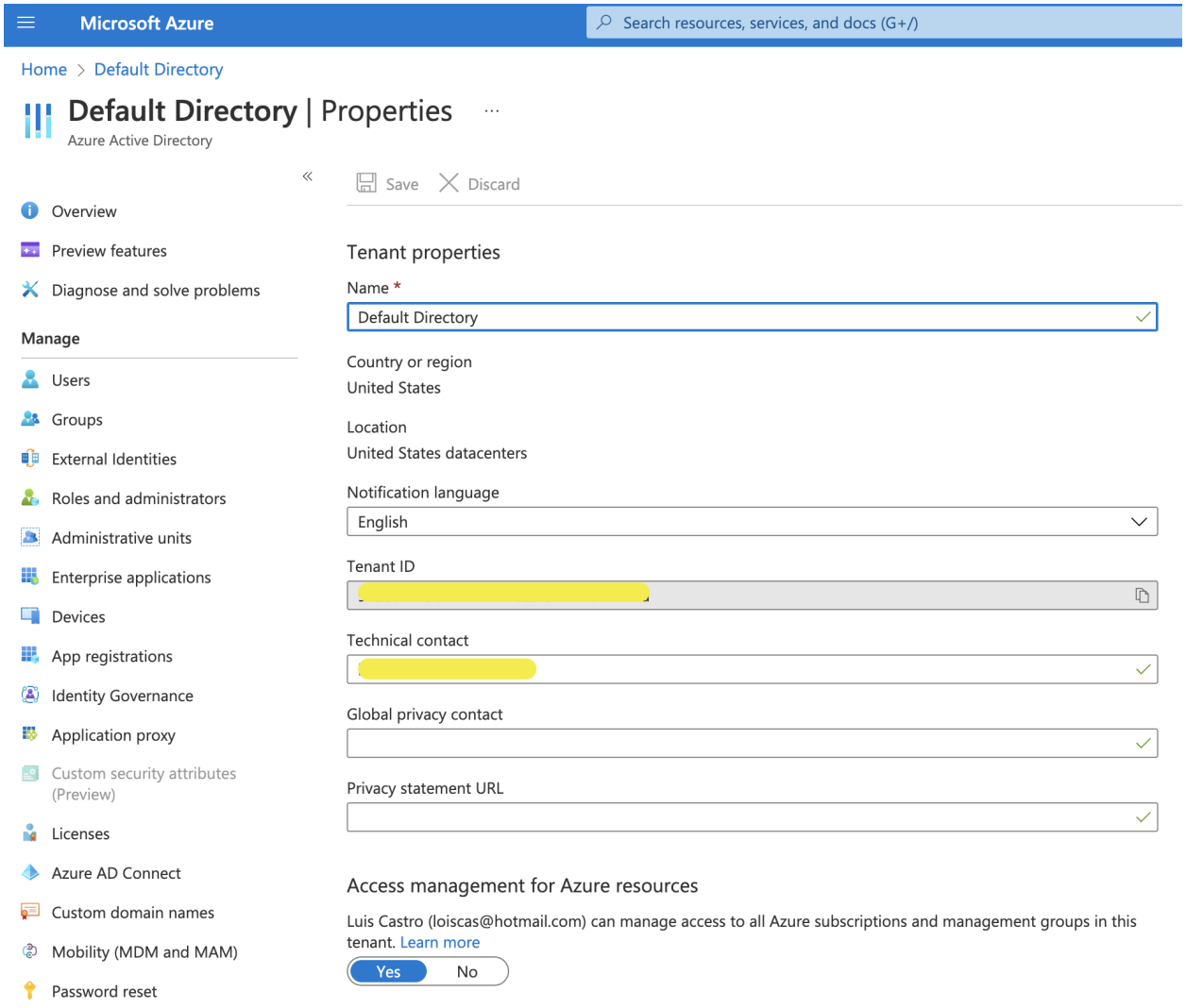


If Tenant Root Group is not clickable, it means your user does not have access to the Root Group



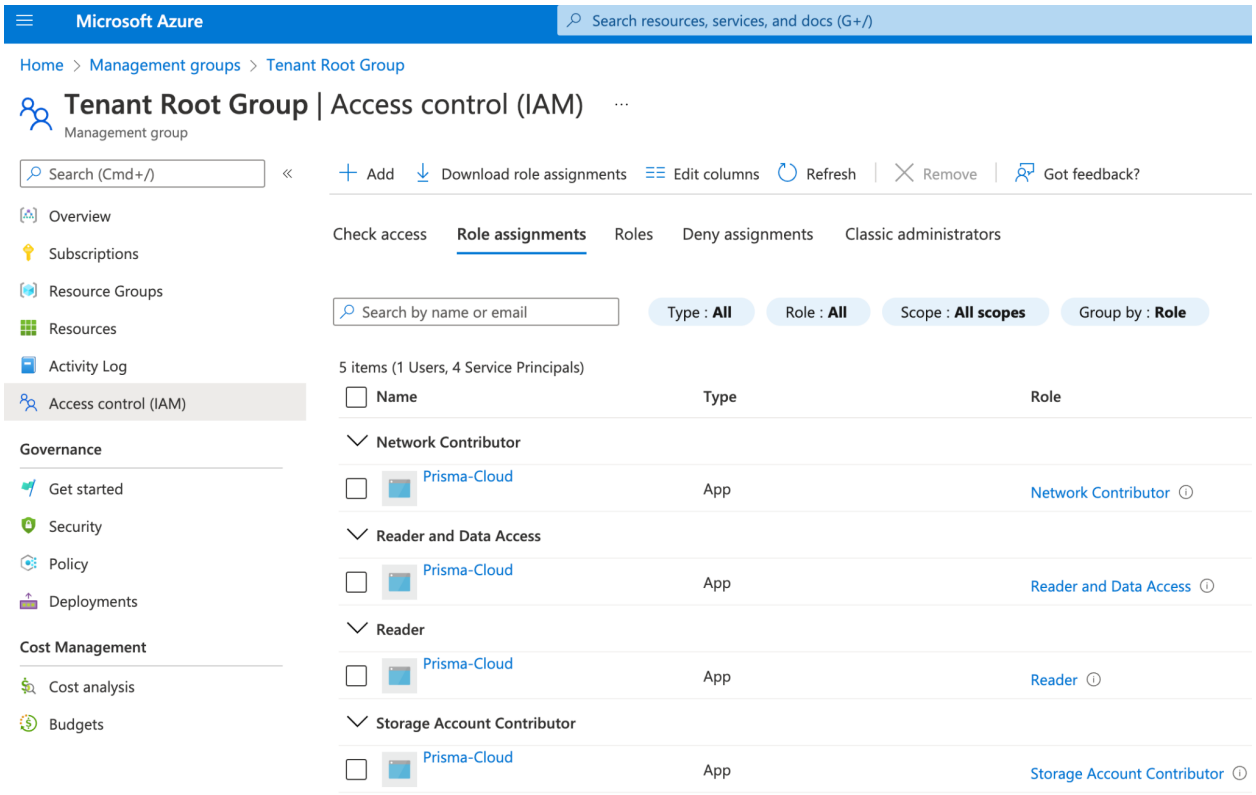
To gain access to Tenant Root Group follow the next steps:
Go to Azure Active Directory > Properties > Access management for Azure resources > Select Yes

Note: This requires a high level of permission inside the Azure Tenant which might require approval from the Tenant Administrator



Step 3 Inside Tenant Root Group IAM go to Role Assignment and add the following roles to your Application Prisma Cloud

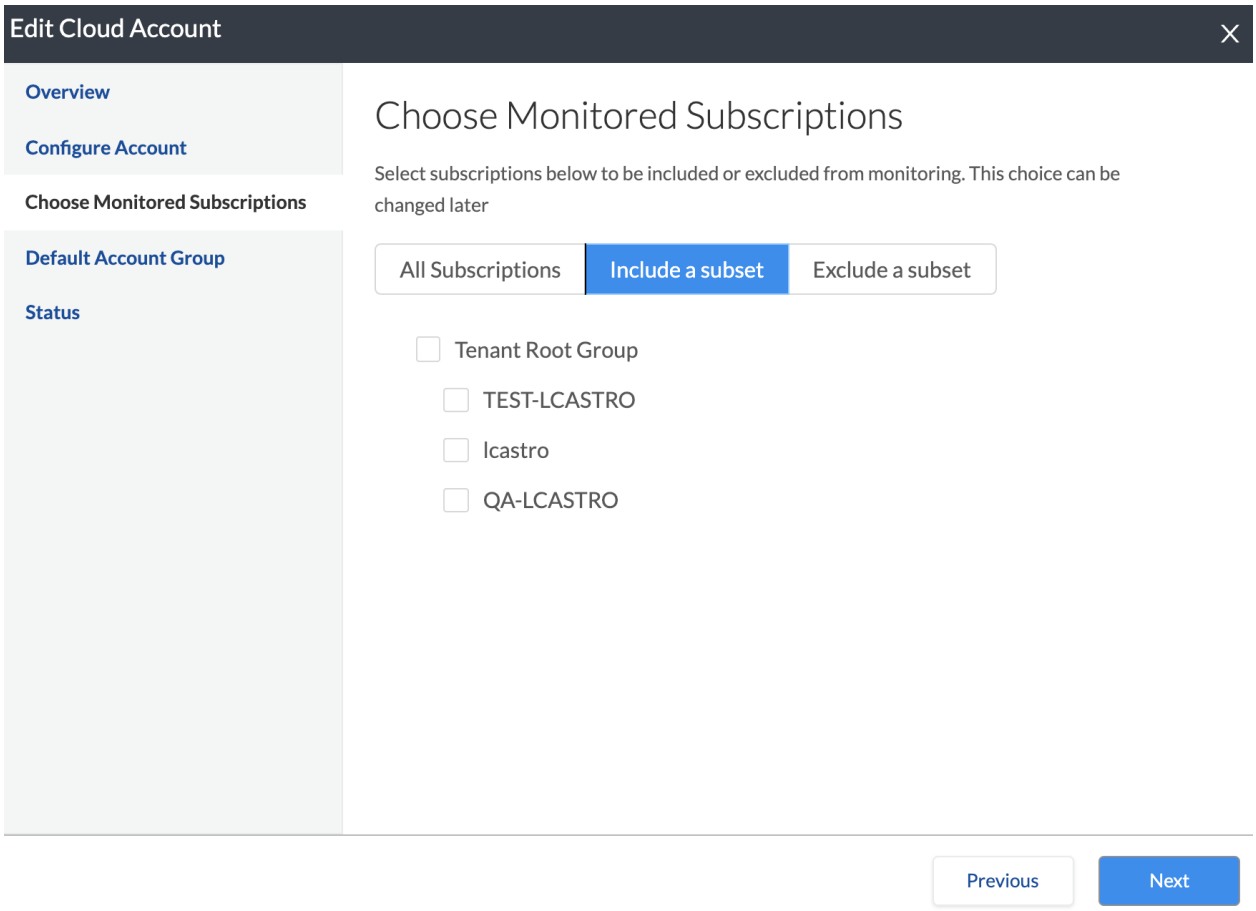
- Reader
- Reader and Data Access
- Network Contributor
- Storage Account Contributor



Procedure 3: Select your desired subscriptions

Step 1 You can choose the following:

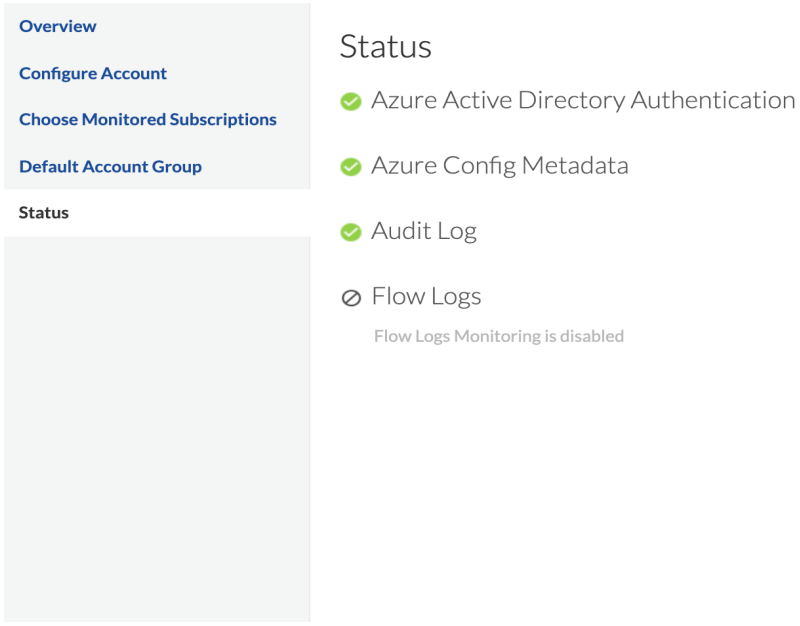
- All Subscriptions
- Include a subset
- Exclude a subset



Step 2 Add your Default Account Group

Step 3 Finalize configuration - Check for the status of the configuration, if ok all should be in green.

If Flow Logs not enable you will see that its disabled



Success!

You successfully updated this cloud account.

Related Information

Register an App on Azure Active Directory

https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/connect-your-cloud-platform-to-prisma-cloud/onboard-your-azure-account/register-an-app-on-azure-active-directory.html#ida0e4567f-7cf3-455b-b755-b2e2072ae0a0

GCP Onboarding

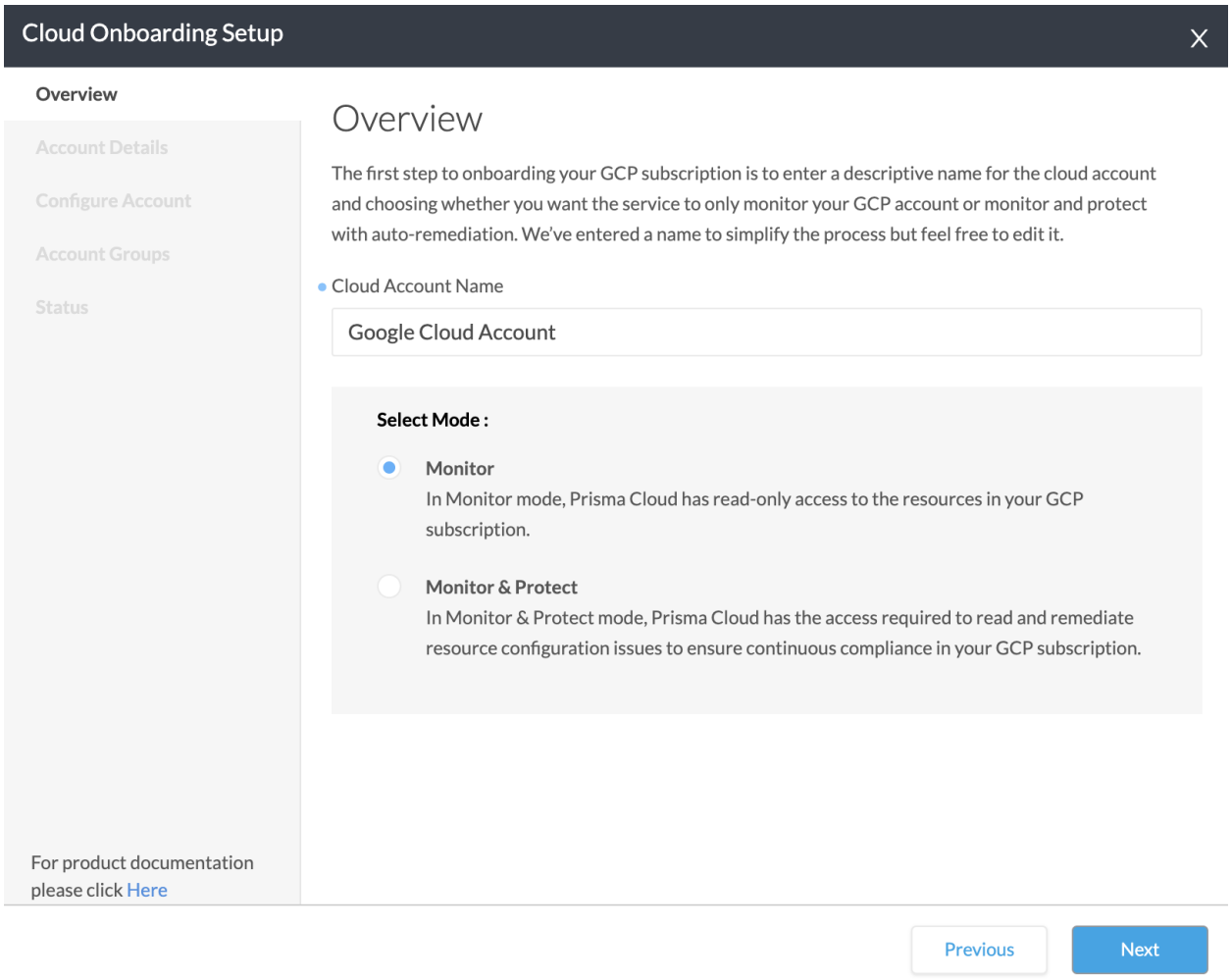
To enable Prisma Cloud to retrieve data on your Google Cloud Platform (GCP) resources and identify potential security risks and compliance issues, you must connect your GCP projects to Prisma Cloud. This document will explain how to onboard GCP folder and GCP projects within (GCP current and future projects) onto Prisma Cloud.

GCP supports flexible resource hierarchy and more details can be found here. In the GCP resource hierarchy, Folders are an additional grouping mechanism on top of projects which customers often use to group a set of GCP projects that belong to a business unit.
If a customer is using G-suite appscripts, then whenever an AppsScript project is created a corresponding GCP project is also created automatically in the background. Why because the G-suite appscripts use the GCP to manage authorization, Advanced services, and other details. In many cases, this leads to a large number of GCP projects [sometimes in 10000+] being created in GCP organization without ANY real IaaS cloud resources within.
In such cases, you may want to exclude the following parent folders for Prisma cloud onboarding.
Organization root > system-gsuite > apps-script | Organization root > system-gsuite

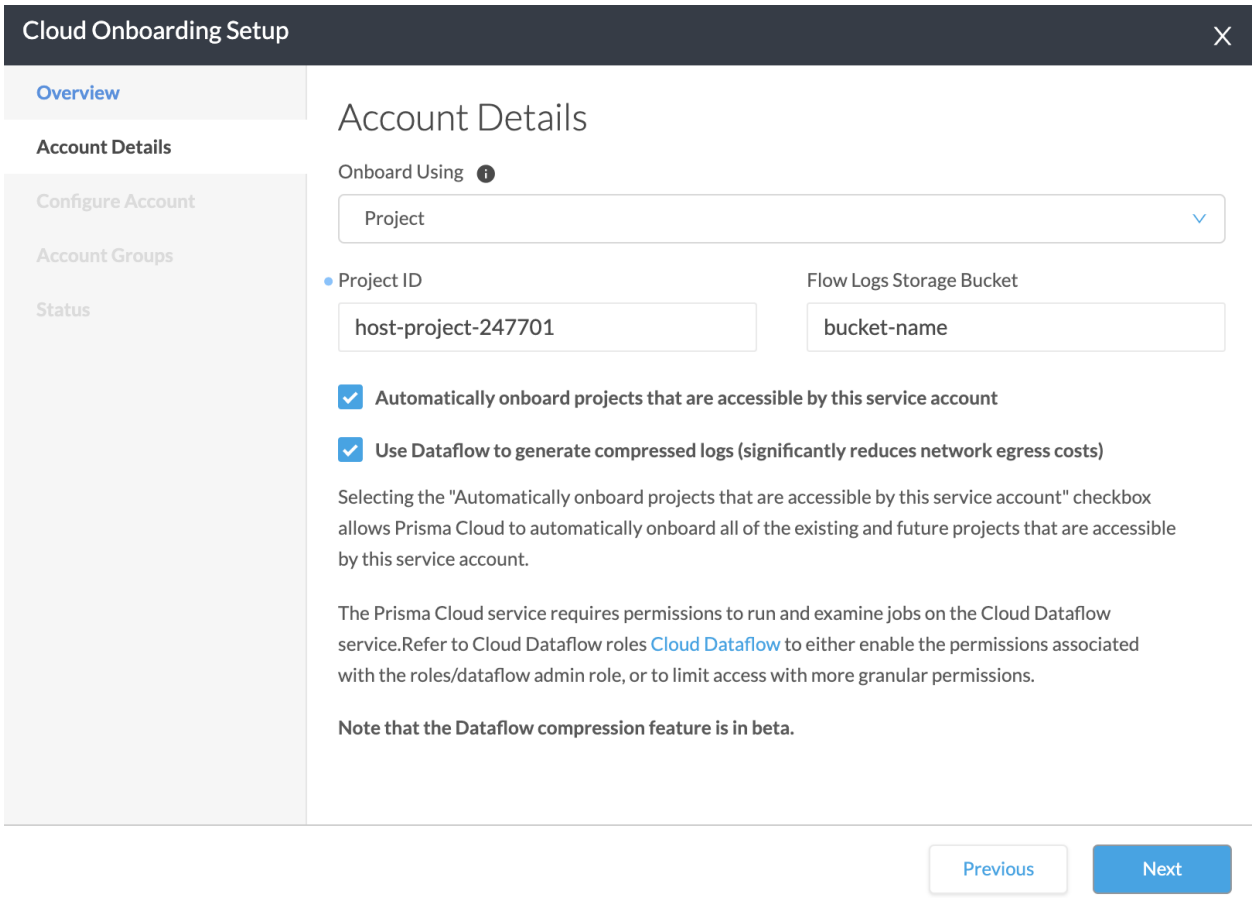
Steps for GCP folder level onboarding

In order to analyze and monitor your Google Cloud Platform (GCP) project, 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.

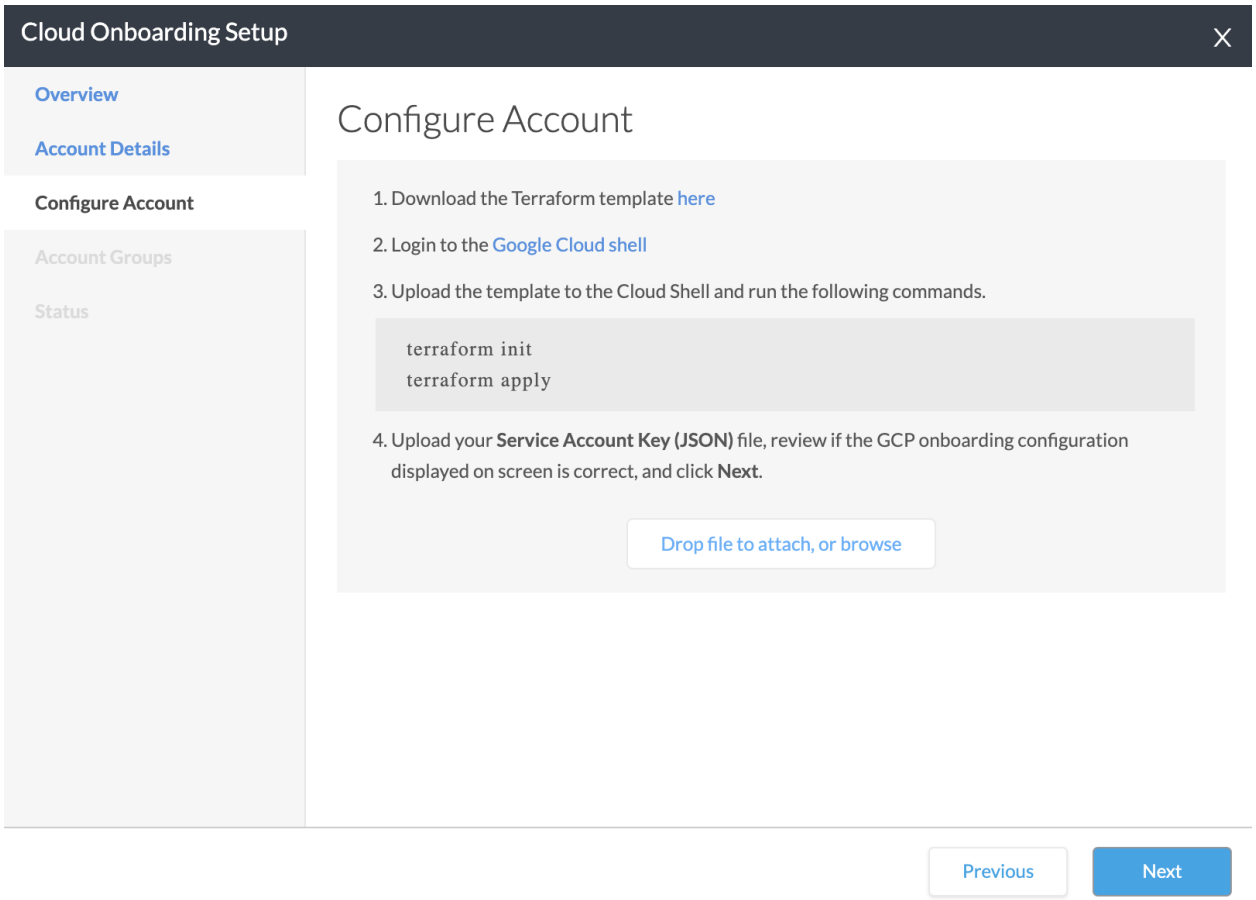
1. Go to https://app3.prismacloud.io/settings/cloud_accounts and click on +ADD NEW



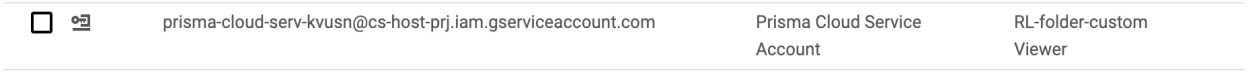
2. After selecting the mode, provide the GCP project info in the account details page. Make sure to select 'Automatically onboard projects that are accessible by this service account'. Dataflow compression and Flog logs are optional here.



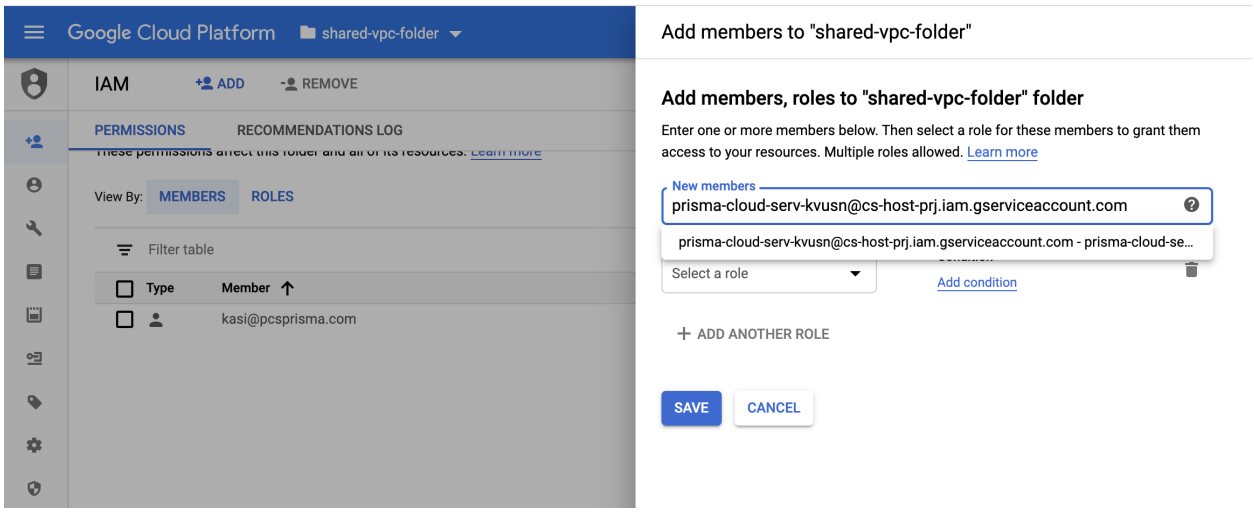
4. Download the Terraform template and run in the 'specified' GCP project (from the previous step). Capture the output and create a .JSON file of the GCP IAM service account credential.



5. In the GCP console, go to the 'specified' GCP project and look for the service-account created and copy the service-account member name.
It will be something like 'prisma-cloud-serv-<randome-value>@<gcp-project-name>.iam.gserviceaccount.com', example below:



6. Now, go to the GCP folder(s) that you wish you onboard to prisma cloud and add this service-account in there and grant one additional permission 'folder viewer' pre-defined role.

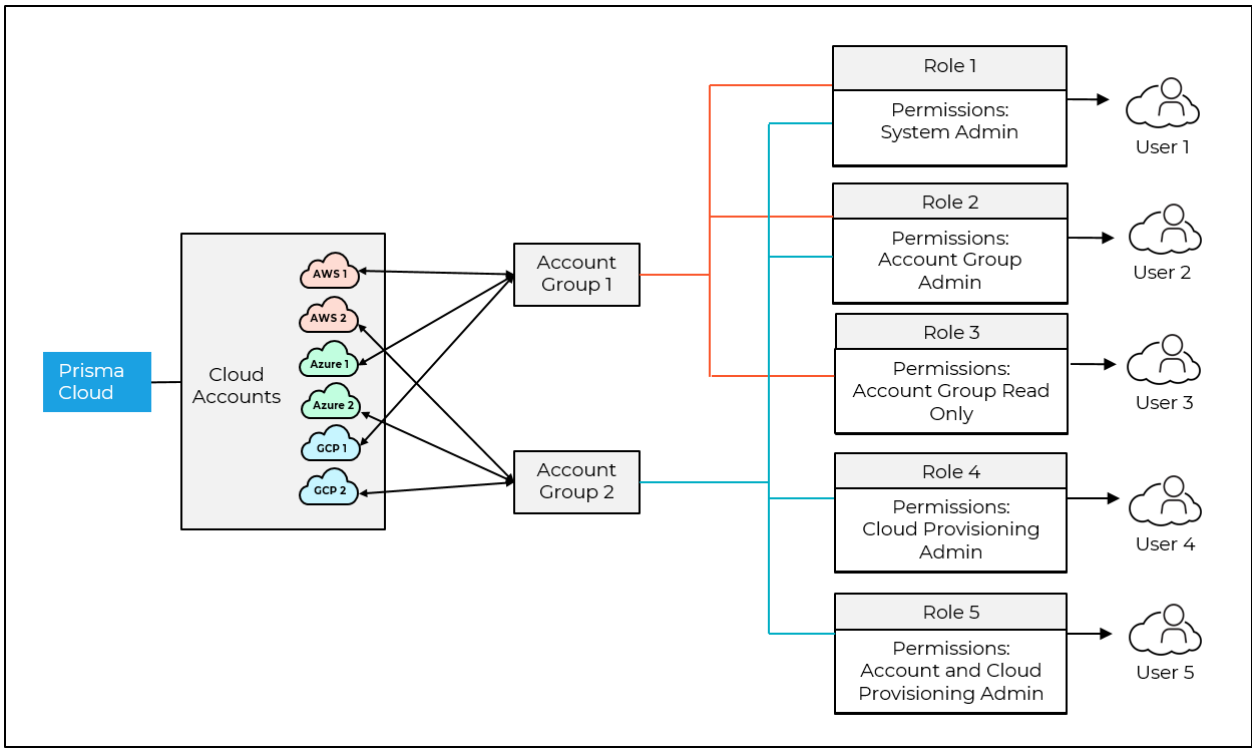


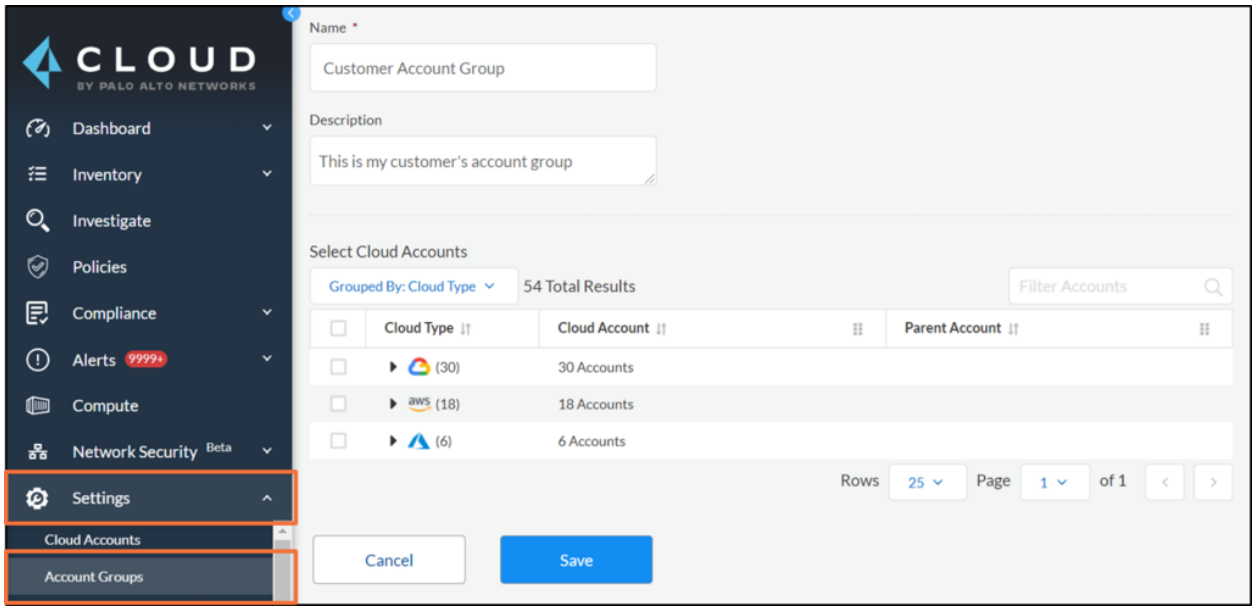
7. Now, go back to the Prisma Cloud onboarding page and supply the service-account JSON credential file to complete the folder level onboarding process.

8. You may check the onboarding status of the GCP folder in the cloud accounts page.

Account Groups

During onboarding, you will also choose the Account Group that you would like the account/hierarchy to belong to. Account Group creation is important to map to your organization's business and security needs. Prisma Cloud by default comes with a "Default Account Group" that contains all of your cloud accounts. It is tied to the "Default Alert Rule", meaning alert me on all enabled policies for all of my cloud accounts. Designating certain accounts into Account Groups such as "AWS Development Accounts", "Compliance Team Accounts", "Production Accounts", etc. will allow you to create meaningful roles with least privilege access, create specific Alert Rules based on Account Groups, Investigate, and filter dashboards/views based on those Account Groups. After specific account groups are created and utilized, the "Default Account Group" may be disabled to reduce noise and incorporate granularity and management of accounts/alerts.





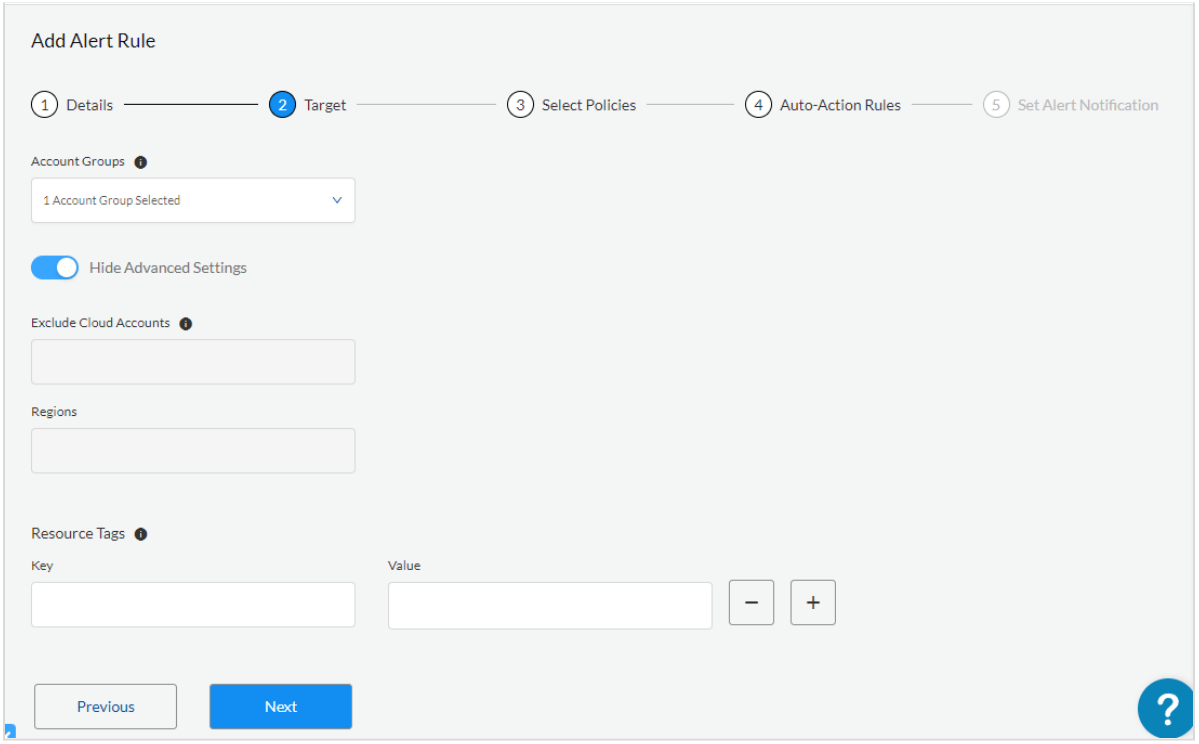
Alert Rules

Prisma Cloud comes with a “Default Alert Rule” as mentioned previously, where the target is the “Default Account Group” or all onboarded cloud accounts and the alerts are for all of the enabled policies within your tenant. Best practice calls for creating specific custom account

groups. Creating specific custom Alert Rules reduces the noise from the Default Alert Rule and allows you to manage your alerts better. Setup Alert Rules based on organization/security use cases and personas as a starting point.

Examples of custom/specific Alert Rules:

- Tagged resources/alerts - IAM Alerts, Security Groups, etc.
- Teams - DevOps (monitoring build policies and IaC), Compliance (monitoring production or critical accounts that are audited)
- Type of account - Production, Testing, Sandbox, Critical



You can filter your Alerts Overview with Alert Rules, allowing you to focus on a specific set at a time. For example, if you have an “AWS Production CIS Compliance” alert rule, you can focus on solely your AWS Production accounts (assuming you’ve created that Account Group) and be alerted on the CIS compliance-related policies within Prisma Cloud.

There are four steps in an Alert Rule configuration:

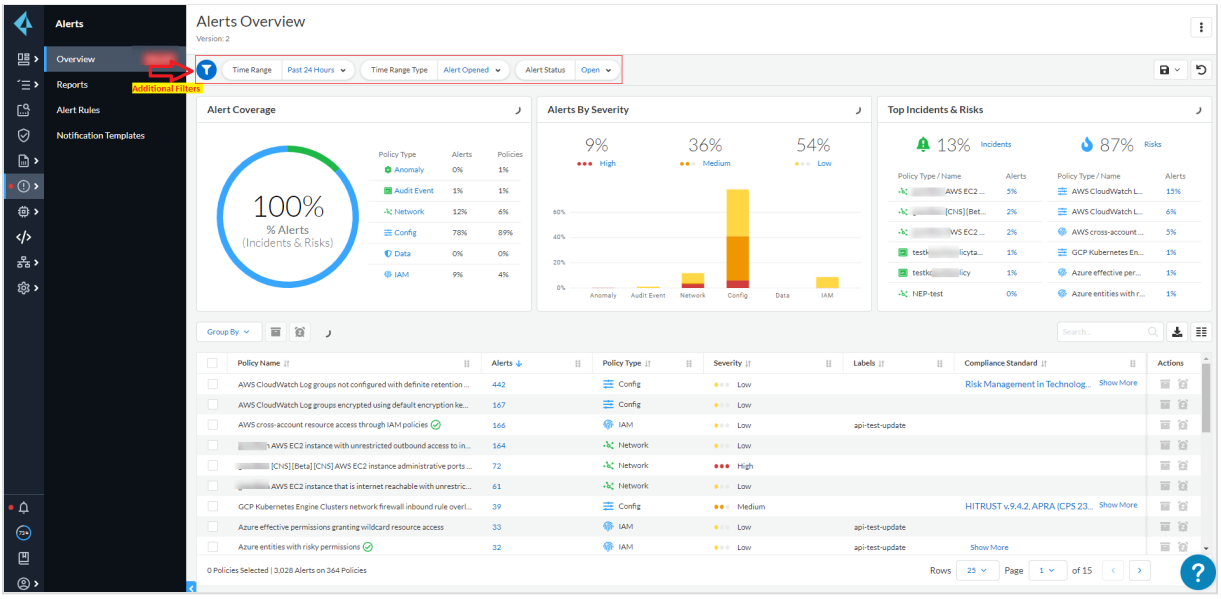
1. Details where you create the name and description
2. Target where you choose the designated Cloud Account Group or individual cloud accounts (utilizing the Advanced Settings to exclude), regions, and resource tags

3. Select Policies where you can filter and choose specific policies to be alerted on, or choose “Select all policies” if you’d like to be alerted on all of them (increases alerts and noise).
4. Alert Notification where you specify how you’d like to be notified on the alerts in this specific Alert Rule. You have the option of third-party integrations, email, or if you do not choose a specific “channel”, it will be an Alert Rule contained just in the console which is still helpful to filter when viewing Alerts and data. Note: New accounts must be updated manually in Alert Rules.

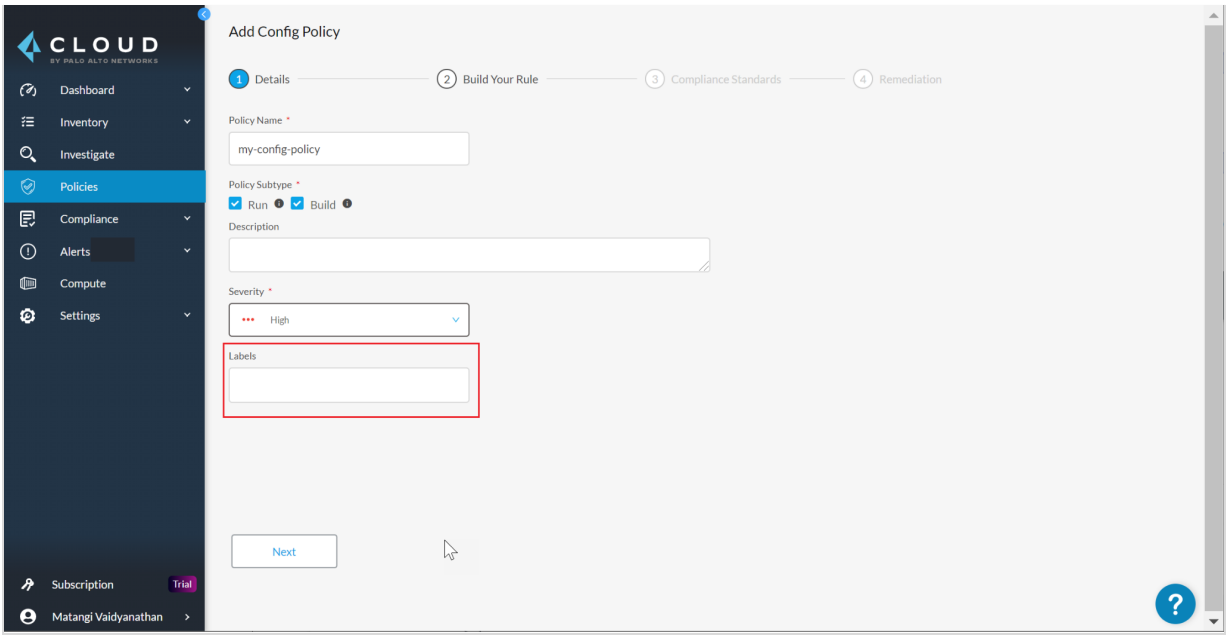
Additional Onboarding/Setup Information

Utilize Filters in *Prisma Cloud Asset Inventory, Policies, Alerts, and Compliance*.

There is typically a lot of data ingested into Prisma Cloud and being able to filter will help you have a more granular view and focus on the relevant information at the time. Understand the use of filters in Policies and Alerts. It helps view more granular/detailed information rather than a flood of Policies and Alerts



Creating meaningful labels for OOTB and custom policies facilitate reviewing data, alerts, and security needs. You will see this option in the first part of a policy, whether it’s a default OOTB or a custom one. See the screenshot below.



Configure

Configure, enable, and customize Prisma Cloud policies. Familiarize yourself with and customize compliance requirements.

Policy walkthrough

Your Prisma Cloud tenant will come with some default out-of-the-box policies already enabled. To view your Prisma Cloud policies, go to your left navigation bar, click on the fourth option which will open your policies. There are hundreds of default policies that apply across multiple cloud providers as well as some that are cloud agnostic. Review the enabled policies as well as the disabled ones to determine if they should be enabled or disabled for your organization. Utilize the filter options mentioned earlier in the Setup/Configuration section of the document to filter by different options such as Cloud Type, Compliance Standard, Severity, and more. There is also a list of recommended policies to enable that are aligned to the CIS Benchmarks for each major cloud provider. See below tables for recommended policies.

Category	AWS Policy	Azure Policy	GCP Policy
Identity			

Management			
	AWS IAM deprecated managed policies in use by User		GCP IAM Service account has admin privileges
	AWS IAM Groups with Administrator Access Permissions		GCP IAM user have overly permissive Cloud KMS roles
	AWS IAM has expired SSL/TLS certificates		GCP IAM user with service account privileges
	AWS IAM password policy allows password reuse		
	AWS IAM password policy does not have a minimum of 14 characters		
	AWS IAM password policy does not have password expiration period		
	AWS IAM Password policy is unsecure		
	AWS IAM policy allows assume role permission across all services		
	AWS IAM policy allows full administrative privileges		
	AWS IAM Roles with Administrator Access Permissions		
	AWS MFA is not enabled on Root account		
	AWS MFA not enabled for IAM users		
	AWS root account configured with Virtual MFA		

Access Management			
	AWS access keys are not rotated for 90 days	Azure Active Directory Guest users found	GCP User managed service account keys are not rotated for

			90 days
	AWS Certificate Manager (ACM) has expired certificates	Azure Custom Role Administering Resource Locks not assigned	GCP VM instance configured with default service account
	AWS Customer Master Key (CMK) rotation is not enabled	Azure subscriptions with custom roles are overly permissive	GCP VM instance using a default service account with full access to all Cloud APIs
	AWS KMS customer managed external key expiring in 30 days or less	SQL servers which do not have Azure Active Directory admin configured	
	AWS KMS Key policy overly permissive	Azure Active Directory Security Defaults is disabled	
	AWS KMS Key scheduled for deletion		
	AWS S3 bucket having policy overly permissive to VPC endpoints		
	AWS SQS queue access policy is overly permissive		
	AWS SNS topic policy overly permissive for publishing		
	AWS SNS topic policy overly permissive for subscription		
	AWS EC2 instance not configured with Instance Metadata Service v2 (IMDSv2)		
	AWS IAM policy is overly permissive to all traffic via condition clause		

Data Protection - Encryption at rest			
	AWS EBS snapshot is not encrypted	Azure Key Vault is not recoverable	GCP GCE Disk snapshot not encrypted with CSEK

	AWS EBS volume region with encryption is disabled	Azure SQL Server advanced data security is disabled	GCP KMS encryption key not rotating in every 90 days
	AWS Elastic File System (EFS) with encryption for data at rest is disabled	SQL databases has encryption disabled	
	AWS RDS DB cluster encryption is disabled		
	AWS RDS DB snapshot is not encrypted		
	AWS RDS instance is not encrypted		
	AWS S3 buckets do not have server side encryption		
	AWS SNS topic with server-side encryption disabled		
	AWS SQS server side encryption not enabled		

Data Protection - Encryption in transit			
	AWS Application Load Balancer (ALB) is not using the latest predefined security policy	Azure ACR HTTPS not enabled for webhook	GCP HTTPS Load balancer is configured with SSL policy having TLS version 1.1 or lower
	AWS CloudFront distribution is using insecure SSL protocols for	Azure App Service Web app doesn't redirect HTTP to HTTPS	

	HTTPS communication		
	AWS CloudFront origin protocol policy does not enforce HTTPS-only	Azure App Service Web app doesn't use latest TLS version	
	AWS CloudFront viewer protocol policy is not configured with HTTPS	Azure Application Gateway allows TLSv1.1 or lower	
	AWS CloudTrail logs are not encrypted using Customer Master Keys (CMKs)	Azure Application gateways listener that allow connection requests over HTTP	
	AWS Elastic Load Balancer (Classic) SSL negotiation policy configured with insecure ciphers	Azure CDN Endpoint Custom domains is not configured with HTTPS	
	AWS Elastic Load Balancer (Classic) SSL negotiation policy configured with vulnerable SSL protocol	Azure CDN Endpoint Custom domains using insecure TLS version	
	AWS Elastic Load Balancer v2 (ELBv2) listener that allow connection requests over HTTP	Azure MySQL Database Server SSL connection is disabled	
	AWS Elastic Load Balancer v2 (ELBv2) SSL negotiation policy configured with weak ciphers	Azure PostgreSQL database server with SSL connection disabled	
	AWS Elastic Load Balancer v2 (ELBv2) with listener TLS/SSL is not configured	Storage Accounts without Secure transfer enabled	
	AWS Elastic Load Balancer with listener TLS/SSL is not configured		
	AWS Network Load Balancer (NLB) is not using the latest predefined security policy		
	AWS S3 bucket not configured with secure data transport policy		
	AWS SNS subscription is not configured with HTTPS		
	AWS SNS topic not configured with secure data transport policy		

Public Exposure			
	AWS Amazon Machine Image (AMI) is publicly accessible	Azure Container registries Public access to All networks is enabled	GCP BigQuery dataset is publicly accessible
	AWS Classic Load Balancer is in		GCP SQL database is assigned

	use for internet-facing applications		with public IP
	AWS CloudTrail bucket is publicly accessible		GCP Storage buckets are publicly accessible to all authenticated users
	AWS EBS Snapshot with access for unmonitored cloud accounts		GCP Storage buckets are publicly accessible to all users
	AWS EBS snapshots are accessible to public		Storage Buckets with publicly accessible Stackdriver logs
	AWS EC2 instance allowing public IP in subnets		
	AWS EC2 instances with Public IP and associated with Security Groups have Internet Access		
	AWS RDS database instance is publicly accessible		
	AWS RDS instance not in private subnet		
	AWS RDS snapshots are accessible to public		
	AWS S3 Bucket Policy allows public access to CloudTrail logs		
	AWS S3 bucket publicly readable		
	AWS S3 bucket publicly writable		
	AWS S3 buckets are accessible to any authenticated user		
	AWS S3 buckets are accessible to public		
	AWS S3 Buckets Block public access setting disabled		
	AWS VPC subnets should not allow automatic public IP assignment		
	AWS SNS topic is exposed to unauthorized access		

Network configuration			
	AWS CloudFront web distribution with AWS Web Application Firewall (AWS WAF) service disabled	Azure Cosmos DB IP range filter not configured	GCP Firewall rule allows all traffic on RDP port (3389)

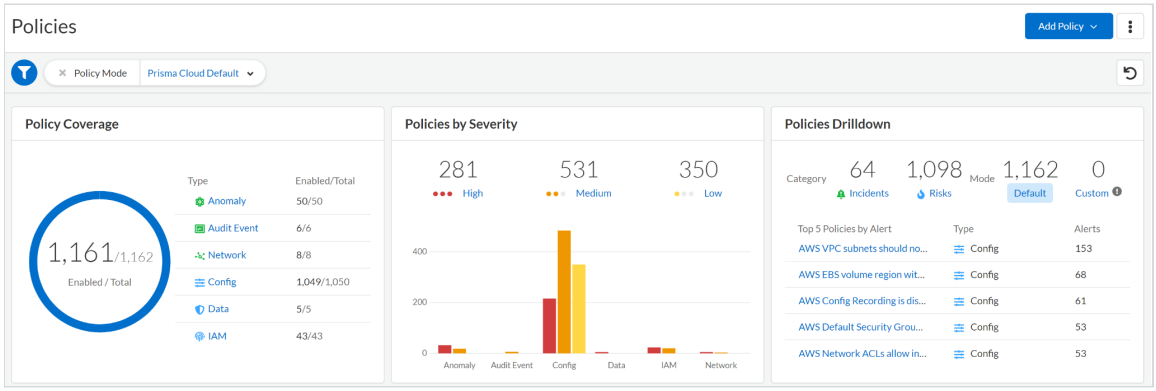
	AWS Default Security Group does not restrict all traffic	Azure Network Security Group allows all traffic on RDP Port 3389	GCP Firewall rule allows all traffic on SSH port (22)
	AWS NAT Gateways are not being utilized for the default route	Azure Network Security Group allows all traffic on SSH port 22	GCP Firewall with Inbound rule overly permissive to All Traffic
	AWS Security Group allows all traffic on RDP port (3389)	Azure Network Security Group having Inbound rule overly permissive to all traffic on any protocol	GCP project is configured with legacy network
	AWS Security Group allows all traffic on SSH port (22)	Azure Network Security Group having Inbound rule overly permissive to all traffic on TCP protocol	GCP VM instances have IP Forwarding enabled
	AWS Security Group Inbound rule overly permissive to all traffic on all protocols (-1)	Azure Network Security Group having Inbound rule overly permissive to all traffic on UDP protocol	GCP VPC Network subnets have Private Google access disabled
	AWS Security Group overly permissive to all traffic	Azure Network Security Group with overly permissive outbound rule	
	AWS VPC allows unauthorized peering	Azure PostgreSQL Database Server 'Allow access to Azure services' enabled	
	Instances exposed to network traffic from the internet	Azure PostgreSQL Database Server Firewall rule allow access to all IPV4 address	
	AWS Application Load Balancer (ALB) not configured with AWS Web Application Firewall v2 (AWS WAFv2)	Azure SQL Servers Firewall rule allow access to all IPV4 address	
		Azure Storage Account default network access is set to 'Allow'	
		Azure storage account has a blob container with public access	
		Azure Virtual Network subnet is not configured with a Network Security Group	
		SQL Server Firewall rules allow access to any Azure internal resources	

Logging			
	AWS Access logging not enabled on S3 buckets	Azure Monitoring log profile is not configured to export	GCP Project audit logging is not configured properly

		activity logs	across all services and all users in a project
	AWS Certificate Manager (ACM) has certificates with Certificate Transparency Logging disabled	Azure Network Watcher Network Security Group (NSG) flow logs retention is less than 90 days	GCP VPC Flow logs for the subnet is set to Off
	AWS CloudFront distribution with access logging disabled	Azure storage account logging for blobs is disabled	
	AWS CloudTrail is not enabled in all regions	Azure storage account logging for queues is disabled	
	AWS CloudTrail logging is disabled	Azure storage account logging for tables is disabled	
	AWS VPC has flow logs disabled		

Others			
	AWS Amazon Machine Image (AMI) infected with mining malware	Azure App Service Web app authentication is off	GCP GCR Container Vulnerability Scanning is disabled
		Azure App Services FTP deployment is All allowed	GCP MySQL instance with local_infile database flag is not disabled
		Azure Application Gateway does not have the Web application firewall (WAF) enabled	GCP SQL database instance is not configured with automated backups
		Azure Security Center Defender set to Off for App Service	GCP VM instance with Shielded VM features disabled
		Azure Security Center Defender set to Off for Azure SQL database servers	VM instances have serial port access enabled
		Azure Security Center Defender set to Off for Key Vault	
		Azure Security Center Defender set to Off for Kubernetes	
		Azure Security Center Defender set to Off for Servers	
		Azure Security Center Defender set to Off for Storage	
		Azure SQL Server ADS	

		Vulnerability Assessment is disabled	
		Threat Detection on SQL databases is set to Off	
		Threat Detection types on SQL databases is misconfigured	

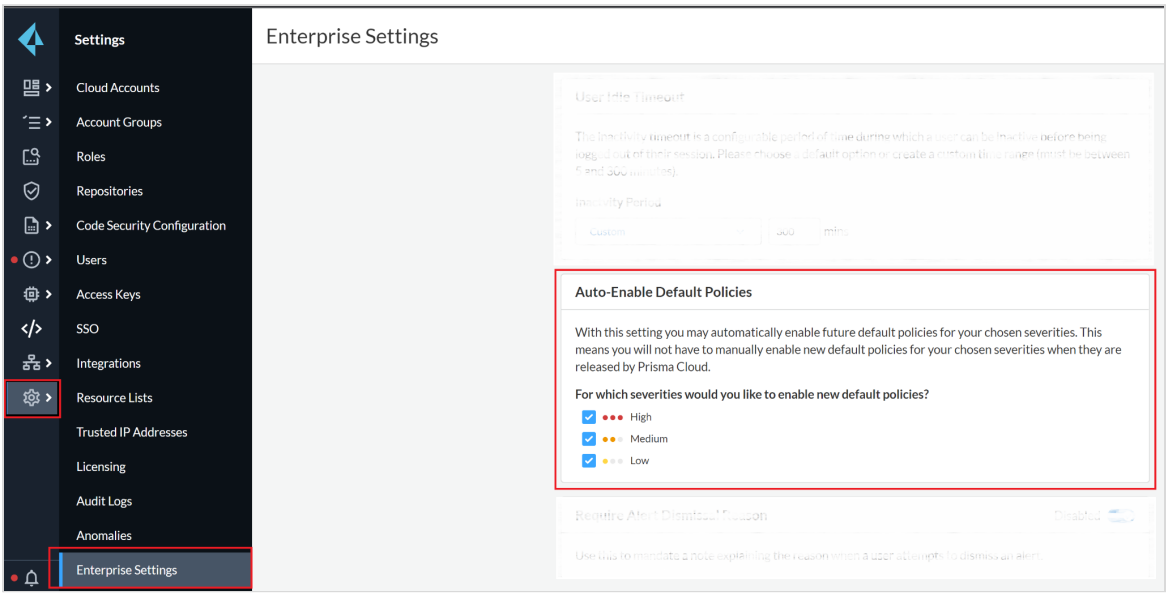


CIS is a Good baseline compliance standards define cloud-specific technical controls and we develop policies per the standard. Other compliance standards map existing policies to high-level requirements.

Policy Types and Classifications

CATEGORY	CLASS	TYPE	SUBTYPE
Incident	Behavioral	Anomaly	UEBA
	Behavioral	Anomaly	Network
	Privileged Activity Monitoring	Audit Event	Audit
	Network Protection	Network	Network Event
Risk	Misconfiguration	Config	Run
	Misconfiguration	Config	Build
	Misconfiguration	Data	Data Classification
	Vulnerability	Data	Malware

In Enterprise Settings (last option in navigation bar, Settings → Enterprise Settings), you can select the option for new default policies to be enabled when they are released with Prisma Cloud updates. You can select the severity of the policies to be enabled, for example, if you'd only like new default high severity policies to be enabled, you can select only the "high" option.



Compliance Standards

1. Setting up custom compliance standards can help users to logically group their RQL policies to fit the scope of what they are trying to achieve. An example of a use case could be to have a compliance standard that is generally used for a specific business unit of a customer environment

Alerts

High Level Steps to Enabling Alerts after Onboarding:

1. Enable Alerts by adding the cloud account to an account group during the onboarding process
2. Create an alert rule for run-time checks that correlates with the account group, selecting the policies you'd like to receive alerts on
3. Verify the alert rule is creating alert notifications in the Alert Overview page

The last step in the Alert Rule window is the Notification. If you don't select a Notification method but still configure the rule, you will see the alerts in the console. If you'd like to receive it via email or send alerts to a third-party tool, you can select it at this stage.

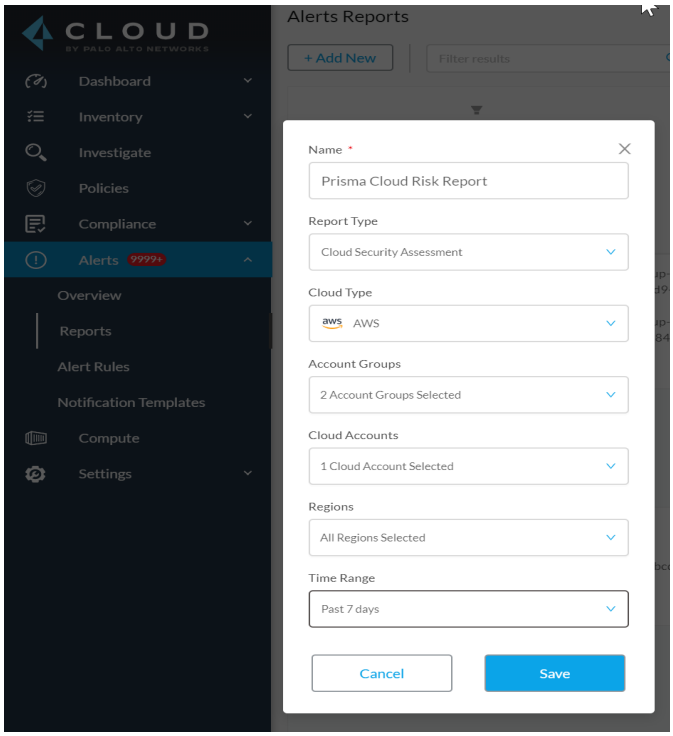
Reporting

There are three different formal reports you can configure from Prisma Cloud; two Alert Reports and one Compliance Report. The two Alert Reports consist of a Cloud Security Assessment Report and a Business Unit Report.

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. This report can be useful to share with management, outside third party organizations for assessment purposes, or just a quick review.

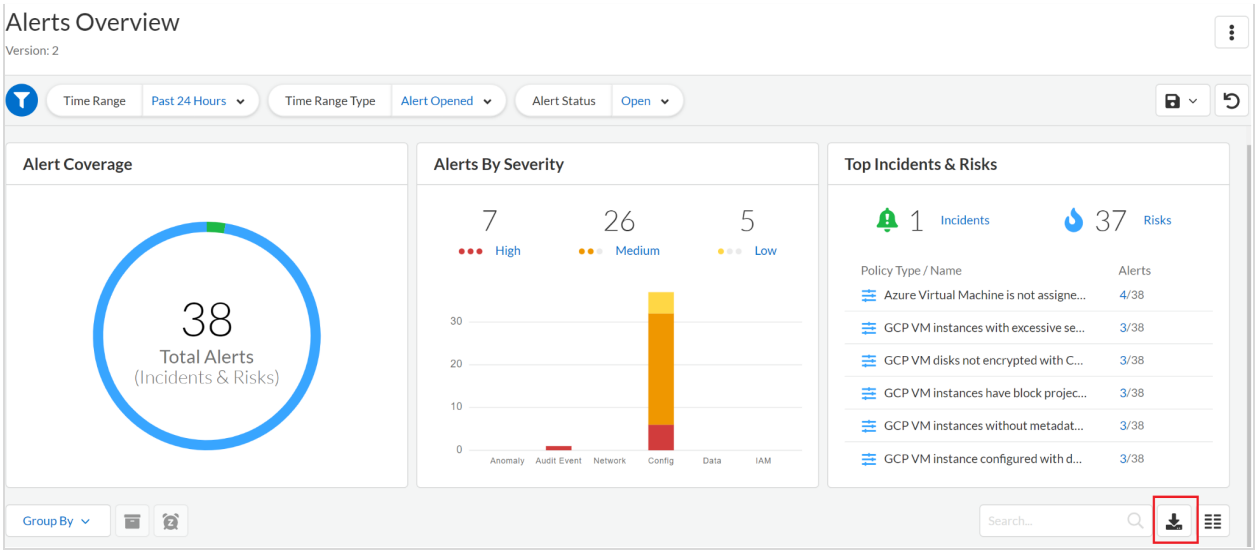
The Business Unit report is a .csv file that includes the total number of resources that have open alerts against policies for any compliance standard, and you can generate the report on-demand or on a recurring schedule. This .csv file allows you to open the report in Microsoft Excel to be able to filter, sort, or utilize any Excel features or you can upload this into a third party tool such as a SIEM tool. 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.

To create an Alert Report, navigate to Alerts → Reports → Add New



Compliance Report: The third type of report that you can generate with Prisma Cloud is a Compliance Report. Prisma Cloud natively includes multiple industry known Compliance Standards such as NIST, CIS, PCI-DSS, HIPAA, and others. 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 of 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. This report can be useful for dedicated Compliance teams, management, or to assist during a security assessment or audit.



Note: You can also download information by clicking on the download option in multiple different views where you see tables of information. Examples include the Asset Inventory page, Alerts Overview, Policies, etc.

Integrate

Configure integrations with third party security tools and SOC workflow tools. Configure alert workflows for notifications and remediation. Organizations already have multiple processes in place when it comes to managing security in the cloud, whether that's a SIEM tool, logging tool, or ticketing system such as ServiceNow.

1. Setup integrations - helps to monitor alerts and send alert notifications to security processes that already exist in an organization or a new process can be created to manage large amounts of alerts/data. Integrations help with the process of keeping Prisma Cloud alerts manageable.
    a. 3rd party integrations (ex. w/Splunk, ServiceNow, other native integrations
    b. Webhooks (non-native integrations such as a SIEM tool)
    c. Discuss Alert Payload info sent to third-party integration - helpful for customers to understand the importance and beneficial to alert remediation/management
2. Create Alert Rules
    a. This is used to generate specific Alerts - specify the target accounts/account groups, the specific policies (or all), as well the notification channel if one is needed

3. Set up alert profiles and integrations to test alert notification functionality with different types of policies that generates alerts

    a. This helps to create a parallel to a DevOps pipeline with alert workflows so that instead of having to test how the alert generates information on the channel that will be used, users can test sending alert information in a standardized way through alert profiles and integrations that mirror the configuration of the production level information streams

Optimize

Ensure end-to-end adoption of the product and full utilization of Custom RQL, Automated Remediation, UEBA, and Compliance Reporting.

Investigate (RQL)

Investigating in Prisma Cloud allows you to see further into security and operational information within your cloud environment(s). Each Prisma Cloud Run policy is built/structured around RQL, or Resource Query Language. RQL is similar to the widely-known SQL and it performs configuration searches into how your cloud resources are deployed. The visibility you gain here saves you time rather than going into your individual cloud provider portal.

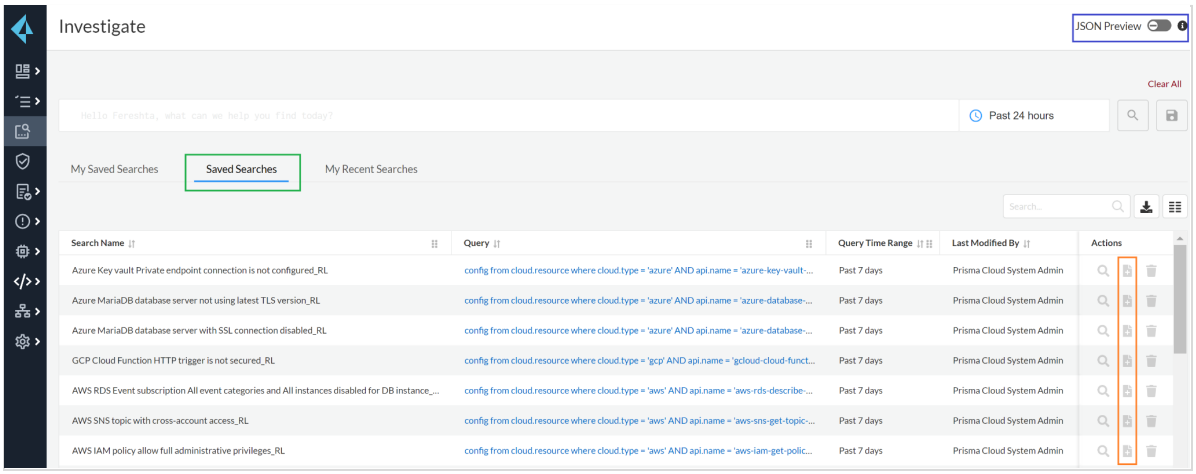
There are multiple types of RQL queries:

1. Config: Config Query to search for the configuration of the cloud resources.
2. Event: Use Event Query to search and audit all the console and API access events in your cloud environment.
3. IAM: Use IAM Query to gain visibility into the permissions of your cloud resources.
4. Network: Use Network Query to search real-time network events in your environment.

Some questions that might come to mind when securing your cloud environment are:

- Do I have any critical S3 buckets that are publicly accessible?
- Are there cloud resources in my environment that are missing critical patches from vulnerabilities?
- What activities has a root user performed that may not have been necessary?

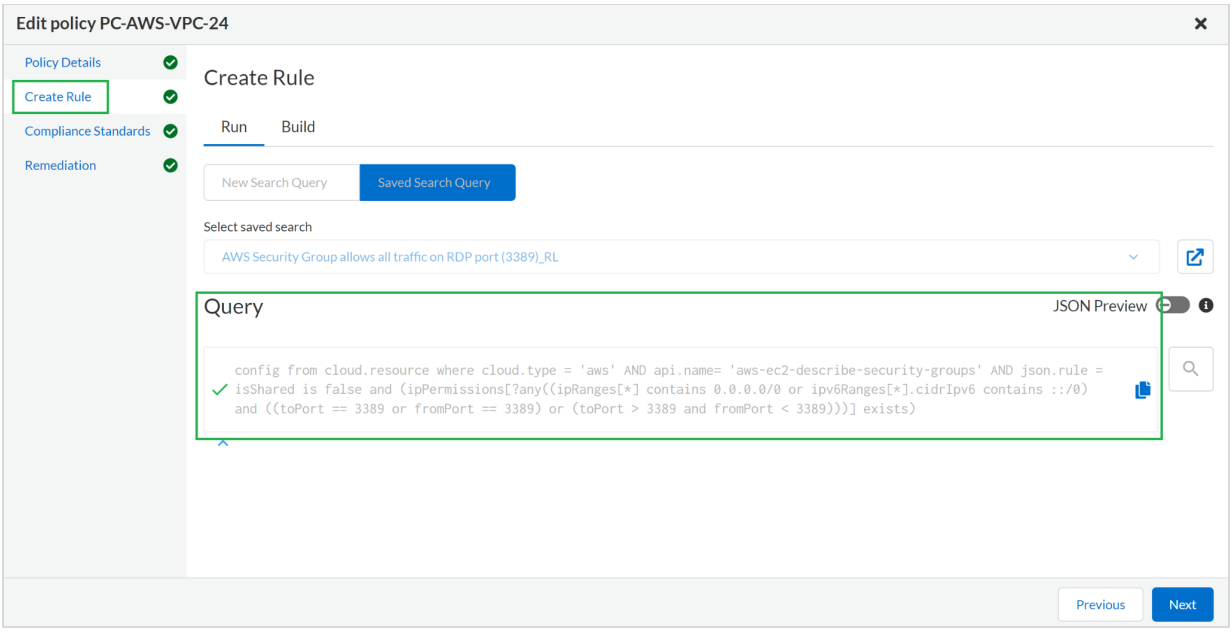
You can create custom RQL queries as well as search into ones that are already structured, such as within a default policy or if you navigate to Investigate → Saved Searches, you will find hundreds of saved searches from not just users within your organization, but also ones that are saved by default within the product that customers and users can find helpful to their organization. See Saved Searches in the following image below in green.



Any helpful Saved Searches can be instantly turned into a policy by selecting the document icon shown in the previous screenshot in orange. RQL queries can be created and start with clicking into the search bar where you're provided drop-down options to build out your query. Selecting the "JSON Preview" option in the top right corner (shown in blue) will show you the drop down menu in the actual .json configuration file view to easily select what object you're looking for. Taking a pre-existing query and modifying it with different objects, operators, and other parameters will allow you to customize your search to exactly what you're looking for.

A query must include an API and you can view if you've built it out correctly when a green checkmark shows at the beginning of your query. A red X signifies that your query is either incomplete or there is a syntax error such as a quotation mark or parentheses missing.

To view the RQL query behind a default policy, navigate to Policies and search for the specific policy you're wanting to look into further. In this example, we'll be looking at "AWS Security Group allows all traffic on RDP port (3389)". Once you find the policy within the policy page, click on the edit icon (indicated by the pencil icon on the far right of the policy). You will see the RQL in the second option of the "Edit Policy" popup, and you can copy the query and go to investigate to search or more easily, just select the blue arrow on the right of the policy/saved search name to open a new window that opens Investigate and populates the query for you.



Examples of Common RQL Searches

Policy Description	RQL
AWS: List EC2 instances with a public IP address	config from cloud.resource where api.name = 'aws-ec2-describe-instances' and json.rule = publicIpAddress exists
Find workloads with vulnerability 'CVE-2015-5600'	network from vpc.flow_record where dest.resource IN (resource where finding.type IN ('Host Vulnerability') AND finding.name = 'CVE-2015-5600') and bytes > 0
Azure SQL instances that allow any IP address to connect to it	config from cloud.resource where cloud.service = 'Azure SQL' AND api.name = 'azure-sql-server-list' AND json.rule = firewallRules[*] contains "0.0.0.0"
List Azure Storage accounts (can be used for Azure flow log checks)	config from cloud.resource where cloud.type = 'azure' AND api.name = 'azure-storage-account-list' addcolumn location
List VPCs that do not have Flow Logs enabled	config from cloud.resource where api.name = 'aws-ec2-describe-vpcs' as X; config from cloud.resource where api.name =

	'aws-ec2-describe-flow-logs' as Y; filter ' not ($.Y.resourceId equals $.X.vpcId)'; show X;

Useful Documentation:

- Review the RQL Reference Guide
- Review the RQL Example Library for useful queries to run and utilize. It allows you to modify easily since you can edit the existing RQLs
- Review the RQL Operators document to understand the different capabilities within RQL searches
- Review the APIs that are ingested for reference to build out custom investigate searches and policies
    - Alibaba APIs Ingested by Prisma Cloud
    - AWS APIs Ingested by Prisma Cloud
    - GCP APIs Ingested by Prisma Cloud
    - Microsoft Azure APIs Ingested by Prisma Cloud
    - OCI APIs Ingested by Prisma Cloud
- Visit the RQL FAQs for additional help and information

Auto remediation

Remediate policies - from Prisma Cloud console can be done manually or by auto-remediation

1. Auto-Remediation requires Read-Write (Monitor and Protect) mode so cloud accounts/hierarchies might need to be updated in terms of onboarding and permissions (review additional permissions required for Read-Write most)
2. Understand how auto-remediation works since it pushes CLI commands automatically once an alert is detected and could possibly create unwanted changes
3. It's helpful to test out auto-remediation in sandbox environments
4. You can view all of the default remediable policies by navigating to Policies and adding the filter "Remediable" and setting it to "True"

The following lists recommend starting policies for Auto Remediation within the three major cloud providers.

AWS

Typically when first starting out with auto-remediation, you will want to focus on low hanging fruit configurations. Down below are 5 AWS policies which we recommend getting started with:

Policy Description	RQL
AWS Security Group allows all traffic on RDP	Used as the remote access port for Microsoft

port (3389)	Windows, it is advised to keep this port open to only trusted IP addresses. This policy checks the configuration of your security groups and ensures that this port is not allowed from any IP address (0.0.0.0/0).
AWS Security Group allows all traffic on SSH port (22)	Most commonly used as the SSH port for Linux, it is highly recommended to lock down access to only trusted IP address ranges. Leaving this port open to 0.0.0.0/0 can expose your instance to brute-force type attacks.
AWS Amazon Machine Image (AMI) is publicly accessible	Unless for very specific use-cases, AMIs should not be made public as they may contain sensitive information. Having a public facing AMI would allow anyone with an AWS account the ability to launch your AMI image.
AWS EBS snapshots are accessible to the public	EBS snapshots are typically used for backups or for security tools. From an EBS snapshot, a volume can be created which can then be attached to an instance, allowing access to the contents of that volume.
AWS CloudTrail logging is disabled	AWS CloudTrail is a service that enables governance, compliance, operational & risk auditing of the AWS account. It is a compliance and security best practice to turn on logging for CloudTrail across different regions to get a complete audit trail of activities across various services.

Azure

Policy Description	RQL
Azure Network Security Group allows all traffic on SSH port 22	As a best practice, restrict SSH solely to known static IP addresses. Limit the access list to include known hosts, services, or specific employees only. This policy will remove the entry for port 22 which allows access from anywhere from your NSG.
Azure Network Security Group allows all	As a best practice, restrict RDP solely to

traffic on RDP Port 3389	known static IP addresses. Limit the access list to include known hosts, services, or specific employees only. This policy will remove the entry for port 3389 which allows access from anywhere from your NSG.
SQL databases have encryption disabled	Transparent data encryption protects Azure database against malicious activity. It performs real-time encryption and decryption of the database, related reinforcements, and exchange log records without requiring any changes to the application. This policy will automatically enable encryption on the databases which have this disabled.
Azure Key Vault is not recoverable	The key vault contains object keys, secrets and certificates. Accidental unavailability of a key vault can cause immediate data loss or loss of security functions (authentication, validation, verification, non-repudiation, etc.) supported by the key vault objects.
	It is recommended the key vault be made recoverable by enabling the "Do Not Purge" and "Soft Delete" functions. This policy enables the soft delete functionality.
	This policy identifies the Azure Security Center policies which have automatic provisioning of monitoring agents and is set to Off. When Automatic provisioning of monitoring agent is turned on, Azure Security Center provisions the Microsoft Monitoring Agent on all existing supported Azure virtual machines and any new ones that are created. This auto-remediation policy can automatically enable this feature for you.

GCP

Policy Description	RQL
GCP Firewall rule allows all traffic on SSH port (22)	Allowing access from arbitrary IP addresses to this port increases the attack surface of

	your network. It is recommended that the SSH port (22) should be allowed to specific IP addresses. This policy can remove the entry exposing port 22 to 0.0.0.0/0 from your firewall rules.
GCP Firewall rule allows all traffic on RDP port (3389)	Allowing access from arbitrary IP addresses to this port increases the attack surface of your network. It is recommended that the RDP port (3389) should be allowed to specific IP addresses. This policy can remove the entry exposing port 3389 to 0.0.0.0/0 from your firewall rules.
GCP Firewall rule allows all traffic on SMTP port (25)	This policy identifies GCP Firewall rules which allow all inbound traffic on SMTP port (25). Allowing access from arbitrary IP addresses to this port increases the attack surface of your network. This policy can remove the entry exposing port 25 to 0.0.0.0/0 from your firewall rules.
GCP Firewall rule logging disabled	This policy identifies GCP firewall rules that are not configured with firewall rule logging. Firewall Rules Logging lets you audit, verify, and analyze the effects of your firewall rules. When you enable logging for a firewall rule, Google Cloud creates an entry called a connection record each time the rule allows or denies traffic. This policy can automatically enable this functionality on the firewall rules.
GCP Storage log buckets have object versioning disabled	This policy identifies Storage log buckets which have object versioning disabled. Enabling object versioning on storage log buckets will protect your cloud storage data from being overwritten or accidentally deleted. This policy can enable object versioning features on all storage buckets where sinks are configured.

Some best practices to keep in mind here:

1. Create custom compliance frameworks if needed for specific organizational needs
2. Enable disposition of new default policies that are added with Prisma Cloud product updates by reviewing Enterprise Setting option
3. Setup Trusted IPs and Prisma Cloud Login IPs to reduce false positive alerts
4. Once specific Alert Rules are created that cover all cloud assets scope, disable Default Alert Rule to reduce noise and add granularity
5. Create and manage access keys as needed for certain cloud tools and third party integrations
6. Policies
    a. What policies are recommended to start with for remediating for customers - reference Copy of Alert_Burn_down (needs updating) ask
        i. Steps on Alert Burndown

            1. Understand what alert burndown means - bringing down the number of alerts so it's manageable (important alerts being looked at and resolved regularly)
            2. Create a plan
                a. Focus on what's important to the organization (high volume alerts or high severity alerts)
            3. Understand alert actions
                a. Dismiss, snooze, remediate, investigate
            4. Effort vs. result
                a. Low effort in lowering many alerts (ex. Audit events)
                b. High volume alerts (think of severity, impact from remediation, complexity of remediation)
                    i. Ex policies:
                        1. Config : AWS Security groups allow internet traffic
                        2. Audit Event : IAM configuration updates
                        3. Anomaly : Unusual user activity
                        4. Network : Internet exposed instances

CSPM Security Modules

IAM Security

The IAM Security Module provides net-effective permissions to cloud infrastructure resources based on policies that are configured out of the box. This allows Prisma Cloud administrators the ability to rightsize these permissions quickly, which reduces the risk of compromise from identity credentials.

The IAM Security module is enabled on the Subscriptions tab in the Prisma Cloud console (click on Learn Mode, and Activate to enable). Once IAM Security has been activated, you will be able to run RQL queries. Verify this on the Investigate tab.

Be sure to save searches that you have created to save time the next time you need to run the query again. If you need to analyze permissions offline you can download the results of a query in CSV format.

Integrate IAM Security with your IdP to calculate permissions for your SSO provider (e.g. Okta, Azure AD).

Manually remediate IAM security alerts by going to:

1. Alerts > Overview
2. Select the violating policy
3. Policies that can be remediated are indicated by a icon.
4. Under the Options column, click the Remediate button.
5. Prisma Cloud will make recommendations for CLI commands to run in your CSP.

Create an Alert Rule for Run-Time Checks and follow the instructions for configuring a custom python script on AWS or Azure; this will allow you to manage auto-remediation for IAM alert rules using the messaging queuing service on the respective CSP.

Data Security

Some best practice guidelines to consider for IAM and Data Security

IAM Security

- Which users have access to resource X?
- What accounts, services and resources does the user name@domain.com have access to?
- What are the cross account permissions between my accounts?
- Can any users outside of group C access resources in region D?
- What roles are not configured according to best practices?
- What resources can be effectively assessed by the public?
- Which compute workloads have permissions that are not actually being used?
- Resolve all over-permissive policies and enforce least-privilege from now on

Data Security

Data security is both the practice and the technology of protecting valuable and sensitive company and customer data, such as personal or financial information. Data security is a

company's protective measures put in place to keep any unauthorized access out of their databases, websites, applications, and computers.

- Create meaningful roles with Permissions to Account Groups.
- Create meaningful labels for OOB policies and custom policies. Use labels in alert rules and report generation.
- If you are using AWS or GCP Orgs, use Prisma Cloud Org support to automatically on-board all the member accounts, folders, projects (Azure mgmt groups - BETA).
- Use API for all bulk operations, such as cloud onboarding, account group management, custom dashboards etc. Review and use Prisma Cloud Terraform provider or Python scripts for leveraging the Prisma Cloud API.
- Create and manage access keys
- Create custom Alert Rules to send alerts only to cloud-account owners. And, then turn off default alert rules.
- Set up integrations with Splunk, ServiceNow, other SIEM/SOAR tools.
- Review cloud accounts in orange/red status to ensure permissions and settings are correct. Goal is for accounts to be in green status.
- Use tags to group your cloud resources, accounts etc.
- Use filters to focus only on specific cloud accounts, alert types, cloud types etc.
- Consider disabling the default alert rule and creating ones with the specific policies that you care about or that you have runbooks setup for so you don't get overwhelmed by alert fatigue or need dedicated alert burndown sessions.
- Consider setting up integrations sooner rather than later as some integrations only get net-new alerts and thus older alerts won't be sent to the integration if you wait to set them up further down the onboarding path.
- When adding new cloud account groups, make sure to include them in alert rules (not automatic)

New Updates and Feature Releases

- Permissions/new APIs being added, edit Prisma Cloud policy in AWS for example to fix config issues
- Prisma Cloud New Features

Resources and References
Prisma Cloud: Monthly Product Overview Webinar Recordings

PRISMA®
BY PALO ALTO NETWORKS

CWPP

Architecture

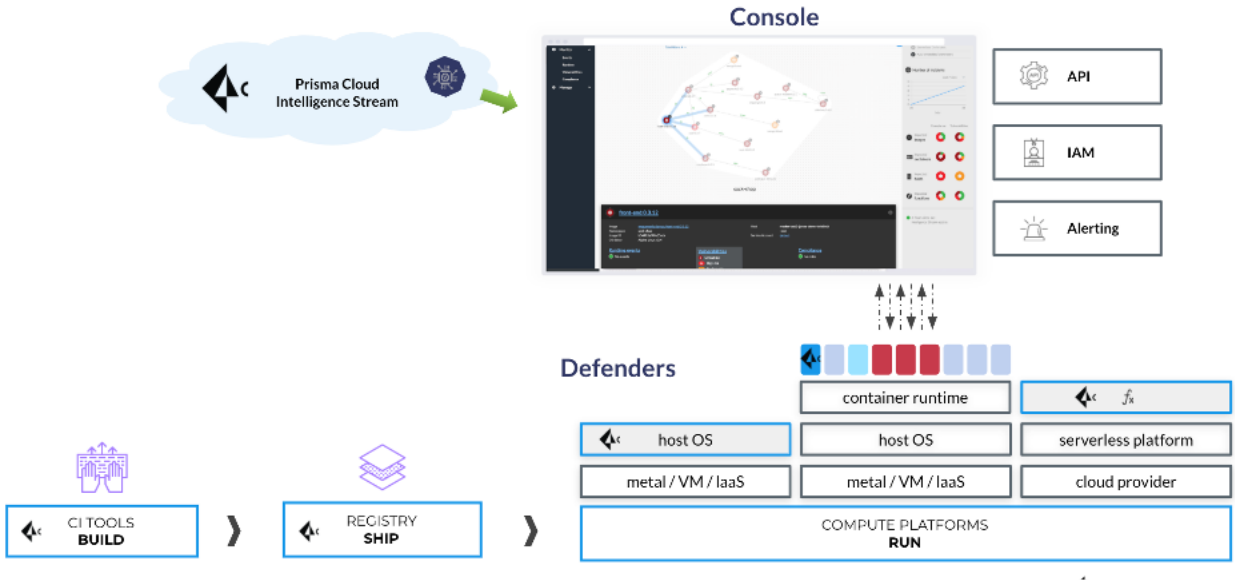
Components of Prisma Cloud Compute

The Console runs as a container within the customer's runtime environment docker, CRI-O and cri-containerd runtimes are supported. There are two operating options:

- Self hosted in which the customer runs the Console within their environment.
- Enterprise edition in which the customer's Console is operated within the Palo Alto Network's Prisma Cloud Google Cloud's GKE cluster. The architectural design can be found here.

The Console's user interface is a java application that the operator's browsers uses to communicate with the Console's API over a TLS v1.2 protected session (default TCP 8083). Due to the increasing size of the API endpoints only a subset of supported endpoints are documented. Every release's tar file contains the openapi.json file which lists all API endpoints and their methods.

Defender runs as a container (there are other Defenders types dependent upon workload) within the customer's environment and monitors the environment in which they are running. The Defender utilizes the shared kernel capabilities that are inherent to containerization. An explanation of the Defender's kernel requirements can be found here. The Defender is a "lights-out" container and receives its command and control from the Console that has deployed the Defender. When a Console is deployed for the first time it will generate a unique set of x.509 certificates and keys that the Console and Defender(s) will use to establish a secure websocket session (default TCP 8084) to exchange data upon.



NAT Configuration (SaaS)

In order to ensure there is connectivity between Prisma Cloud Defenders and the Prisma Cloud Console, NAT addresses must be configured on any firewalls that are in the network path between them. This would also include any on premises security tooling being used as part of the incident Response workflow (e.g. SIEM, SOAR..etc).

The NAT address used will depend on the region that your Prisma Cloud instance is hosted in. Your region can be found by looking at the URL when you log in. For example: https://app2.prismacloud.io/ The "app2" in address indicates your Prisma Cloud tenant region.

A list of NAT addresses that map to these regions are located at this link: Prisma Cloud Regional NAT Address List.

Setup

Familiarize yourself with the application. Inventory cloud accounts need to be secured, and users need access to the platform and their functions. Onboard them to the Prisma Cloud platform.

Console Setup

The Console must be deployed first. The initiation of the Console will ensure that only the Defenders that a Console deploys will only be controlled by the Console. Automation (e.g. ansible, Operators, etc.) can be used to deploy Compute. The Console must be running and

licensed before any further configuration can be implemented (e.g. deploy Defenders). The most common methods of deployment are:

- Onebox - simple single docker host deployment. The twistlock.sh (*$ twistlock.sh -sy onebox*) bash script that is included within the release tar will deploy a Console and Defender on the node. Note: Onebox will only deploy the Console on a RHEL node running podman.
- Kubernetes - Most favors of K8s are supported. Every release is tested on several versions of K8s and they are listed here.
- OpenShift - basically it is Kubernetes but there are some slight nuances (e.g. external router, OpenShift internal registry, <=v3.11 - docker, >=v4.0 - cri-o)

Structure for CI/CD pipeline

integrate Prisma cloud scanning in the pipelines) for these technologies:

- Azure devops
- Gitlab CI/CD
- Jenkins
- Any other pipeline that we don't integrate into (Twistcli)

Best Practices for “Other” Pipeline Integration

For CI/CD pipeline tools that we do not natively support an integration with, TwistCLI can be used instead and should be embedded in the pipeline at a stage that is before the container/image is deployed. This allows for the container/image to be scanned for vulnerabilities and blocked as needed per the CI rules in Prisma Cloud Compute.

The following Code Block below was pulled from a Gitlab pipeline task and demonstrates how to pull down and configure TwistCLI on the pipeline job runner agent, run the image scans using credentials and URL pertinent to the customer's tenant, and publish the results both in the pipeline output and to the console.

```
prisma-cloud-compute-scan:
  stage: build
  variables:
    prisma_cloud_compute_url: ""
    prisma_cloud_compute_username: ""
    prisma_cloud_compute_password: ""
    prisma_cloud_scan_image: node:lts-alpine
  before_script:
    - apk update && apk add --no-cache docker-cli
    - docker version
    - apk --no-cache add curl
    - apk add --no-cache --upgrade bash
    - |
```

```
      if ! /tmp/twistcli --version 2> /dev/null; then
        echo "Download twistcli binary file ..."
        curl -k -u
${prisma_cloud_compute_username}:${prisma_cloud_compute_password} \
          --output /tmp/twistcli
${prisma_cloud_compute_url}/api/v1/util/twistcli
        chmod +x /tmp/twistcli
      fi
      /tmp/twistcli --version
    - |
      echo "Create image scan helper script image_scan.sh ..."
      cat > ./image_scan.sh << EOF
      #!/bin/bash
      set +e
      /tmp/twistcli images scan --details --address \$prisma_cloud_compute_url
\
        --user=\$prisma_cloud_compute_username
--password=\$prisma_cloud_compute_password \
        --output-file twistcli.json \$prisma_cloud_scan_image
      rc=\$?
      if [ -f twistcli.json ]; then
        mkdir -p report/image_scan
        touch report/image_scan/results.xml
        docker run --rm \
          -v \$PWD/twistcli.json:/tmp/twistcli.json \
          -v \$PWD/report/image_scan/results.xml:/tmp/results.xml \
          redlock/pcs-sl-scanner pcs_compute_junit_report
      fi
      exit \$rc
      EOF
      chmod +x ./image_scan.sh
  script:
    # if script is defined in extended job, make sure below command is added
    -  bash ./image_scan.sh
  artifacts:
    when : always
    paths:
      - report/image_scan/results.xml
    reports:
      junit:
        - report/image_scan/results.xml
  tags:
    - shell
```

Defender deployment strategy

Automated deployment of the Defender Agent

Automated Defender Agents are possible with these technologies

- Everything (serveless, server, host..)
- Openshift
- K8's
  Terraform and Kubernetes:
  - https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/daemonset
  - Depending on your use case you can couple that daemonset deployment in Terraform with this API call: https://prisma.pan.dev/api/cloud/cwpp/defenders#operation/post-defenders-daemonset.yaml

- Defender baked in AMI
  - Adding curl command into EC2 Image Builder to install a defender in the AMI (checking EC2 Image Builder API also) (example: curl -sSL -k --header "authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYXZraW5nX3BhbG9hbHRvbmV0d29ya3NfY29tIiwicm9sZSI6ImFkbWluIiwiZ3JvdXBzIjpbImFkbWlucyIsImRldm9wcyJdLCJyb2xlUGVybXMiOltbMjU1LDI1NSwyNTUsMjU1LDI1NSwxMjcsMV0sWzI1NSwyNTUsMjU1LDI1NSwyNTUsMTI3LDFdXSwicGVybWlzc2lvbnMiOlt7InByb2plY3QiOiJDZW50cmFsIENvbnNvbGUifV0sInNlc3Npb25UaW1lb3V0U2VjIjo4NjQwMCwiZXhwIjoxNjQ1MTMzMjIzLCJpc3MiOiJ0d2lzdGxvY2sifQ.86IPNZwEHmIvGenl8SZR1wyIi85g1xa1ZAbjVvfbsbc" -X POST https://127.0.0.1:8083/api/v1/scripts/defender.sh | sudo bash -s -- -c "127.0.0.1" -d "none" -m)
- Fargate
  - https://prisma.pan.dev/api/cloud/cwpp/defenders#operation/post-defenders-fargate.json
- Detailed doc / github for twistcli for deployment of images
  - Code Repo - github (reference links)
    - https://github.com/PaloAltoNetworks/prisma-cloud-compute-operator
    - https://github.com/PaloAltoNetworks/terraform-provider-prismacloudcompute
    - https://github.com/twistlock/sample-code/tree/master/automated-deployments
    - https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin-compute/vulnerability_management/code_repo_scanning.html

Upgrade and Redeployment

Upgrade the console (Self-Hosted)

You should have kept good notes when initially installing Prisma Cloud. The configuration options set in twistlock.cfg and the parameters passed to twistcli in the initial install are used to generate working configurations for the upgrade.

Prerequisites: Document and save all options set in twistcli.cfg and parameters passed to twistcli during the install.

The console upgrade is the one way to upgrade the higher version, meaning you cannot downgrade to the current version. Due to the restore only support to the same version, it's strongly recommended to test the upgrade with the dev or secondary console first and validate the all utilized functionality.

To safely upgrade the console, it is required to keep the "Default - ignore Twistlock components", in the vulnerability to defend the policy. If this rule is disabled or deleted, there is a chance that an upgrade will fail.

Upgrade/Redeploy the defenders

After the console upgrade, either SaaS or Self-hosted, the defenders need to be upgraded by the customer.

It is recommended to automate the defender deployment process. With automation, the defender upgrade will be much smoother in a large-scale environment.

If the customer has an automation process for the container deployment, the customer should integrate the defender deployment process into the existing pipeline process. Defender deployment can be automated with API calls (defender yaml, helm chart, or fargate defender) as referenced.

Kubernetes Operator, another method to automate the defender deployment, can be considered to deploy and upgrade defenders.

App-Embedded Defenders and serverless defenders need to be upgraded manually. These defenders are also recommended to integrate to the pipeline of task deployment or serveless script deployment process.

Backup and Restore (Self-Hosted)

Prisma Cloud automatically backs up all data and configuration files periodically. You can view all backups, make new backups, and restore specific backups from the Console UI. You can also restore specific backups using the twistcli command line utility. You can also manually backup any point of the time from the console.

Prisma Cloud is implemented with containers that cleanly separate the application from its state and configuration data. To back up a Prisma Cloud installation, only the files in the data directory need to be archived. Because Prisma Cloud containers read their state from the files in the data directory, Prisma Cloud containers do not need to be backed up, and they can be installed and restarted from scratch.

You can only restore Console from a backup file whose version exactly matches the current running version of Console. If the console is unresponsive, you can use twistcli to restore the console.

Supported life cycle for connected components

Any supported version of Defender, twistcli, and the Jenkins plugin can connect to Console. Prisma Cloud supports the latest release and the previous two releases (n, n-1, and n-2).

There are some exceptions to this policy as we roll out this new capability.

For Defenders:

- 21.08 supports n and n-1 (21.04) only.
- Starting with the next release (Joule), there will be full support for n, n-1, and n-2.

For twistcli and the Jenkins plugin:

- 21.08 supports itself (n) only.
- In the next release (Joule), Console will support n and n-1.
    - In release after Joule (Kepler), Console will support n, n-1, n-2.

Configure

Configure, enable, and customize Prisma Cloud policies. Familiarize yourself with and customize compliance requirements.

Runtime Models

One key goal is minimizing the amount of work you're required to do to manage runtime defense. Leverage the models that Prisma Cloud can automatically create and manage. Because behavioral learning for model creation is mature technology for Prisma Cloud, in most cases, you won't need to create auxiliary rules to augment model behavior. There will be some exceptions. For example, a long-running container that changes its behavior throughout its lifecycle might need some manually created rules to fully capture all valid behaviors. This is atypical for most environments, however, as containers that need to be upgraded are typically destroyed and reprovisioned with new images.

If you do need to create runtime rules, here are some best practices for doing so:

Minimize the number of rules—Creating static rules requires time and effort to build and maintain; only create rules where necessary and allow the autonomous models to provide most of the the protection.

Precisely target rules—Be cautious of creating rules that apply to broad sets of images or containers. Providing wide ranging runtime exceptions can lower your overall security by making rules too permissive. Instead, target only the specific containers and images necessary. Don't use wildcard (*) in the whitelist or blacklist because it can interrupt the execution of legitimate services.

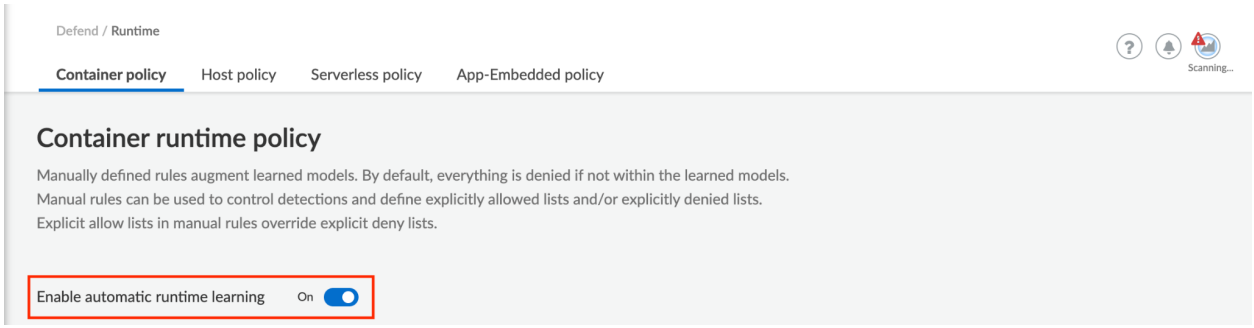
Name rules consistently—Because rule names are used in audit events, choose consistent, descriptive names for any rules you create. This simplifies incident response and investigation. Also, consider using Prisma Cloud's alert profile feature to alert specific teams to specific types of events that are detected.

Rules in alert action —It is recommended to start configuring the runtime rules in alert action and after you are comfortable with the outputs you can change the action to prevent or block .

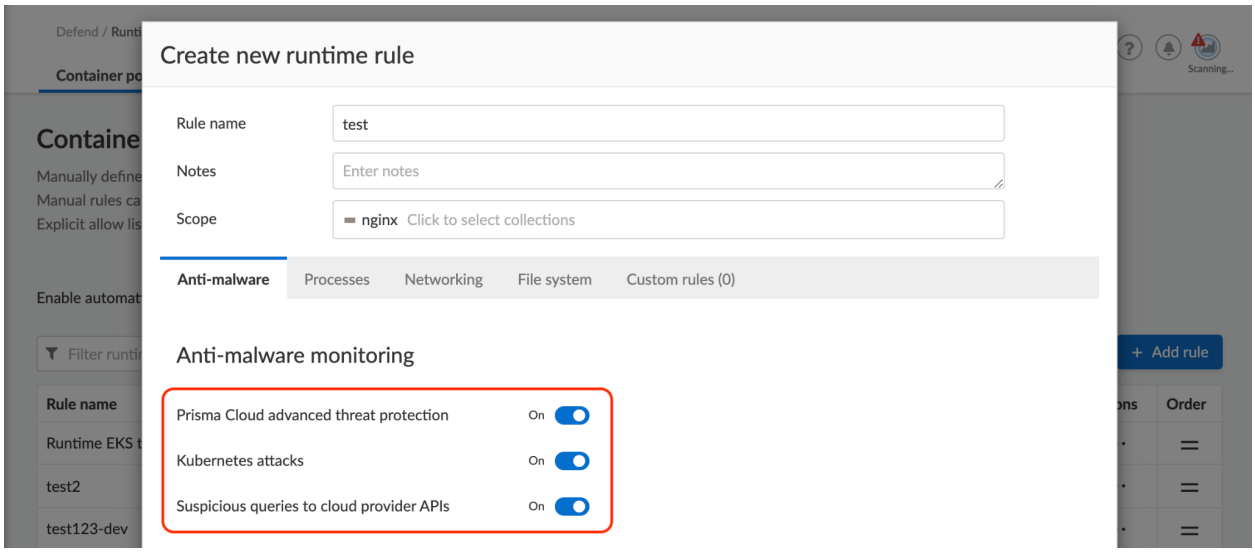
Rules testing —In case the customer wants to implement a runtime policy with block or prevent actions. It is recommended to test this policy behavior in a test environment before pushing it to production. For example testing Kubernetes cluster or test namespace.

Runtime Policy Configuration

- Enable the use of ML models in case of container policy by enabling the automatic runtime learning option.



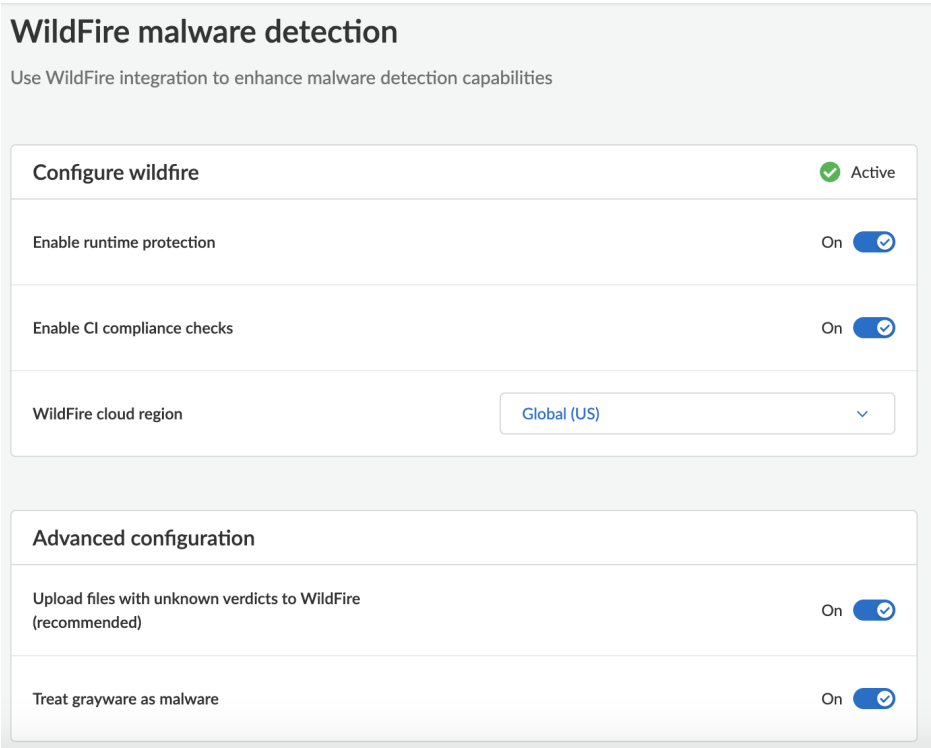
- Provide a descriptive rule name.
- Avoid using a wide scope based on a cluster name for example and make it more specific by providing a namespace name and image or container value.
- Use **Prisma Cloud Advanced Threat Protection** intelligence feed, to apply malware prevention techniques across processes, networking and filesystem.
- In case the container is running inside a Kubernetes cluster it is better to enable the **Kubernetes attack** option to monitor attempts to directly access Kubernetes infrastructure from within a running container. In case a container is developed for some use case to communicate with Kubernetes API, it needs to be excluded from the selected scope to avoid false positive alarms.
- **Suspicious queries to cloud provider APIs** can be enabled to monitor access to cloud provider metadata API from within a running container. In case a container is developed for some use case to communicate with a cloud provider API, it needs to be excluded from the selected scope to avoid false positive alarms.



- Use **Advanced Malware Analysis** based on Wildfire malware analysis engine, to detect malware. Currently Prisma Cloud Compute uses WildFire for file verdicts only in the following scenarios for Container runtime / CI:
    - ELF files written to a linux container file system in runtime.
    - Shared objects are not examined via WildFire.
    - File must be smaller than 100MB (WildFire limit).
    - You can submit up to 5000 files per day, and get up to 50,000 verdicts on your submissions to the WildFire service.
    - Wildfire is supported on Linux only. Windows containers and hosts aren't currently supported.

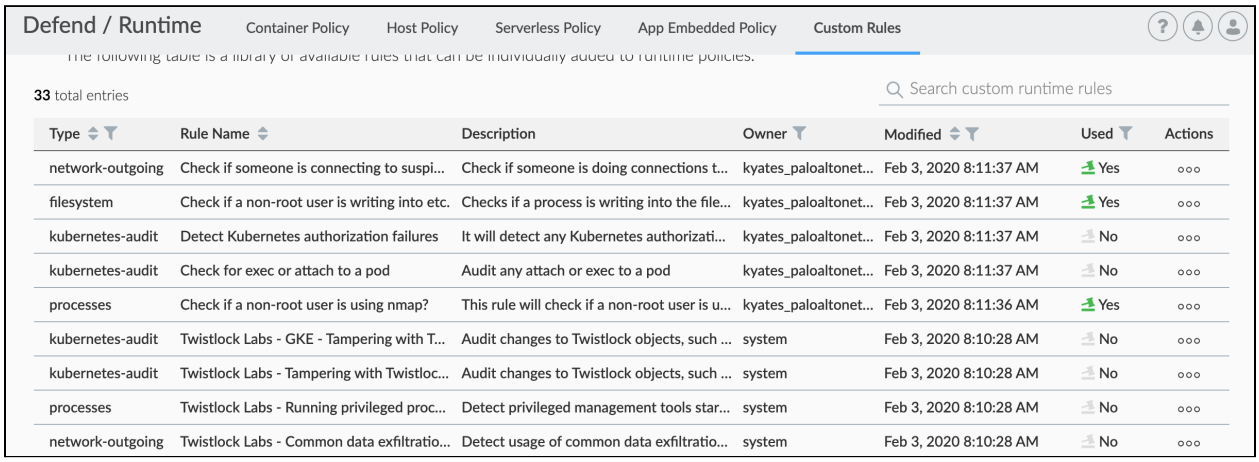
WildFire malware detection

- **Use WildFire for runtime protection**—Enable WildFire malware scanning in runtime for containers and hosts.
- **Use WildFire for CI compliance checks**—Enable WildFire malware scanning for containers CI checks.
- Choose the closest WildFire cloud region
- **Upload files with unknown verdicts to WildFire**—Determines whether files with unknown verdict will be sent to WildFire for full analysis. When off, WildFire will only provide verdict for files that have been uploaded to WildFire via a different client.
- **Treat grayware as malware**—Use a more restrictive approach and treat files with grayware verdict as malware.



- Processes:
  - Review the learned processes in the container model and whitelist or blacklist the process in the rule based on the business need.
  - Configure the Anti-malware and exploit prevention option on alert mode for testing and then change to prevent or block mode.
- Networking:
  - Review the learned networking ports and domains in the container model and whitelist or blacklist it in the rule based on the business need.
  - Configure the Anti-malware and exploit prevention option on alert mode for testing and then change to prevent or block mode
- File System:
  - Review the learned file system paths in the container model and whitelist or blacklist it in the rule based on the business need.
  - Configure the Anti-malware and exploit prevention option on alert mode for testing and then change to prevent or block mode

- Custom Runtime Rules:
  - Precise way to describe and detect specific runtime behaviors.
  - Can help fill in a lot of gaps on hosts since our model is more focused on services
    - Example: Preventing writes to a particular file system on a host.
  - Make sure to test specific use cases before telling customers what you can and can't do.
  - Sample built-in runtime rules are usually good enough in POC.

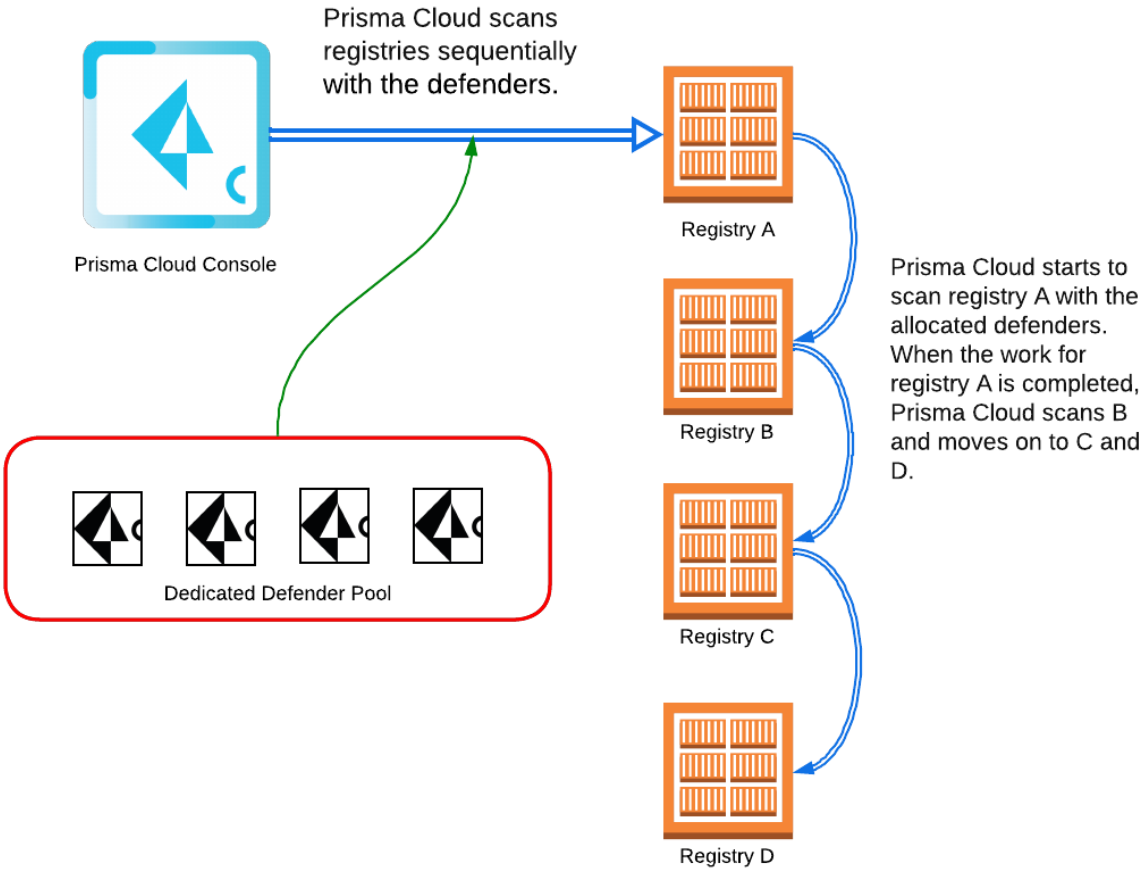


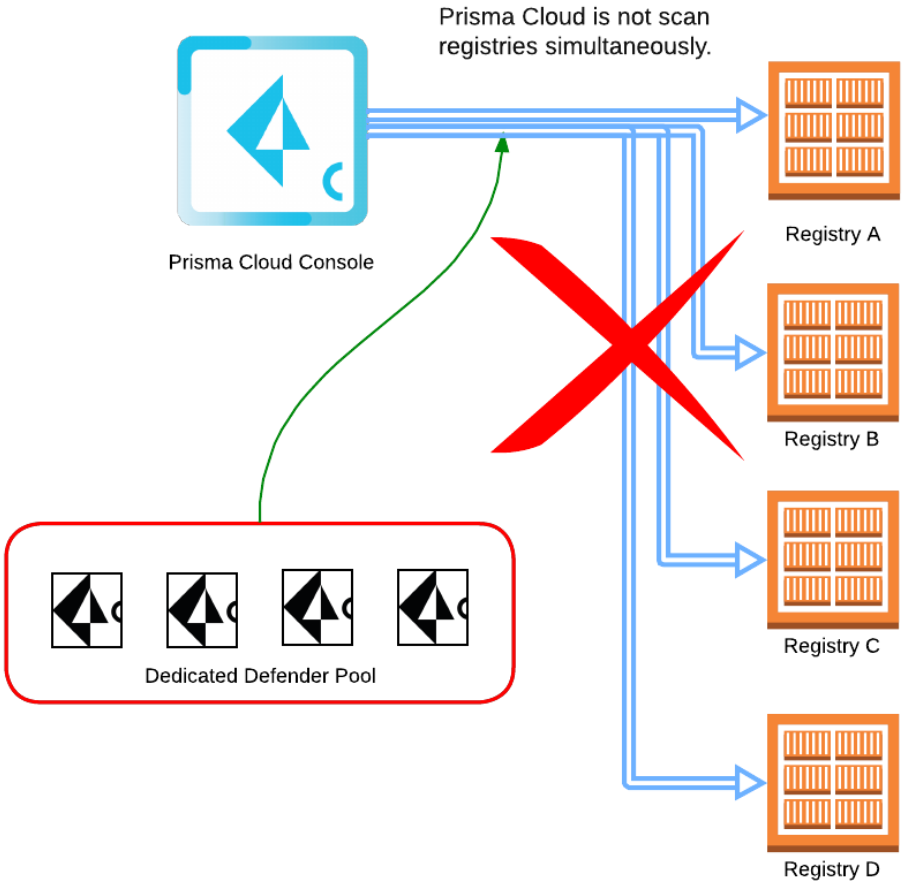
Registry Scanning

Prisma Cloud can scan container images in public and private repositories on public and private registries. When you configure Prisma Cloud to scan a registry, you can select the scope of defenders that will be used for performing the scan job.

Registry Scan Behavior

Prisma Console controls the registry scan with assigned defenders. Console scans one registry at a time. If multiple registries are configured to scan, Prisma Cloud console will scan one registry. Once completed the first registry scanning, then move to the following registry. This registry scanning behavior is not configurable.


Prisma Cloud scans registries sequentially with the defenders.
Prisma Cloud Console
Registry A
Registry B
Registry C
Registry D
Prisma Cloud starts to scan registry A with the allocated defenders. When the work for registry A is completed, Prisma Cloud scans B and moves on to C and D.
Dedicated Defender Pool

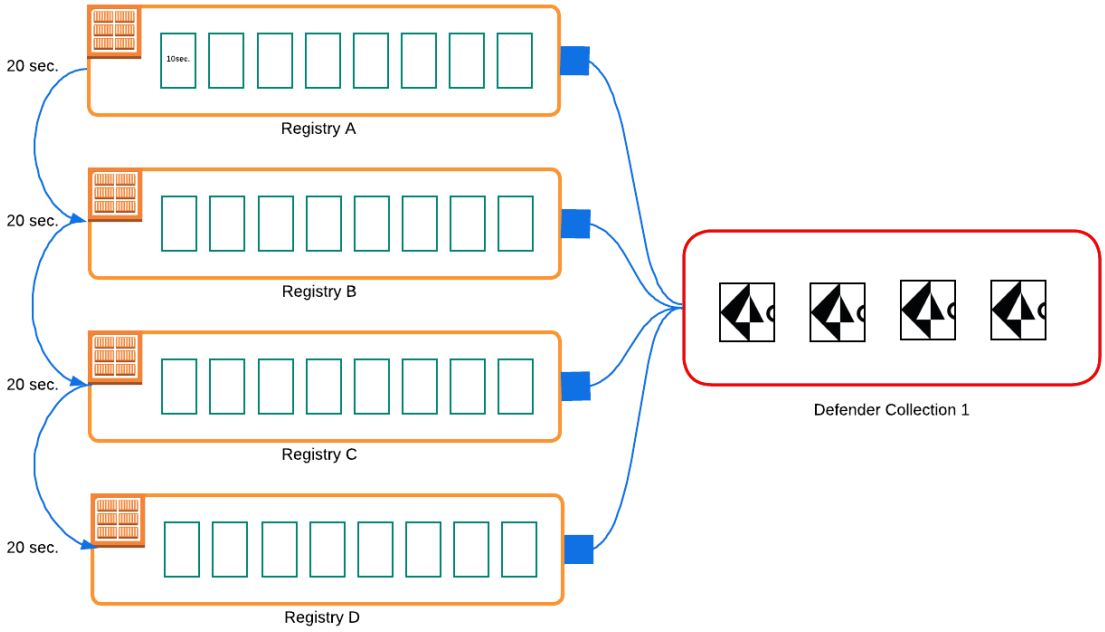


For example, here are three scenarios to configure multiple registries scan with Prisma Cloud. Depending on the configuration, you can compare the total scanning time. Let's assume the following conditions:

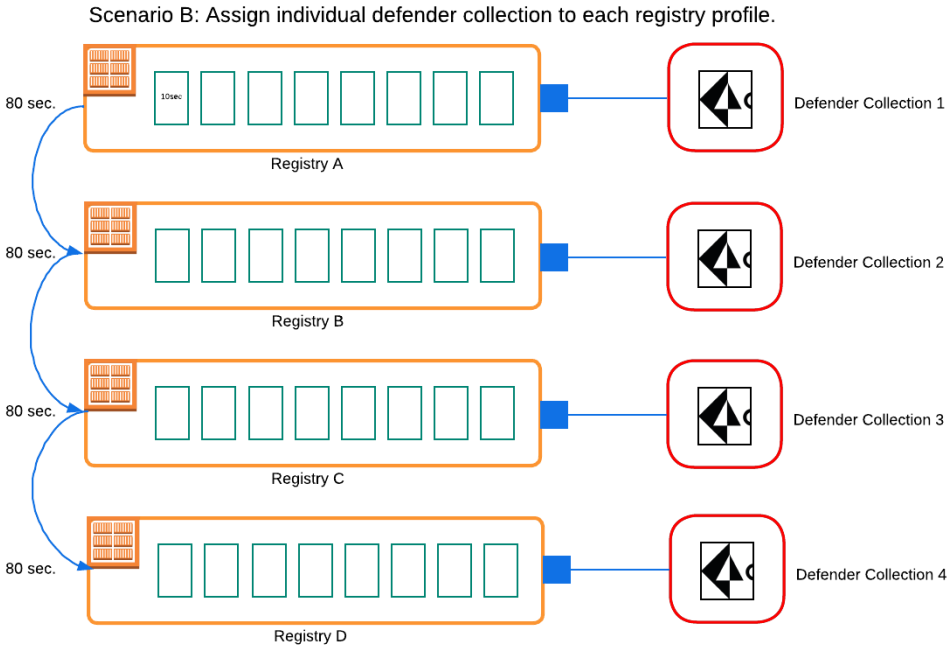
- There are four registries to scan.
- Each registry has eight images.
- It takes 10 seconds to scan the image.

Scenario A:
Build a defender collection for registry scan and assign four defenders. The configures all registry scan profiles with this defender collection. With this scenario, all four defenders scan the images in the registry in parallel. Prisma cloud would then pick registry A and scan it first. It will take 20 seconds to complete. When the work for registry A is done, Prisma Cloud scans B and moves to C and D. The total time to finish the four registry scans is 80 seconds.

Scenario A: Assign Defender Collection 1 to the scope of the each regisery scan profile.

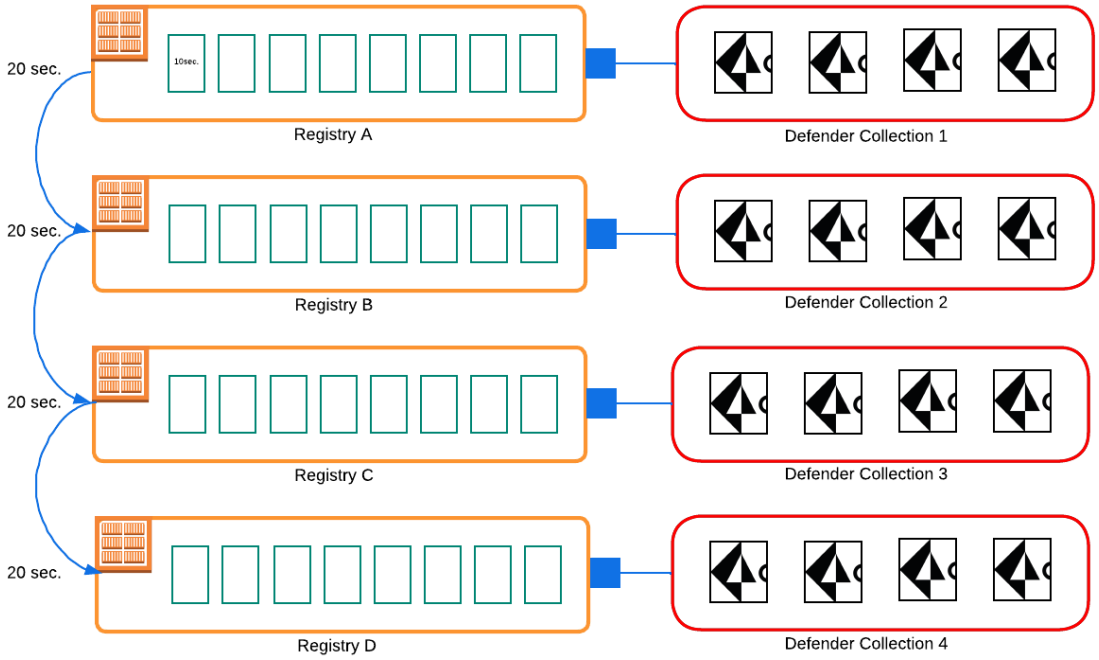


Scenario B:
Build 4 defender collections with a member of the defender. Configure each registry scan with a specific defender collection. One defender will scan the assigned registry. Prisma cloud picks the registry A to scan with the defender collection 1. It will take 80 sec to complete the scan. Then Prisma cloud scans B and moves to C and D. The total time for the scan is 320 seconds.



Scenario C:
Build 4 defender collections with four defenders. Configure each registry scan with a specific defender collection. First, Prisma cloud picks registry A to scan with defender collection 1. It will take 20 sec to complete the scan with four defenders. Then, Prisma cloud scans B and moves to C and D. The total time for the scan will be 80 seconds, which is the same as scenario A. However, because Prisma cloud scans registry sequentially, while defender collection 1 is scanning registry A, the other 12 defenders in collections 2, 3, and 4 are running but scanning.

Scenario C: Assign individual defender collection with 4 defenders to each registry profile.



In the above example, scenario A is the most efficient method. It is best to deploy a dedicated defender pool for scanner purposes and create a dedicated scanner collection. For multiple large registries, it's recommended to assign the defender collection to all registries and increase the number of defenders in the collection to improve throughput and reduce scan time. The console will not scan registries simultaneously, but sequentially, so creating multiple dedicated defender pools is not recommended.

Trusted Images

As organizations get more familiar with their images and environment, they typically leverage our Trusted Images feature to control developer access to a specific registry or even specific images or layers. Trusted Images ensure that developers are using verified or approved sources for their images, as well as provide a straightforward way to implement the best practices for container security.

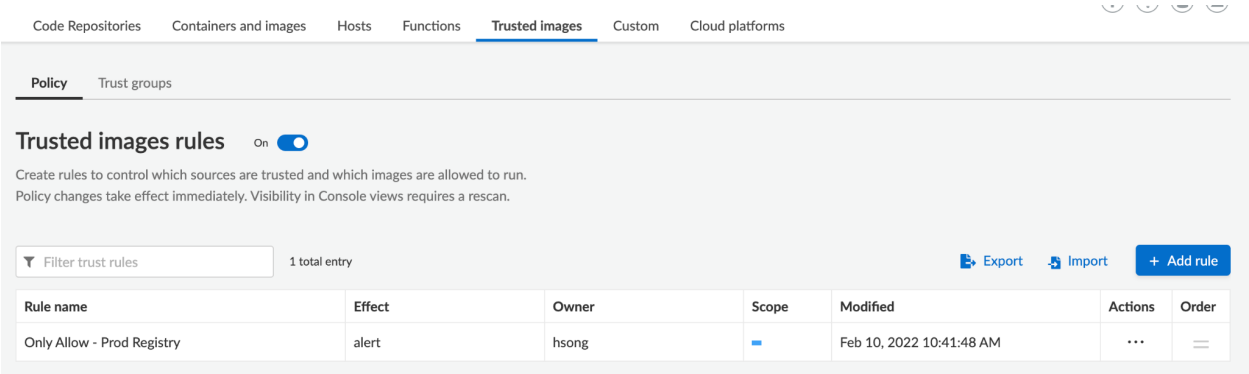
The trusted image function lets you explicitly define which images are permitted to run in your environment. If an untrusted image runs, Prisma Cloud emits an audit, raises an alert, and optionally blocks the container from running.

It is recommended to specify the images it trusts. Declare trust using objects called Trust Groups. Trust Groups collect related registries, repositories, and images in a single entity. Then, for writing policy rules. It's recommended to use registries or repositories for the trusted groups if they are an organizations' standard golden images.

As a best practice, the default rule, Default - alert all should be maintained, and it should be the last rule in your policy as a catchall rule. The default rule matches all clusters and hosts (*). It will alert the images that aren't captured by any other rule in your policy.

Assuming the default rule is in place, the policy is evaluated as follows:

- A rule is matched: The rule is evaluated.
- A rule is matched, but no trust group is matched: The image is considered untrusted. Prisma Cloud takes the same action as if it were explicitly denied.
- No rule match is found: The default rule is evaluated, and an alert is raised for the image that was started. The default rule is always matched because the cluster and hostname are set to a wildcard



Trusted Image Policy

Base Images

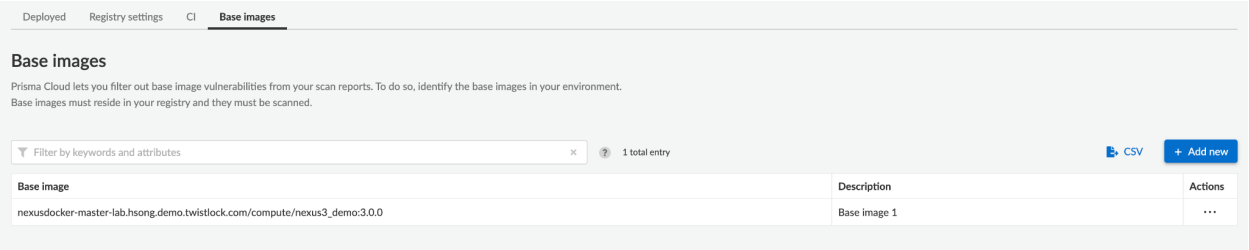
Filtering out vulnerabilities whose source is the base image can help your teams focus on the vulnerabilities relevant for them to fix. For Prisma Cloud to be able to exclude base image vulnerabilities, first identify the base images in your environment.

The base image should be specified in the following format: registry/repo:tag. You can use wildcards for the tag's definition. Excluding base image vulnerabilities is currently not supported on Windows images.

It is recommended to select base Images for the most commonly used throughout the different projects/applications and add them to the Base Images.

Once the base images are identified, they can be filtered out from the reports by using the 'Exclude base images vulns' filter.

The maximum number of base images that can be in scope is 50, where each base image is represented by a digest. If there are 50 base images in scope, and the scanner discovers a new base image, the oldest is purged and replaced with the newest.



Base Images

Radar Utilization

Cloud Radar View

Allows user to view geographic locations of onboarded cloud accounts as well as filter these geographic locations by CSP region, CSPs, resources that are protected within each cloud account including (functions, clusters, registries, app embedded and hosts), and individual cloud accounts

Upon selecting a specific location's resources the user can view the compliance posture of the resources within that location with options to protect the resources if available should they not be compliant. Users can select a resource through the list of available resources to view the resource details including (name, version, runtime, ARN, and protected status), as well as the option to view listed compliance violations with detailed descriptions of each violation upon individual selection of the unprotected resource. The view of non-compliant resources includes the resource's onboarded cloud credential, ID, severity, result, title and associated collection

Host Radar View

Allows users to view available hosts that Prisma has access to. There is an option to color code the hosts by vulnerability severity, runtime behavior compliance, and overall compliance. Users can filter by specific collections, clusters, CSP regions, CSP hosts, only connected hosts, and overall severity level

Container Radar View

Allows user to view individual clusters of containers with available egress/ingress connections

Upon selecting a cluster, individual nodes can be viewed in detail. The console provides a view into the risk summary, environment, and networking information

Risk Summary:

Provides view of vulnerabilities, runtime, compliance, and WAAS for individual container selected with a link to that part of the console within Compute

Shows Image, Image ID, Cluster, Namespace, and Service Account

Environment:

Shows containers and hosts of selected cluster's selected node

Networking Information:

Shows the connected inbound and outbound ports and protocols to the node as well as the outbound IP addresses

Serverless Radar View:

Allows user to view connected Serverless functions within cloud environments

Upon selection of a node representing a serverless function, the user can view the services that the function has permission to as well as general info including the function's CSP, region, and runtime name.

Upon selection of permitted services associated with the serverless function the user can view the resource association (AWS ARN or equivalent) within the service as well as the associated permissions that the function can perform within the associated services

There is also an option, if applicable, to see the scanning levels that are not natively associated with the serverless function within Prisma Cloud Compute. As an example, if the serverless function is not scanned by vulnerabilities or associated with a compliance standard, there will be an option to protect the function with the natively available infrastructure not yet associated within PCC

Settings:

Provides the user with the options to toggle container network monitoring and host network monitoring

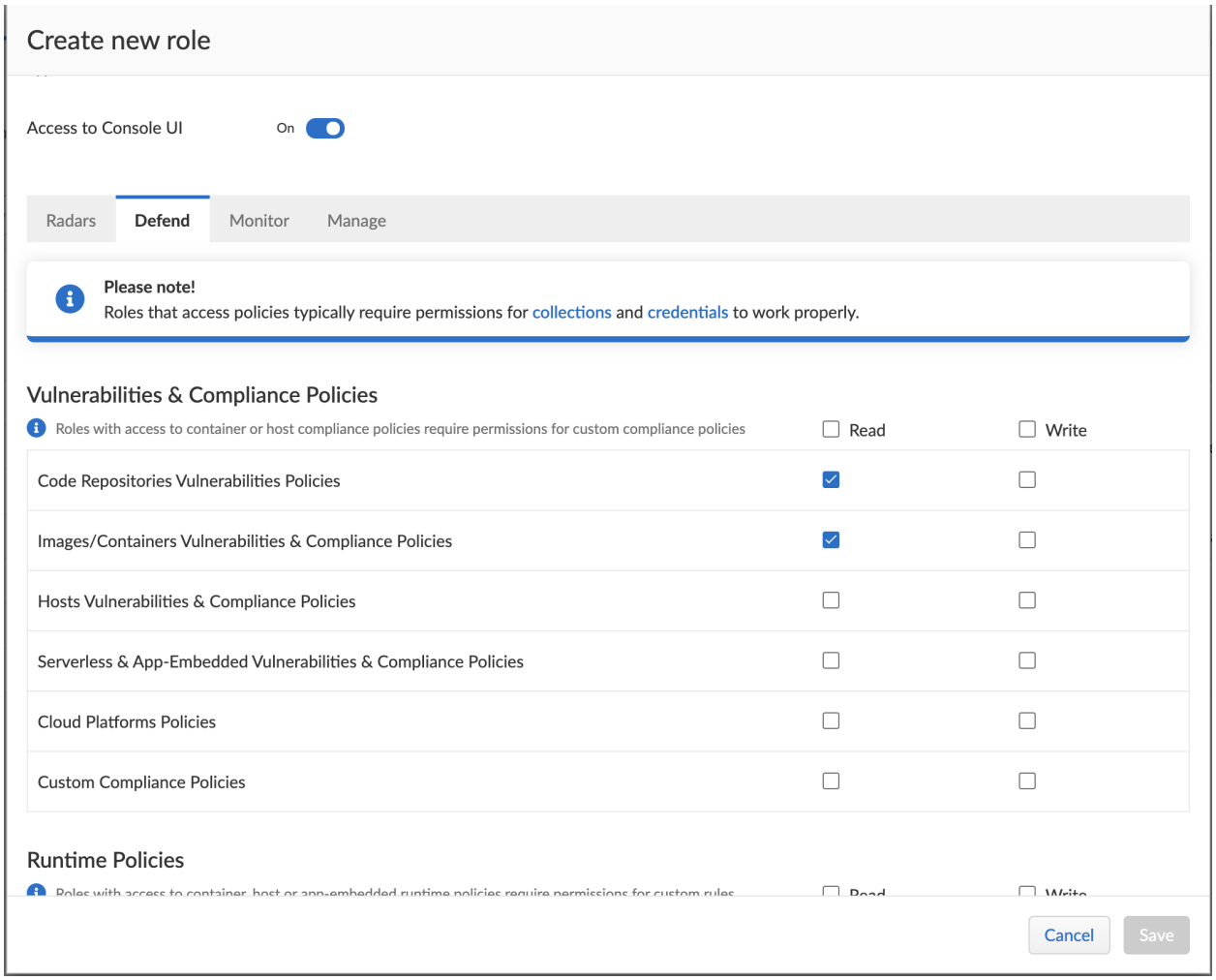
Allows the user to configure network objects by adding subnets to scan that include the network object name and the CIDR block of the associated subnet

Collections

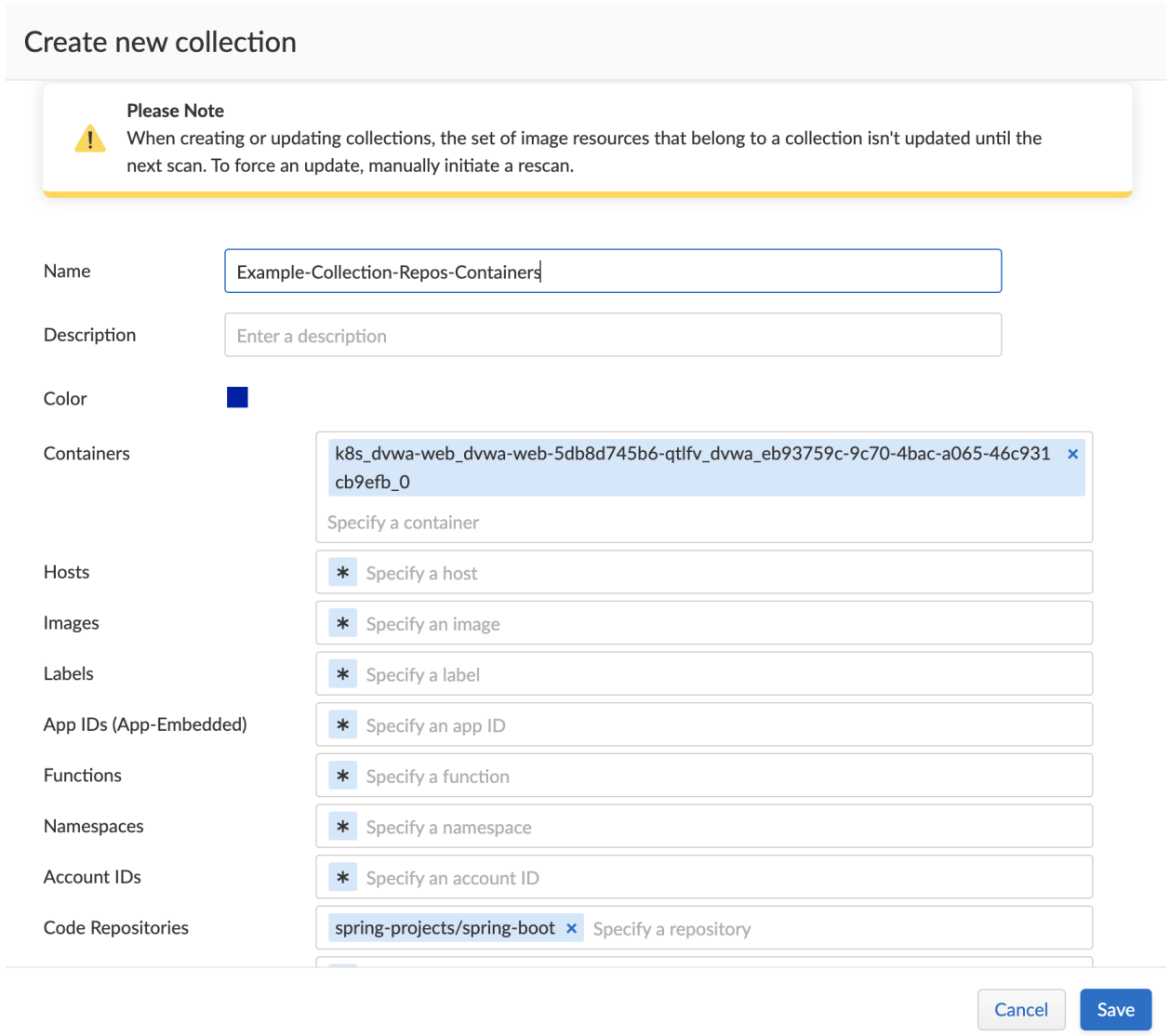
You will need access to an existing SaaS or self-hosted tenant that has the compute capabilities enabled. This means that a user correlated to a role that has sufficient permissions has been assigned to you and you have the ability to authenticate with the console.

Collection Functionality Overview in Compute

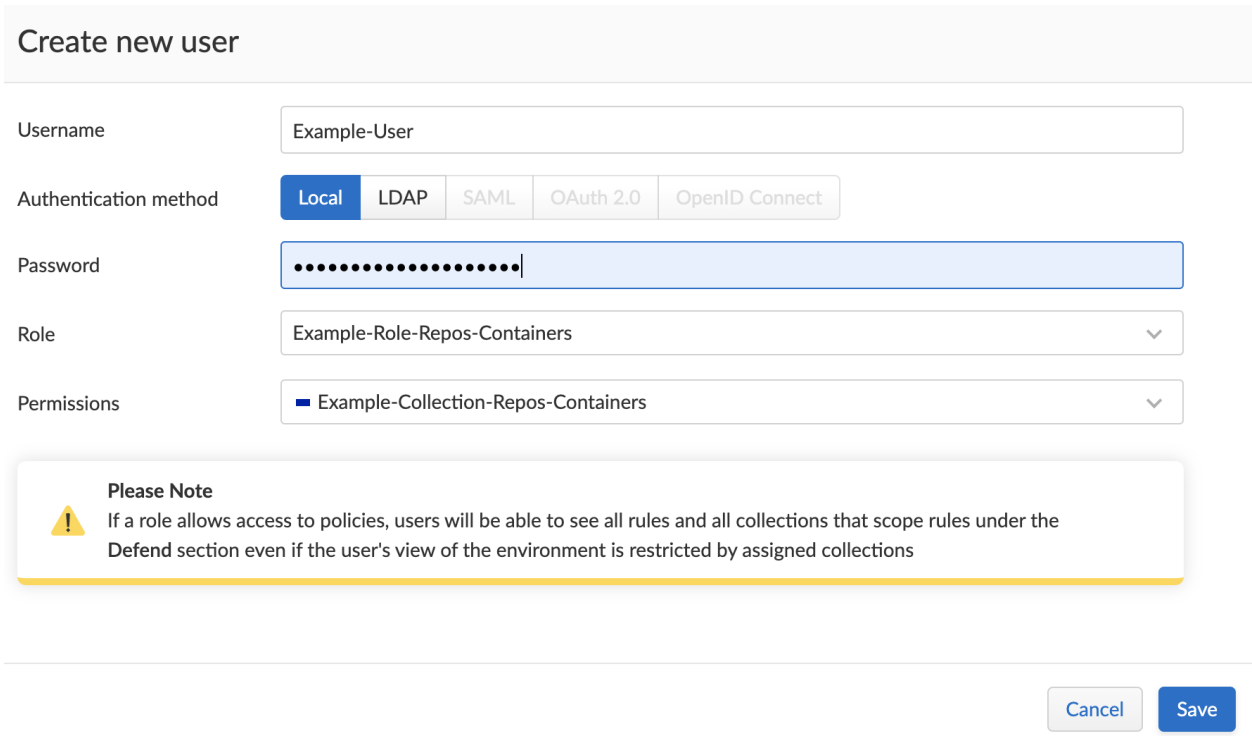
In terms of collection scoping within the console, the first step will be to create a role that limits user access to the specific modules that they need. In this example, access has been limited to the Defend. Vulnerabilities and Compliance modules through the role that were created:



Next a collection was created to associate with the users that would be encompassed by this role, and only included a code repository and an image:

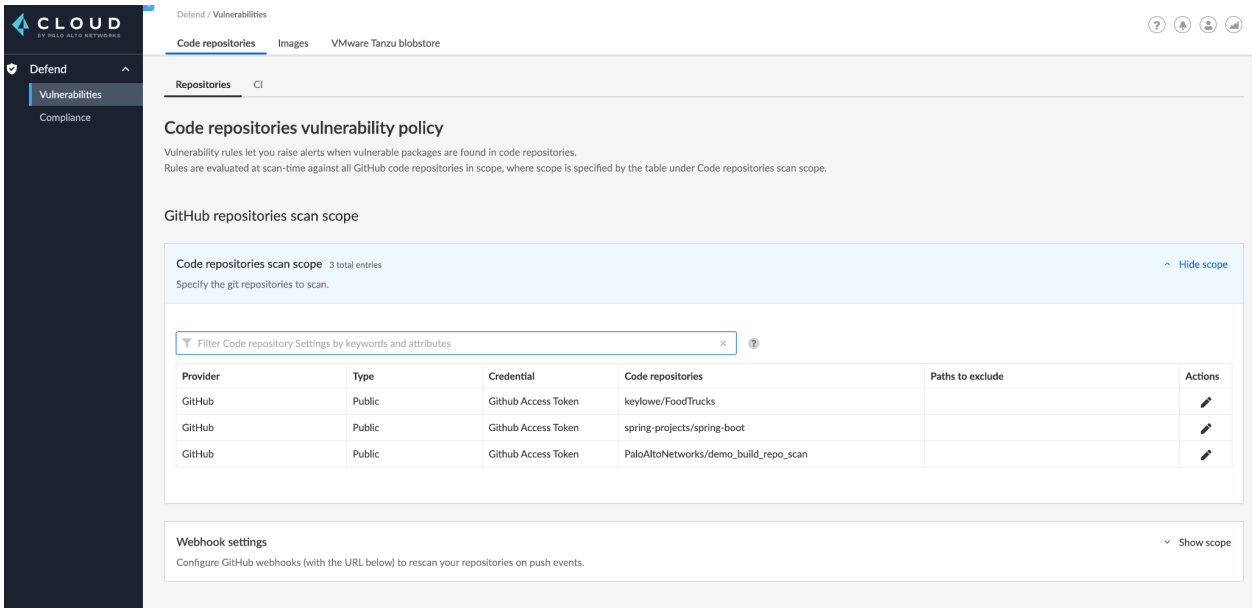


Then the collection and the role were created with a new user:

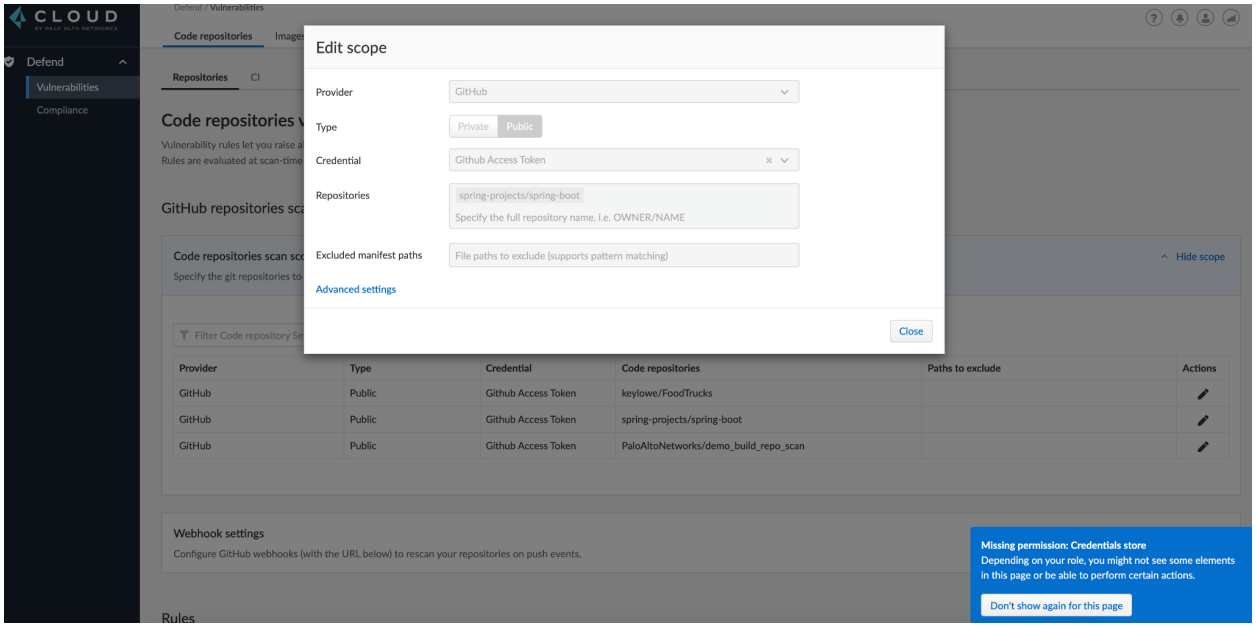


As you can see by the warning sign, in the Defend module of the console where rules are set, one limitation of the console is that specifically within this module users are able to view all rules regardless of assigned collections. This is only within this module and will be shown in a subsequent step. The other modules are completely filtered by the collections that are assigned to a user.

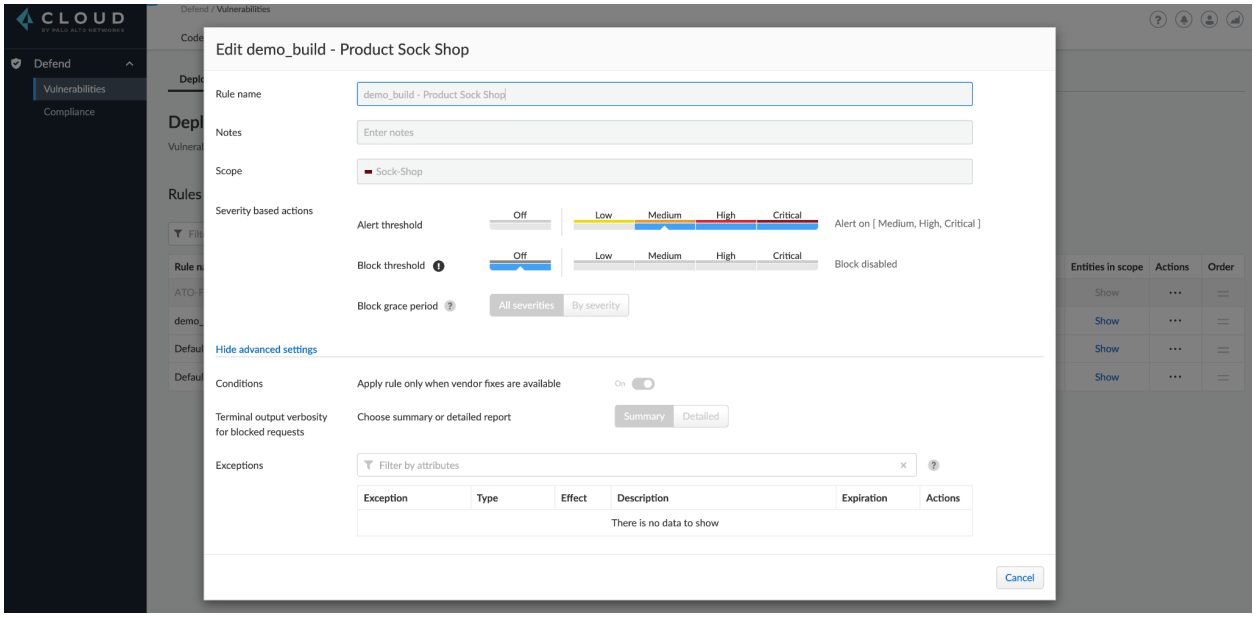
One thing to note is that this will allow console users to see all rules with associated infrastructure in terms of the associated views within the Defend module. For the Radar and Monitor modules, the views are filtered by the individual collections assigned to a user. Once this role was assigned with a collection, the user got the following view of the console when logged in with the new user:



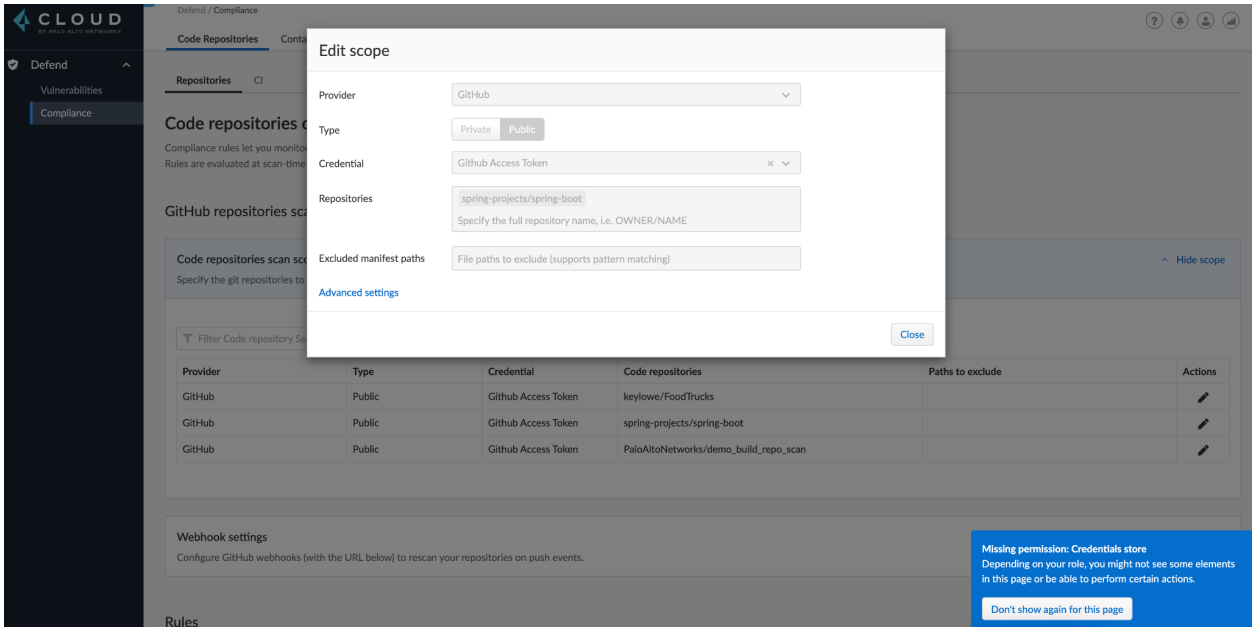
As previously mentioned, in this specific module users, when assigned to have access to this module via role, can see all of the rules regardless of the collection that they are assigned to. In the console's current design the Defend module is meant to be managed and reviewed by vulnerability management teams or managers. Only read access was granted to this module. When we click into editing a scope, we get a non-editable view as can be seen below:

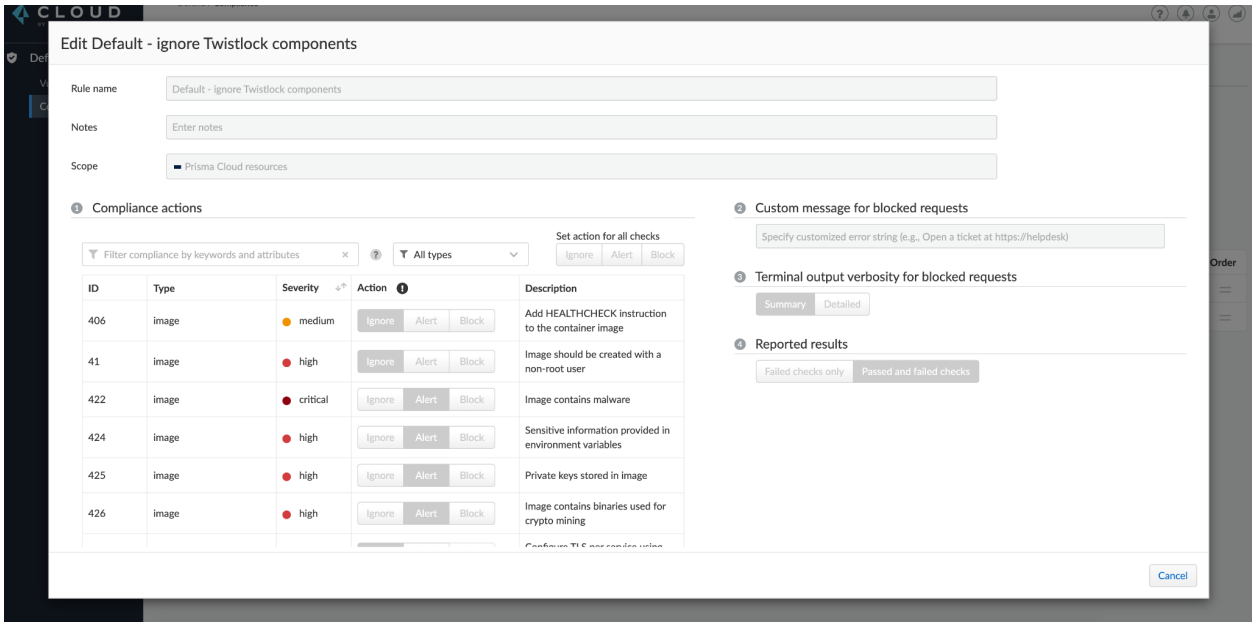


BY pivoting into the Images tab within the Defend module, we can see the rules associated with images but, again, since this role has read-only access to this module, the fields are not editable:

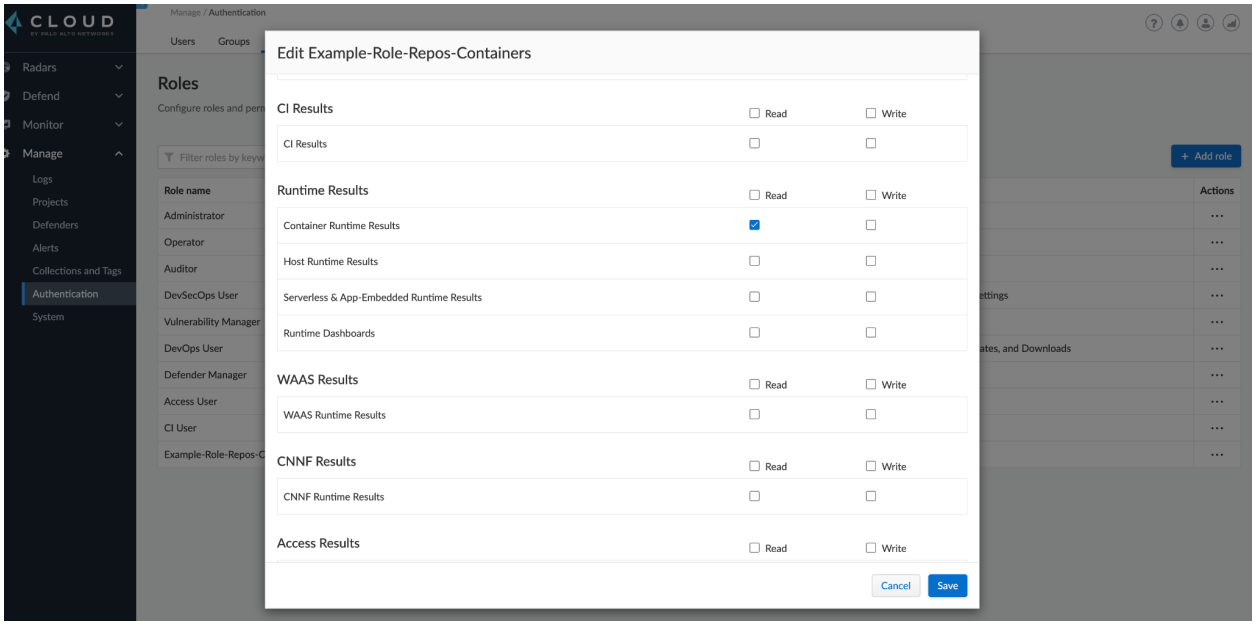


This behavior is also mirrored within the Compliance view of the Defend module for code repositories as well as containers:

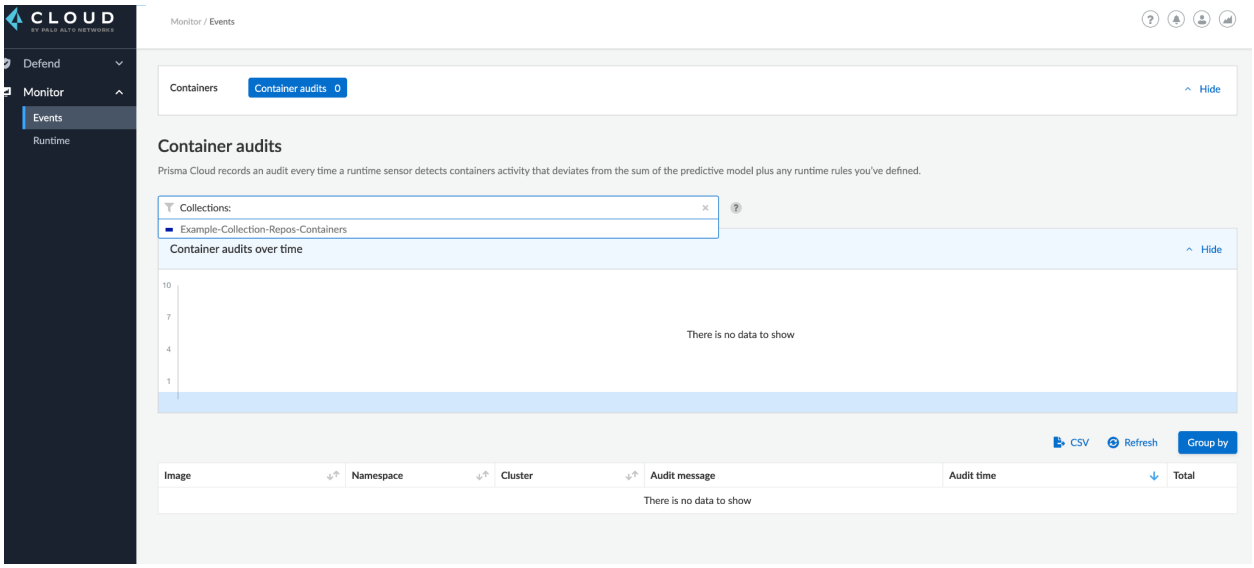




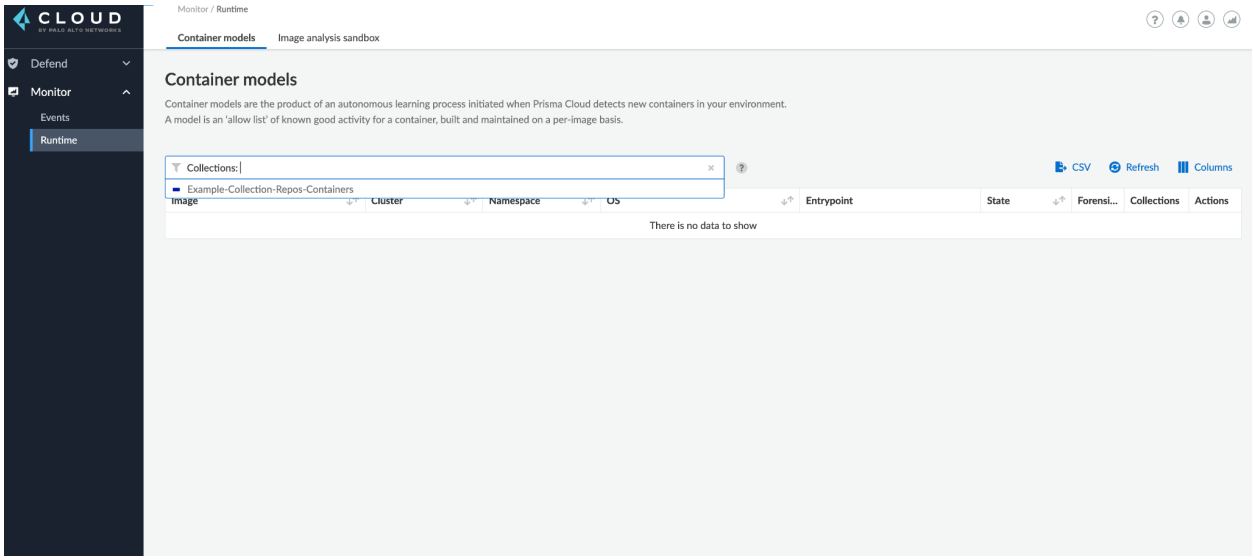
Next, to show how the Monitor module filters by collection, read access was assigned to the image runtime behavior view of the Monitor module for the user's role that was used to log in..



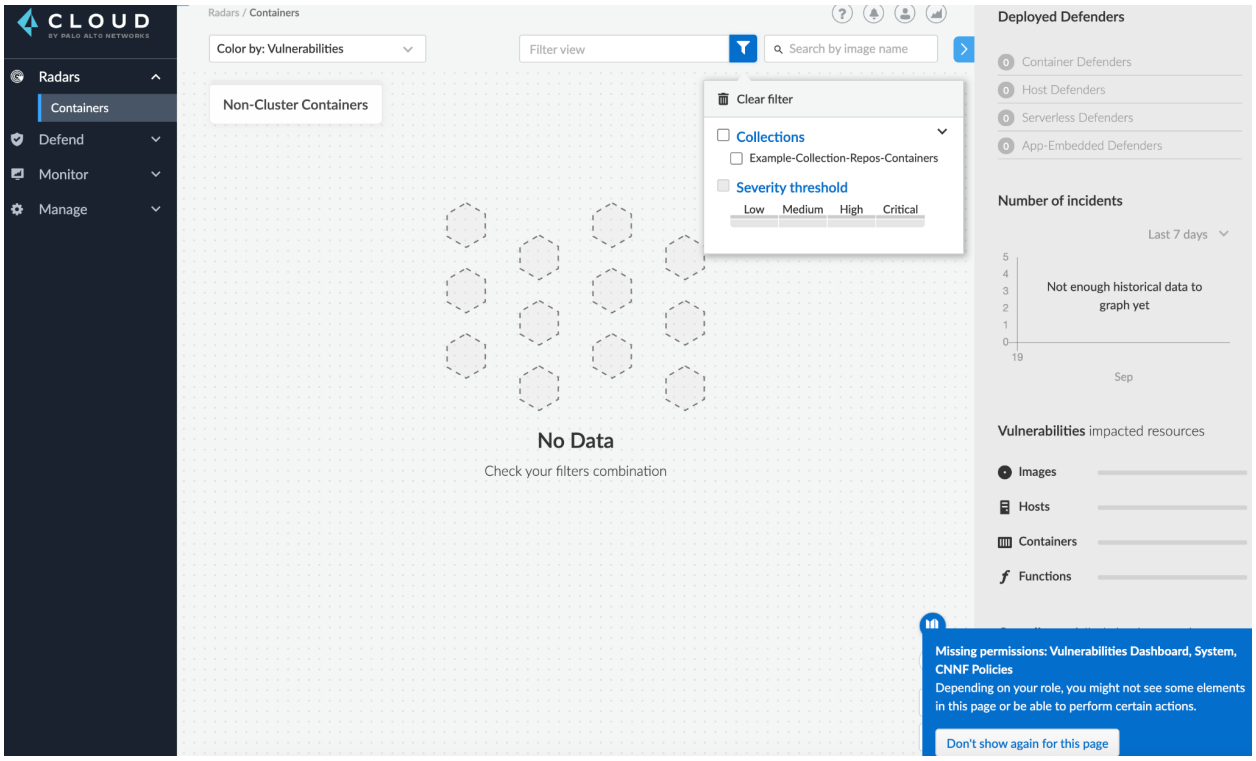
After logging back in with the new user's account, we can see that the Monitor module and the runtime view have been added to the new user's view of the console:



You can also see that the only collection that is available within this view is the collection assigned to the user. This shows how module access can be limited by the scope of the collections that a user is assigned to. In addition to the events view this collection scope has also limited the runtime view within the Monitor module.



As an additional example the Radar module was added, and the container view within this module to the new user's role. When logging back into the console using the new user's credentials, the radar module was limited in functionality to the collection that was assigned to the user.



To summarize, assigned collections combined with roles can limit the scope of what users are able to view within the console. The limitation of the scope of this filtering is the Defend module which, in the present state of the console, is meant to be only assigned to security managers and vulnerability management teams who set the threshold of the risk appetites of environments through severity thresholds.

Hopefully this information helps with your use cases. In terms of newly defended resources, they will be assigned to the OOTB “All” collection that is accessible by system administrators and can be further assigned to the collection associated with a user, team, or environment by system administrators or roles that are granted write access to the collections view of the System module.

In the console defenders are the primary point of ingesting data from the cloud resources on which they are deployed. They push updates via port 8084 to the console and the results of what are pushed are compared to the console’s threat intelligence stream in addition to the various vulnerability threshold rules, runtime rules, and WAAS rules associated with how the defender is scoped into collections. Defenders are the base level of information ingestion into the console, with typically a one-to-one mapping to various resources deployed in a customer cloud environment, such as containers, hosts, registries, and serverless functions.

The information ingested into the console from defenders is made available on a need to know basis through the role associated with the user that is viewing the console as well as the collection that the infrastructure has the defender is deployed to. The role associated with the user can grant read or write permissions to individual modules of the console, as well as individual views within each module. The collection associated with the infrastructure can further limit the information that each user is privy to within the console. All defenders scan their respective infrastructure that they are deployed to but not all of that information is available to each user.

When a user is associated with a role, collections can be associated with either the role that the user is assigned to or with an individual user. As an example, in a DevOps environment there could be a development, QA/testing, and production environment with different business units interacting with each environment. A role could be created for the development team that is associated only with collections that encompass infrastructure in the development environment. As an additional example, there could be a use case of an audit, where an auditing team would need access to the production environment. In this use case a role could be created with full read access to every collection associated with the production environment. Through creating the scope of what defender streams are able to be viewed in the console through collections console administrators can limit the information available to each business unit's use case.

Defenders are built so that the information ingested to the console is tailored to the type of infrastructure on which the defender is deployed. Associating a collection with a single defender would necessitate that each resource in the collection would have to be the same (i.e. all hosts, containers, registries, or serverless functions). Collections allow for the information ingested by defenders to be available regardless of the resource type. However, in the use case of a defender being associated with a collection, the ability for the collection to encompass multiple types of defended infrastructure would be limited by how the defenders are configured for different types of resources. The granular, one-to-one mapping of defenders with individual cloud resources in customer infrastructure allows for, in the present state of the console. the information ingested by the defenders to be grouped on a need to know basis via collections or roles associated with specific collections.

General Best Practices When Implementing Collections

- Naming conventions (code words, hashes, etc.)
- Designing scope with information streams (upstream and downstream) as well as levels of scope in mind (business unit, team, application, environment)
- Looking at available options with the API
- Creating collections for specific incidents/events

Integrate

Configure integrations with 3rd party security tools and SOC workflow tools. Configure alert workflows for notifications and remediation.

Third Party Integrations

Prisma Cloud Compute supports the following Third Party tool integrations out of the box:

- Email
- JIRA
- Slack
- Splunk
- PagerDuty
- Webhooks
- Google Cloud Security Command Center - Only available for onboarded PC accounts.
- AWS Security Hub - Only available for onboarded PC accounts.
- ServiceNow - Only Incident Response

Best practice is to first determine if a customer is using any of the above tools as a SOC tool that can ingest Prisma Cloud alert data. If there is no available out-of-the-box integration for your customer's tool, try to figure out if their tool supports Webhooks ingestion. If Webhooks ingestion is not a probability, there is a possibility that Professional Services can help set up a custom integration with their tool using our APIs.

Related Links:
prisma-cloud-docs/compute/admin_guide at master · PaloAltoNetworks/prisma-cloud-docs (github.com)
https://github.com/PaloAltoNetworks/prisma-cloud-docs/tree/master/compute/admin_guide/alerts

Optimize

Ensure end-to-end adoption of the product and full utilization of Custom RQL, Automated Remediation, UEBA, and Compliance Reporting.

Compute Compliance Functionality

The Center for Internet Security (CIS) publishes recommendations or best practices that should be adhered to when setting up machine images, servers, containers, etc. In addition to these CIS compliance standards the twistlock team has added best practices As an example, to be HIPAA compliant you may be required to implement certain CIS standards.

Computers can fail or block builds to adhere to compliance standards but there is no remediation in place.

Example CIS Kubernetes Compliance Benchmark:

- Ensure admin.conf file permissions are 644 or more restrictive
- Audit: run the stat command to display the permissions
- Remediation: run chmod to set the appropriate permissions

Prisma Compute automates the audit steps checking for misconfigurations in various environments and exposes each benchmark check as a discrete configuration item in a compliance rule allowing for individual lower level checks to be enforced in an environment.

- Critical and high severity alerts only -- medium and low are ignored by default
- Rule creation compliance standard are in the top right corner of creating an alert
- Compliance checks are set by PANW not CIS, customer needs to verify compliance checks based on need
- Anything the customer can script out to run as a compliance check can be integrated as a compliance check (powershell or bash) to be run against each image
- Checks can be organization specific or standard hardening guidelines
- Checks are safely executed against new container instance in a sandbox environment

Trusted Images:

- Allow for specific images, registries, or image base layers to be deemed as trusted
- Allows for choice of which image can be trusted
- Rules can be setup so that the use of untrusted images trigger an alert or the image is blocked by being prevented from launching
- Feature allows organizations to protect against the deployment of arbitrary and potentially malicious images from the internet such as from Docker Hub
- ‘Not a get out of jail free card’
    - Even if image is trusted it can still fail at run time based on compliance checks and rules

Cloud Platform Discovery and Compliance

- Check and scan resources based on customer cloud provider credentials
    - Checks if there is a scan in the CSP that coordinates with a Prisma Cloud Scan
        - Scans:
            - Managed Kubernetes
            - Image Registries
            - Serverless Functions
        - Runs set of Twistlock Lab created checks against a cloud environment

Cloud Compliance is defense in depth:

1. Scan vulnerabilities (check critical or high CVEs)
2. Compliance checks for unpublished CVEs

a. Available dashboard in UI -- Defend Compliance -- Rule matching is the same (top to bottom)
    i. Checks high and critical severity -- can create alert, turns off, blocks image spin up, or fails -- checks that nothing is set to block

Defend Tab -- Compliance Tab -- Cloud Platform Tab:

- Used for onboarding cloud providers
- Initially scans for resources to show if there is a protection or not

Compliance Tab:

- Compliance Explorer: shows all non-compliance critical and high alerts
- Clicking on compliance check will show resources that are not compliant
- Container level view shows each container in environment -- name, image, host, cluster, compliance heat map
    - Open up to see best practice within compliance -- make sure to check version number for compliance
    - Check compliance standard vendor with description in Prisma Compliance Container View
    - Compliance vendors will show audits, remediation, etc.
    - Our Audits in Prisma are written in Go so they are not available to see in UI
- Image Level view -- container compliance is different from image level compliance
    - Can be custom, twistlock, CIS (most CIS are written at the container level)
    - Every image scan for vulnerability follows compliance at the same time following the compliance rules
    - Typically, vulnerabilities are scanned first, the written into rules or policies for compliance
- Function Level View
- Trusted Images: list of images with trust status
- Cloud discovery: scans all resources, picks out environments' resources (AWS: EC2, registries, lambda functions, EKS, ECS) count in each region and how many are scanned by twistlock or compute
    - Protect button will begin to start scanning

WAAS

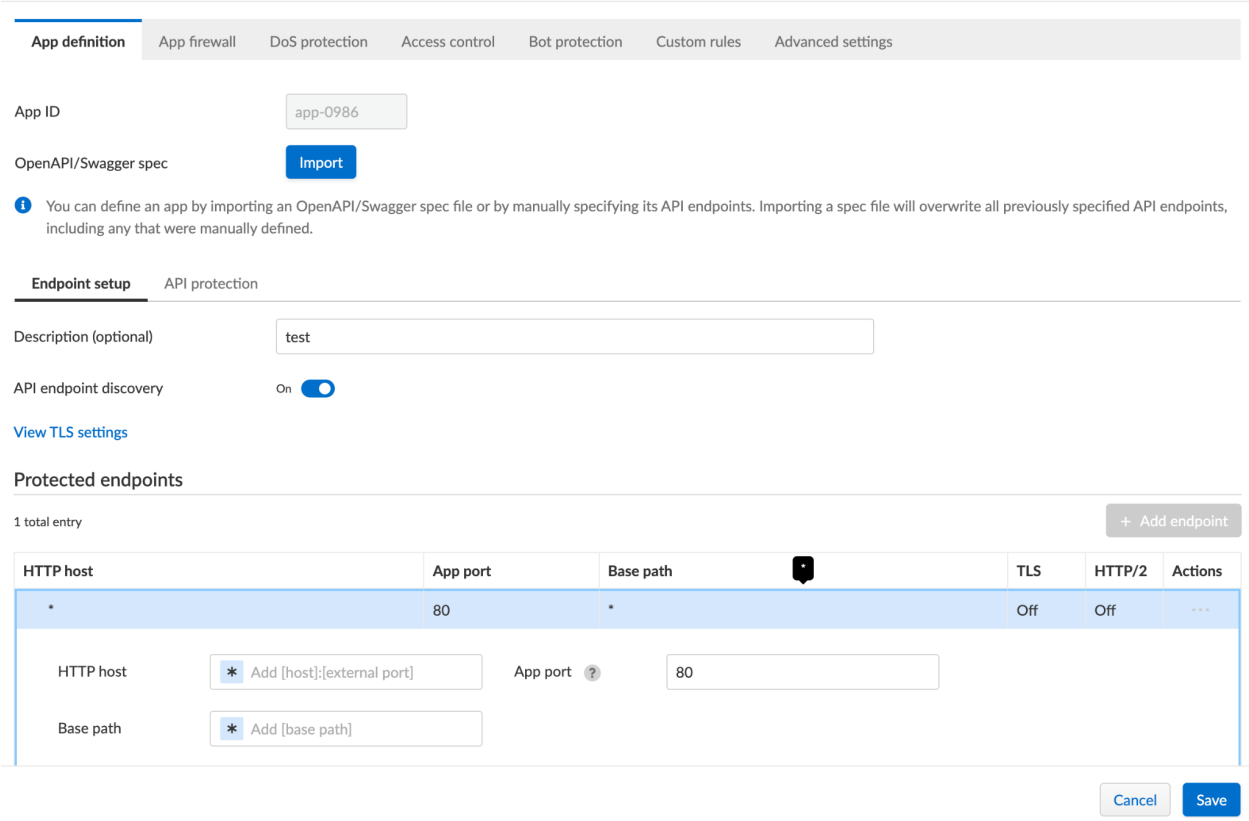
App Definition

Utilizing Open API / Swagger documents to create the General App Setup and API protection is highly recommended. Not only do Open API and Swagger documents make it easy and consistent to deploy application definitions, they also support the following DevOps philosophy:

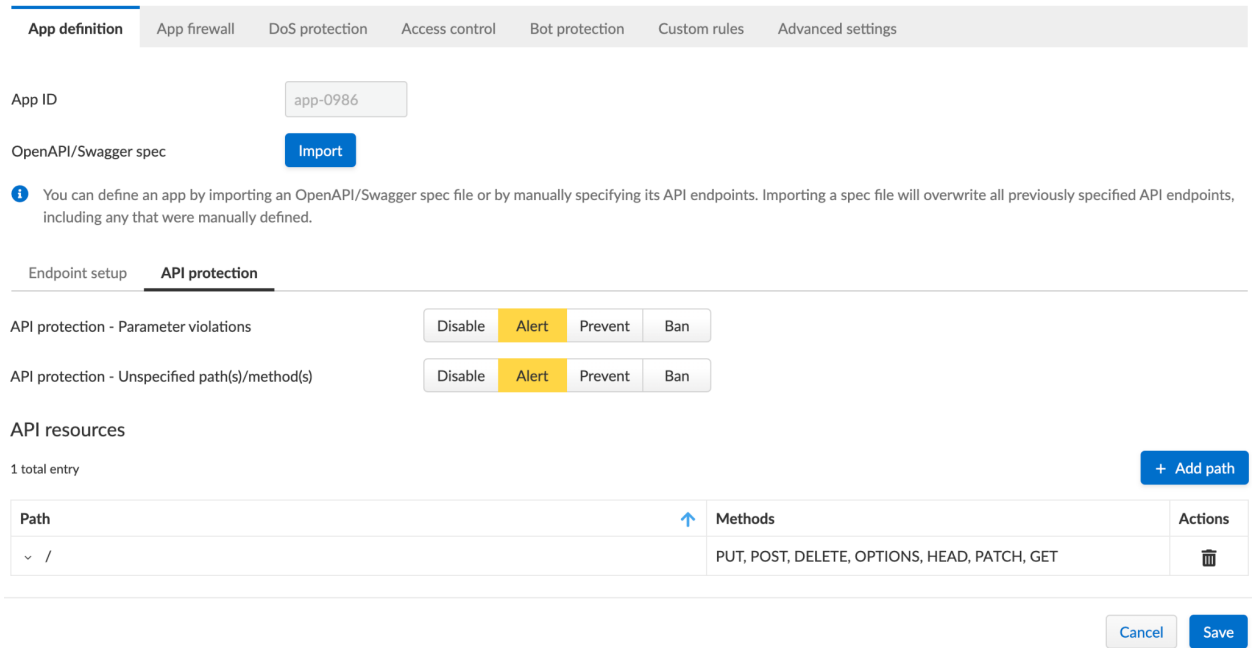
1.) Open API / Swagger is a good form of documentation on Restful APIs for any given developer group and organization
2.) Support automated updating as API and infrastructure changes.
3.) Allow for automation of deployment in a consistent manner.
4.) Allow versioning for documentation and rollback of configuration if there is a problem with the deployment.

If Open API / Swagger documentation is not available, define the API functions manually. To enable API protection, define the base path and the App port. The App port is the port that the application is listening on rather than the external port that the calling application will use. The more specific to describe your application, there is a better chance of protecting your application. Rather than defining a host, use the scoping mechanism to clearly identify the application to protect.

With the base path, the standard path for RestAPI is "/api/v1".



Specify your APIs by defining the signature of your endpoint as far as what is acceptable input parameter.

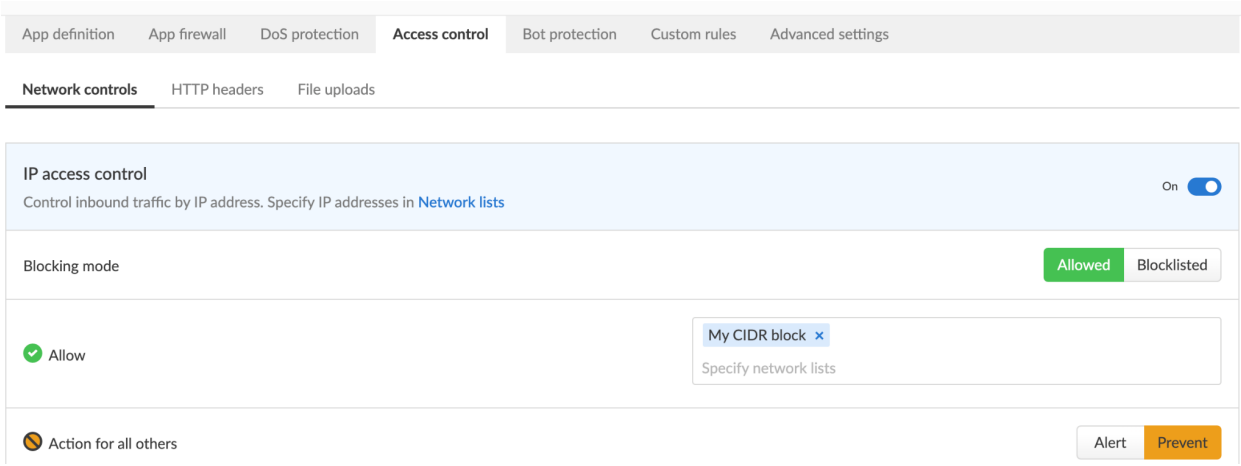


API Protection Scoping

To properly profile your application, clearly defining a scope is very important. Think broad enough to encompass the application, yet specific enough to define the application. For example, specify an image and even specific enough that you add labels that Prisma Cloud will use to identify the application. If the API protection rules are all the same for any given image, use one API protection policy. If you need behavior to be different based on a label, or a cluster where the container will be running, create a different scope and apply the given policy to that scope.

Network Controls

Allowing only sources that need access to the application reduces the attack radius of a given application. In order to do this, define a CIDR block that is allowed to access the application, and allow only that CIDR block. Prevent for all other CIDR blocks by setting the “Prevent” action for all others.



HTTP Headers

With http header inspection, it is looking for a key value pair to inspect for every single http call to the application. The header name is case insensitive and the value could be either allow or deny. For the values, they can be either explicit or a wildcard character (*) can be used for the following three use cases:

- Begins With
  - E.g., “Mozilla/5.0*”
- Contains
  - E.g., “*(X11; Linux x86_64)*”
- Ends With
  - E.g., “*Safari/537.36”

File Uploads

Application Profiling

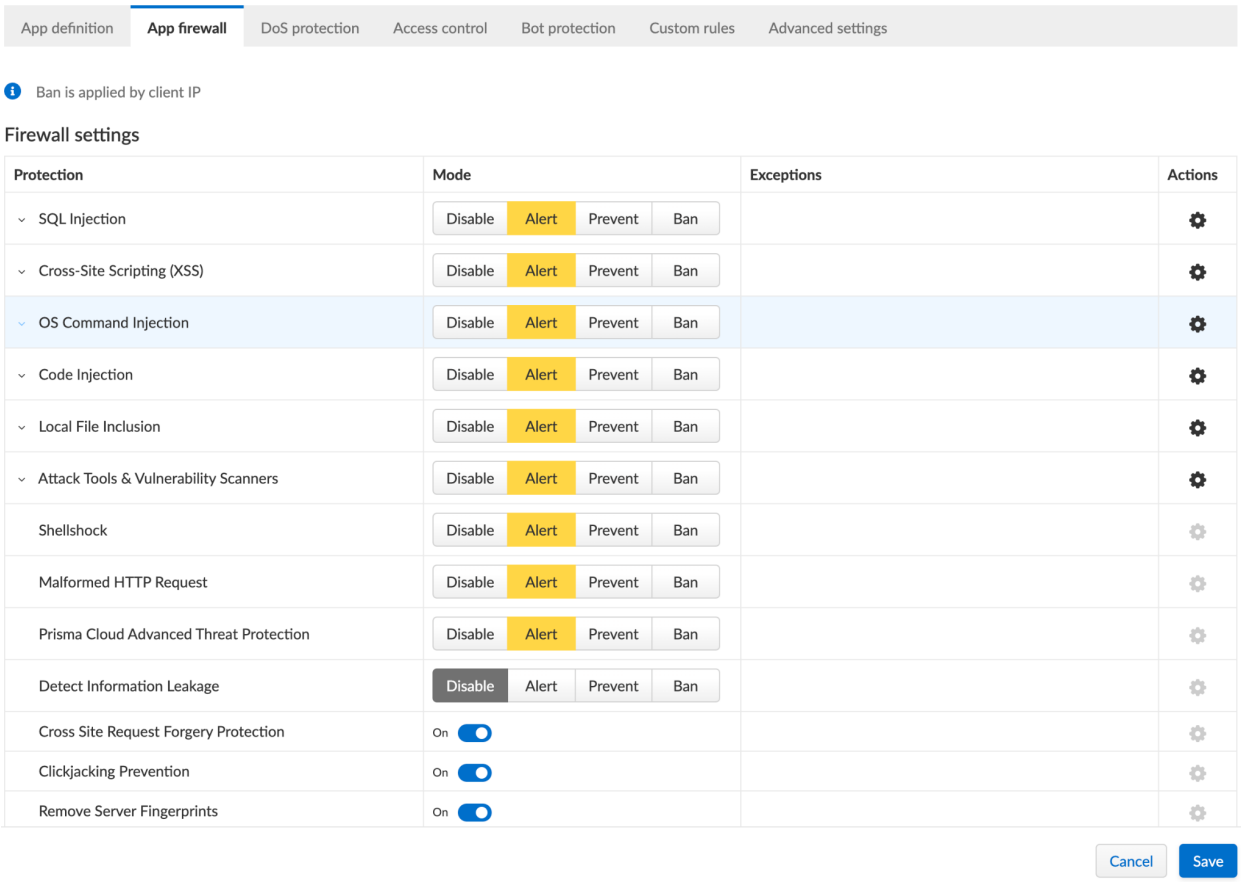
The best way to protect an application is to be as specific in the application profiling. For example, if you are looking for a person at an airport and you have never met this person before, the more accurate detail that describes the person will clearly identify the person with the least amount of false positives. Likewise with an application, the greater detail the WAAS component has to profile the application, the better it will be in blocking attacks. To enforce specific headers such as a specific token type to retrieve data, you inspect the header for a specific token format by means of a regular expression. For example, if in any given request to an application you wanted the following:

- Specific headers with specific values
  - The headers can be inspected with the exact values
- An application token in the header with specific format (e.g., “s.34x7797bxe3p118923”)

    - The header value can be inspected with a regular expression such as the following would match the above token:
        - /^s\.[0-9a-zA-Z]+/g
    - Utilizing an online regex tool such as the following link will assist in creating an application profile that accurately describes your application.
- Body content
    - The body content can look for specific values using regular expressions to block malicious intent
    - Regular expression can be used to inspect body content to ensure it is a valid web request and reject it if it is not.
    - The body content can look for payload that is only valid
    - The body content can be inspected in combination with an action. For example the payload will only be inspected if the http request action is a POST or PUT and will not be examined if the action is a GET or DELETE.

Scoping

Rules are basic building blocks to enforce a specific action but scopes are used to bind a rule to a specific application. Building proper scopes



Load Balancers

Configuration for load balancers need to have the HTTP Header X-Forwarded-For in each request for WAAS to be able to determine if the HTTP call needs to be blocked based on source country origin. Most load balancers have that header enabled but without that header country origin blocking will not work.

Alerting To Blocking

Having clear visibility into which vulnerabilities to focus on is crucial to identify four items: 1.) the criticality of the alarm, 2.) the owners of an image resource, 3.) the actual image resource that is triggering the vulnerability alarm and 4.) if there is a fix for the vulnerability. In order to facilitate visibility, the GO code to pull the images and sort the images by owner tag and image can be found here.

Without measurability, it is impossible to manage in a proper manner. In addition, scanning images in the CD / CI pipeline is crucial in preventing further vulnerabilities from polluting the environment. Once the report is generated by the GO, it will be clear who the

maintainers are, and the images they own. The report will also show the vulnerabilities and their severity levels. When initially generating a report for a customer who has a non-existent vulnerability management policy, their vulnerabilities with critical and high severity levels can be over 1 million alerts. The GO code is flexible enough to filter out only specific vulnerability levels and fix dates. When filtered, to simply critical, the list is much more manageable. Also the report will show that across multiple images, the same package is triggering alarms. Therefore updating one package across multiple images could reduce the vulnerabilities significantly.

Owners

It is vital to know who the owners are for resources. Many customers have no visibility into these cloud resources because they do not have an enforcement of tags in their environment. If ownership is not clear, it will be a huge challenge to manage the environment. Also, the tags can be utilized in defining scopes in Prisma Cloud for blocking vulnerabilities from polluting the environment.

Socialization

It is important to socialize the project of converting from alerting to blocking. Having key stakeholders onboard with the objective of the project is important. To that end, having a regular cadence call multiple times a week is critical to review which images and which packages are triggering alarms. In these meetings, the package updates can be measured to ensure readiness of blocking. If alarms are still not being addressed, during the meeting those issues can be discussed and determined if the blocking for that image is a go or no go for specific business issues.

Change Control

As with any good business process, it is important to have a good documentation system to know what changes affect end consumers of an application. With a change control, it also outlines a backout plan if an outage occurs.

Scope

Defining scopes that you can place the images in to block is important. Having one for alerting only and one for blocking is very important. When converting images to blocking mode causes an outage, move the image back to the scope of alerting and triage the situation later. Once the image has been moved to blocking mode, the CI pipeline scanning is critical in keeping the environment unpolluted.