



# Tenable Cloud Security User Guide

---

Last Revised: August 31, 2023



# Table of Contents

<b>Welcome to Tenable Cloud Security</b> .....	<b>10</b>
System Requirements .....	12
Role-Based Access Control .....	13
User Role Mapping between Tenable Vulnerability Management and Tenable Cloud Security .....	16
Access Tenable Cloud Security .....	17
Access the Workspace .....	18
<b>Getting Started with Tenable Cloud Security</b> .....	<b>20</b>
Cloud Scan Workflow .....	22
IaC Scan Workflow .....	24
<b>Create a Project</b> .....	<b>28</b>
Projects and Connections .....	28
View Projects and Connections .....	29
Manage Projects .....	32
Manage Repositories .....	35
<b>Connect Cloud Accounts</b> .....	<b>37</b>
Onboard AWS Accounts .....	38
Set Up Read-Only Access to the AWS Account .....	39
Permissions and Supported Resources for AWS ReadOnlyAccess Policy .....	48
Onboard an AWS Organization .....	81
Onboard an AWS Account .....	87
Onboard an Azure Account .....	89
Create an Azure Service Principal Role .....	90



Register an application with Azure .....	92
Create a custom role and assign it to the service principal .....	93
Assign the Reader role to the service principal .....	100
Create a client secret .....	104
Onboard a GCP Service Account .....	106
Create a GCP Service Account .....	108
Discover Cloud Accounts .....	115
Cloud Account Statuses .....	116
Discover and Onboard AWS Accounts .....	117
Discover Azure Accounts .....	120
Discover GCP Accounts .....	123
Manage Cloud Accounts .....	125
View Cloud Accounts .....	126
Edit the Configuration of a Cloud Account .....	129
Ignore a Cloud Account .....	131
Delete a Cloud Account .....	132
Cloud Account Discovery FAQ .....	133
<b>Cloud Scans .....</b>	<b>136</b>
Create a Scan Profile .....	137
Schedule a Scan .....	139
Run a Cloud Scan .....	140
Manage Scan Profiles .....	140
View Scan Profiles .....	142
Set a Default Scan Profile .....	144



Edit a Scan Profile .....	145
Copy a Scan Profile .....	146
Delete a Scan Profile .....	147
View Scan History .....	148
<b>Agentless Assessment .....</b>	<b>152</b>
Live Results for Agentless Assessment .....	154
AWS Agentless Assessment Workflow .....	155
Agentless Assessment Requirements for AWS .....	157
AWS IAM Role for Agentless Assessment .....	160
Create AWS Snapshot .....	161
Create AWS Snapshot Manually .....	163
Automate Snapshot Creation with AWS Data Lifecycle Manager (DLM) .....	166
Configure Vulnerability Scan using Agentless Assessment for AWS .....	167
Azure Agentless Assessment Workflow .....	169
Agentless Assessment Requirements for Azure .....	171
Azure Service Principal Role for Agentless Assessment .....	174
Create an Azure Virtual Machine Snapshot .....	174
Create Azure Virtual Machine Snapshot Manually .....	175
Automate Azure Virtual Machine Snapshot Creation .....	177
Configure Vulnerability Scan using Agentless Assessment for Azure .....	178
Agentless Assessment FAQ .....	180
Troubleshooting Issues with Agentless Assessment .....	182
<b>Connect Repositories .....</b>	<b>184</b>
Repository Configuration Parameters .....	190



IaC Engine Types .....	194
Integrate with GitHub .....	195
Integrate with Bitbucket .....	198
Integrate with GitLab .....	202
Integrate with Azure DevOps .....	205
Integrate with AWS CodeCommit .....	209
Set Up Write Access for AWS CodeCommit .....	211
<b>Scan Kubernetes Cluster Environments .....</b>	<b>213</b>
<b>Set up Code Analysis Using CLI .....</b>	<b>218</b>
Download Configuration File .....	219
Install or Upgrade the CLI .....	220
Scan IaC Files Using CLI .....	222
Scan IaC Files in the CLI Local Mode .....	226
Tenable Cloud Security Commands and Options .....	228
<b>Container Security with Tenable Cloud Security .....</b>	<b>233</b>
Install Tenable Cloud Security CLI for Tenable Container Security .....	235
Scan a Container Image .....	237
Integrate Tenable Cloud Security CLI with SCM and CI/CD Pipelines .....	245
Scan a Container Registry .....	248
Container Registry Scan using CLI .....	250
Scan an Amazon Elastic Container Registry (ECR) .....	254
Scan a Quay Container Registry .....	256
Scan a JFrog Container Registry .....	257
Scan an Azure Container Registry .....	257



Generate a Report of Images in a Container Registry .....	260
Scan a Container Registry using Tenable Cloud Security Docker Image .....	262
Tenable Cloud Security Container Security Commands and Options .....	263
Commands .....	264
Global Scan Options for Image and Registry Scans (tcs consec command) .....	265
Scan Options for Container Images (tcs consec image command) .....	267
Scan Options for Container Registries (tcs consec registry command) .....	269
Scan with Environment Variables .....	272
Script Options .....	273
View the Containers Dashboard .....	274
<b>Configure CI/CD Integrations .....</b>	<b>276</b>
Generate API Tokens .....	277
Integrate with Terraform Cloud .....	279
Integrate with Jenkins Pipeline .....	283
Integrate with GitHub Action .....	284
Integrate with Azure DevOps Pipeline .....	289
Set Up Policy Guardrails (CI/CD) .....	293
<b>Use an On-Premises Code Scanner .....</b>	<b>297</b>
Deploy an On-Premises Code Scanner .....	299
Use an On-Premises Code Scanner to Scan GitHub Enterprise IaCs .....	304
Use an On-Premises Code Scanner to Scan Bitbucket Server IaCs .....	308
Use an On-Premises Code Scanner to Scan GitLab Server IaCs .....	313
Configure an On-Premise Code Scanner to Use Self-Signed Certificate .....	317
Viewing the Logs from an On-Premises Code Scanner .....	318



<b>Policies and Policy Groups</b> .....	<b>319</b>
How Policies Work in Tenable Cloud Security .....	320
Manage Policies .....	322
Policy Modes .....	323
Create a Custom Policy .....	325
View and Download Policies .....	327
Edit a Policy .....	330
Delete a Policy .....	332
Manage Policy Groups .....	333
Create a Custom Policy Group .....	334
View Policy Groups .....	336
Edit a Policy Group .....	338
Delete Policy Groups .....	339
Associate Policies with a Project .....	340
<b>Set up Drift Analysis</b> .....	<b>341</b>
Set a Baseline for a Project .....	342
View Cloud Drifts .....	343
View IaC Drifts .....	346
Review Drifts .....	349
Remediate Drifts .....	351
<b>Configure Alerts</b> .....	<b>353</b>
Configure Email Alerts .....	354
Configure Slack Alerts .....	355
Configure Microsoft Teams Alerts .....	357



Configure Splunk Alerts .....	358
Configure AWS SNS Alerts .....	362
Set up a Role for AWS SNS Alerts .....	364
Integrate with Atlassian Jira .....	367
View Alerts .....	370
Configure Alert Rules .....	371
Alerts Page Information .....	372
Set an Alert Rule for a Policy .....	374
<b>View Findings .....</b>	<b>375</b>
View Misconfigurations .....	376
View Vulnerabilities .....	381
View Ignored Misconfigurations .....	384
View Resource Configuration .....	386
Compare Resource Configurations .....	387
<b>View Resources .....</b>	<b>388</b>
View Resource Details .....	391
<b>Remediate Issues .....</b>	<b>394</b>
Set up Auto-Remediation .....	395
Set up Inline Reviews .....	398
Escalate or Share an Issue .....	401
Create a Pull Request for an Issue .....	403
Create a Ticket for an Issue .....	407
Ignore Misconfigurations .....	409
Unignore an Issue .....	414





---

View and Remediate the Line of Change in IaC .....	415
Fix a Configuration Violation for a Project .....	417
<b>View Tenable Cloud Security Dashboards and Reports .....</b>	<b>418</b>
View the Misconfigurations Dashboard .....	419
View the Vulnerabilities Dashboard .....	423
View and Download Compliance Report .....	425
<b>Tenable Cloud Security Settings .....</b>	<b>428</b>
<b>Troubleshooting Issues with Tenable Cloud Security .....</b>	<b>430</b>
Not Able to Find your Repository? .....	431
Seeing Duplicate Repositories? .....	435
Cloud Accounts cannot be Associated with this Project .....	436
Auto-Remediation not Working with On-Premises Scanner .....	437



---

# Welcome to Tenable Cloud Security

---

Tenable Cloud Security (formerly known as Tenable.cs) is designed to scan short-lived and long-lived multi-cloud instances and the infrastructure-as-code (IaC) you use to provision them. Tenable Cloud Security displays the vulnerabilities, misconfigurations, policy violations, breach paths, configuration drift, and remediation steps in unified dashboards that all DevSecOps teams can use.

## How Tenable Cloud Security Works

Tenable Cloud Security connects to your cloud providers to scan your assets. Tenable Cloud Security scans your cloud for security risks and compliance violations without installing any agents into your runtime infrastructure. It also monitors the infrastructure deployments across AWS, Microsoft Azure, and GCP to alert any changes in production that can introduce cloud posture drift.

Connections to code repositories allow you to scan provisioning code and runtimes together.

The key features of Tenable Cloud Security are:

- **Agentless Assessment** – Tenable Cloud Security scans AWS workloads for security risks, compliance violations, and configuration drift without installing any agents into your runtime infrastructure. It securely scans your instance resources inside your own environment. For more information, see [Agentless Assessment](#).
- **Cloud Security Posture Management (CSPM)** – Tenable Cloud Security continuously monitors cloud infrastructure for vulnerabilities, policy gaps, and configuration problems. For more information, see [Connect Cloud Accounts](#).
- **Code scanning** – Tenable Cloud Security scans Terraform and other code used to provision cloud systems on developers' machines before it is checked in to code repositories (GitHub, Bitbucket, GitLab) or in the code repositories themselves. For more information, see [Connect Repositories](#).
- **CI/CD integration** – Tenable Cloud Security integrates with Jenkins, Jira, and other CI/CD tools to monitor builds and prevent misconfigurations before code is built or deployed. For more information, see [Configure CI/CD Integrations](#).
- **Tenable Vulnerability Management integration** – Tenable Cloud Security sends the scan data to Tenable Vulnerability Management to display the results on the reporting and



remediation dashboards of Tenable Vulnerability Management. For more information, see [Findings in Tenable Vulnerability Management](#).

For more information about using Tenable Cloud Security, see [Getting Started with Tenable Cloud Security](#).

**Note:** Tenable Cloud Security can be purchased alone or as part of the Tenable One package. For more information, see [Tenable One](#).

## Tenable One Exposure Management Platform

Tenable One is an Exposure Management Platform to help organizations gain visibility across the modern attack surface, focus efforts to prevent likely attacks and accurately communicate cyber risk to support optimal business performance.

The platform combines the broadest vulnerability coverage spanning IT assets, cloud resources, containers, web apps and identity systems, builds on the speed and breadth of vulnerability coverage from Tenable Research and adds comprehensive analytics to prioritize actions and communicate cyber risk. Tenable One allows organizations to:

- Gain comprehensive visibility across the modern attack surface
- Anticipate threats and prioritize efforts to prevent attacks
- Communicate cyber risk to make better decisions

Tenable Cloud Security exists as a standalone product, or can be purchased as part of the Tenable One Exposure Management platform.

**Tip:** For additional information on getting started with Tenable One products, check out the [Tenable One Deployment Guide](#).



# System Requirements

This topic lists the system requirements for Tenable Cloud Security.

## Display Settings

Supported Browsers	Minimum Screen Resolution
Google Chrome	1440 x 1024

## On-Premise Code Scanner Display Settings

Supported Browsers	Minimum Screen Resolution
Google Chrome	1440 x 1024
Microsoft Edge	1440 x 1024

## Hardware Requirements for On-Premise Code Scanner

- You must have a virtual machine or system with the following minimum requirements:
  - A virtual machine with 4 GB RAM
  - 20 GB Solid State Drive (SSD)
  - Ubuntu 18 or later

Examples of virtual machine include Amazon Elastic Compute Cloud (Amazon EC2) instance, Azure virtual machine, VMware, and so on.

## Command Line Interface (CLI) Requirements

- macOS 10.15 (Catalina or later)
- Microsoft Windows 10 or later
- Linux
- [Terraform](#)
- [Terrascan](#)



## Role-Based Access Control

Role-Based Access Control (RBAC) defines the activities that a user can perform in the associated projects and on the Tenable Cloud Security console. Create users for Tenable Cloud Security and then assign roles to the users from Tenable Vulnerability Management. For more information about user roles in Tenable Vulnerability Management, see [User Roles](#).

Entity	Task	Viewer	Operator	Administrator
Project	Create		✓	✓
	Modify		✓	✓
	Delete		✓	✓
	View	✓	✓	✓
Policies and Policy Groups	View	✓	✓	✓
	Export	✓	✓	✓
Custom policies	Create			✓
	Modify			✓
	Delete			✓
	View	✓	✓	✓
Cloud accounts	Add		✓	✓
	Remove		✓	✓
Repositories	Add		✓	✓
	Remove		✓	✓
Pipeline	Run	✓	✓	✓
Kubernetes cluster	Scan using CLI, Helm charts		✓	✓



<b>Integrations</b>	Add		✓	✓
	Remove		✓	✓
<b>Scans</b>	Run		✓	✓
	Schedule		✓	✓
<b>Findings (mis-configurations and vulnerabilities)</b>	View findings, tickets, pull requests	✓	✓	✓
	Ignore		✓	✓
	Unignore		✓	✓
	Create Ticket		✓	✓
	Create Pull Request		✓	✓
	Export	✓	✓	✓
<b>Alerts and Alert Rules</b>	Configure		✓	✓
	View	✓	✓	✓
<b>Dashboards</b>	View	✓	✓	✓
<b>Reports</b>	View	✓	✓	✓
	Export to CSV		✓	✓
<b>User Management</b>	Not applicable for Tenable Cloud Security. <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"><b>Note:</b> You must create and manage users for Tenable Cloud Security from Tenable Vul-</div>	NA	NA	NA



nerability Management. For more information about how user roles in Tenable Cloud Security map to corresponding roles in Tenable Vulnerability Management, see [User Role Mapping between Tenable Vulnerability Management and Tenable Cloud Security](#).



---

## User Role Mapping between Tenable Vulnerability Management and Tenable Cloud Security

---

User roles in Tenable Cloud Security map to corresponding roles in Tenable Vulnerability Management.

User Role Name in Tenable Vulnerability Management	User Role Name in Tenable Cloud Security
Basic	Viewer
Scan Operator	Operator
Standard	Operator
Scan Manager	Operator
Administrator	Administrator

For user role permissions in Tenable Cloud Security, see [Role-Based Access Control](#).

For user role permissions in Tenable Vulnerability Management, see [User Roles](#).





---

# Access Tenable Cloud Security

---

You can connect to Tenable Cloud Security from the **Workspace** page.

Before you begin:

- Obtain credentials for your Tenable Cloud Security user account.
- Review the Tenable Cloud Security requirements described in [System Requirements](#).

To connect to Tenable Cloud Security:

1. In a supported browser, navigate to <https://cloud.tenable.com/>.


The login page appears.

2. Type your **Username** and **Password** credentials.
3. Click **Login**.

The **Workspace** page appears.

4. Click the **Tenable Cloud Security** tile.

The **Tenable Cloud Security** page opens. By default, a dashboard appears that displays various statistics.

**Note:** To access the **Workspace** menu from any page in any Tenable cloud product, in the upper-right corner, click the  button.



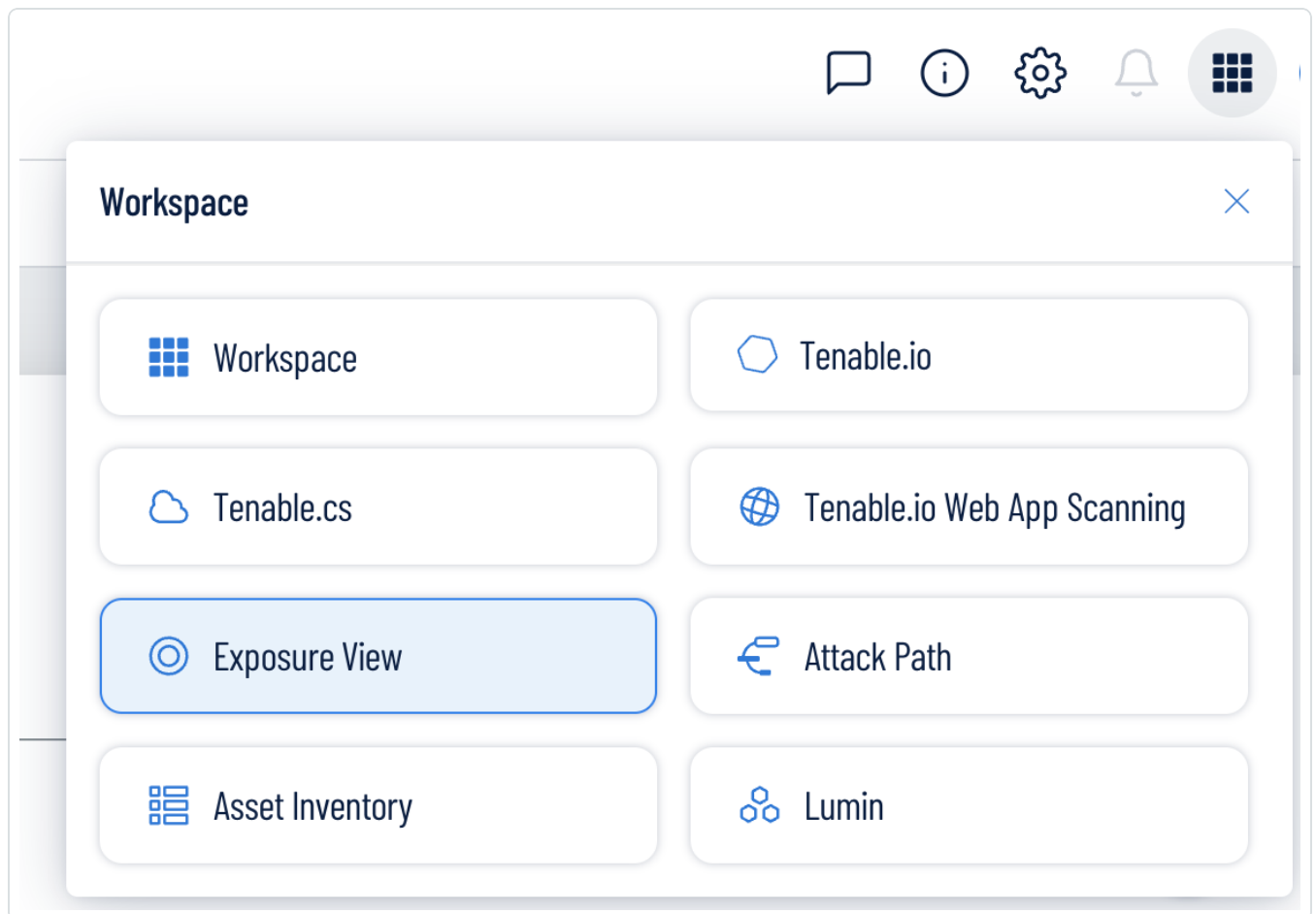
## Access the Workspace

On the **Workspace** page and in the **Workspace** menu, you can view and access all of your Tenable products in one location.

To access the Workspace menu:

1. On any page, in the upper-right corner, click the  button.

The **Workspace** menu appears and displays all of your Tenable products.



2. Click on a product name to navigate to that product's home page.

To access the full Workspace page:

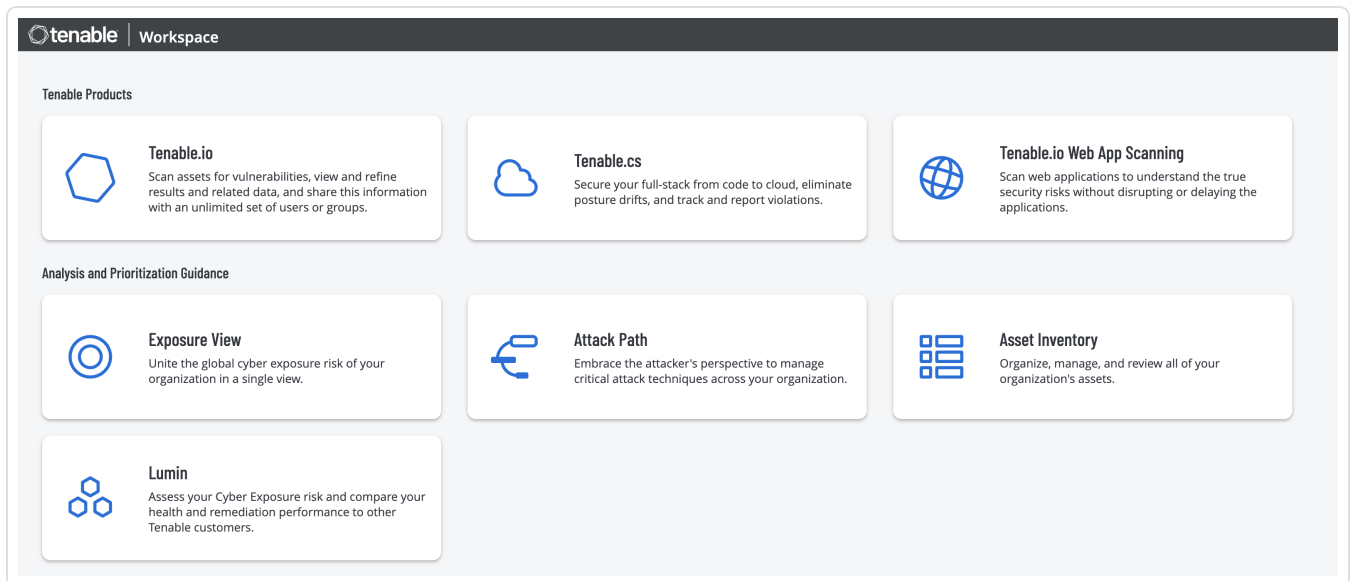


1. Do one of the following:

- [Log in](#) to Tenable Cloud Security.
- [Access](#) the **Workspace** menu.

a. In the **Workspace** menu, click  **Workspace**.

The full **Workspace** page appears and displays all of your Tenable products.



2. Click on a product name to navigate to that product's home page.



---

# Getting Started with Tenable Cloud Security

---

This section provides the getting started sequence to perform cloud and IaC scans in Tenable Cloud Security.

Before you begin:

- Review the following requirements:
  - [System Requirements](#)
  - [Role-Based Access Control](#)
- Ensure that you have provided the required permissions and access for onboarding your cloud accounts.

For more information, see [Connect Cloud Accounts](#).

For any type of scan, perform the following initial steps:

1. [Create a Project](#).

In Tenable Cloud Security, you can group resources, such as repositories and cloud accounts, into projects. Projects allow you to monitor, analyze, and manage all your resources at once.

2. Configure [policies](#) for your scan projects.

Tenable Cloud Security uses policies to identify vulnerabilities present on cloud resources. Tenable Cloud Security comes with built-in policies and policy groups for all cloud providers. By default, Tenable Cloud Security associates policies to your project depending on the resources added to the project. You can also [associate other policies](#) to your project or [create custom policies](#).

3. [Integrate with alert and notification systems](#).

Tenable Cloud Security provides options for you to set up alerts in every project. With alerts, you can enable Tenable Cloud Security to notify users with a summary of key events of the project. Tenable Cloud Security allows you to integrate with email, Slack, Splunk, Microsoft Teams, Jira, and AWS SNS.

What to do next:

Depending on the type of resources, do one or more of the following:



---

- [Cloud Scan Workflow](#)

Tenable Cloud Security scans your cloud resources for security compliance and identify violations. Tenable Cloud Security supports connecting to AWS, Microsoft Azure, and Google Cloud Platform cloud service providers.

- [IaC Scan Workflow](#)

Infrastructure as Code (IaC) scan is scanning your IaC configuration files for known vulnerabilities. Tenable Cloud Security supports IaC scan for Terraform, Terragrunt, CloudFormation, Kubernetes YAML, Kustomize YAML, Helm Chart, and Azure Resource Manager (ARM).



---

# Cloud Scan Workflow

---

Tenable Cloud Security scans your cloud resources for security compliance and identify violations. When you connect your cloud services, you can select the required virtual private clouds (VPCs).

For a detailed workflow for onboarding cloud accounts, see the following Quick Reference Guides:

- [Onboarding AWS Accounts](#)
- [Onboarding Azure Accounts](#)
- [Onboarding GCP Accounts](#)

For vulnerability scanning, perform an [Agentless Assessment](#).

Before you begin:

- Perform the steps in [Getting Started with Tenable Cloud Security](#).

To perform a cloud scan:

1. [Connect your cloud accounts](#).

You can connect the following cloud services to Tenable Cloud Security:

- [Connect AWS](#).
  - [Connect Microsoft Azure](#).
  - [Connect Google Cloud Platform \(GCP\)](#).
2. (Recommended) [Configure cloud scan](#) to define the resources to scan and to schedule scan intervals.
  3. View the [Tenable Cloud Security dashboard](#) to see the analytics for all projects and timelines.
  4. [Analyze the failing policies](#).

Tenable Cloud Security displays failing policies when resources fail to comply with the configured policies.

**Tip:** You can also view the vulnerability findings for your cloud resources from Tenable Vulnerability Management. For more information, see [Vulnerabilities](#).



---

5. Perform workflow actions for the impacted resources. Workflow actions allow organizational users to configure and manage alerting and ticketing.

- [Escalate an Issue](#)
- [Create a Ticket](#)
- [Ignore Misconfigurations](#)

6. [View cloud to cloud drifts.](#)

The changes you make to the configuration of any unmapped resource in the cloud account create a cloud-to-cloud drift. An unmapped resource is any resource in the cloud that does not have a matching configuration in IaC. For unmapped resources, your cloud configuration may differ from the previous configuration on the cloud, which creates a cloud-to-cloud drift.

7. [View compliance reports.](#)

The Tenable Cloud Security **Reports** page displays the compliance reports for all resources.



---

# laC Scan Workflow

---

Infrastructure as Code (laC) scan is scanning your laC configuration files for known vulnerabilities. Tenable Cloud Security supports laC scan for Terraform, Terragrunt, CloudFormation, Kubernetes YAML, Kustomize YAML, Helm Chart, and Azure Resource Manager (ARM).

Before you begin:

- Perform the steps in [Getting Started with Tenable Cloud Security](#).

To perform an laC scan:

The laC scan workflow consists of the following high-level steps:

1. [Integrate with Repositories](#).
2. [Analyze and Remediate laC Scan Issues](#).

## Integrate with Repositories

First integrate your laC repository with Tenable Cloud Security. Tenable Cloud Security allows you to perform laC scans for the following types of repositories:

- **Code repositories:** You can scan your laC files in your code repositories by connecting to your Source Code Management (SCM) providers. Tenable Cloud Security supports the laC scans for Bitbucket, GitHub, GitLab, Azure DevOps, and AWS CodeCommit.
- **CI/CD applications:** Tenable Cloud Security integrates with your CI/CD provider and scans your laC files for violations in your build pipeline. Tenable Cloud Security supports integration with Terraform Cloud, Jenkins, Azure DevOps, and CircleCI.
- **On-premises code repositories:** If your code repositories are behind the firewall, you can use Tenable Cloud Security on-premises code scanner to connect to the repository. The Tenable Cloud Security code scanner scans the repository within the firewall-bound network and sends the processed data to Tenable Cloud Security services for reporting in Tenable Cloud Security.
- **Local repositories:** You can use the Tenable Cloud Security CLI to scan the code in your local machine.

The following table provides the steps for integrating repositories with Tenable Cloud Security.





Repository	Integration Procedure
Code repositories	<ol style="list-style-type: none"><li>1. <a href="#">Connect your repositories</a> and grant Tenable Cloud Security access to your repository.</li></ol> <p>Tenable Cloud Security supports the following SCM providers:</p> <ul style="list-style-type: none"><li>• <a href="#">Bitbucket</a></li><li>• <a href="#">GitHub</a></li><li>• <a href="#">GitLab</a></li><li>• <a href="#">Azure DevOps</a></li><li>• <a href="#">AWS Code Commit</a></li></ul>
CI/CD applications	<ol style="list-style-type: none"><li>1. If you do not want your CI/CD tool to deploy cloud resources in case Tenable Cloud Security detects violations in your IaC, create a policy with the <b>Enforce</b> mode. For more information, see <a href="#">Policy Modes</a>.</li><li>2. <a href="#">Generate an API token</a> to authenticate your CI/CD application with Tenable Cloud Security.</li><li>3. Integrate with the CI/CD tool. Tenable Cloud Security supports integration with the following tools:<ul style="list-style-type: none"><li>• <a href="#">Integrate with Terraform Cloud</a></li><li>• <a href="#">Integrate with Jenkins Pipeline</a></li><li>• <a href="#">Integrate with Azure DevOps Pipeline</a></li></ul></li></ol>
On-premises repositories	<ol style="list-style-type: none"><li>1. <a href="#">Deploy an On-Premises Code Scanner</a>.</li></ol> <p>Tenable Cloud Security also supports the on-premises scanning of the following enterprise IaCs:</p> <ul style="list-style-type: none"><li>• <a href="#">Use an On-Premises Code Scanner to Scan Bitbucket Server IaCs</a></li><li>• <a href="#">Use an On-Premises Code Scanner to Scan GitHub Enterprise IaCs</a></li></ul>



Repository	Integration Procedure
	<ul style="list-style-type: none"><li>• <a href="#">Use an On-Premises Code Scanner to Scan GitLab Server IaCs</a></li></ul>
Local repositories	1. Install and set up the command-line interface. <a href="#">Set up Code Analysis Using CLI</a>

## Analyze and Remediate IaC Scan Issues

After you have integrated your repositories with Tenable Cloud Security, you can perform the following steps to monitor, analyze, and remediate your IaC scans.

1. View the [Tenable Cloud Security dashboard](#) to see the analytics for all projects and timelines.
2. [Analyze the failing policies.](#)

Tenable Cloud Security displays failing policies when resources fail to comply with the configured policies.

3. Perform workflow actions and remediate the impacted resources.

Workflow actions allow organizational users to configure and manage alerting and ticketing. You can also generate pull requests with proposed fixes to remediate build-time issues.

- [Escalate an Issue](#)
- [Create a Ticket](#)
- [Ignore Misconfigurations](#)
- [Auto-Remediation](#)
- [Inline Reviews](#)
- [Create a Pull Request for an Issue](#)
- [View and Remediate the Line of Change in IaC](#)

4. [View code to cloud drifts.](#)

Tenable Cloud Security maps your IaC resources to the corresponding cloud resources in your cloud account. For mapped resources, your IaC code configuration may differ from that on the cloud, which raises a code to cloud drift.



---

5. [View compliance reports.](#)

The Tenable Cloud Security **Reports** page displays the compliance reports for all resources.



---

## Create a Project

---

In Tenable Cloud Security Console, you can group resources, such as repositories and cloud accounts, into projects. Projects allow you to monitor, analyze, and manage all your resources at once.

To create a project:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click  > **Project**.
3. In the **Give the project a name** section, type a name for your project.

**Note:** A project name can have a maximum of 25 characters.

4. Click **Continue**.
5. In the **Choose provider** section, select the cloud service provider.
6. Click **Create**.

A confirmation message appears and Tenable Cloud Security creates the project. You can view the new project on the **Projects & Connections** page.

For more information on setting up projects, see [Connect Cloud Accounts](#) and [Connect Repositories](#).

---

## Projects and Connections

---


On the **Projects & Connections** page, you can view the details of projects, repositories, cloud accounts, Kubernetes clusters, and pipelines.



# View Projects and Connections

1. From the home page, click the **Projects & Connections** tab.

The **Projects & Connections** page appears and shows the following tabs:

Tab	Description
<b>Projects</b>	<p>The <b>Projects</b> tab lists the following information:</p> <ul style="list-style-type: none"><li>• <b>Projects</b> – List of all projects.</li><li>• <b>Resources</b> – The total number of IaC and cloud resources in that project. Hover over the total resources to view the number of IaC and cloud resources.</li><li>• <b>Vulnerabilities</b> – Vulnerabilities detected during the Agentless Assessment of AWS EC2 instances and Azure virtual machines. For more information, see <a href="#">View Misconfigurations</a>.</li><li>• <b>Misconfigurations</b> – The number of non-compliant policies for resources in that project.</li><li>• <b>Drifts</b> – The number of IaC and cloud drifts. For more information, see <a href="#">Set up Drift Analysis</a>.</li><li>• <b>Scan status</b> – The scan have one of the following statuses:<ul style="list-style-type: none"><li>• <b>Canceled</b> – Scan was canceled when it was running.</li><li>• <b>Completed with exceptions</b> – Scan completed, but with errors.</li></ul>To re-assess or rescan a project that is completed with errors:<ol style="list-style-type: none"><li>1. In the <b>Projects</b> tab, click the  icon on the <b>Status</b> column.</li></ol>The <b>Exceptions</b> dialog box appears with the list of</li></ul>



	<p>failed resources grouped by <b>Account ID</b>. You can view and sort the exceptions in each cloud account by <b>Failed resource type</b>, <b>Resource group</b> (for Azure), <b>Region</b> (for AWS), and <b>Resource count</b>.</p> <p>2. Click <b>Re-assess</b> to rescan the selected cloud account or <b>Re-assess all</b> to rescan all failed cloud accounts.</p> <p>Re-assessing scans the following if the scan has exceptions across these parameters:</p> <ul style="list-style-type: none"><li>• Accounts, projects, and subscriptions</li><li>• Resource type</li><li>• Regions or Resource group</li></ul> <ul style="list-style-type: none"><li>• <b>Failed</b> – Scan has failed.</li><li>• <b>In progress</b> – Scan is in progress.</li><li>• <b>Not scanned</b> – Project is empty and has not been scanned.</li><li>• <b>Successful</b> – Scan is completed successfully.</li></ul> <p>To manage projects, see <a href="#">Manage Projects</a>.</p>
<b>Repositories</b>	<p>The <b>Repositories</b> tab shows the list of repositories, folder path, all projects associated with the repository, the number of misconfigurations, and the resources they contain. To manage repositories, see <a href="#">Manage Repositories</a>.</p>
<b>Cloud Accounts</b>	<p>The <b>Cloud Accounts</b> tab lists all cloud accounts, management unit, the project they are linked to, the number of resources in each account, the number of failing policies, and the current status of the scan. Tenable Cloud Security allows you to discover cloud accounts automatically. For more information, see <a href="#">Discover Cloud Accounts</a>.</p>




<b>K8s Clusters</b>	The <b>Kubernetes Clusters</b> tab lists all the clusters, their parent project, the number of failing policies, resources, and the associated cloud account.
<b>Pipelines</b>	The <b>Pipelines</b> tab lists all the pipeline repositories, the repository owner, the number of failures, and the last run status of the scan.



# Manage Projects

To filter a project:

1. Click the  **Filters** icon to open the **Filter Projects** box.
2. Select the following filters as needed.

Filter	Description
Cloud provider	Filters the list by cloud providers – AWS, Azure, and GCP.
Cloud accounts	Filters the list by cloud accounts.
Scan status	Filters by the scan status.

3. Click **Apply**.

Tenable Cloud Security shows the list of projects after applying the filter criteria.

To start a scan for a project:


You can run two types of scans in Tenable Cloud Security:

- IaC scan – [Connect a repository](#) to the project to run an IaC scan.
- Cloud scan – [Connect cloud accounts](#) to the project and run [cloud scans](#).

To edit a project:

1. Click the project that you want to edit.

The project details pane appears.

2. Click the  icon and edit any of the following configurations for the project:
  - Name of the project
  - Repositories
  - Cloud accounts





**Note:** When you remove a cloud account from a project, the cloud account and the associated findings are not deleted from Tenable Cloud Security. The cloud account is only disassociated from the project. You can view the cloud account in the **Cloud accounts** tab. To delete a cloud account, [see Delete a Cloud Account](#).

- K8s Clusters
- Active policy groups
- [Alerts](#)
- Exclude resources
- Exclude drifts for selected resources

**Note:** After editing a project, rescan the project to update the findings based on the current settings.

To edit policies associated with a project:

1. Select the check box next to the project that you want to edit.

Tenable Cloud Security enables the **More Actions** button.

2. Click **More Actions > Manage Policies**.

The **Edit policy group** window appears.

3. Select the required policies from the list.

4. Click **Save**.

A confirmation message appears.

5. Confirm the policy additions.

Tenable Cloud Security initiates the scan with the newly added policies.

To delete a project:

1. Select one or more projects that you want to delete.

Tenable Cloud Security enables the **More Actions** button.

2. Click **More Actions > Delete**.



A confirmation message appears.

3. Click **Yes**.

A confirmation message appears and Tenable Cloud Security deletes the project. When Tenable Cloud Security deletes a project, Tenable Cloud Security deletes all findings associated with the project.

To set or reset a baseline:

Tenable Cloud Security allows you to set a baseline for a project by recording the time stamp of the scan when the baseline is set. A baseline allows you to compare and identify cloud-to-cloud drifts between scans. For more information, see [Set a Baseline for a Project](#).



---

## Manage Repositories

---

To start a scan for a repository:

1. Select the repository that you want to scan.
2. In the **Scans** column, click **Run Scan**.

When the scan completes, a confirmation message appears.

To assign or unassign a repository to a project:

1. Do one of the following:
  - To assign or unassign a project for a single repository:
    - a. In the row corresponding to the repository to which you want to edit the assigned projects, click **⋮ > Manage project**.
  - To assign multiple repositories:
    - a. Select one or more repository that you want to assign.

Tenable Cloud Security enables the **More Actions** button.
    - b. Click **More Actions > Manage projects**.

The **Projects** dialog box appears.
2. Select the required project to assign to the repository or clear the check box corresponding to a selected project to unassign the repository.
3. Click **Save**.

Tenable Cloud Security assigns the selected repository to the project.

To delete a repository:



1. Do one of the following:

- To delete a single repository:
  - a. In the row corresponding to the repository that you want to delete, click **⋮ > Delete repository**.
- To delete multiple repositories:
  - a. Select one or more repository that you want to delete.  
Tenable Cloud Security enables the **More Actions** button.
  - b. Click **More Actions > Delete repositories**.

A confirmation message appears

2. Click **Yes** to confirm.

Tenable Cloud Security deletes the repository and its associated data.



---

## Connect Cloud Accounts

---

To scan cloud resources for security compliance, you must connect your cloud services to Tenable Cloud Security Console. When you connect your cloud services, you can select the required virtual private clouds (VPCs).

Tenable Cloud Security provides the following flows for onboarding cloud accounts:

- Auto-discovery: [Discover Cloud Accounts](#)
- On-demand basis: For more information, see the following topics:
  - [Onboard AWS Accounts](#)
  - [Onboard an Azure Account](#)
  - [Onboard a GCP Service Account](#)



---

## Onboard AWS Accounts

---

You can connect your single, multiple or all Amazon Web Services (AWS) accounts as a part of your AWS project. For a detailed workflow for onboarding AWS accounts, see the [Tenable Cloud Security Quick Reference Guide: Onboarding AWS Accounts](#).

To onboard AWS accounts in Tenable Cloud Security, each AWS account being onboarded must be associated with a role granting the **ReadOnlyAccess** policy to the Tenable AWS account. Tenable Cloud Security requires the Role ARN and External ID to onboard the AWS account. When onboarding an AWS Organization, Tenable Cloud Security provides you with a StackSet that recursively adds that role to all accounts under the organization. Tenable Cloud Security requires the StackSet ARN to onboard the organization. For more information, see the following topics:

- To connect multiple or all AWS accounts, see [Onboard an AWS Organization](#).
- To connect a single AWS account, see [Onboard an AWS Account](#).



---

# Set Up Read-Only Access to the AWS Account

---

To read the resources in the Amazon Web Services (AWS) cloud account, Tenable Cloud Security requires appropriate permissions. Tenable Cloud Security recommends provisioning an IAM (Identity and Access Management) role in the target AWS cloud account and configuring it for Tenable Cloud Security to read the resources in the same account. When onboarding an AWS organization account, create an IAM role for the management account.

You can create the role in the following ways:

- [Create a read-only role manually](#)
- [Create a read-only role using a script](#)
- [Create a read-only role using a CloudFormation Template](#)

## Create a read-only role manually

You can create a read-only role manually from the AWS management console.

Before you begin:

- Log in to the AWS web console with a user account with permission to create IAM roles.

For more information about IAM roles, see Amazon's [AWS Identity and Access Management User Guide](#).

To create a read-only role manually:

1. In the AWS web console, go to **Identity and Access Management (IAM)**.
2. On the left navigation pane, click **Roles**.  
The **Roles** page appears.
3. Click **Create Role**.  
The Create Role wizard appears.
4. In the **Select trusted entity** page, do the following:



- a. In the **Trusted entity type** section, select **AWS Account**.
- b. In the **An AWS Account** section, select **Another AWS Account**.
- c. In the **Account ID** box, type **012615275169**.

**Note:** 012615275169 is the account ID of the Tenable AWS account that you are establishing a trust relationship with to support AWS role delegation.

- d. Under **Options**, click the **Require External ID** check box and type your Tenable Vulnerability Management Container UUID in the External ID box.

**Note:** In Tenable Vulnerability Management, navigate to **Settings > License** to get your container UUID. For more information, see [View Information about Your Tenable Vulnerability Management Instance](#).

- e. Click **Next**.

Step 2  
Add permissions

Step 3  
Name, review, and create

### Trusted entity type

- AWS service  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

### An AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

- This account (576993307204)
- Another AWS account

Account ID  
Identifier of the account that can use this role

012615275169

Account ID is a 12-digit number.

Options

- Require external ID (Best practice when a third party will assume this role)  
You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. [Learn more](#)

External ID

<insertTicContainerUUID>

**i** Important: The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account iam:AssumeRole calls. [Learn more](#)

- Require MFA  
Requires that the assuming entity use multi-factor authentication.

5. On the **Add permissions** page, perform the following:





- a. Search for **ReadOnlyAccess** in the search box.

**Tip:** Filtering for “ReadOnlyAccess” by role name might return many entries. Apply the “Used as: Used as permissions policy” filter along with the role name “ReadOnlyAccess” to narrow down the search results.

- b. Select the **ReadOnlyAccess** check box.

Add permissions

**Permissions policies** (Selected 1/878)  
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter 11 matches

“ReadOnlyAccess” × Used as: Used as permissions policy × Clear filters

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	ADM-POL-ArtifactReadOnlyAccess	Customer managed	This policy gives read-only access to pull reports from artifact
<input type="checkbox"/>	AmazonEC2ReadOnlyAccess	AWS managed	Provides read only access to Amazon EC2 via the AWS Management Console.
<input type="checkbox"/>	AmazonVPCReadOnlyAccess	AWS managed	Provides read only access to Amazon VPC via the AWS Management Console.
<input checked="" type="checkbox"/>	ReadOnlyAccess	AWS managed	Provides read-only access to AWS services and resources.
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets via the AWS Management Console.
<input type="checkbox"/>	ResourceGroupsandTagEditorReadOnlyAc...	AWS managed	Provides access to use Resource Groups and Tag Editor, but does not allow editing of tags via the Tag Editor.
<input type="checkbox"/>	AWSCloudFormationReadOnlyAccess	AWS managed	Provides access to AWS CloudFormation via the AWS Management Console.

For the list of permissions and AWS resources scanned by Tenable Cloud Security with this policy, see [Permissions and Supported Resources for AWS ReadOnlyAccess Policy](#).

- c. For vulnerability scanning with Agentless Assessment, create an inline policy with the following JSON to provide Elastic Block Store permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ebs:List*",
        "ebs:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```



```
} ]
```

d. Select the required policies for the IAM role and click **Next**.

**Note:** The new policy might take some time to get created. Refresh your browser if you do not see the policy in the list of policies.

For information about creating IAM policies, see the [AWS documentation](#).

6. In the **Name, review, and create** page, do the following:
  - a. In the **Role Details** section, type a **Role Name** for the role.
  - b. (Optional) Add a role description in the **Description** box.
  - c. (Optional) Click **Add Tags** to add key-value pairs to AWS resources.
  - d. Click **Create Role**.

The screenshot shows the 'Name, review, and create' page in the AWS IAM console. The role name is 'TenableReadOnlyTrustRole'. The description field is empty. Under 'Step 1: Select trusted entities', a JSON policy is displayed, allowing the role to assume the 'ReadOnlyAccess' role. Under 'Step 2: Add permissions', a table shows the 'ReadOnlyAccess' policy is attached. The 'Tags' section is empty, and the 'Add tag' button is visible. At the bottom right, there are 'Cancel', 'Previous', and 'Create role' buttons.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole",
7       "Principal": {
8         "AWS": "*"
9       }
10      "Condition": {
11        "StringEquals": {
12          "sts:ExternalId": "*"
13        }
14      }
15    }
16  ]
17 }
```

Policy name	Type	Attached as
ReadOnlyAccess	AWS managed	Permissions policy

Tenable Cloud Security now has read-only access to your AWS account.



7. To get the **Role ARN** and **External ID** of this new role for Tenable Cloud Security, do the following:
  - a. On the left navigation pane, click **Roles**.
  - b. Search for the role that you created.
  - c. In the **Summary** section, note the **Role ARN** value.
  - d. Click the **Trust Relationships** tab and note the value of the **ExternalId** field.

IAM > Roles > TenableReadOnlyTrustRole

## TenableReadOnlyTrustRole

### Summary

Creation date  
January 27, 2022, 03:39 (UTC+05:30)

Last activity  
14 days ago

ARN  
[Redacted]

Maximum session duration  
1 hour

Permissions | **Trust relationships** | Tags | Access Advisor | Revoke sessions

### Trusted entities

Entities that can assume this role under specified conditions.

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Principal": {
7-         "AWS": "arn:aws:iam::[Redacted]:role/[Redacted]"
8-       },
9-       "Action": "sts:AssumeRole",
10-      "Condition": {
11-        "StringEquals": {
12-          "sts:ExternalId": "[Redacted]"
13-        }
14-      }
15-    }
16-  ]
17- }
```

8. Note down the following values:
  - **Role ARN**
  - **External ID**

You need these values when onboarding AWS accounts in Tenable Cloud Security.



## Create a Read-Only Role Using a Script

You can run the script provided by Tenable Cloud Security to create an AWS read-only role.

Before you begin:

- You must have the following:
  - Terraform version 12 or higher
  - AWS access key
  - AWS secret key

To create a read-only role using a script:

1. Run the following command:

```
/bin/bash -c "$(curl https://downloads.accurics.com/downloads/io/create_tcs_aws_readonly_role.sh)"
```

2. Provide values for the following parameters, when prompted:

- (Required) **AWS\_ACCESS\_KEY\_ID**: Access key of the AWS account.
- (Required) **AWS\_SECRET\_ACCESS\_KEY**: Secret key of the AWS account.
- (Optional) **Role name suffix**: By default, Tenable Cloud Security creates a role with the name *TenableReadOnlyTrustRole*. Provide an optional suffix to append to this role name. For example, if you provide ACME, the role name is *TenableReadOnlyTrustRoleACME*.
- (Required) **ExternalId**: Provide an alphanumeric string to be used as the External ID of the role. The External ID can contain a minimum of 4 chars and a maximum of 1224 characters. Tenable recommends providing your Tenable Vulnerability Management Container UUID for the External ID.

3. When prompted "**Do you want to perform these actions?**", type **yes** to continue.

Tenable Cloud Security executes the script and creates the read-only role.



```
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_iam_role.read_only: Creating...
aws_iam_role.read_only: Creation complete after 3s [id=TenableReadOnlyTrustRoleTEST]
aws_iam_role_policy_attachment.read_only: Creating...
aws_iam_role_policy_attachment.read_only: Creation complete after 0s [id=TenableReadOnlyTrustRoleTEST-20220930064810139700000001]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:
CustomerExternalId = "XXXXXXX"
role_arn = "arn:aws:iam::XXXXXXXXXXXX:role/TenableReadOnlyTrustRole"
-e \Read Only AWS Role Creation Successful.

Please use the ARN and CUSTOMER_EXTERNAL_ID printed in the terraform output to enable Tenable.cs Cloud Scan.
```

#### 4. Note down the following values:

- Role ARN
- External ID

You need these values when onboarding accounts in AWS.

## Create a read-only role using a CloudFormation Accurics

You can deploy the Tenable Cloud Security stackset to create a read-only role.

Before you begin:

- Log in to the AWS web console.

To create a read-only role using a CloudFormation Accurics:

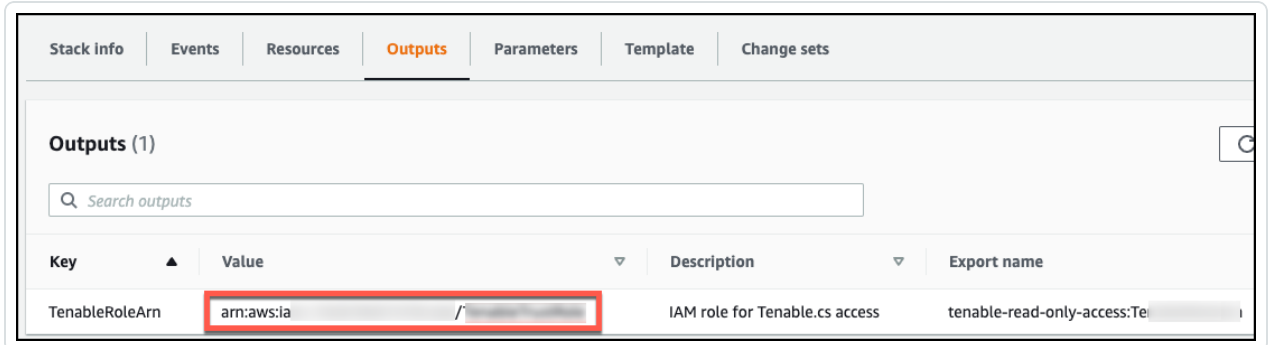
1. Click [here](#) to open the CloudFormation template to deploy a read-only role in AWS.  
Tenable Cloud Security redirects you to the **Quick create stack** page in AWS.
2. Review the parameters in the stack template and update, if required.
3. In the **Capabilities** section, select the **I acknowledge that AWS CloudFormation might create IAM resources with custom names**. check box to confirm creating the IAM resources with required permissions.
4. Click **Create stack**.



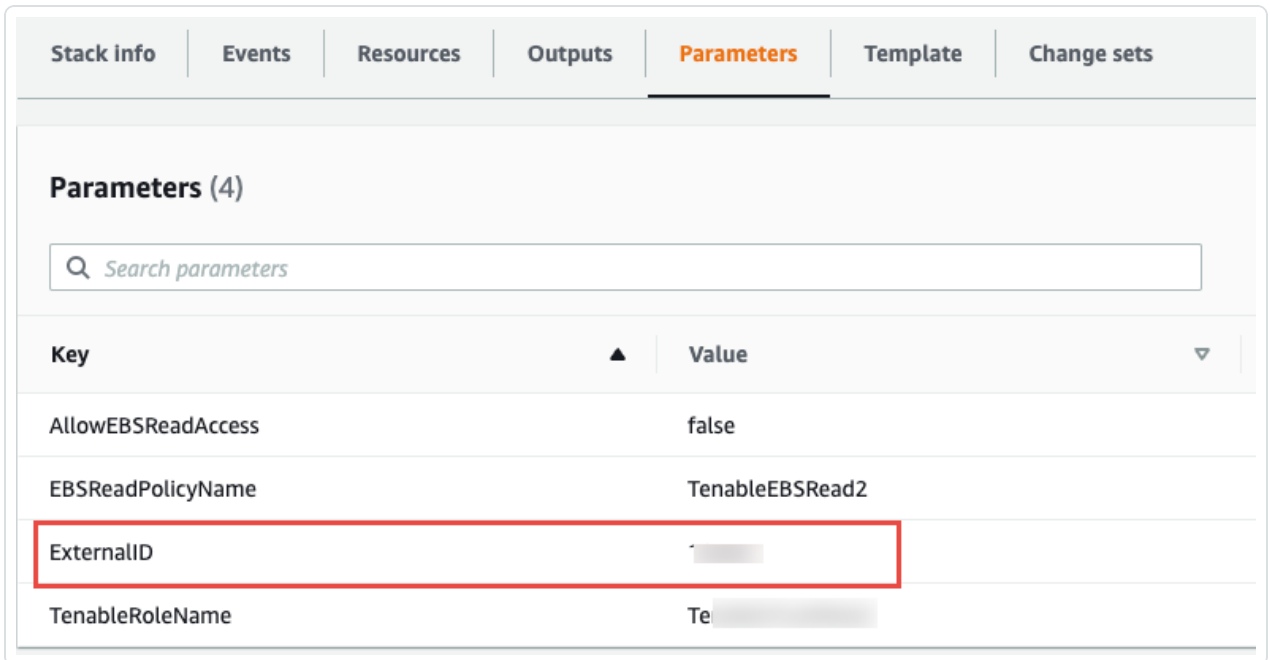
Wait for the stack to get created and its status to become **CREATE\_COMPLETE**.

5. Note down the following values:

- **Role ARN:** Copy the stack ARN of the deployed stack from the **Outputs** tab.



- **External ID:** Copy the **ExternalID** from the **Parameters** tab.



You need these values when onboarding AWS accounts in Tenable Cloud Security.

What to do next:

[Onboard AWS Accounts](#)

You must have the following values for onboarding the AWS account in Tenable Cloud Security:



- Role ARN
- External ID



# Permissions and Supported Resources for AWS ReadOnlyAccess Policy

Tenable Cloud Security requires a read-only role with the **ReadOnlyAccess** policy to scan the resources for misconfigurations. You can also assign the SecurityAudit policy to the read-only role; however, some resources are not scanned when the SecurityAudit policy is used. See the following sections for:

- [Comparison of resources scanned by Tenable Cloud Security for the ReadOnlyAccess and SecurityAudit policies.](#)
- [Permissions defined for the ReadOnlyAccess policy.](#)

## Supported Resources with the ReadOnlyAccess and SecurityAudit Policies

The [ReadOnlyAccess](#) and [SecurityAudit](#) policies are AWS managed policies that can be assigned to the Tenable Cloud Security read-only role. The following table shows the comparison of the resources scanned using the Tenable Cloud Security read-only role when associated with either of these policies:

Terraform Resource	ReadOnlyAccess	SecurityAudit
aws_acmpca_certificate_authority	Yes	Yes
aws_api_gateway_rest_api_policy	Yes	No
aws_apigatewayv2_api	Yes	No
aws_apigatewayv2_stage	Yes	Yes
aws_appautoscaling_policy	Yes	Yes
aws_appmesh_mesh	Yes	Yes
aws_athena_database	Yes	No
aws_athena_workgroup	Yes	Yes





aws_autoscaling_attachment	Yes	Yes
aws_autoscaling_group	Yes	Yes
aws_backup_vault	Yes	Yes
aws_backup_vault_policy	Yes	Yes
aws_budgets_budget	Yes	No
aws_cloudformation_stack	Yes	Yes
aws_cloudtrail	Yes	Yes
aws_cloudwatch_metric_alarm	Yes	Yes
aws_codebuild_project	Yes	Yes
aws_codecommit_repository	Yes	Yes
aws_codedeploy_app	Yes	Yes
aws_codepipeline	Yes	No
aws_codepipeline_webhook	Yes	No
aws_db_instance	Yes	Yes
aws_devicefarm_project	Yes	No
aws_dynamodb_table	Yes	Yes
aws_ebs_snapshot	Yes	Yes
aws_ebs_volume	Yes	Yes
aws_ec2_transit_gateway	Yes	Yes
aws_ec2_transit_gateway_route_table	Yes	Yes
aws_ec2_transit_gateway_	Yes	Yes



vpc_attachment		
aws_ecr_lifecycle_policy	Yes	Yes
aws_ecrpublic_repository	Yes	Yes
aws_ecr_repository	Yes	Yes
aws_ecr_repository_policy	Yes	Yes
aws_ecs_cluster	Yes	Yes
aws_ecs_service	Yes	Yes
aws_ecs_task_definition	Yes	Yes
aws_efs_backup_policy	Yes	<b>No</b>
aws_eip	Yes	Yes
aws_eks_cluster	Yes	Yes
aws_eks_node_group	Yes	Yes
aws_elastic_beanstalk_application	Yes	<b>No</b>
aws_elastic_beanstalk_environment	Yes	<b>No</b>
aws_flow_log	Yes	Yes
aws_iam_access_key	Yes	Yes
aws_iam_account_password_policy	Yes	Yes
aws_iam_group	Yes	Yes
aws_iam_group_policy	Yes	Yes
aws_iam_instance_profile	Yes	Yes
aws_iam_policy	Yes	Yes



aws_iam_user_login_profile	Yes	Yes
aws_iam_user_policy_attachment	Yes	Yes
aws_instance	Yes	Yes
aws_internet_gateway	Yes	Yes
aws_kms_alias	Yes	Yes
aws_kms_key	Yes	Yes
aws_lambda_function	Yes	<b>No</b>
aws_lb	Yes	Yes
aws_lb_listener	Yes	Yes
aws_lb_listener_rule	Yes	Yes
aws_lb_target_group	Yes	Yes
aws_main_route_table_association	Yes	Yes
aws_mq_broker	Yes	Yes
aws_nat_gateway	Yes	Yes
aws_neptune_cluster	Yes	Yes
aws_neptune_cluster_instance	Yes	Yes
aws_network_acl	Yes	Yes
aws_organizations_organization	Yes	Yes
aws_ram_resource_share	Yes	<b>No</b>
aws_rds_cluster	Yes	Yes



aws_rds_cluster_instance	Yes	Yes
aws_redshift_parameter_group	Yes	Yes
aws_route53_query_log	Yes	Yes
aws_route53_record	Yes	Yes
aws_route53_zone	Yes	Yes
aws_route_table	Yes	Yes
aws_route_table_association	Yes	Yes
aws_s3_bucket	Yes	No
aws_s3_bucket_policy	Yes	Yes
aws_sagemaker_notebook_instance	Yes	Yes
aws_secretsmanager_secret	Yes	Yes
aws_security_group	Yes	Yes
aws_ses_configuration_set	Yes	No
aws_ses_email_identity	Yes	Yes
aws_sns_topic	Yes	Yes
aws_sqs_queue	Yes	Yes
aws_storagegateway_nfs_file_share	Yes	Yes
aws_subnet	Yes	Yes
aws_volume_attachment	Yes	Yes
aws_vpc	Yes	Yes



aws_xray_encryption_con- fig	Yes	Yes
---------------------------------	-----	-----

## Permissions for the ReadOnlyAccess Policy

The following JSON lists the permissions for the **ReadOnlyAccess** policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "a4b:Get*",
        "a4b:List*",
        "a4b:Search*",
        "access-analyzer:GetAccessPreview",
        "access-analyzer:GetAnalyzedResource",
        "access-analyzer:GetAnalyzer",
        "access-analyzer:GetArchiveRule",
        "access-analyzer:GetFinding",
        "access-analyzer:GetGeneratedPolicy",
        "access-analyzer:ListAccessPreviewFindings",
        "access-analyzer:ListAccessPreviews",
        "access-analyzer:ListAnalyzedResources",
        "access-analyzer:ListAnalyzers",
        "access-analyzer:ListArchiveRules",
        "access-analyzer:ListFindings",
        "access-analyzer:ListPolicyGenerations",
        "access-analyzer:ListTagsForResource",
        "access-analyzer:ValidatePolicy",
        "account:GetAlternateContact",
        "account:GetContactInformation",
        "account:GetRegionOptStatus",
        "account:ListRegions",
        "acm-pca:Describe*",
        "acm-pca:Get*",
        "acm-pca:List*",
        "acm:Describe*",
        "acm:Get*",
        "acm:List*",
        "airflow:ListEnvironments",
        "airflow:ListTagsForResource",
        "amplify:GetApp",
        "amplify:GetBranch",
        "amplify:GetDomainAssociation",
        "amplify:GetJob",
        "amplify:ListApps",
        "amplify:ListBranches",
        "amplify:ListDomainAssociations",
        "amplify:ListJobs",
        "apigateway:GET",
        "appconfig:GetApplication",
        "appconfig:GetConfiguration",
        "appconfig:GetConfigurationProfile",
```



```
"appconfig:GetDeployment",
"appconfig:GetDeploymentStrategy",
"appconfig:GetEnvironment",
"appconfig:GetHostedConfigurationVersion",
"appconfig:ListApplications",
"appconfig:ListConfigurationProfiles",
"appconfig:ListDeployments",
"appconfig:ListDeploymentStrategies",
"appconfig:ListEnvironments",
"appconfig:ListHostedConfigurationVersions",
"appconfig:ListTagsForResource",
"appflow:DescribeConnector",
"appflow:DescribeConnectorEntity",
"appflow:DescribeConnectorFields",
"appflow:DescribeConnectorProfiles",
"appflow:DescribeConnectors",
"appflow:DescribeFlow",
"appflow:DescribeFlowExecution",
"appflow:DescribeFlowExecutionRecords",
"appflow:DescribeFlows",
"appflow:ListConnectorEntities",
"appflow:ListConnectorFields",
"appflow:ListConnectors",
"appflow:ListFlows",
"appflow:ListTagsForResource",
"application-autoscaling:Describe*",
"applicationinsights:Describe*",
"applicationinsights:List*",
"appmesh:Describe*",
"appmesh:List*",
"apprunner:DescribeAutoScalingConfiguration",
"apprunner:DescribeCustomDomains",
"apprunner:DescribeObservabilityConfiguration",
"apprunner:DescribeService",
"apprunner:DescribeVpcConnector",
"apprunner:DescribeVpcIngressConnection",
"apprunner:ListAutoScalingConfigurations",
"apprunner:ListConnections",
"apprunner:ListObservabilityConfigurations",
"apprunner:ListOperations",
"apprunner:ListServices",
"apprunner:ListTagsForResource",
"apprunner:ListVpcConnectors",
"apprunner:ListVpcIngressConnections",
"appstream:Describe*",
"appstream:List*",
"appsync:Get*",
"appsync:List*",
"aps:DescribeAlertManagerDefinition",
"aps:DescribeRuleGroupsNamespace",
"aps:DescribeWorkspace",
"aps:GetAlertManagerSilence",
"aps:GetAlertManagerStatus",
"aps:GetLabels",
"aps:GetMetricMetadata",
"aps:GetSeries",
"aps:ListAlertManagerAlertGroups",
"aps:ListAlertManagerAlerts",
"aps:ListAlertManagerReceivers",
"aps:ListAlertManagerSilences",
```



```
"aps:ListAlerts",
"aps:ListRuleGroupsNamespaces",
"aps:ListRules",
"aps:ListTagsForResource",
"aps:ListWorkspaces",
"aps:QueryMetrics",
"arc-zonal-shift:GetManagedResource",
"arc-zonal-shift:ListManagedResources",
"arc-zonal-shift:ListZonalShifts",
"artifact:GetReport",
"artifact:GetReportMetadata",
"artifact:GetTermForReport",
"artifact:ListReports",
"athena:Batch*",
"athena:Get*",
"athena:List*",
"auditmanager:GetAccountStatus",
"auditmanager:GetAssessment",
"auditmanager:GetAssessmentFramework",
"auditmanager:GetAssessmentReportUrl",
"auditmanager:GetChangeLogs",
"auditmanager:GetControl",
"auditmanager:GetDelegations",
"auditmanager:GetEvidence",
"auditmanager:GetEvidenceByEvidenceFolder",
"auditmanager:GetEvidenceFolder",
"auditmanager:GetEvidenceFoldersByAssessment",
"auditmanager:GetEvidenceFoldersByAssessmentControl",
"auditmanager:GetOrganizationAdminAccount",
"auditmanager:GetServicesInScope",
"auditmanager:GetSettings",
"auditmanager:ListAssessmentFrameworks",
"auditmanager:ListAssessmentReports",
"auditmanager:ListAssessments",
"auditmanager:ListControls",
"auditmanager:ListKeywordsForDataSource",
"auditmanager:ListNotifications",
"auditmanager:ListTagsForResource",
"auditmanager:ValidateAssessmentReportIntegrity",
"autoscaling-plans:Describe*",
"autoscaling-plans:GetScalingPlanResourceForecastData",
"autoscaling:Describe*",
"autoscaling:GetPredictiveScalingForecast",
"aws-portal:View*",
"backup-gateway:ListGateways",
"backup-gateway:ListHypervisors",
"backup-gateway:ListTagsForResource",
"backup-gateway:ListVirtualMachines",
"backup:Describe*",
"backup:Get*",
"backup:List*",
"batch:Describe*",
"batch:List*",
"billing:GetBillingData",
"billing:GetBillingDetails",
"billing:GetBillingNotifications",
"billing:GetBillingPreferences",
"billing:GetContractInformation",
"billing:GetCredits",
"billing:GetIAMAccessPreference",
```



```
"billing:GetSellerOfRecord",
"billing:ListBillingViews",
"billingconductor:ListAccountAssociations",
"billingconductor:ListBillingGroupCostReports",
"billingconductor:ListBillingGroups",
"billingconductor:ListCustomLineItems",
"billingconductor:ListCustomLineItemVersions",
"billingconductor:ListPricingPlans",
"billingconductor:ListPricingPlansAssociatedWithPricingRule",
"billingconductor:ListPricingRules",
"billingconductor:ListPricingRulesAssociatedToPricingPlan",
"billingconductor:ListResourcesAssociatedToCustomLineItem",
"billingconductor:ListTagsForResource",
"braket:GetDevice",
"braket:GetQuantumTask",
"braket:SearchDevices",
"braket:SearchQuantumTasks",
"budgets:Describe*",
"budgets:View*",
"cassandra:Select",
"ce:DescribeCostCategoryDefinition",
"ce:DescribeNotificationSubscription",
"ce:DescribeReport",
"ce:GetAnomalies",
"ce:GetAnomalyMonitors",
"ce:GetAnomalySubscriptions",
"ce:GetCostAndUsage",
"ce:GetCostAndUsageWithResources",
"ce:GetCostCategories",
"ce:GetCostForecast",
"ce:GetDimensionValues",
"ce:GetPreferences",
"ce:GetReservationCoverage",
"ce:GetReservationPurchaseRecommendation",
"ce:GetReservationUtilization",
"ce:GetRightsizingRecommendation",
"ce:GetSavingsPlansCoverage",
"ce:GetSavingsPlansPurchaseRecommendation",
"ce:GetSavingsPlansUtilization",
"ce:GetSavingsPlansUtilizationDetails",
"ce:GetTags",
"ce:GetUsageForecast",
"ce:ListCostCategoryDefinitions",
"ce:ListSavingsPlansPurchaseRecommendationGeneration",
"ce:ListTagsForResource",
"chatbot:Describe*",
"chatbot:Get*",
"chime:Get*",
"chime:List*",
"chime:Retrieve*",
"chime:Search*",
"chime:Validate*",
"cleanrooms:BatchGetSchema",
"cleanrooms:GetCollaboration",
"cleanrooms:GetConfiguredTable",
"cleanrooms:GetConfiguredTableAnalysisRule",
"cleanrooms:GetConfiguredTableAssociation",
"cleanrooms:GetMembership",
"cleanrooms:GetProtectedQuery",
"cleanrooms:GetSchema",
```





```
"cleanrooms:GetSchemaAnalysisRule",
"cleanrooms:ListCollaborations",
"cleanrooms:ListConfiguredTableAssociations",
"cleanrooms:ListConfiguredTables",
"cleanrooms:ListMembers",
"cleanrooms:ListMemberships",
"cleanrooms:ListProtectedQueries",
"cleanrooms:ListSchemas",
"cloud9:Describe*",
"cloud9:List*",
"clouddirectory:BatchRead",
"clouddirectory:Get*",
"clouddirectory:List*",
"clouddirectory:LookupPolicy",
"cloudformation:Describe*",
"cloudformation:Detect*",
"cloudformation:Estimate*",
"cloudformation:Get*",
"cloudformation:List*",
"cloudfront:DescribeFunction",
"cloudfront:Get*",
"cloudfront:List*",
"cloudhsm:Describe*",
"cloudhsm:Get*",
"cloudhsm:List*",
"cloudsearch:Describe*",
"cloudsearch:List*",
"cloudtrail:Describe*",
"cloudtrail:Get*",
"cloudtrail:List*",
"cloudtrail:LookupEvents",
"cloudwatch:Describe*",
"cloudwatch:Get*",
"cloudwatch:List*",
"codeartifact:DescribeDomain",
"codeartifact:DescribePackage",
"codeartifact:DescribePackageVersion",
"codeartifact:DescribeRepository",
"codeartifact:GetAuthorizationToken",
"codeartifact:GetDomainPermissionsPolicy",
"codeartifact:GetPackageVersionAsset",
"codeartifact:GetPackageVersionReadme",
"codeartifact:GetRepositoryEndpoint",
"codeartifact:GetRepositoryPermissionsPolicy",
"codeartifact:ListDomains",
"codeartifact:ListPackages",
"codeartifact:ListPackageVersionAssets",
"codeartifact:ListPackageVersionDependencies",
"codeartifact:ListPackageVersions",
"codeartifact:ListRepositories",
"codeartifact:ListRepositoriesInDomain",
"codeartifact:ListTagsForResource",
"codeartifact:ReadFromRepository",
"codebuild:BatchGet*",
"codebuild:DescribeCodeCoverages",
"codebuild:DescribeTestCases",
"codebuild:List*",
"codecommit:BatchGet*",
"codecommit:Describe*",
"codecommit:Get*",
```



```
"codecommit:GitPull",
"codecommit:List*",
"codedeploy:BatchGet*",
"codedeploy:Get*",
"codedeploy:List*",
"codeguru-profiler:Describe*",
"codeguru-profiler:Get*",
"codeguru-profiler:List*",
"codeguru-reviewer:Describe*",
"codeguru-reviewer:Get*",
"codeguru-reviewer:List*",
"codepipeline:Get*",
"codepipeline:List*",
"codestar-connections:GetConnection",
"codestar-connections:GetHost",
"codestar-connections:ListConnections",
"codestar-connections:ListHosts",
"codestar-connections:ListTagsForResource",
"codestar-notifications:describeNotificationRule",
"codestar-notifications:listEventTypes",
"codestar-notifications:listNotificationRules",
"codestar-notifications:listTagsForResource",
"codestar-notifications:ListTargets",
"codestar:Describe*",
"codestar:Get*",
"codestar:List*",
"codestar:Verify*",
"cognito-identity:Describe*",
"cognito-identity:GetCredentialsForIdentity",
"cognito-identity:GetIdentityPoolRoles",
"cognito-identity:GetOpenIdToken",
"cognito-identity:GetOpenIdTokenForDeveloperIdentity",
"cognito-identity:List*",
"cognito-identity:Lookup*",
"cognito-idp:AdminGet*",
"cognito-idp:AdminList*",
"cognito-idp:Describe*",
"cognito-idp:Get*",
"cognito-idp:List*",
"cognito-sync:Describe*",
"cognito-sync:Get*",
"cognito-sync:List*",
"cognito-sync:QueryRecords",
"comprehend:BatchDetect*",
"comprehend:Classify*",
"comprehend:Contains*",
"comprehend:Describe*",
"comprehend:Detect*",
"comprehend:List*",
"compute-optimizer:DescribeRecommendationExportJobs",
"compute-optimizer:GetAutoScalingGroupRecommendations",
"compute-optimizer:GetEBSVolumeRecommendations",
"compute-optimizer:GetEC2InstanceRecommendations",
"compute-optimizer:GetEC2RecommendationProjectedMetrics",
"compute-optimizer:GetECSServiceRecommendationProjectedMetrics",
"compute-optimizer:GetECSServiceRecommendations",
"compute-optimizer:GetEffectiveRecommendationPreferences",
"compute-optimizer:GetEnrollmentStatus",
"compute-optimizer:GetEnrollmentStatusesForOrganization",
"compute-optimizer:GetLambdaFunctionRecommendations",
```



```
"compute-optimizer:GetRecommendationPreferences",
"compute-optimizer:GetRecommendationSummaries",
"config:BatchGetAggregateResourceConfig",
"config:BatchGetResourceConfig",
"config:Deliver*",
"config:Describe*",
"config:Get*",
"config:List*",
"config>SelectAggregateResourceConfig",
"config>SelectResourceConfig",
"connect:Describe*",
"connect:GetFederationToken",
"connect:List*",
"consolidatedbilling:GetAccountBillingRole",
"consolidatedbilling>ListLinkedAccounts",
"databrew:DescribeDataset",
"databrew:DescribeJob",
"databrew:DescribeJobRun",
"databrew:DescribeProject",
"databrew:DescribeRecipe",
"databrew:DescribeRuleset",
"databrew:DescribeSchedule",
"databrew>ListDatasets",
"databrew>ListJobRuns",
"databrew>ListJobs",
"databrew>ListProjects",
"databrew>ListRecipes",
"databrew>ListRecipeVersions",
"databrew>ListRulesets",
"databrew>ListSchedules",
"databrew>ListTagsForResource",
"dataexchange:Get*",
"dataexchange>List*",
"datapipeline:Describe*",
"datapipeline:EvaluateExpression",
"datapipeline:Get*",
"datapipeline>List*",
"datapipeline:QueryObjects",
"datapipeline:Validate*",
"datasync:Describe*",
"datasync>List*",
"dax:BatchGetItem",
"dax:Describe*",
"dax:GetItem",
"dax>ListTags",
"dax:Query",
"dax:Scan",
"deepcomposer:GetComposition",
"deepcomposer:GetModel",
"deepcomposer:GetSampleModel",
"deepcomposer>ListCompositions",
"deepcomposer>ListModels",
"deepcomposer>ListSampleModels",
"deepcomposer>ListTrainingTopics",
"detective:BatchGetGraphMemberDatasources",
"detective:BatchGetMembershipDatasources",
"detective:Get*",
"detective>List*",
"detective:SearchGraph",
"devicefarm:Get*",
```



```
"devicefarm:List*",
"devops-guru:DescribeAccountHealth",
"devops-guru:DescribeAccountOverview",
"devops-guru:DescribeAnomaly",
"devops-guru:DescribeEventSourcesConfig",
"devops-guru:DescribeFeedback",
"devops-guru:DescribeInsight",
"devops-guru:DescribeOrganizationHealth",
"devops-guru:DescribeOrganizationOverview",
"devops-guru:DescribeOrganizationResourceCollectionHealth",
"devops-guru:DescribeResourceCollectionHealth",
"devops-guru:DescribeServiceIntegration",
"devops-guru:GetCostEstimation",
"devops-guru:GetResourceCollection",
"devops-guru:ListAnomaliesForInsight",
"devops-guru:ListAnomalousLogGroups",
"devops-guru:ListEvents",
"devops-guru:ListInsights",
"devops-guru:ListMonitoredResources",
"devops-guru:ListNotificationChannels",
"devops-guru:ListOrganizationInsights",
"devops-guru:ListRecommendations",
"devops-guru:SearchInsights",
"devops-guru:StartCostEstimation",
"directconnect:Describe*",
"discovery:Describe*",
"discovery:Get*",
"discovery:List*",
"dlm:Get*",
"dms:Describe*",
"dms:List*",
"dms:Test*",
"drs:DescribeJobLogItems",
"drs:DescribeJobs",
"drs:DescribeRecoveryInstances",
"drs:DescribeRecoverySnapshots",
"drs:DescribeReplicationConfigurationTemplates",
"drs:DescribeSourceServers",
"drs:GetFailbackReplicationConfiguration",
"drs:GetLaunchConfiguration",
"drs:GetReplicationConfiguration",
"drs:ListExtensibleSourceServers",
"drs:ListStagingAccounts",
"drs:ListTagsForResource",
"ds:Check*",
"ds:Describe*",
"ds:Get*",
"ds:List*",
"ds:Verify*",
"dynamodb:BatchGet*",
"dynamodb:Describe*",
"dynamodb:Get*",
"dynamodb:List*",
"dynamodb: PartiQLSelect",
"dynamodb:Query",
"dynamodb:Scan",
"ec2:Describe*",
"ec2:Get*",
"ec2:ListImagesInRecycleBin",
"ec2:ListSnapshotsInRecycleBin",
```



```
"ec2:SearchLocalGatewayRoutes",
"ec2:SearchTransitGatewayRoutes",
"ec2messages:Get*",
"ecr-public:BatchCheckLayerAvailability",
"ecr-public:DescribeImages",
"ecr-public:DescribeImageTags",
"ecr-public:DescribeRegistries",
"ecr-public:DescribeRepositories",
"ecr-public:GetAuthorizationToken",
"ecr-public:GetRegistryCatalogData",
"ecr-public:GetRepositoryCatalogData",
"ecr-public:GetRepositoryPolicy",
"ecr-public:ListTagsForResource",
"ecr:BatchCheck*",
"ecr:BatchGet*",
"ecr:Describe*",
"ecr:Get*",
"ecr:List*",
"ecs:Describe*",
"ecs:List*",
"eks:Describe*",
"eks:List*",
"elastic-inference:DescribeAcceleratorOfferings",
"elastic-inference:DescribeAccelerators",
"elastic-inference:DescribeAcceleratorTypes",
"elastic-inference:ListTagsForResource",
"elasticache:Describe*",
"elasticache:List*",
"elasticbeanstalk:Check*",
"elasticbeanstalk:Describe*",
"elasticbeanstalk:List*",
"elasticbeanstalk:Request*",
"elasticbeanstalk:Retrieve*",
"elasticbeanstalk:Validate*",
"elasticfilesystem:Describe*",
"elasticloadbalancing:Describe*",
"elasticmapreduce:Describe*",
"elasticmapreduce:GetBlockPublicAccessConfiguration",
"elasticmapreduce:List*",
"elasticmapreduce:View*",
"elastictranscoder:List*",
"elastictranscoder:Read*",
"elemental-appliances-software:Get*",
"elemental-appliances-software:List*",
"emr-containers:DescribeJobRun",
"emr-containers:DescribeManagedEndpoint",
"emr-containers:DescribeVirtualCluster",
"emr-containers:ListJobRuns",
"emr-containers:ListManagedEndpoints",
"emr-containers:ListTagsForResource",
"emr-containers:ListVirtualClusters",
"es:Describe*",
"es:ESHttpGet",
"es:ESHttpHead",
"es:Get*",
"es:List*",
"events:Describe*",
"events:List*",
"events:Test*",
"evidently:GetExperiment",
```



```
"evidently:GetExperimentResults",
"evidently:GetFeature",
"evidently:GetLaunch",
"evidently:GetProject",
"evidently:GetSegment",
"evidently:ListExperiments",
"evidently:ListFeatures",
"evidently:ListLaunches",
"evidently:ListProjects",
"evidently:ListSegmentReferences",
"evidently:ListSegments",
"evidently:ListTagsForResource",
"evidently:TestSegmentPattern",
"firehose:Describe*",
"firehose:List*",
"fis:GetAction",
"fis:GetExperiment",
"fis:GetExperimentTemplate",
"fis:GetTargetResourceType",
"fis:ListActions",
"fis:ListExperiments",
"fis:ListExperimentTemplates",
"fis:ListTagsForResource",
"fis:ListTargetResourceTypes",
"fms:GetAdminAccount",
"fms:GetAppsList",
"fms:GetComplianceDetail",
"fms:GetNotificationChannel",
"fms:GetPolicy",
"fms:GetProtectionStatus",
"fms:GetProtocolsList",
"fms:GetViolationDetails",
"fms:ListAppsLists",
"fms:ListComplianceStatus",
"fms:ListMemberAccounts",
"fms:ListPolicies",
"fms:ListProtocolsLists",
"fms:ListTagsForResource",
"forecast:DescribeAutoPredictor",
"forecast:DescribeDataset",
"forecast:DescribeDatasetGroup",
"forecast:DescribeDatasetImportJob",
"forecast:DescribeExplainability",
"forecast:DescribeExplainabilityExport",
"forecast:DescribeForecast",
"forecast:DescribeForecastExportJob",
"forecast:DescribeMonitor",
"forecast:DescribePredictor",
"forecast:DescribePredictorBacktestExportJob",
"forecast:DescribeWhatIfAnalysis",
"forecast:DescribeWhatIfForecast",
"forecast:DescribeWhatIfForecastExport",
"forecast:GetAccuracyMetrics",
"forecast:ListDatasetGroups",
"forecast:ListDatasetImportJobs",
"forecast:ListDatasets",
"forecast:ListExplainabilities",
"forecast:ListExplainabilityExports",
"forecast:ListForecastExportJobs",
"forecast:ListForecasts",
```



```
"forecast:ListMonitorEvaluations",
"forecast:ListMonitors",
"forecast:ListPredictorBacktestExportJobs",
"forecast:ListPredictors",
"forecast:ListWhatIfAnalyses",
"forecast:ListWhatIfForecastExports",
"forecast:ListWhatIfForecasts",
"forecast:QueryForecast",
"forecast:QueryWhatIfForecast",
"frauddetector:BatchGetVariable",
"frauddetector:DescribeDetector",
"frauddetector:DescribeModelVersions",
"frauddetector:GetBatchImportJobs",
"frauddetector:GetBatchPredictionJobs",
"frauddetector:GetDeleteEventsByEventTypeStatus",
"frauddetector:GetDetectors",
"frauddetector:GetDetectorVersion",
"frauddetector:GetEntityTypes",
"frauddetector:GetEvent",
"frauddetector:GetEventPredictionMetadata",
"frauddetector:GetEventTypes",
"frauddetector:GetExternalModels",
"frauddetector:GetKMSEncryptionKey",
"frauddetector:GetLabels",
"frauddetector:GetListElements",
"frauddetector:GetListsMetadata",
"frauddetector:GetModels",
"frauddetector:GetModelVersion",
"frauddetector:GetOutcomes",
"frauddetector:GetRules",
"frauddetector:GetVariables",
"frauddetector:ListEventPredictions",
"frauddetector:ListTagsForResource",
"freertos:Describe*",
"freertos:List*",
"freetier:GetFreeTierAlertPreference",
"freetier:GetFreeTierUsage",
"fsx:Describe*",
"fsx:List*",
"gamelift:Describe*",
"gamelift:Get*",
"gamelift:List*",
"gamelift:ResolveAlias",
"gamelift:Search*",
"gamesparks:GetExtension",
"gamesparks:GetExtensionVersion",
"gamesparks:GetGame",
"gamesparks:GetGameConfiguration",
"gamesparks:GetGeneratedCodeJob",
"gamesparks:GetPlayerConnectionStatus",
"gamesparks:GetSnapshot",
"gamesparks:GetStage",
"gamesparks:GetStageDeployment",
"gamesparks:ListExtensions",
"gamesparks:ListExtensionVersions",
"gamesparks:ListGames",
"gamesparks:ListGeneratedCodeJobs",
"gamesparks:ListSnapshots",
"gamesparks:ListStageDeployments",
"gamesparks:ListStages",
```



```
"gamesparks:ListTagsForResource",
"glacier:Describe*",
"glacier:Get*",
"glacier:List*",
"globalaccelerator:Describe*",
"globalaccelerator:List*",
"glue:BatchGetCrawlers",
"glue:BatchGetDevEndpoints",
"glue:BatchGetJobs",
"glue:BatchGetPartition",
"glue:BatchGetTriggers",
"glue:BatchGetWorkflows",
"glue:CheckSchemaVersionValidity",
"glue:GetCatalogImportStatus",
"glue:GetClassifier",
"glue:GetClassifiers",
"glue:GetCrawler",
"glue:GetCrawlerMetrics",
"glue:GetCrawlers",
"glue:GetDatabase",
"glue:GetDatabases",
"glue:GetDataCatalogEncryptionSettings",
"glue:GetDataflowGraph",
"glue:GetDevEndpoint",
"glue:GetDevEndpoints",
"glue:GetJob",
"glue:GetJobBookmark",
"glue:GetJobRun",
"glue:GetJobRuns",
"glue:GetJobs",
"glue:GetMapping",
"glue:GetMLTaskRun",
"glue:GetMLTaskRuns",
"glue:GetMLTransform",
"glue:GetMLTransforms",
"glue:GetPartition",
"glue:GetPartitions",
"glue:GetPlan",
"glue:GetRegistry",
"glue:GetResourcePolicy",
"glue:GetSchema",
"glue:GetSchemaByDefinition",
"glue:GetSchemaVersion",
"glue:GetSchemaVersionsDiff",
"glue:GetSecurityConfiguration",
"glue:GetSecurityConfigurations",
"glue:GetTable",
"glue:GetTables",
"glue:GetTableVersion",
"glue:GetTableVersions",
"glue:GetTags",
"glue:GetTrigger",
"glue:GetTriggers",
"glue:GetUserDefinedFunction",
"glue:GetUserDefinedFunctions",
"glue:GetWorkflow",
"glue:GetWorkflowRun",
"glue:GetWorkflowRunProperties",
"glue:GetWorkflowRuns",
"glue:ListCrawlers",
```





```
"glue:ListCrawls",
"glue:ListDevEndpoints",
"glue:ListJobs",
"glue:ListMLTransforms",
"glue:ListRegistries",
"glue:ListSchemas",
"glue:ListSchemaVersions",
"glue:ListTriggers",
"glue:ListWorkflows",
"glue:QuerySchemaVersionMetadata",
"glue:SearchTables",
"grafana:DescribeWorkspace",
"grafana:ListWorkspaces",
"greengrass:DescribeComponent",
"greengrass:Get*",
"greengrass:List*",
"groundstation:DescribeContact",
"groundstation:GetConfig",
"groundstation:GetDataflowEndpointGroup",
"groundstation:GetMinuteUsage",
"groundstation:GetMissionProfile",
"groundstation:GetSatellite",
"groundstation:ListConfigs",
"groundstation:ListContacts",
"groundstation:ListDataflowEndpointGroups",
"groundstation:ListGroundStations",
"groundstation:ListMissionProfiles",
"groundstation:ListSatellites",
"groundstation:ListTagsForResource",
"guardduty:Describe*",
"guardduty:Get*",
"guardduty:List*",
"health:Describe*",
"healthlake:DescribeFHIRDatastore",
"healthlake:DescribeFHIRExportJob",
"healthlake:DescribeFHIRImportJob",
"healthlake:GetCapabilities",
"healthlake:ListFHIRDatastores",
"healthlake:ListFHIRExportJobs",
"healthlake:ListFHIRImportJobs",
"healthlake:ListTagsForResource",
"healthlake:ReadResource",
"healthlake:SearchWithGet",
"healthlake:SearchWithPost",
"iam:Generate*",
"iam:Get*",
"iam:List*",
"iam:Simulate*",
"identity-sync:GetSyncProfile",
"identity-sync:GetSyncTarget",
"identity-sync:ListSyncFilters",
"identitystore-auth:BatchGetSession",
"identitystore-auth:ListSessions",
"imagebuilder:Get*",
"imagebuilder:List*",
"importexport:Get*",
"importexport:List*",
"inspector:Describe*",
"inspector:Get*",
"inspector:List*",
```



```
"inspector:Preview*",
"inspector2:BatchGetAccountStatus",
"inspector2:BatchGetFreeTrialInfo",
"inspector2:DescribeOrganizationConfiguration",
"inspector2:GetDelegatedAdminAccount",
"inspector2:GetFindingsReportStatus",
"inspector2:GetMember",
"inspector2:ListAccountPermissions",
"inspector2:ListCoverage",
"inspector2:ListCoverageStatistics",
"inspector2:ListDelegatedAdminAccounts",
"inspector2:ListFilters",
"inspector2:ListFindingAggregations",
"inspector2:ListFindings",
"inspector2:ListMembers",
"inspector2:ListTagsForResource",
"inspector2:ListUsageTotals",
"internetmonitor:GetHealthEvent",
"internetmonitor:GetMonitor",
"internetmonitor:ListHealthEvents",
"internetmonitor:ListMonitors",
"internetmonitor:ListTagsForResource",
"invoicing:GetInvoiceEmailDeliveryPreferences",
"invoicing:GetInvoicePDF",
"invoicing:ListInvoiceSummaries",
"iot:Describe*",
"iot:Get*",
"iot:List*",
"iot1click:DescribeDevice",
"iot1click:DescribePlacement",
"iot1click:DescribeProject",
"iot1click:GetDeviceMethods",
"iot1click:GetDevicesInPlacement",
"iot1click:ListDeviceEvents",
"iot1click:ListDevices",
"iot1click:ListPlacements",
"iot1click:ListProjects",
"iot1click:ListTagsForResource",
"iotanalytics:Describe*",
"iotanalytics:Get*",
"iotanalytics:List*",
"iotanalytics:SampleChannelData",
"iotevents:DescribeAlarm",
"iotevents:DescribeAlarmModel",
"iotevents:DescribeDetector",
"iotevents:DescribeDetectorModel",
"iotevents:DescribeInput",
"iotevents:DescribeLoggingOptions",
"iotevents:ListAlarmModels",
"iotevents:ListAlarmModelVersions",
"iotevents:ListAlarms",
"iotevents:ListDetectorModels",
"iotevents:ListDetectorModelVersions",
"iotevents:ListDetectors",
"iotevents:ListInputs",
"iotevents:ListTagsForResource",
"iotfleethub:DescribeApplication",
"iotfleethub:ListApplications",
"iotfleetwise:GetCampaign",
"iotfleetwise:GetDecoderManifest",
```



```
"iotfleetwise:GetFleet",
"iotfleetwise:GetLoggingOptions",
"iotfleetwise:GetModelManifest",
"iotfleetwise:GetRegisterAccountStatus",
"iotfleetwise:GetSignalCatalog",
"iotfleetwise:GetVehicle",
"iotfleetwise:GetVehicleStatus",
"iotfleetwise:ListCampaigns",
"iotfleetwise:ListDecoderManifestNetworkInterfaces",
"iotfleetwise:ListDecoderManifests",
"iotfleetwise:ListDecoderManifestSignals",
"iotfleetwise:ListFleets",
"iotfleetwise:ListFleetsForVehicle",
"iotfleetwise:ListModelManifestNodes",
"iotfleetwise:ListModelManifests",
"iotfleetwise:ListSignalCatalogNodes",
"iotfleetwise:ListSignalCatalogs",
"iotfleetwise:ListTagsForResource",
"iotfleetwise:ListVehicles",
"iotfleetwise:ListVehiclesInFleet",
"iotroborunner:GetDestination",
"iotroborunner:GetSite",
"iotroborunner:GetWorker",
"iotroborunner:GetWorkerFleet",
"iotroborunner:ListDestinations",
"iotroborunner:ListSites",
"iotroborunner:ListWorkerFleets",
"iotroborunner:ListWorkers",
"iotsitewise:Describe*",
"iotsitewise:Get*",
"iotsitewise:List*",
"iotwireless:GetDestination",
"iotwireless:GetDeviceProfile",
"iotwireless:GetPartnerAccount",
"iotwireless:GetServiceEndpoint",
"iotwireless:GetServiceProfile",
"iotwireless:GetWirelessDevice",
"iotwireless:GetWirelessDeviceStatistics",
"iotwireless:GetWirelessGateway",
"iotwireless:GetWirelessGatewayCertificate",
"iotwireless:GetWirelessGatewayFirmwareInformation",
"iotwireless:GetWirelessGatewayStatistics",
"iotwireless:GetWirelessGatewayTask",
"iotwireless:GetWirelessGatewayTaskDefinition",
"iotwireless:ListDestinations",
"iotwireless:ListDeviceProfiles",
"iotwireless:ListPartnerAccounts",
"iotwireless:ListServiceProfiles",
"iotwireless:ListTagsForResource",
"iotwireless:ListWirelessDevices",
"iotwireless:ListWirelessGateways",
"iotwireless:ListWirelessGatewayTaskDefinitions",
"ivs:BatchGetChannel",
"ivs:GetChannel",
"ivs:GetPlaybackKeyPair",
"ivs:GetRecordingConfiguration",
"ivs:GetStreamSession",
"ivs:ListChannels",
"ivs:ListPlaybackKeyPairs",
"ivs:ListRecordingConfigurations",
```



```
"ivs:ListStreams",
"ivs:ListStreamSessions",
"ivs:ListTagsForResource",
"ivschat:GetLoggingConfiguration",
"ivschat:GetRoom",
"ivschat:ListLoggingConfigurations",
"ivschat:ListRooms",
"ivschat:ListTagsForResource",
"kafka:Describe*",
"kafka:DescribeCluster",
"kafka:DescribeClusterOperation",
"kafka:DescribeClusterV2",
"kafka:DescribeConfiguration",
"kafka:DescribeConfigurationRevision",
"kafka:Get*",
"kafka:GetBootstrapBrokers",
"kafka:GetCompatibleKafkaVersions",
"kafka:List*",
"kafka:ListClusterOperations",
"kafka:ListClusters",
"kafka:ListClustersV2",
"kafka:ListConfigurationRevisions",
"kafka:ListConfigurations",
"kafka:ListKafkaVersions",
"kafka:ListNodes",
"kafka:ListTagsForResource",
"kafkaconnect:DescribeConnector",
"kafkaconnect:DescribeCustomPlugin",
"kafkaconnect:DescribeWorkerConfiguration",
"kafkaconnect:ListConnectors",
"kafkaconnect:ListCustomPlugins",
"kafkaconnect:ListWorkerConfigurations",
"kendra:BatchGetDocumentStatus",
"kendra:DescribeDataSource",
"kendra:DescribeExperience",
"kendra:DescribeFaq",
"kendra:DescribeIndex",
"kendra:DescribePrincipalMapping",
"kendra:DescribeQuerySuggestionsBlockList",
"kendra:DescribeQuerySuggestionsConfig",
"kendra:DescribeThesaurus",
"kendra:GetQuerySuggestions",
"kendra:GetSnapshots",
"kendra:ListDataSources",
"kendra:ListDataSourceSyncJobs",
"kendra:ListEntityPersonas",
"kendra:ListExperienceEntities",
"kendra:ListExperiences",
"kendra:ListFaqs",
"kendra:ListGroupsOlderThanOrderingId",
"kendra:ListIndices",
"kendra:ListQuerySuggestionsBlockLists",
"kendra:ListTagsForResource",
"kendra:ListThesauri",
"kendra:Query",
"kinesis:Describe*",
"kinesis:Get*",
"kinesis:List*",
"kinesisanalytics:Describe*",
"kinesisanalytics:Discover*",
```



```
"kinesisanalytics:Get*",
"kinesisanalytics:List*",
"kinesisvideo:Describe*",
"kinesisvideo:Get*",
"kinesisvideo:List*",
"kms:Describe*",
"kms:Get*",
"kms:List*",
"lambda:Get*",
"lambda:List*",
"lex:DescribeBot",
"lex:DescribeBotAlias",
"lex:DescribeBotChannel",
"lex:DescribeBotLocale",
"lex:DescribeBotVersion",
"lex:DescribeExport",
"lex:DescribeImport",
"lex:DescribeIntent",
"lex:DescribeResourcePolicy",
"lex:DescribeSlot",
"lex:DescribeSlotType",
"lex:Get*",
"lex:ListBotAliases",
"lex:ListBotChannels",
"lex:ListBotLocales",
"lex:ListBots",
"lex:ListBotVersions",
"lex:ListBuiltInIntents",
"lex:ListBuiltInSlotTypes",
"lex:ListExports",
"lex:ListImports",
"lex:ListIntents",
"lex:ListSlots",
"lex:ListSlotTypes",
"lex:ListTagsForResource",
"license-manager:Get*",
"license-manager:List*",
"lightsail:GetActiveNames",
"lightsail:GetAlarms",
"lightsail:GetAutoSnapshots",
"lightsail:GetBlueprints",
"lightsail:GetBucketAccessKeys",
"lightsail:GetBucketBundles",
"lightsail:GetBucketMetricData",
"lightsail:GetBuckets",
"lightsail:GetBundles",
"lightsail:GetCertificates",
"lightsail:GetCloudFormationStackRecords",
"lightsail:GetContainerAPIMetadata",
"lightsail:GetContainerImages",
"lightsail:GetContainerServiceDeployments",
"lightsail:GetContainerServiceMetricData",
"lightsail:GetContainerServicePowers",
"lightsail:GetContainerServices",
"lightsail:GetDisk",
"lightsail:GetDisks",
"lightsail:GetDiskSnapshot",
"lightsail:GetDiskSnapshots",
"lightsail:GetDistributionBundles",
"lightsail:GetDistributionLatestCacheReset",
```



```
"lightsail:GetDistributionMetricData",
"lightsail:GetDistributions",
"lightsail:GetDomain",
"lightsail:GetDomains",
"lightsail:GetExportSnapshotRecords",
"lightsail:GetInstance",
"lightsail:GetInstanceMetricData",
"lightsail:GetInstancePortStates",
"lightsail:GetInstances",
"lightsail:GetInstanceSnapshot",
"lightsail:GetInstanceSnapshots",
"lightsail:GetInstanceState",
"lightsail:GetKeyPair",
"lightsail:GetKeyPairs",
"lightsail:GetLoadBalancer",
"lightsail:GetLoadBalancerMetricData",
"lightsail:GetLoadBalancers",
"lightsail:GetLoadBalancerTlsCertificates",
"lightsail:GetOperation",
"lightsail:GetOperations",
"lightsail:GetOperationsForResource",
"lightsail:GetRegions",
"lightsail:GetRelationalDatabase",
"lightsail:GetRelationalDatabaseBlueprints",
"lightsail:GetRelationalDatabaseBundles",
"lightsail:GetRelationalDatabaseEvents",
"lightsail:GetRelationalDatabaseLogEvents",
"lightsail:GetRelationalDatabaseLogStreams",
"lightsail:GetRelationalDatabaseMetricData",
"lightsail:GetRelationalDatabaseParameters",
"lightsail:GetRelationalDatabases",
"lightsail:GetRelationalDatabaseSnapshot",
"lightsail:GetRelationalDatabaseSnapshots",
"lightsail:GetStaticIp",
"lightsail:GetStaticIps",
"lightsail:Is*",
"logs:Describe*",
"logs:FilterLogEvents",
"logs:Get*",
"logs:ListTagsForResource",
"logs:ListTagsLogGroup",
"logs:StartQuery",
"logs:StopQuery",
"logs:TestMetricFilter",
"lookoutequipment:DescribeDataIngestionJob",
"lookoutequipment:DescribeDataset",
"lookoutequipment:DescribeInferenceScheduler",
"lookoutequipment:DescribeLabel",
"lookoutequipment:DescribeLabelGroup",
"lookoutequipment:DescribeModel",
"lookoutequipment:ListDataIngestionJobs",
"lookoutequipment:ListDatasets",
"lookoutequipment:ListInferenceEvents",
"lookoutequipment:ListInferenceExecutions",
"lookoutequipment:ListInferenceSchedulers",
"lookoutequipment:ListLabelGroups",
"lookoutequipment:ListLabels",
"lookoutequipment:ListModels",
"lookoutequipment:ListSensorStatistics",
"lookoutequipment:ListTagsForResource",
```



```
"lookoutmetrics:Describe*",
"lookoutmetrics:Get*",
"lookoutmetrics:List*",
"lookoutvision:DescribeDataset",
"lookoutvision:DescribeModel",
"lookoutvision:DescribeModelPackagingJob",
"lookoutvision:DescribeProject",
"lookoutvision:ListDatasetEntries",
"lookoutvision:ListModelPackagingJobs",
"lookoutvision:ListModels",
"lookoutvision:ListProjects",
"lookoutvision:ListTagsForResource",
"m2:GetApplication",
"m2:GetApplicationVersion",
"m2:GetBatchJobExecution",
"m2:GetDataSetDetails",
"m2:GetDataSetImportTask",
"m2:GetDeployment",
"m2:GetEnvironment",
"m2:ListApplications",
"m2:ListApplicationVersions",
"m2:ListBatchJobDefinitions",
"m2:ListBatchJobExecutions",
"m2:ListDataSetImportHistory",
"m2:ListDataSets",
"m2:ListDeployments",
"m2:ListEngineVersions",
"m2:ListEnvironments",
"m2:ListTagsForResource",
"machinelearning:Describe*",
"machinelearning:Get*",
"macie2:BatchGetCustomDataIdentifiers",
"macie2:DescribeBuckets",
"macie2:DescribeClassificationJob",
"macie2:DescribeOrganizationConfiguration",
"macie2:GetAdministratorAccount",
"macie2:GetAllowList",
"macie2:GetAutomatedDiscoveryConfiguration",
"macie2:GetBucketStatistics",
"macie2:GetClassificationExportConfiguration",
"macie2:GetClassificationScope",
"macie2:GetCustomDataIdentifier",
"macie2:GetFindings",
"macie2:GetFindingsFilter",
"macie2:GetFindingsPublicationConfiguration",
"macie2:GetFindingStatistics",
"macie2:GetInvitationsCount",
"macie2:GetMacieSession",
"macie2:GetMember",
"macie2:GetResourceProfile",
"macie2:GetRevealConfiguration",
"macie2:GetSensitiveDataOccurrencesAvailability",
"macie2:GetSensitivityInspectionTemplate",
"macie2:GetUsageStatistics",
"macie2:GetUsageTotals",
"macie2:ListAllowLists",
"macie2:ListClassificationJobs",
"macie2:ListClassificationScopes",
"macie2:ListCustomDataIdentifiers",
"macie2:ListFindings",
```



```
"macie2:ListFindingsFilters",
"macie2:ListInvitations",
"macie2:ListMembers",
"macie2:ListOrganizationAdminAccounts",
"macie2:ListResourceProfileArtifacts",
"macie2:ListResourceProfileDetections",
"macie2:ListSensitivityInspectionTemplates",
"macie2:ListTagsForResource",
"macie2:SearchResources",
"managedblockchain:GetMember",
"managedblockchain:GetNetwork",
"managedblockchain:GetNode",
"managedblockchain:GetProposal",
"managedblockchain:ListInvitations",
"managedblockchain:ListMembers",
"managedblockchain:ListNetworks",
"managedblockchain:ListNodes",
"managedblockchain:ListProposals",
"managedblockchain:ListProposalVotes",
"managedblockchain:ListTagsForResource",
"mediaconnect:DescribeFlow",
"mediaconnect:DescribeOffering",
"mediaconnect:DescribeReservation",
"mediaconnect:ListEntitlements",
"mediaconnect:ListFlows",
"mediaconnect:ListOfferings",
"mediaconnect:ListReservations",
"mediaconnect:ListTagsForResource",
"mediaconvert:DescribeEndpoints",
"mediaconvert:Get*",
"mediaconvert:List*",
"medialive:DescribeChannel",
"medialive:DescribeInput",
"medialive:DescribeInputDevice",
"medialive:DescribeInputDeviceThumbnail",
"medialive:DescribeInputSecurityGroup",
"medialive:DescribeMultiplex",
"medialive:DescribeMultiplexProgram",
"medialive:DescribeOffering",
"medialive:DescribeReservation",
"medialive:DescribeSchedule",
"medialive:ListChannels",
"medialive:ListInputDevices",
"medialive:ListInputDeviceTransfers",
"medialive:ListInputs",
"medialive:ListInputSecurityGroups",
"medialive:ListMultiplexes",
"medialive:ListMultiplexPrograms",
"medialive:ListOfferings",
"medialive:ListReservations",
"medialive:ListTagsForResource",
"mediapackage-vod:Describe*",
"mediapackage-vod:List*",
"mediapackage:Describe*",
"mediapackage:List*",
"mediastore:DescribeContainer",
"mediastore:DescribeObject",
"mediastore:GetContainerPolicy",
"mediastore:GetCorsPolicy",
"mediastore:GetLifecyclePolicy",
```





```
"mediastore:GetMetricPolicy",
"mediastore:GetObject",
"mediastore:ListContainers",
"mediastore:ListItems",
"mediastore:ListTagsForResource",
"memorydb:DescribeClusters",
"memorydb:DescribeParameterGroups",
"memorydb:DescribeParameters",
"memorydb:ListTags",
"mgh:Describe*",
"mgh:GetHomeRegion",
"mgh:List*",
"mgn:DescribeJobLogItems",
"mgn:DescribeJobs",
"mgn:DescribeLaunchConfigurationTemplates",
"mgn:DescribeReplicationConfigurationTemplates",
"mgn:DescribeSourceServers",
"mgn:DescribeVcenterClients",
"mgn:GetLaunchConfiguration",
"mgn:GetReplicationConfiguration",
"mgn:ListApplications",
"mgn:ListSourceServerActions",
"mgn:ListTemplateActions",
"mgn:ListWaves",
"mobileanalytics:Get*",
"mobiletargeting:Get*",
"mobiletargeting:List*",
"monitron:GetProject",
"monitron:GetProjectAdminUser",
"monitron:ListProjects",
"monitron:ListTagsForResource",
"mq:Describe*",
"mq:List*",
"network-firewall:DescribeFirewall",
"network-firewall:DescribeFirewallPolicy",
"network-firewall:DescribeLoggingConfiguration",
"network-firewall:DescribeResourcePolicy",
"network-firewall:DescribeRuleGroup",
"network-firewall:DescribeRuleGroupMetadata",
"network-firewall:ListFirewallPolicies",
"network-firewall:ListFirewalls",
"network-firewall:ListRuleGroups",
"network-firewall:ListTagsForResource",
"networkmanager:DescribeGlobalNetworks",
"networkmanager:GetConnectAttachment",
"networkmanager:GetConnections",
"networkmanager:GetConnectPeer",
"networkmanager:GetConnectPeerAssociations",
"networkmanager:GetCoreNetwork",
"networkmanager:GetCoreNetworkChangeEvents",
"networkmanager:GetCoreNetworkChangeSet",
"networkmanager:GetCoreNetworkPolicy",
"networkmanager:GetCustomerGatewayAssociations",
"networkmanager:GetDevices",
"networkmanager:GetLinkAssociations",
"networkmanager:GetLinks",
"networkmanager:GetNetworkResourceCounts",
"networkmanager:GetNetworkResourceRelationships",
"networkmanager:GetNetworkResources",
"networkmanager:GetNetworkRoutes",
```



```
"networkmanager:GetNetworkTelemetry",
"networkmanager:GetResourcePolicy",
"networkmanager:GetRouteAnalysis",
"networkmanager:GetSites",
"networkmanager:GetSiteToSiteVpnAttachment",
"networkmanager:GetTransitGatewayConnectPeerAssociations",
"networkmanager:GetTransitGatewayPeering",
"networkmanager:GetTransitGatewayRegistrations",
"networkmanager:GetTransitGatewayRouteTableAttachment",
"networkmanager:GetVpcAttachment",
"networkmanager:ListAttachments",
"networkmanager:ListConnectPeers",
"networkmanager:ListCoreNetworkPolicyVersions",
"networkmanager:ListCoreNetworks",
"networkmanager:ListPeerings",
"networkmanager:ListTagsForResource",
"nimble:GetEula",
"nimble:GetFeatureMap",
"nimble:GetLaunchProfile",
"nimble:GetLaunchProfileDetails",
"nimble:GetLaunchProfileInitialization",
"nimble:GetLaunchProfileMember",
"nimble:GetStreamingImage",
"nimble:GetStreamingSession",
"nimble:GetStudio",
"nimble:GetStudioComponent",
"nimble:GetStudioMember",
"nimble:ListEulaAcceptances",
"nimble:ListEulas",
"nimble:ListLaunchProfileMembers",
"nimble:ListLaunchProfiles",
"nimble:ListStreamingImages",
"nimble:ListStreamingSessions",
"nimble:ListStudioComponents",
"nimble:ListStudioMembers",
"nimble:ListStudios",
"nimble:ListTagsForResource",
"oam:GetLink",
"oam:GetSink",
"oam:GetSinkPolicy",
"oam:ListAttachedLinks",
"oam:ListLinks",
"oam:ListSinks",
"opsworks-cm:Describe*",
"opsworks-cm:List*",
"opsworks:Describe*",
"opsworks:Get*",
"organizations:Describe*",
"organizations:List*",
"outposts:Get*",
"outposts:List*",
"payments:GetPaymentInstrument",
"payments:GetPaymentStatus",
"payments:ListPaymentPreferences",
"personalize:Describe*",
"personalize:Get*",
"personalize:List*",
"pi:DescribeDimensionKeys",
"pi:GetDimensionKeyDetails",
"pi:GetResourceMetadata",
```



```
"pi:GetResourceMetrics",
"pi:ListAvailableResourceDimensions",
"pi:ListAvailableResourceMetrics",
"polly:Describe*",
"polly:Get*",
"polly:List*",
"polly:SynthesizeSpeech",
"proton:GetEnvironment",
"proton:GetEnvironmentTemplate",
"proton:GetEnvironmentTemplateVersion",
"proton:GetService",
"proton:GetServiceInstance",
"proton:GetServiceTemplate",
"proton:GetServiceTemplateVersion",
"proton:ListEnvironmentAccountConnections",
"proton:ListEnvironments",
"proton:ListEnvironmentTemplates",
"proton:ListServiceInstances",
"proton:ListServices",
"proton:ListServiceTemplates",
"proton:ListTagsForResource",
"purchase-orders:GetPurchaseOrder",
"purchase-orders:ListPurchaseOrderInvoices",
"purchase-orders:ListPurchaseOrders",
"purchase-orders:ViewPurchaseOrders",
"qldb:DescribeJournalS3Export",
"qldb:DescribeLedger",
"qldb:GetBlock",
"qldb:GetDigest",
"qldb:GetRevision",
"qldb:ListJournalS3Exports",
"qldb:ListJournalS3ExportsForLedger",
"qldb:ListLedgers",
"qldb:ListTagsForResource",
"ram:Get*",
"ram:List*",
"rbin:GetRule",
"rbin:ListRules",
"rbin:ListTagsForResource",
"rds:Describe*",
"rds:Download*",
"rds:List*",
"redshift:Describe*",
"redshift:GetReservedNodeExchangeOfferings",
"redshift:View*",
"refactor-spaces:GetApplication",
"refactor-spaces:GetEnvironment",
"refactor-spaces:GetResourcePolicy",
"refactor-spaces:GetRoute",
"refactor-spaces:GetService",
"refactor-spaces:ListApplications",
"refactor-spaces:ListEnvironments",
"refactor-spaces:ListEnvironmentVpcs",
"refactor-spaces:ListRoutes",
"refactor-spaces:ListServices",
"refactor-spaces:ListTagsForResource",
"rekognition:CompareFaces",
"rekognition:Detect*",
"rekognition:List*",
"rekognition:Search*",
```



```
"resiliencehub:DescribeApp",
"resiliencehub:DescribeAppAssessment",
"resiliencehub:DescribeAppVersionResourcesResolutionStatus",
"resiliencehub:DescribeAppVersionTemplate",
"resiliencehub:DescribeDraftAppVersionResourcesImportStatus",
"resiliencehub:DescribeResiliencyPolicy",
"resiliencehub:ListAlarmRecommendations",
"resiliencehub:ListAppAssessments",
"resiliencehub:ListAppComponentCompliances",
"resiliencehub:ListAppComponentRecommendations",
"resiliencehub:ListApps",
"resiliencehub:ListAppVersionResourceMappings",
"resiliencehub:ListAppVersionResources",
"resiliencehub:ListAppVersions",
"resiliencehub:ListRecommendationTemplates",
"resiliencehub:ListResiliencyPolicies",
"resiliencehub:ListSopRecommendations",
"resiliencehub:ListSuggestedResiliencyPolicies",
"resiliencehub:ListTagsForResource",
"resiliencehub:ListTestRecommendations",
"resiliencehub:ListUnsupportedAppVersionResources",
"resource-explorer-2:BatchGetView",
"resource-explorer-2:GetDefaultView",
"resource-explorer-2:GetIndex",
"resource-explorer-2:GetView",
"resource-explorer-2:ListIndexes",
"resource-explorer-2:ListSupportedResourceTypes",
"resource-explorer-2:ListTagsForResource",
"resource-explorer-2:ListViews",
"resource-explorer-2:Search",
"resource-groups:Get*",
"resource-groups:List*",
"resource-groups:Search*",
"robomaker:BatchDescribe*",
"robomaker:Describe*",
"robomaker:Get*",
"robomaker:List*",
"route53-recovery-cluster:Get*",
"route53-recovery-cluster:ListRoutingControls",
"route53-recovery-control-config:Describe*",
"route53-recovery-control-config:List*",
"route53-recovery-readiness:Get*",
"route53-recovery-readiness:List*",
"route53:Get*",
"route53:List*",
"route53:Test*",
"route53domains:Check*",
"route53domains:Get*",
"route53domains:List*",
"route53domains:View*",
"route53resolver:Get*",
"route53resolver:List*",
"rum:GetAppMonitor",
"rum:GetAppMonitorData",
"rum:ListAppMonitors",
"s3-object-lambda:GetObject",
"s3-object-lambda:GetObjectAcl",
"s3-object-lambda:GetObjectLegalHold",
"s3-object-lambda:GetObjectRetention",
"s3-object-lambda:GetObjectTagging",
```



```
"s3-object-lambda:GetObjectVersion",
"s3-object-lambda:GetObjectVersionAcl",
"s3-object-lambda:GetObjectVersionTagging",
"s3-object-lambda:ListBucket",
"s3-object-lambda:ListBucketMultipartUploads",
"s3-object-lambda:ListBucketVersions",
"s3-object-lambda:ListMultipartUploadParts",
"s3:DescribeJob",
"s3:Get*",
"s3:List*",
"sagemaker-groundtruth-synthetic:GetAccountDetails",
"sagemaker-groundtruth-synthetic:GetBatch",
"sagemaker-groundtruth-synthetic:GetProject",
"sagemaker-groundtruth-synthetic:ListBatchDataTransfers",
"sagemaker-groundtruth-synthetic:ListBatchSummaries",
"sagemaker-groundtruth-synthetic:ListProjectDataTransfers",
"sagemaker-groundtruth-synthetic:ListProjectSummaries",
"sagemaker:Describe*",
"sagemaker:GetSearchSuggestions",
"sagemaker:List*",
"sagemaker:Search",
"savingsplans:DescribeSavingsPlanRates",
"savingsplans:DescribeSavingsPlans",
"savingsplans:DescribeSavingsPlansOfferingRates",
"savingsplans:DescribeSavingsPlansOfferings",
"savingsplans:ListTagsForResource",
"scheduler:GetSchedule",
"scheduler:GetScheduleGroup",
"scheduler:ListScheduleGroups",
"scheduler:ListSchedules",
"scheduler:ListTagsForResource",
"schemas:Describe*",
"schemas:Get*",
"schemas:List*",
"schemas:Search*",
"sdb:Get*",
"sdb:List*",
"sdb:Select*",
"secretsmanager:Describe*",
"secretsmanager:GetResourcePolicy",
"secretsmanager:List*",
"securityhub:BatchGetStandardsControlAssociations",
"securityhub:Describe*",
"securityhub:Get*",
"securityhub:List*",
"serverlessrepo:Get*",
"serverlessrepo:List*",
"serverlessrepo:SearchApplications",
"servicecatalog:Describe*",
"servicecatalog:GetApplication",
"servicecatalog:GetAttributeGroup",
"servicecatalog:List*",
"servicecatalog:Scan*",
"servicecatalog:Search*",
"servicediscovery:Get*",
"servicediscovery:List*",
"servicequotas:GetAssociationForServiceQuotaTemplate",
"servicequotas:GetAWSDefaultServiceQuota",
"servicequotas:GetRequestedServiceQuotaChange",
"servicequotas:GetServiceQuota",
```



```
"servicequotas:GetServiceQuotaIncreaseRequestFromTemplate",
"servicequotas:ListAWSDefaultServiceQuotas",
"servicequotas:ListRequestedServiceQuotaChangeHistory",
"servicequotas:ListRequestedServiceQuotaChangeHistoryByQuota",
"servicequotas:ListServiceQuotaIncreaseRequestsInTemplate",
"servicequotas:ListServiceQuotas",
"servicequotas:ListServices",
"ses:BatchGetMetricData",
"ses:Describe*",
"ses:Get*",
"ses:List*",
"shield:Describe*",
"shield:Get*",
"shield:List*",
"signer:DescribeSigningJob",
"signer:GetSigningPlatform",
"signer:GetSigningProfile",
"signer:ListProfilePermissions",
"signer:ListSigningJobs",
"signer:ListSigningPlatforms",
"signer:ListSigningProfiles",
"signer:ListTagsForResource",
"sms-voice:DescribeAccountAttributes",
"sms-voice:DescribeAccountLimits",
"sms-voice:DescribeConfigurationSets",
"sms-voice:DescribeKeywords",
"sms-voice:DescribeOptedOutNumbers",
"sms-voice:DescribeOptOutLists",
"sms-voice:DescribePhoneNumbers",
"sms-voice:DescribePools",
"sms-voice:DescribeSenderId",
"sms-voice:DescribeSpendLimits",
"sms-voice:ListPoolOriginationIdentities",
"sms-voice:ListTagsForResource",
"snowball:Describe*",
"snowball:Get*",
"snowball:List*",
"sns:Check*",
"sns:Get*",
"sns:List*",
"sqs:Get*",
"sqs:List*",
"sqs:Receive*",
"ssm-contacts:DescribeEngagement",
"ssm-contacts:DescribePage",
"ssm-contacts:GetContact",
"ssm-contacts:GetContactChannel",
"ssm-contacts:ListContactChannels",
"ssm-contacts:ListContacts",
"ssm-contacts:ListEngagements",
"ssm-contacts:ListPageReceipts",
"ssm-contacts:ListPagesByContact",
"ssm-contacts:ListPagesByEngagement",
"ssm-incidents:GetIncidentRecord",
"ssm-incidents:GetReplicationSet",
"ssm-incidents:GetResourcePolicies",
"ssm-incidents:GetResponsePlan",
"ssm-incidents:GetTimelineEvent",
"ssm-incidents:ListIncidentRecords",
"ssm-incidents:ListRelatedItems",
```



```
"ssm-incidents:ListReplicationSets",
"ssm-incidents:ListResponsePlans",
"ssm-incidents:ListTagsForResource",
"ssm-incidents:ListTimelineEvents",
"ssm:Describe*",
"ssm:Get*",
"ssm:List*",
"sso-directory:Describe*",
"sso-directory:List*",
"sso-directory:Search*",
"sso:Describe*",
"sso:Get*",
"sso:List*",
"sso:Search*",
"states:Describe*",
"states:GetExecutionHistory",
"states:List*",
"storagegateway:Describe*",
"storagegateway:List*",
"sts:GetAccessKeyInfo",
"sts:GetCallerIdentity",
"sts:GetSessionToken",
"support:DescribeAttachment",
"support:DescribeCases",
"support:DescribeCommunications",
"support:DescribeServices",
"support:DescribeSeverityLevels",
"support:DescribeTrustedAdvisorCheckRefreshStatuses",
"support:DescribeTrustedAdvisorCheckResult",
"support:DescribeTrustedAdvisorChecks",
"support:DescribeTrustedAdvisorCheckSummaries",
"supportplans:GetSupportPlan",
"supportplans:GetSupportPlanUpdateStatus",
"sustainability:GetCarbonFootprintSummary",
"swf:Count*",
"swf:Describe*",
"swf:Get*",
"swf:List*",
"synthetics:Describe*",
"synthetics:Get*",
"synthetics:List*",
>tag:DescribeReportCreation",
>tag:Get*",
"tax:GetExemptions",
"tax:GetExemptions",
"tax:GetTaxInheritance",
"tax:GetTaxInterview",
"tax:GetTaxRegistration",
"tax:GetTaxRegistrationDocument",
"tax:ListTaxRegistrations",
"timestream:DescribeBatchLoadTask",
"timestream:DescribeDatabase",
"timestream:DescribeEndpoints",
"timestream:DescribeTable",
"timestream:ListBatchLoadTasks",
"timestream:ListDatabases",
"timestream:ListMeasures",
"timestream:ListTables",
"timestream:ListTagsForResource",
"transcribe:Get*",
```



```
"transcribe:List*",
"transfer:Describe*",
"transfer:List*",
"transfer:TestIdentityProvider",
"translate:DescribeTextTranslationJob",
"translate:GetParallelData",
"translate:GetTerminology",
"translate:ListParallelData",
"translate:ListTerminologies",
"translate:ListTextTranslationJobs",
"trustedadvisor:Describe*",
"waf-regional:Get*",
"waf-regional:List*",
"waf:Get*",
"waf:List*",
"wafv2:CheckCapacity",
"wafv2:Describe*",
"wafv2:Get*",
"wafv2:List*",
"workdocs:CheckAlias",
"workdocs:Describe*",
"workdocs:Get*",
"workmail:Describe*",
"workmail:Get*",
"workmail:List*",
"workmail:Search*",
"workspaces:Describe*",
"xray:BatchGet*",
"xray:Get*"
  ],
  "Resource": "*"
}
]
```





# Onboard an AWS Organization

Tenable Cloud Security can connect to your AWS organization's management account to discover all the member accounts under that account. This is the recommended method when you want to onboard all of your AWS accounts in Tenable Cloud Security for security assessment. You must have the required permissions to deploy a CloudFormation stack for setting up access roles in each of the member accounts.

**Tip:** For more information about AWS organizations, see Amazon's [AWS Organizations User Guide](#).


Before you begin:

You must have the following details for the read-only role in your AWS account:

- Role ARN
- External ID

For more information, see [Set Up Read-Only Access to the AWS Account](#).

To connect to an AWS organization account:

1. In the left navigation bar, click  > **Connection** > **AWS account**.
2. In the **Choose a workflow to discover AWS account(s)** section, select **Onboard AWS organization**.
3. Click **Continue**.

The **Configure management account** section appears.

4. Type the appropriate **Read Only Role ARN** and **External ID**.
5. Click **Continue**.

The **Configure member accounts** section appears.

6. Configure member accounts by performing the following actions:



- a. In the **Configure member accounts** section, in the first step, click [here](#).

Tenable Cloud Security redirects you to the **Create StackSet** wizard in the AWS Management Console. Follow these steps to [deploy the stackset](#) that creates the role for all member accounts.

**To deploy the StackSet to create a read-only role for a member account:**

- a. Sign in to the AWS management account of the target organization.
- b. Copy the appropriate URL from the **Configure member accounts** section.
- c. On the **Choose a template** page, do the following:
  - i. In the **Permissions** section, ensure that the **Service-managed permissions** option is selected.
  - ii. In the **Prerequisite - Prepare template** section, ensure that the **Template is ready** option is selected.
  - iii. In the **Template source** section, click **Amazon S3 URL**.
  - iv. In the **Amazon S3 URL** box, copy the template URL from the Tenable Cloud Security Console and paste it.
  - v. Click **Next**.
- d. On the **Specify StackSet details** page, do the following:
  - i. In the **StackSet name** section, type a name for the StackSet.

**Tip:** Choose a meaningful name because the Tenable Cloud Security role name is used for all the member accounts of the organization.
  - ii. In the **StackSet description** section, type a description for the current StackSet.
  - iii. In the **Parameters** section, type the appropriate management account ID.
  - iv. Click **Next**.
- e. On the **Configure StackSet Options** page, do the following:



- i. (Optional) In the **Tags** section, click **Add new tag** and provide a **Key** and a **Value** to specify the tag.

Tags are arbitrary key-value pairs that can be used to identify your stack. Tags that you apply to stack sets are applied to all resources created by your stacks.

- ii. For **Execution configuration**, choose **Active** so that StackSets performs non-conflicting operations concurrently and queues conflicting operations. After conflicting operations finish, StackSets starts queued operations in request order.
- iii. Click **Next**.

- f. On the **Set deployment options** page, do the following:

- i. In the **Deployment targets** section, click one of the following:

- **Deploy to organization** – Creates the role in all the member AWS accounts for the organization.
- **Deploy to organizational units (OUs)** – Creates the role in all the member AWS accounts for selected organizations.

- ii. In **Automatic deployment**, click **Enabled**.

- iii. In **Account removal behavior**, click the required option.

- g. In the **Specify regions** section, add a region available across all member accounts.

**Caution:** Select only one region. If you specify multiple regions, stack deployment succeeds only for one region and fails for others and can cause issues.

**Note:** If the selected region is not available under a particular member account, the stackset deployment fails.

- h. In the **Deployment options** section, do the following:



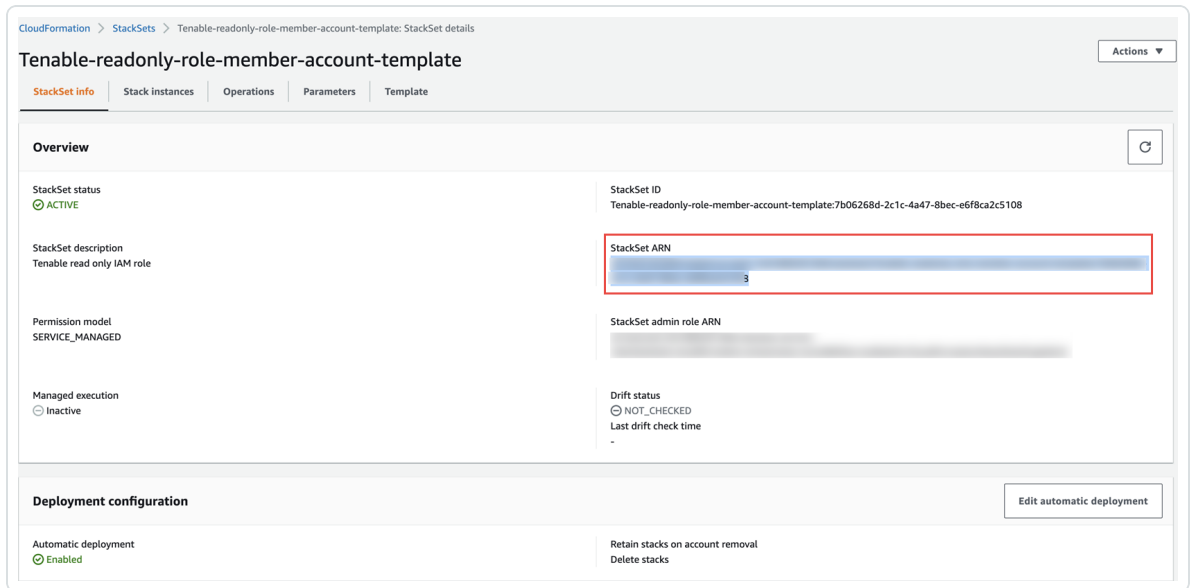
- i. In the **Maximum concurrent accounts - optional** drop-down box, select **Percentage**, and set the value to **100**.
- ii. In the **Failure tolerance - optional** drop-down box, select **Percentage**, and set the value to **100**.
- iii. In the **Regional Concurrency** section, click **Sequential**.
- iv. Click **Next**.
- i. In the **Capabilities** section, select the **I acknowledge that AWS CloudFormation might create IAM resources with custom names**.check box to confirm.
- j. Click **Submit**.

The **StackSet details** page appears. Wait for the status of the StackSet to change to **Succeeded**.

The screenshot shows the AWS CloudFormation console interface for a StackSet. The breadcrumb trail is 'CloudFormation > StackSets > Tenable-readonly-role-member-account-template: StackSet details'. The title is 'Tenable-readonly-role-member-account-template'. There are tabs for 'StackSet info', 'Stack instances', 'Operations', 'Parameters', and 'Template'. The 'Operations' tab is active, showing a search bar and a table with one entry. The table has columns for 'Operation ID', 'Type', 'Status', 'Created time', and 'Completed time'. The entry shows a 'CREATE' operation that has 'SUCCEEDED'.

Operation ID	Type	Status	Created time	Completed time
1778681c-fd4a-8d0d-4fd5-46b0da2556d0	CREATE	SUCCEEDED	2022-06-29 18:00:11 UTC+0530	2022-06-29 18:01:36 UTC+0530

- k. Click the **StackSet Info** tab and copy the **StackSet ARN**.



- b. In the Tenable Cloud Security Console, paste the Stacksets ARN copied in the previous step in the **Stacksets ARN** box.
- c. Click **Continue**.

The **Discover and onboard member accounts** section appears. Tenable Cloud Security deploys the StackSet used to create a Tenable Cloud Security role for each member account.

## 7. Onboard member accounts.

- a. In the **Discover and onboard member accounts** section, in the list, select the cloud member accounts that you want to onboard.

**Tip:** You can also search for specific cloud accounts and filter the list by organizations.

- b. (Optional) To create a new project automatically for the AWS organization, select the **Map accounts automatically** check box.

Tenable Cloud Security creates a new project for the AWS organization and links all AWS member accounts with the project.

8. In the **Choose prerequisites** section, select the check boxes:



- Ensure that you have granted all permissions.
- Ensure that you already have snapshots or followed the provided instructions to create snapshots for the instances you wish to scan.

Click the links to view documentation for providing permissions to Tenable Cloud Security for scanning and creating snapshots for Agentless Assessment.

9. Click **Onboard accounts**.

On the **Projects & Connections** page, the AWS project links to the connected AWS organization's account and the selected VPCs.



---

# Onboard an AWS Account

---

You can connect your Amazon Web Services (AWS) account as part of your AWS project. Use this method if you want to onboard each of your AWS account manually without deploying a CloudFormation template.


Before you begin:

You must have the following details for the read-only role in for your AWS account:

- Role ARN
- External ID

For more information, see [Set Up Read-Only Access to the AWS Account](#).

To connect an AWS account:

1. In the left navigation bar of the Tenable Cloud Security page, click  > **Connection > AWS account**.
2. In the **Choose a workflow to discover AWS accounts** section, click **Onboard AWS account**.
3. Click **Continue**.

The **Configure AWS account** section appears.

4. Type the appropriate **Read Only Role ARN** and **External ID**.
5. Click **Continue**.
6. In the **Choose projects to add the AWS account(s) to** section, select the project that you created for the AWS account.

For more information, see [Create a Project](#).

7. In the **Choose prerequisites** section, select the check boxes:
  - Ensure that you have granted all permissions.
  - Ensure that you already have snapshots or followed the provided instructions to create snapshots for the instances you wish to scan.



---

Click the links to view documentation for providing permissions to Tenable Cloud Security for scanning and creating snapshots for Agentless Assessment.

8. Click **Connect Cloud Account**.

You can view the AWS project linked to the connected AWS account and the selected VPCs on the **Projects & Connections** page.





# Onboard an Azure Account


In Tenable Cloud Security, you can connect your Microsoft Azure cloud account using a service principal. In Microsoft Azure, a service principal is an entity that requires access to the resources secured by a Microsoft Entra ID tenant.

Before you begin:

- Ensure you have the following Azure values:
  - Client ID
  - Secret value
  - Tenant ID

For more information, see [Create an Azure Service Principal Role](#).

To connect an Azure subscription with a service principal:

1. In the left navigation bar, click  > **Connection** > **Azure subscription**.
2. In the **Choose a workflow to discover Azure subscriptions** section, click **Service principal (recommended)**.
3. Click **Continue**.
4. In the **Discover Azure subscription(s)** section, enter your **Client ID**, **Secret value**, and **Tenant ID**.
5. Click **Continue**.

Tenable Cloud Security connects to your Microsoft Azure account using the specified credentials, and displays the list of subscriptions.

6. In the **Choose Azure subscription(s)** section, select the required subscriptions.
7. Click **Continue**.
8. For the selected subscriptions, in the **Choose resource group(s)** section, do one of the following:



- To select all available resource groups, click **All (recommended)**.
- To select specific resource groups, click **Specific**, and select a resource group in the list.

**Tip:** You can search for specific resource groups, and filter the list by subscriptions.

9. Click **Continue**.

10. (Optional) In the **Choose projects to add the Azure project(s) to** section, create or select a project for the Azure subscription.

- To create a new project for your Azure account, click **Add a project**. For more information, see [Create a Project](#).
- Select a project from the list.

**Tip:** You can also search for specific projects.

11. In the **Choose prerequisites** section, select the check boxes:

- Ensure that you have granted all permissions.
- Ensure that you already have snapshots or or followed the provided instructions to create snapshots for the instances you wish to scan.

Click the links to view documentation for providing permissions to Tenable Cloud Security for scanning and creating snapshots for Agentless Assessment.

12. Click **Connect Cloud Account**.

On the **Projects & Connections** page, you can view the Azure project with the connected Azure account and view the selected VPCs.

## Create an Azure Service Principal Role

Tenable Cloud Security requires adequate permissions to read the resources in your Azure subscription. Provision a service principal role in the target Azure subscription and configure it for Tenable Cloud Security to read the resources in the same account.

The following permissions are required for a vulnerability scan of Azure virtual machines:



- Reader
- Disk Snapshot Contributor

Follow these steps to create a service principal and assign a role to it:

1. [Register an application with Azure to create the service principal.](#)
2. Choose one of the following options to assign a role to the service principal for accessing the resources in your subscription:
  - [Create and assign a custom role with expanded Read access \(comprehensive\) to the service principal.](#)
  - [Assign the built-in Reader role \(limited\) to the service principal.](#)
3. [Create a client secret for authenticating the service principal from Tenable Cloud Security.](#)



# Register an application with Azure

When you register an application through the Azure portal, Azure automatically creates an application object and service principal in your tenant. For more information on the relationship between application registration, application objects, and service principals, see [Application and service principal objects in Microsoft Entra ID](#).

To create a service principal role in Azure:

1. Log in to the [Microsoft Azure portal](#).
2. In the home page, click **App registrations**.

The **App registrations** page appears.

3. Click **New registration**.

The **Register an application** page appears.

4. Type a name for the application you want to register.
5. Click **Register**.

The application details page appears.

Home > App registrations >

tenablecs-app

Search (Ctrl+/) << Delete Endpoints Preview features

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Essentials

Display name : [tenablecs-app](#)

Application (client) ID : [redacted]

Object ID : [redacted]

Directory (tenant) ID : [redacted]

Supported account types : [My organization only](#)

Client credentials : [Add a certificate or secret](#)

Redirect URIs : [Add a Redirect URI](#)

Application ID URI : [Add an Application ID URI](#)

Managed application in L... : [tenablecs-app](#)

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

6. Note down the following values. You need these values when onboarding the service account in Tenable Cloud Security:

- **Application (client) ID:** This is the client ID requested by Tenable Cloud Security.
- **Directory (tenant) ID:** This is the Tenant ID requested by Tenable Cloud Security.



## Create a custom role and assign it to the service principal

For a comprehensive Azure cloud scan for resources such as Storage Account, Kubernetes Cluster, Cosmos DB, Function App resources, create a custom role with expanded read access including the list APIs access. Additionally, Agentless Assessment requires the Disk Snapshot Contributor role along with the Reader role for scanning virtual machine snapshots.

For more information about these permissions, see [Azure built-in roles](#) in Azure documentation.

To create a custom role and assign it to the service principal:

1. On the home page of the Azure portal, do one of the following:
  - To create a role for a management group, click **Management groups**.  
The **Management groups** page appears.
  - To create a role for a subscription, click **Subscriptions**.  
The **Subscriptions** page appears.


**Note:** To enable Tenable Cloud Security to discover all subscriptions under a management group, ensure that the service principal role is assigned to the management group. You can also assign the role to a root management group to discover all subscriptions under the root management group.

2. On the left navigation bar, click **Access Control (IAM)**.  
The **Access control (IAM)** page for your subscription appears.
3. In the **Create a custom role** section, click **Add**.  
The **Create a custom role** page appears.



Home > Subscriptions > tenable-acc-test | Access control (IAM) >


## Create a custom role ...

 Got feedback?

**Basics**   Permissions   Assignable scopes   JSON   Review + create

To create a custom role for Azure resources, fill out some basic information. [Learn more](#) 

\* Custom role name ⓘ

Tenablecs-ReaderPlusStorageAccountRead 

Description


Custom role for Tenable.cs

Baseline permissions ⓘ

Clone a role

Start from scratch

Start from JSON

"tenablecs\_customrole.json" 

**Review + create**

Previous

Next

4. In **Baseline permissions**, select the **Start from JSON** option.


You can create a custom role in the following ways:



- **Clone a role:** Create a custom role by cloning an existing role and modifying the role, as required.
- **Start from scratch:** Create a custom role by using the Azure user interface.
- **Start from JSON:** Create a custom role by uploading a JSON file with the required permissions.

For more information about these methods, see [Create or update Azure custom roles using the Azure portal](#) in Azure documentation.

**Note:** This procedure describes how to create a custom role using a JSON file.

5. Click  to upload a JSON file that has the required permissions.

Azure validates the JSON file and uploads the file for role creation.

The following sample JSON file creates a role with read permissions along with the list APIs for the Storage Accounts, Kubernetes cluster, Cosmos DB, and Function App services for a **subscription**:

```
{
  "properties": {
    "roleName": "Tenablecs-ReaderPlusStorageAccountRead",
    "description": "Custom role for Tenable Cloud Security",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "*/read",
          "Microsoft.Storage/storageAccounts/listkeys/action",
          "Microsoft.Storage/storageAccounts/listAccountSas/action",
          "Microsoft.Storage/storageAccounts/listServiceSas/action",
          "Microsoft.Storage/storageAccounts/localusers/listKeys/action",
          "Microsoft.ContainerService/managedClusters/accessProfiles/listCredential/a-
action",
          "Microsoft.DocumentDB/databaseAccounts/listKeys/action",
          "Microsoft.DocumentDB/databaseAccounts/readonlykeys/action",
          "Microsoft.DocumentDB/databaseAccounts/listConnectionStrings/action",
          "Microsoft.Web/sites/config/list/action"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```



```
]
}
}
```

The following sample JSON file creates a role with read permissions along with the list APIs for the Storage Accounts, Kubernetes cluster, Cosmos DB, and Function App services for a **management group**:

```
{
  "properties": {
    "roleName": "Tenablecs-ReaderPlusStorageAccountRead",
    "description": "Custom role for Tenable Cloud Security",
    "assignableScopes": [
      "/providers/Microsoft.Management/managementGroups/<management-group-ID>"
    ],
    "permissions": [
      {
        "actions": [
          "*/read",
          "Microsoft.Storage/storageAccounts/listkeys/action",
          "Microsoft.Storage/storageAccounts/listAccountSas/action",
          "Microsoft.Storage/storageAccounts/listServiceSas/action",
          "Microsoft.Storage/storageAccounts/localusers/listKeys/action",
          "Microsoft.ContainerService/managedClusters/accessProfiles/listCredential/a-
action",
          "Microsoft.DocumentDB/databaseAccounts/listKeys/action",
          "Microsoft.DocumentDB/databaseAccounts/readonlykeys/action",
          "Microsoft.DocumentDB/databaseAccounts/listConnectionStrings/action",
          "Microsoft.Web/sites/config/list/action"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```

The following sample JSON file creates a custom role with read permissions along with permissions to access snapshots at a subscription-level, which is required for Agentless Assessment:

```
{
  "properties": {
    "roleName": "Tenablecs-ReaderPlusDiskSnapshotContributor",
    "description": "Custom role for Tenable Cloud Security",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ]
  }
}
```






```
    ],
    "permissions": [
      {
        "actions": [
          "*/read",
          "Microsoft.Storage/storageAccounts/listkeys/action",
          "Microsoft.Storage/storageAccounts/listAccountSas/action",
          "Microsoft.Storage/storageAccounts/listServiceSas/action",
          "Microsoft.Storage/storageAccounts/localusers/listKeys/action",
          "Microsoft.Compute/snapshots/beginGetAccess/action"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```

where <subscription-id> is your Azure subscription ID.

6. Click **Review + create**.

The **Review + create** tab appears.

## Create a custom role ...

 Got feedback?

[Basics](#) [Permissions](#) [Assignable scopes](#) [JSON](#) [Review + create](#)

### Basics

Role name Tenablecs-ReaderPlusStorageAccountRead

Role description Custom role for Tenable.cs

### Permissions

Action \*/read

Action Microsoft.Storage/storageAccounts/listkeys/action

Action Microsoft.Storage/storageAccounts/listAccountSas/action

Action Microsoft.Storage/storageAccounts/listServiceSas/action

Action Microsoft.Storage/storageAccounts/localusers/listKeys/action

Action Microsoft.ContainerService/managedClusters/accessProfiles/listCredential/action

Action Microsoft.DocumentDB/databaseAccounts/listKeys/action

Action Microsoft.DocumentDB/databaseAccounts/readonlykeys/action

Action Microsoft.DocumentDB/databaseAccounts/listConnectionStrings/action

[Create](#)

[Previous](#)

### 7. Click **Create**.

Azure creates the custom role and redirects you to the **Access control (IAM)** page.

### 8. In the **Grant access to this resource** section, click **Add role assignment** to assign the custom role to the service principal.

The **Add role assignment** page appears.



9. On the **Role** tab, search for the custom role you created.
10. Select the custom role and click **Next**.

The **Members** tab appears.

11. On the **Members** tab, do the following:

- a. Click **Select Members**.
- b. In the **Select members** window, search for the application you created.
- c. Select the application.

The application appears under **Selected members**.

- d. Click **Select**.

Azure adds the application for assigning the selected custom role.

- e. Click **Next**.

The **Review + assign** tab appears.

12. Review the details of the role and click **Review + assign**.

### Add role assignment ...

Got feedback?

[Role](#) [Members](#) [Review + assign](#)

**Role** Tenablecs-ReaderPlusStorageAccountRead

**Scope** /subscriptions/ [redacted]

Members	Name	Object ID	Type
	tenablecs-app	[redacted]	App

**Description** No description

Azure assigns the custom role to the service principal of the application and redirects you to the **Access control (IAM)** page.



---

## Assign the Reader role to the service principal

---

Tenable Cloud Security requires the **Reader** role for accessing the resources for a cloud scan. This role provides limited permissions to the service principal. If you want to perform a comprehensive scan including managed clusters and storage accounts, [create a custom role with expanded read permissions](#).

1. On the home page of the Azure portal, do one of the following:

- To assign the role to a management group, click **Management groups**.

The **Management groups** page appears.

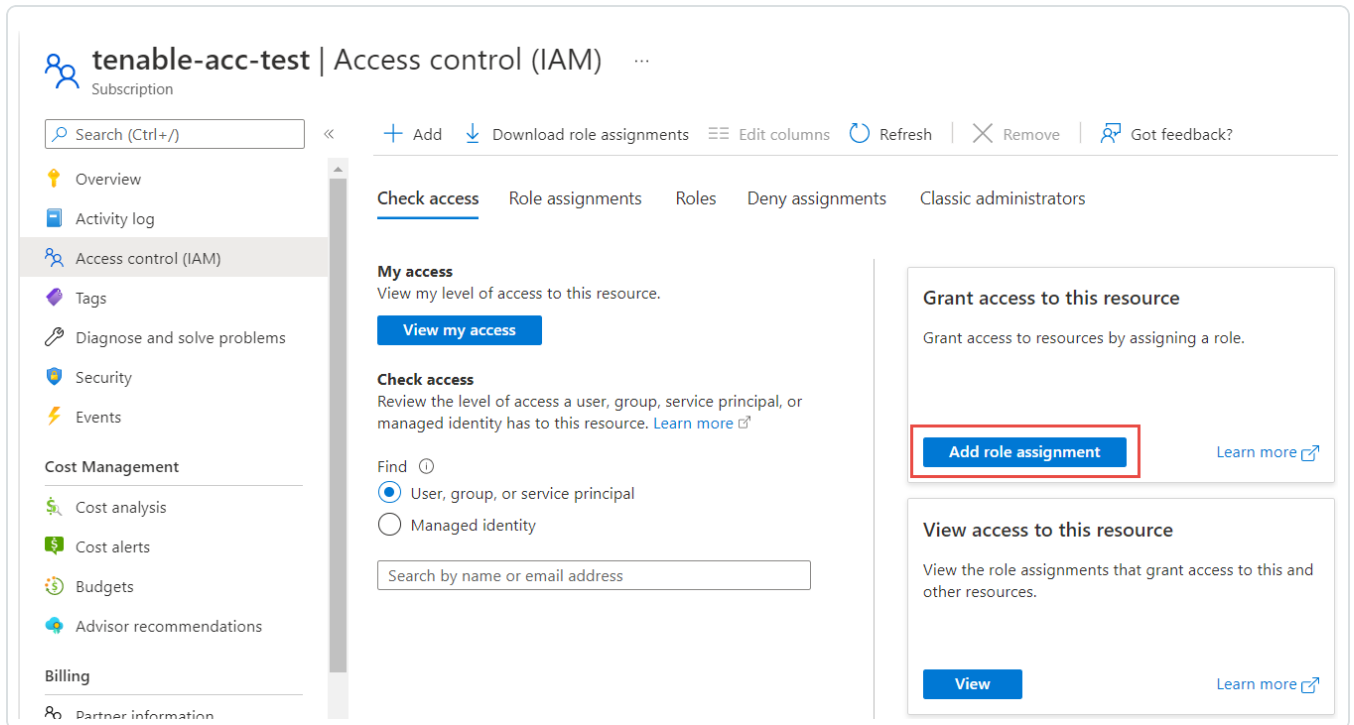
**Note:** To enable Tenable Cloud Security to discover all subscriptions under a management group, ensure that the service principal role is assigned to the management group. You can also assign the role to a root management group to discover all subscriptions under the root management group.

- To create a role for a subscription, click **Subscriptions**.

The **Subscriptions** page appears.

2. On the left navigation bar, click **Access Control (IAM)**.

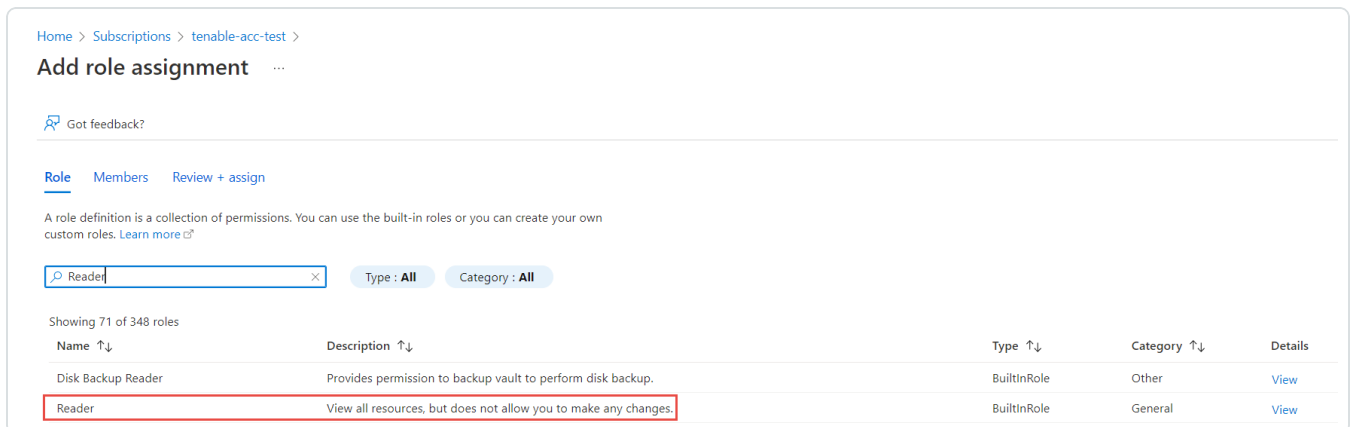
The **Access control (IAM)** page for your subscription appears.



3. In the **Grant access to this resource** section, click **Add role assignment**.

The **Add role assignment** page appears.

4. On the **Role** tab, search for the **Reader** role.



5. Select the **Reader** role and click **Next**.

The **Members** tab appears.

6. On the **Members** tab, do the following:



- a. Click **Select Members**.
- b. In the **Select members** window, search for the application you created.
- c. Select the application.

The application appears under **Selected members**.

Home > Subscriptions > tenable-acc-test >  
**Add role assignment** ...

Got feedback?

Role **Members** Review + assign

**Assign access to**  User, group, or service principal  
 Managed identity

**Members** + Select members

Name	Object ID	Type
No members selected		

**Description** Optional

**Select members**

Select

- TenableCS
- Tenablecsappreggarnation

**Selected members:**

- tenables-app**

- d. Click **Select**.

Azure adds the application for assigning the **Reader** role.

Home > Subscriptions > tenable-acc-test >  
**Add role assignment** ...

Got feedback?

Role **Members** Review + assign

**Assign access to**  User, group, or service principal  
 Managed identity

**Members** + Select members

Name	Object ID	Type
tenables-app	[REDACTED]	App

**Description** Optional

- e. Click **Next**.

The **Review + assign** tab appears.



7. Review the details of the role and click **Review + assign**.

Home > Subscriptions > tenable-acc-test >

## Add role assignment ...

Got feedback?

---

[Role](#) [Members](#) [Review + assign](#)

**Role** Reader

**Scope** [Redacted]

<b>Members</b>		Name	Object ID	Type
		tenablecs-app	[Redacted]	App

**Description** No description

Azure assigns the role to the service principal of the application and redirects you to the **Access control (IAM)** page.



# Create a client secret

You can create a new application secret to authenticate the service principal.

1. On the home page of the Azure portal, click **App Registrations**.
2. Click the application that you created for Tenable Cloud Security.
3. On the left navigation bar, click **Certificates & secrets**.

The **Certificates & secrets** page appears.

4. Click **New client secret**.

The **Add a client secret** page appears.

## Add a client secret ✕

Description

Expires

5. Provide a relevant description for the secret. For example, Tenable Cloud Security Scan.
6. Set an expiration for the client secret.
7. Click **Add**.

The client secret value and ID appear.

Certificates (0)	<b>Client secrets (1)</b>	Federated credentials (0)	
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.			
<a href="#">+ New client secret</a>			
Description	Expires	Value ⓘ	Secret ID
Password uploaded on Wed Jul 27 2022	1/27/2023	<input type="text" value=""/>	<input type="text" value=""/>





8. Record the **Value** of this client secret.

**Note:** You cannot view this value again because Azure masks this value.

What to do next:

### [Onboard an Azure Account](#)

You must have the following values for onboarding the Azure account in Tenable Cloud Security:

- Client ID
- Tenant ID
- Secret value



---

# Onboard a GCP Service Account

---

You can connect your Google Cloud Platform (GCP) account using a Google service account in Tenable Cloud Security. For a detailed workflow for onboarding GCP accounts, see the [Tenable Cloud Security Quick Reference Guide: Onboarding GCP Accounts](#).

Before you begin:

- Make sure you have the private key or GCP credentials file (JSON) for your service account and activated your service account.

For more information, see [Create a GCP Service Account](#) and [Activate the GCP Service Account](#).

To connect to a GCP service account from Tenable Cloud Security:

1. [Log in](#) to Tenable Vulnerability Management.

2. In the left navigation bar, click **Cloud Security**.

The Tenable Cloud Security page opens. By default, a dashboard appears that shows various statistics.

3. In the left navigation bar, click  > **Connection** > **GCP service account**.

4. In the **Choose a workflow to discover GCP service account(s)** section, click **Service account credentials (recommended)**.

5. Click **Continue**.

6. To upload the service account credential file, in the **Discover GCP service account(s)** section, click **Upload** and select the private key JSON file.

7. Click **Continue**.

8. For the discovered account, in the **Choose GCP project(s)** section, do one of the following:



- To select all available GCP projects, click **All (recommended)**.
- To select specific projects, click **Specific**, then select a GCP project.

**Tip:** You can search for a specific project.

9. Click **Continue**.
10. (Optional) In the **Choose projects to add the GCP project(s) to** section, create or select a project for the GCP instance.
  - To create a new project for your GCP account, click **Add a project**. For more information, see [Create a Project](#).
  - Select a project from the list.
11. Click **Connect Cloud Account**.

You can view the GCP projects linked to the connected GCP account on the **Projects & Connections** page.

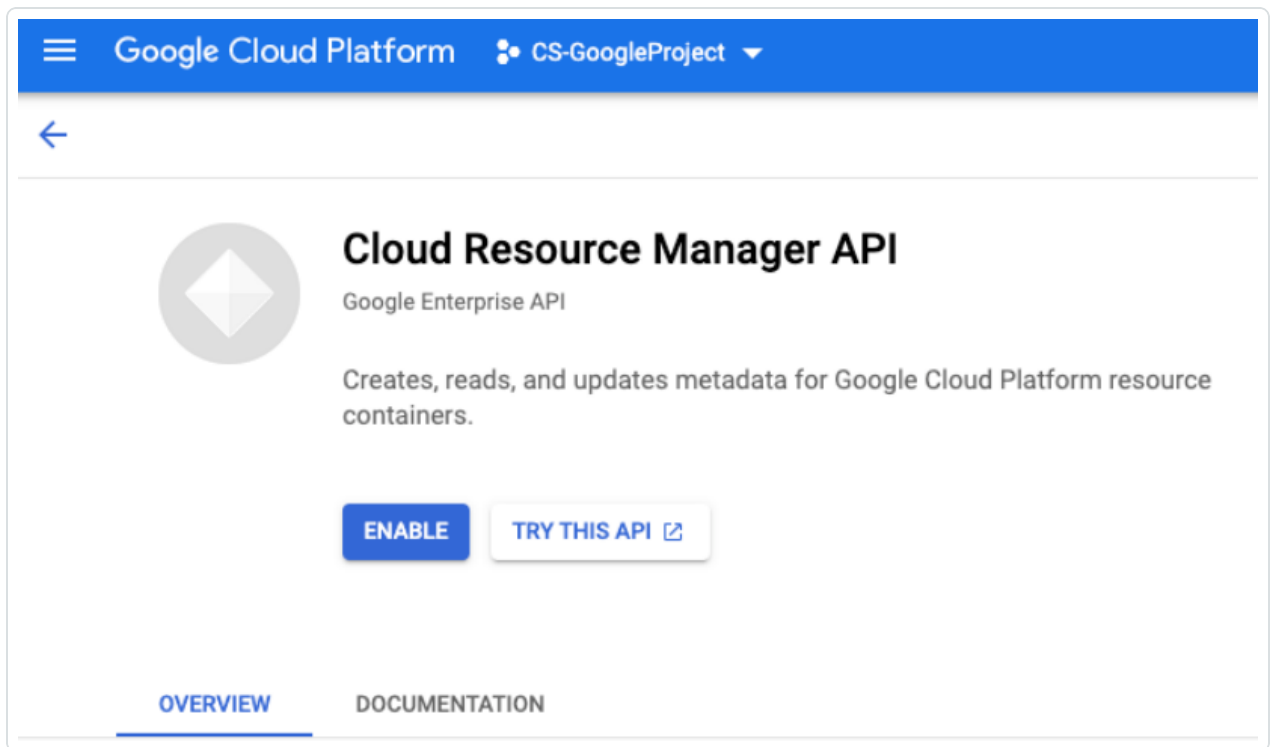


# Create a GCP Service Account

Create a service account for Tenable Cloud Security in Google cloud and then provide read-only access for this service account to your Google cloud project. This provides Tenable Cloud Security with authorized access to the resources in the Google cloud project.

To create a GCP service account:

1. Log in to the Google Cloud console.
2. Select your GCP project from the drop-down box in the top panel.
3. Enable the **Cloud Resource Manager API** service.
  - a. Search for **Cloud Resource Manager API** in the search box.
  - b. Click **Enable**.



4. On the left navigation bar of the the Google Cloud dashboard, click **IAM & Admin > Service Accounts**.

The **Service accounts** page appears.



5. Click **+ Create Service Account** to create the service account.

The **Create service account** page appears.

6. In the **Service account details** section, provide the following information:

- **Service account name:** Name of the service account you are creating.
- **Service account ID:** The **Service account ID** box populates automatically with the name of the service account. The email address of the service account uses this ID. Change the ID, if required.
- **Service account description:** A description for the service account.

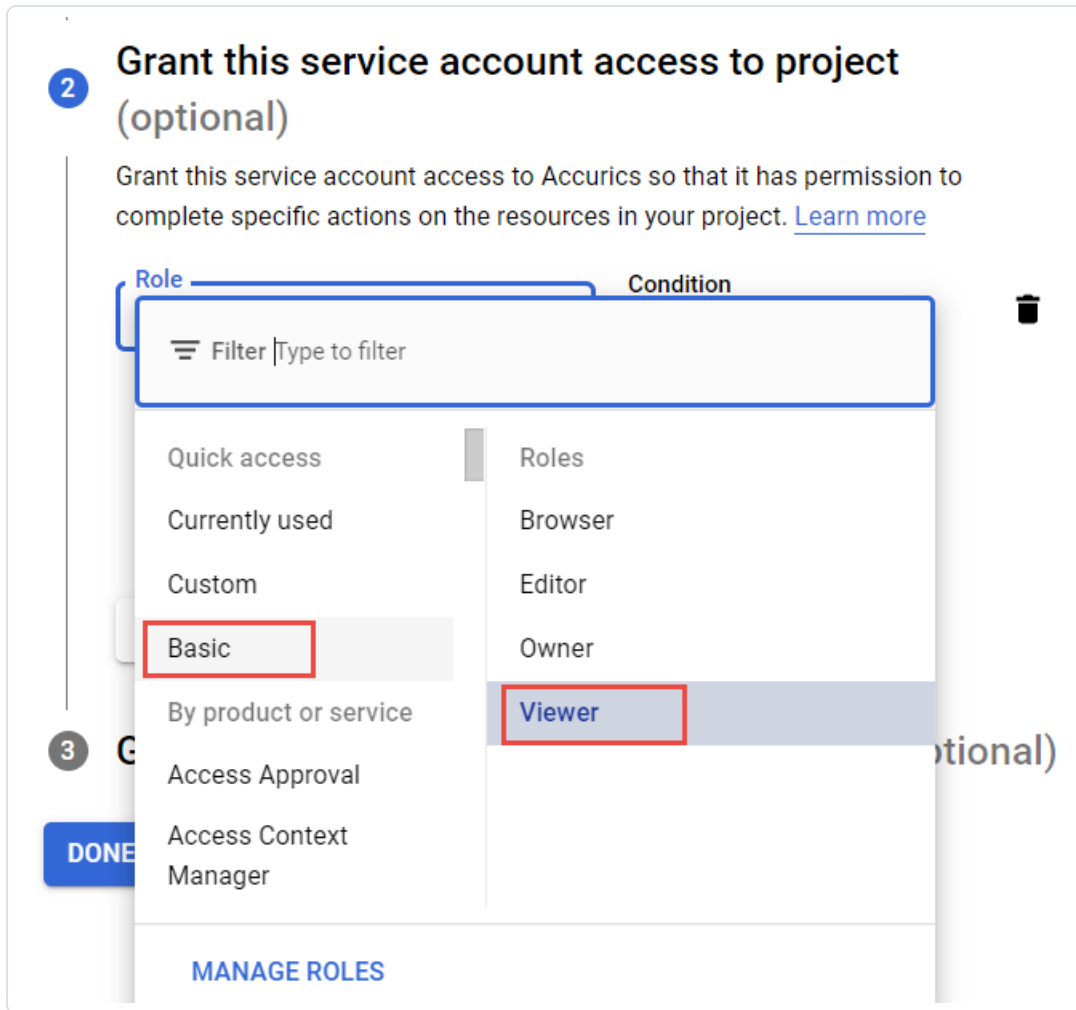
The screenshot shows the Google Cloud IAM & Admin console. The left sidebar is titled 'IAM & Admin' and includes a list of navigation items: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (highlighted), Workload Identity Federat..., Labels, and Tags. The main content area is titled 'Create service account' and features a 'Service account details' section with a blue '1' icon. The form contains the following fields: 'Service account name' with the value 'tenablecssvc' and a subtext 'Display name for this service account'; 'Service account ID \*' with the value 'tenablecssvc' and icons for 'X' and refresh; 'Email address' with the value 'tenablecssvc@accurics.iam.gserviceaccount.com' and a copy icon; and 'Service account description' with the value 'Service account for Tenable.cs' and a subtext 'Describe what this service account will do'. A 'CREATE AND CONTINUE' button is located at the bottom of the form.

7. Click **Create and Continue**.

Google Cloud displays a confirmation message that the service account creation is complete.

8. In the **Grant this service account access to project (optional)** section, provide the service account with access to the GCP project by adding the following role:

- **Viewer:** Click **Basic > Viewer** in the **Role** drop-down box.



This role provides access to Tenable Cloud Security to view most Google Cloud resources. For more information about basic roles, see [Basic roles](#) in Google documentation. You can see the list of included permissions for the **Viewer** role from the **Roles** page.

The screenshot displays the Google Cloud IAM & Admin console interface. On the left is a navigation sidebar with the following items: IAM & Admin (selected), Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts, Workload Identity Federat..., Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Roles (highlighted), Audit Logs, Manage Resources, Release Notes, and a back arrow. The main content area is titled 'Viewer' and includes a back arrow, '+ EDIT ROLE', and 'CREATE FROM ROLE' buttons. Below the title, there is a table with two rows: 'ID' with value 'roles/viewer' and 'Role launch stage' with value 'General Availability'. A 'Description' section follows, stating 'View most Google Cloud resources. See the list of included permissions.' Below that, a section titled '2692 assigned permissions' lists various permissions such as 'accessapproval.requests.get', 'accesscontextmanager.accessLevels.get', and 'aiplatform.annotationSpecs.get'.

9. Click **Continue**.

Google Cloud displays a confirmation message that the policy update is complete.

10. (Optional) In the **Grant users access to this service account (optional)** section, add users or groups that need access to this service account.

11. Click **Done**.

The **Service accounts** page appears with the list of service accounts.



### Service accounts for project "Accurics"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter Enter property name or value

<input type="checkbox"/>	Email	Status	Name ↑	Description	Key ID	Key creati	Actions
<input type="checkbox"/>	tenablecssvc@accurics.iam.gserviceaccount.com		tenablecssvc	Service account for Tenable.cs	No keys		

12. Click the service account that you created.

The **Service account details** page for the service account appears.

13. Click the **Keys** tab.

The **Keys** page appears.

← tenablecssvc

DETAILS PERMISSIONS **KEYS** METRICS LOGS

### Keys

Service account keys could pose a security risk if compromised. We recommend you about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).  
[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

- Create new key
- Upload existing key

Key creation date	Key expiration date
-------------------	---------------------

14. Click **Add Key > Create new key**.

The **Create private key** page appears.





## Create private key for "tenablecssvc"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

### Key type

JSON

Recommended

P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

15. In the **Key type** section, select **JSON** and click **Create**.

A confirmation message appears that the private key JSON file is saved to your computer.

16. Click **Close** to close the confirmation message.

The new private key and its details appear.

← tenablecssvc HELP ASSISTANT

DETAILS PERMISSIONS **KEYS** METRICS LOGS

### Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).  
[Learn more about setting organization policies for service accounts](#)

**ADD KEY** ▾

Type	Status	Key	Key creation date	Key expiration date	
	Active		Jul 5, 2022	Jan 1, 10000	

What to do next:

[Activate the GCP Service Account.](#)

## Activate the GCP Service Account



After creating the service account for Tenable Cloud Security, you must authorize this service account to access the Google Cloud resources using the Google Cloud CLI. Use the `gcloud auth activate-service-account` command to import the credentials from the JSON file with the private authorization key for the service account and activate it for use.

Before you begin:

- Install the `gcloud` CLI.

For more information, see [Install the gcloud CLI](#).

To activate the GCP service account:

1. From the `gcloud` CLI, run the following command:

```
gcloud auth activate-service-account --key-file=<KEY_FILE>
```

Where:

- **KEY\_FILE** is the path to the JSON key file for the service account. For more information, see [Create a GCP Service Account](#).

```
$ gcloud auth activate-service-account --key-file="C:\tenablecs-0cf0be2a244e.json"
Activated service account credentials for: [tenablecssvc@tenablecs.iam.gserviceaccount.com]
```

2. Verify that you can list the GCP project with the service account credentials:

```
gcloud projects list --sort-by=projectId
```

```
$ gcloud projects list --sort-by=projectId
PROJECT_ID  NAME                PROJECT_NUMBER
tenablecs   CS-GoogleProject   XXXXXXXXXXXXX
```

What to do next:

[Onboard a GCP Service Account](#)



---

## Discover Cloud Accounts

---

Tenable Cloud Security can automatically discover your cloud accounts and onboard them. Tenable Cloud Security provides a new cloud account onboarding flow that supports the following:

- **Single account onboarding** – Provide the credentials of an AWS, Azure, or GCP account to onboard the account.
- **Multiple account discovery and onboarding** – Tenable Cloud Security can automatically discover the following:
  - AWS: Provide the credentials of the AWS management account and Tenable Cloud Security automatically discovers all member accounts in that AWS organization.
  - Azure: Provide the tenant-level credentials and Tenable Cloud Security automatically discovers all Azure subscriptions in that tenant.
  - GCP: Provide the credentials of the GCP organization account and Tenable Cloud Security automatically discovers all projects in that organization.

Tenable Cloud Security schedules discovery every 24 hours and automatically discovers any new member accounts in the organization created after the initial onboarding. No user intervention is required to onboard single accounts after onboarding the management account.

For more information, see the following topics:

[Cloud Account Statuses](#)

[Discover and Onboard AWS Accounts](#)

[Discover Azure Accounts](#)

[Discover GCP Accounts](#)

[Manage Cloud Accounts](#)

[Cloud Account Discovery FAQ](#)



## Cloud Account Statuses

The cloud accounts can have one of the following status depending on the discovery and scan status. The tasks that you can perform on a cloud account also depends on its status.

Status	Description	Actions Allowed
<b>Discovered</b>	This status indicates one of the following: <ul style="list-style-type: none"><li>• Discovered accounts that must be configured before Tenable Cloud Security can scan these accounts.</li><li>• Configured accounts that are not yet scanned.</li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Configure</a></li><li>• <a href="#">Ignore</a></li></ul>
<b>Assessing</b>	Scan is in progress.	None
<b>Assessed</b>	Scan is successful for the account.	<ul style="list-style-type: none"><li>• <a href="#">Configure</a></li><li>• <a href="#">Ignore</a></li></ul>
<b>Failed</b>	Scanning of the account failed.	<ul style="list-style-type: none"><li>• <a href="#">Configure</a></li><li>• <a href="#">Ignore</a></li></ul>
<b>Ignored</b>	The account is excluded from the scan. If you configure an ignored account, the account status remains <b>Ignored</b> . Configure and scan these accounts to move the status to <b>Assessed</b> or <b>Failed</b> , depending on the scan status.	<ul style="list-style-type: none"><li>• <a href="#">Configure</a></li></ul>
<b>Suspended</b>	The account is suspended by the cloud provider and can be scanned only after it is reactivated.	None
<b>Deleted</b>	The account has been deleted.	<ul style="list-style-type: none"><li>• <a href="#">Configure</a></li></ul>



---

## Discover and Onboard AWS Accounts

---

Cloud discovery in Tenable Cloud Security supports onboarding a single AWS account and an AWS organization. To onboard an AWS organization, provide the role details of the management account and Tenable Cloud Security automatically discovers the member accounts. After Tenable Cloud Security discovers the member accounts, you must configure the member accounts by providing the credentials before you can run a cloud scan for detecting vulnerabilities and misconfigurations in the cloud account. To onboard AWS accounts, perform the following tasks:

1. [Discover AWS Accounts](#).
2. [Configure AWS Member Accounts](#).

### Discover AWS Accounts

Before you begin:

- Create an IAM role with read access to the AWS account that you are onboarding. For more information, see [Set Up Read-Only Access to the AWS Account](#).

To discover an AWS cloud account:

1. Click **Projects and Connections**.
2. Click **Cloud Accounts**.

The **Cloud Accounts** page lists all onboarded cloud accounts.


3. Click **Discover accounts > AWS**.


The **Configure AWS Management Account(s)** window appears.


4. Onboard a single AWS account or an AWS organization.
  - a. In the **Account type** section, select one of the following:
    - **Single** for onboarding a single AWS account.
    - **Multiple** for onboarding an AWS organization.
  - b. Enter the **Read-only Role ARN** and **External ID** of the AWS account.



If you selected **Multiple** account type, provide the credentials of the AWS management account.

- c. (Optional) Click  to add more accounts.
- d. Click **Discover**.

For multiple accounts, Tenable Cloud Security discovers and shows all member accounts under the management account with the status as **Discovered**. The management account appears with the  icon next to it.

For AWS management account, Tenable Cloud Security schedules discovery every 24 hours and automatically discovers any new member accounts in the AWS organization. All accounts discovered in the last 7 days show the  label until they are configured or ignored.


## Configure AWS Accounts

Configure the discovered accounts before you can run a cloud scan to assess the resources in the account for misconfigurations and vulnerabilities. To configure an AWS account, provide the read-only role ARN and external ID of the AWS account and assign the account to a project.

Before you begin:

- For multiple account type, you must have the credentials (Role ARN and External ID) of the member accounts.

To configure an AWS member account:

1. Click  > **Configure** in the row for the account that you want to configure.

The **Configure AWS Account** window appears.

2. Provide the **Read-only Role ARN** and **External ID** for the AWS account.
3. Click **Next**.
4. In the **Assign a Project or Create a New Project** section, do one of the following:

- Select a project from the list of AWS projects.

You can search for a project in the **Search projects** box.



- Click **New Project** to create a new AWS project.
  - a. Type a **Project name** for your new project.
  - b. Select AWS for the provider.
  - c. Click **Create New Project**.

Tenable Cloud Security creates the new project and Tenable Cloud Security automatically selects this project for onboarding the AWS account.

5. Click **Save**.

The **Cloud Accounts** page appears and shows the project assigned to the account.

6. Repeat these steps for all the discovered GCP projects you want to configure.

What to do next:

- Go to the **Projects** tab and run a cloud scan for the project. For more information, see [Run a Cloud Scan](#).



---

## Discover Azure Accounts

---

Cloud discovery in Tenable Cloud Security supports onboarding single and multiple Azure subscriptions. To onboard an Azure tenant, provide the role details of the Azure tenant-level credentials and Tenable Cloud Security automatically discovers the subscriptions in that tenant. After Tenable Cloud Security discovers the subscriptions, you must configure them by providing the credentials before you can run a cloud scan for detecting misconfigurations in the cloud account. To onboard Azure subscriptions, perform the following tasks:

1. [Discover Azure Subscriptions.](#)
2. [Configure Azure Subscriptions.](#)

## Discover Azure Subscriptions

Before you begin:

- Create an Azure service principal role with read access to the Azure subscriptions you are onboarding. For more information, see [Create an Azure Service Principal Role.](#)

**Note:** Ensure that you create the service principal role at the correct scope (management group or subscription) and provide required permissions for Tenable Cloud Security cloud account autodiscovery to work. For example, to discover all subscriptions under a tenant, ensure that the service principal role has access to the management group.

To discover Azure subscriptions:

1. Click **Projects and Connections.**
2. Click **Cloud Accounts.**

The list of all onboarded cloud accounts appears.

3. Click **Discover accounts > Azure.**

The **Configure Azure Service Principals or Root Management Group(s)** window appears.


4. Discover a single or multiple Azure subscriptions.






- a. In the **Subscription type** toggle, select one of the following:
  - **Single** for onboarding a single subscription.
  - **Multiple** for onboarding multiple subscriptions in a tenant.
- b. Provide the following details of the service principal or root management group:
  - **Client ID** – Application ID of your subscription.
  - **Secret Key** – Value of the secret key for authentication of the service principal or root management group.
  - **Tenant ID** – Directory (tenant) ID of the service principal or root management group.
  - **Subscription ID** – Subscription ID of the service principal or root management group.

**Note:** The subscription ID is optional for **Multiple** subscription type.

- c. (Optional) Click  to add more subscriptions.
- d. Click **Discover**.

For multiple subscription type, Tenable Cloud Security discovers all subscriptions under the tenant with the status as **Discovered**.

**Note:** For multiple subscription type, Tenable Cloud Security schedules discovery every 24 hours and automatically discovers any new subscriptions in the tenant. All subscriptions discovered in the last 7 days show the  label until they are configured or ignored.

## Configure Azure Subscriptions

Configure the discovered subscriptions before you can run a cloud scan to assess the resources in the subscription for misconfigurations. To configure an Azure subscription, provide the client ID, secret key, tenant ID and assign the subscription to a project.

To configure Azure subscriptions:



1. Click **> Configure** in the row for the subscription you want to configure.

The **Configure Azure Account** window appears.

2. Type the **Client ID**, **Secret Key**, and **Tenant ID** for the subscription, if required.
3. Click **Next**.
4. In the **Assign a Project or Create a New Project** section, do one of the following:

- Select a project from the list of Azure projects.

You can search for a project in the **Search projects** box.

- Click **New Project** to create a new Azure project.
  - a. Type a **Project name** for your new project.
  - b. Select Azure for the provider.
  - c. Click **Create New Project**.

Tenable Cloud Security creates the new project and Tenable Cloud Security automatically selects this project for onboarding the Azure subscription.

5. Click **Save**.

The **Cloud Accounts** page appears and shows the project assigned to the subscription.

6. Repeat these steps for all the discovered subscriptions you want to configure.

What to do next:

- Go to the **Projects** tab and run a cloud scan for the project. For more information, see [Run a Cloud Scan](#).



---

## Discover GCP Accounts

---

Tenable Cloud Security can automatically discover your GCP projects. If you want to onboard all projects in a GCP organization, provide the role details of the organization administrator for Tenable Cloud Security to automatically discover the projects in the organization. After Tenable Cloud Security discovers the GCP projects in that organization, you must configure them by providing the credentials before you can run a cloud scan for detecting misconfigurations in the cloud account. To onboard GCP projects, perform the following tasks:

1. [Discover GCP Accounts](#).
2. [Configure GCP Accounts](#).

To discover GCP cloud accounts:

Before you begin:


- Create a service account for Tenable Cloud Security in Google cloud and then provide read-only access for this service account to your Google cloud project. For more information, see [Create a GCP Service Account](#) and [Activate the GCP Service Account](#).

1. Click **Projects and Connections**.
2. Click **Cloud Accounts**.

The **Cloud Accounts** page appears listing all onboarded cloud accounts.


3. Click **Discover accounts > GCP**.


The **Configure GCP Service Account(s)** window appears.

4. Onboard a single or multiple GCP service accounts.
  - a. In the **Project type** section, select one of the following:
    - **Single** for onboarding a single GCP project.
    - **Multiple** for onboarding all projects in a GCP organization.
  - b. Click **Upload** to upload the service account credential file in the JSON format.
  - c. (Optional) Click  to add more accounts.



d. Click **Discover**.

For multiple accounts, Tenable Cloud Security discovers and shows all projects under the GCP organization account with the status as **Discovered**. The GCP organization account appears with the  icon next to it.

For multiple project type, Tenable Cloud Security schedules discovery every 24 hours and automatically discovers any new project in the GCP organization. All accounts discovered in the last 7 days show the  label until they are configured or ignored.

## Configure discovered GCP accounts

Configure the discovered GCP projects before you can run a cloud scan to assess the resources in the project for misconfigurations. To configure a GCP project, upload the credentials file and assign it to a project in Tenable Cloud Security.

1. In the row of service account you want to configure, click  > **Configure**.

The **Configure GCP Account** window appears.

2. Click **Upload** to upload the service account credential file in the JSON format.
3. Click **Next**.
4. In the **Assign a Project or Create a New Project** section, do one of the following:

- Select a project from the list of GCP projects.

You can search for a project in the **Search projects** box.

- Click **New Project** to create a new GCP project.
  - a. Enter a **Project name** for your new project.
  - b. Select GCP for the provider.
  - c. Click **Create New Project**.

Tenable Cloud Security creates the new project and Tenable Cloud Security automatically selects this project for onboarding the GCP project.

5. Click **Save**.



The **Cloud Accounts** page appears and shows the project assigned to the GCP project.

6. Repeat these steps for all the discovered accounts you want to configure.

What to do next:

- Go to the **Projects** tab and run a cloud scan for the project. For more information, see [Run a Cloud Scan](#).

## Manage Cloud Accounts

---

You can view and manage all cloud accounts on the **Cloud Accounts** tab of the **Projects & Connections** tab.



## View Cloud Accounts

The **Cloud Accounts** tab in the **Projects and Connections** page shows all the cloud accounts onboarded in Tenable Cloud Security. The cloud accounts can be onboarded manually or using autodiscovery.

To view the cloud accounts:

1. [Access Tenable Cloud Security.](#)


The **Dashboard** page appears.

2. Click **Projects and Connections**.

The **Projects** tab appears by default.

3. Click **Cloud Accounts**.

The **Cloud Accounts** page appears with a list of all onboarded cloud accounts. The page shows the following details about cloud accounts:


**Note:** Not all the following columns appear in the table by default. To view columns that do not appear by default, click the  icon and select the required columns.

Column	Description
<b>Name</b>	Name of the cloud account. An icon next to the name shows the cloud provider of the account.
<b>Management Unit</b>	If the account type is <b>Multiple</b> , the management unit is the name of the management account for AWS, management group for Azure, and Google group for GCP.
<b>Status</b>	The cloud account status – <b>Ignored, Needs Configuration, Not Scanned, Scanned, Assessed, Failed</b> , and <b>Suspended</b> . For more information, see <a href="#">Cloud Account Statuses</a> .
<b>Resources</b>	The number of resources in the cloud account.



<b>Findings</b>	The number of vulnerabilities and misconfigurations. Misconfigurations are results from a Misconfiguration Scan. Vulnerabilities are results from <a href="#">Agentless Assessment</a> .
<b>Projects</b>	The project that you assign the cloud account to. You can only assign configured cloud accounts to projects.
<b>Account ID</b>	The cloud account ID.
<b>Tags</b>	The cloud tag or label associated with the resource by the cloud provider.
<b>Created By</b>	Email ID used for creating the cloud account.
<b>Discovered On</b>	Time elapsed after Tenable Cloud Security discovered the account.

4. Do one or more of the following:

- Click **Discover accounts** to discover cloud accounts automatically for your provider. For more information, see the following:
  - [Discover and Onboard AWS Accounts](#)
  - [Discover Azure Accounts](#)
  - [Discover GCP Accounts](#)
- Use the **Search accounts** box to search by cloud account name.
- Click the  **Filters** icon to open the **Filter Cloud Accounts** box. Select the following filters as needed.

Filter	Description
<b>Cloud Account Name</b>	Filters by the cloud account name.
<b>Cloud Providers</b>	Filters by the cloud provider – AWS, Azure, and GCP.
<b>Management Unit</b>	Filters by management unit. The management unit is



	the management account for AWS, management group for Azure, and Google group for GCP.
<b>Projects</b>	Filters by the project.
<b>Cloud Account ID</b>	Filters by the cloud account ID.
<b>Status</b>	Filters by the cloud account status – <b>Discovered, Assessed, Failed, Ignored, Deleted, and Suspended.</b>
<b>Cloud Account Alias</b>	Filters by the cloud account alias.
<b>Discovered On</b>	Search based on when the cloud account was discovered: <ul style="list-style-type: none"><li>• Last 24 hours</li><li>• Last 7 days</li><li>• Last 30 days</li><li>• Last 3 months</li><li>• Last 6 months</li><li>• Last 1 year</li></ul>





---

## Edit the Configuration of a Cloud Account

---

You can edit the credentials of cloud accounts after Tenable Cloud Security discovers or assesses the accounts. You can edit the configuration of a cloud account in any of the following states — **Discovered**, **Assessed**, **Failed**, or **Ignored**.

To edit the configuration of a cloud account:

1. Click **Projects and Connections**.


The **Projects** tab appears.

2. Click **Cloud Accounts**.

The **Cloud Accounts** page lists all onboarded cloud accounts.

3. Click **⋮ > Configure** in the row for the member account you want to configure.

The **Configure Account** window for the selected cloud provider appears.

4. Click the  icon and edit the credentials for your cloud account.

- **AWS** – Edit the **Read-only Role ARN** and **External ID** for the AWS account.
- **Azure** – Provide the following values:
  - **Client ID** – Application ID of your subscription.
  - **Secret Key** – Value of the secret key for authentication of the service principal or root management group.
  - **Tenant ID** – Directory (tenant) ID of the service principal or root management group.
- **GCP** – Click **Upload** to upload the service account credential file in the JSON format.

5. Click  to save the account configuration.

6. Click **Next**.

7. In the **Assign a Project or Create a New Project** section, click **Done**.



**Note:** You cannot edit the project assigned to the cloud account from this window. To add or remove a project assigned to the cloud account, go to the **Projects** tab.



## Ignore a Cloud Account

You can exclude a cloud account from the scan. You can ignore a cloud account in any of the following states — **Discovered**, **Assessed**, **Failed**, or **Ignored**. Tenable Cloud Security dissociates an ignored cloud account from the project and no longer includes the account in any future scans. All findings related to the cloud account are removed from Tenable Cloud Security.

**Note:** Ignoring a cloud account does not remove the account from Tenable Cloud Security, it is only excluded from scanning.

To ignore a cloud account:

1. Click **Projects and Connections**.

The **Projects** tab appears.

2. Click **Cloud Accounts**.

The **Cloud Accounts** page lists all onboarded cloud accounts.

3. Click **⋮ > Ignore** in the row for the cloud account you want to ignore.

A confirmation dialog appears to confirm whether you want to ignore the account.

4. Click **Yes** to confirm.

The cloud account status changes to **Ignored**.

**Note:** You can configure an ignored account. However, the status of an ignored account changes only after scanning. For more information, see [Cloud Account Statuses](#).



## Delete a Cloud Account

Tenable Cloud Security allows you to delete cloud accounts. You can delete a cloud account in any of the following states — **Discovered**, **Assessed**, **Failed**, or **Ignored**. You can onboard the account again as a new connection.

**Note:** Tenable recommends not to delete management or organization accounts.

To delete a cloud account:

1. Click **Projects and Connections**.

The **Projects** tab appears.

2. Click **Cloud Accounts**.

The **Cloud Accounts** page lists all onboarded cloud accounts.

3. Click **:** > **Delete** in the row for the member account you want to delete.

The **Delete Account** window appears asking you to confirm the account deletion.

4. Click **Yes** to confirm.


Tenable Cloud Security removes the account and all findings related to that account.



**Note:** To view deleted cloud accounts, filter cloud accounts by Status as **Deleted**. You can configure and onboard a deleted account again. For more information, see [Edit the Configuration of a Cloud Account](#).

---

## Cloud Account Discovery FAQ

---

Why cloud account discovery and how is it different from the current cloud onboarding flow (via  > Connection) in Tenable Cloud Security?

Cloud Onboarding Flow via  > Connection	Cloud Account Discovery Flow
Primarily for onboarding single accounts, except for an AWS organization. This flow supports only single account onboarding for Azure and GCP.	Supports multiple account discovery for all cloud providers – AWS, Azure, and GCP. Tenable Cloud Security can automatically discover the following: <ul style="list-style-type: none"><li>• All member accounts in an AWS organization.</li><li>• All Azure subscriptions in a management group.</li><li>• All projects in a GCP organization.</li></ul>
Tenable Cloud Security can onboard all accounts in an AWS organization. However, if you create a new account after onboarding, you must onboard that single account manually.	With cloud account discovery, Tenable Cloud Security schedules discovery every 24 hours and automatically discovers any new member accounts in the management account. The new account has the  icon next to it, indicating that it is newly discovered. All newly discovered accounts must be configured by providing the credentials and associated with projects before they can be assessed.

What happens when you provide a single member account in the cloud account discovery flow?

Cloud discovery supports single account onboarding. To onboard a single account, select the account type as **Single** and provide the credentials of the account in the cloud discovery flow. Tenable Cloud Security onboards the account, but no further discovery happens.

What is the cloud account discovery schedule frequency? Can the schedule be configured?



The default schedule for the account discovery process is every 24 hours. The schedule cannot be configured.

### How to delete a cloud account after it is onboarded?

You cannot delete a cloud account from Tenable Cloud Security; but, you can ignore the cloud account that you no longer want to scan. Ignoring an account results in the disassociation of the cloud account from the project and stops any future assessment. Tenable Cloud Security removes all findings related to the cloud account. The account still appears in the user interface with the **Ignored** status, but is not deleted. For more information, see [Ignore a Cloud Account](#).


### What happens to an Ignored account?

Tenable Cloud Security dissociates an ignored cloud account from the project and no longer includes the account in any future scans. All findings related to the cloud account are removed from Tenable Cloud Security.


### How to onboard an Ignored cloud account again and make it available for discovery?

You can onboard an **Ignored** account again with the **Configure** option. Provide the credentials of the ignored cloud account and the associated project. After you save this configuration, the cloud account status changes to **Discovered** and the account is ready for assessment. For more information, see [Edit the Configuration of a Cloud Account](#).

### Will the cloud onboarding flow via > Connection to onboard cloud accounts be removed or replaced?

You can still use the existing cloud onboarding flow via  > **Connection** to onboard cloud accounts. For onboarding multiple accounts automatically, Tenable recommends using the cloud account discovery flow.


### If I have already onboarded a cloud account using the cloud onboarding flow via > Connection, can I set it up for cloud account discovery?

Yes. Any management account that is onboarded via the  > **Connection** flow shows as a member account in the **Cloud accounts** tab. To enable cloud account discovery for that account, onboard the account again (with account type as **Multiple**) using the cloud discovery flow. This enables



automatic discovery of all member accounts.

### After you onboard an AWS organization via the > Connection flow, does Tenable Cloud Security discover any new member cloud accounts added to the organization?

Tenable Cloud Security does not discover any new member accounts created after the organization onboarding via the  > **Connection** flow. Manually onboard those new member accounts.

### Why is my new member cloud account not discovered and shown in the Cloud accounts tab?

Verify if cloud account discovery failed due to any of the following conditions:

- The cloud account credentials used for the discovery have changed. Update the credentials of your cloud account in Tenable Cloud Security.
- The cloud account expired or the cloud service provider deactivated the account. Activate your cloud account to enable cloud account discovery.

### Why is there only the Ignore option and no way to *delete* a cloud account?

With the **Ignore** option, you can exclude an account from any future scan, but you can still view the account in Tenable Cloud Security as long as the account is active in the cloud. Tenable Cloud Security does not provide the option to delete a cloud account because of the potential security risk when an active cloud account is deleted from Tenable Cloud Security unintentionally. Another advantage of the **Ignore** option is that it is much easier to re-onboard an ignored cloud account and make it available for assessment.

### What happens to the cloud accounts that are decommissioned, closed, canceled, shut, or terminated by the cloud service provider?

Any cloud account that is decommissioned, closed, canceled, shut, or terminated by the cloud provider appears with the **Suspended** state in Tenable Cloud Security. All such cloud accounts are deleted by the cloud provider after a certain waiting period or post-closure period, which varies for each cloud service provider. Tenable Cloud Security then automatically removes such deleted cloud accounts and they no longer appear in the user interface.



---

## Cloud Scans

---

To run a cloud scan after onboarding your cloud accounts, you must select and run a scan profile. Tenable Cloud Security provides a default scan profile for each cloud provider. You can also create your custom scan profiles. After creating a scan profile, you can run the following types of cloud scans:

- **Misconfiguration Scan:** Scans for policy violations in IaC repositories and cloud resources. You can view the scan results on the **Findings > [Misconfigurations](#)** page. The Misconfigurations Scan is supported for all cloud providers - AWS, Azure, and GCP.
- **Vulnerability Scan:** Scans for known vulnerabilities (CVEs) in workloads, such as operating systems, images, containers, and software based on plugins. You can view these vulnerabilities on the **Findings > [Vulnerabilities](#)** page in Tenable Cloud Security and the **[Findings](#)** page in Tenable Vulnerability Management. For more information, see [Configure Vulnerability Scan using Agentless Assessment for AWS](#).

To configure and run a cloud scan:

1. [Create a Scan Profile](#).
2. (Optional) [Schedule a Scan](#).
3. [Run a Cloud Scan](#).





## Create a Scan Profile

Scan profiles allow you to group the scan operations of different cloud resources and schedule scans according to your needs. You can create different scan profiles to run scans targeting different resources. For example, you can create a scan profile to run a scan targeting only Vulnerability Scans of EC2 instances.

**Note:** You can create a maximum of 10 scan profiles.

Before you begin:

To run a vulnerability scan using Agentless Assessment, see the following:

- [Configure Vulnerability Scan using Agentless Assessment for AWS](#)
- [Configure Vulnerability Scan using Agentless Assessment for Azure](#)

To create a scan profile:

1. Click **Projects & Connections**.

Tenable Cloud Security lists all the projects in the **Projects** tab.

2. In the row for the project for which you are creating the scan profile, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears with the default scan profile.

**Note:** You can use the default scan profile to perform a scan. Click the default scan profile to view the resources that get scanned. Vulnerability scan with agentless assessment is enabled by default for the default scan profile.

3. Click **New Scan Profile**.

The **Create new scan profile for cloud** window appears.

**Note:** To create a scan profile from an existing scan profile, create a copy of the scan profile and then edit the profile.

4. In the **Scan profile name** box, type a name for the scan profile or retain the default name.



5. In **Step 1 Cloud config assessment options**, retain the default selections or do one of the following:
  - Select the check box next to the option to select all the resources within a category.
  - Click the drop-down arrow  $\vee$  to show all the available resources in the category. Select the check boxes as needed.

**Note:** The count next to the drop-down arrow  $\vee$  shows: Number of resources available / Number of resources selected.

- Select a resource by searching for it in the **Search resources** box.
6. (Optional) In **Step 2**, click the **Enable Vulnerability Scan** toggle to enable vulnerability assessment.

**Note:** The vulnerability scan option is available only for AWS EC2 Instances and Azure Virtual Machines. When you enable vulnerability scan, Tenable Cloud Security starts scanning for vulnerabilities after the misconfiguration scan completes.

7. Click **Preview** to view the resources selected in the cloud scan profile.
8. Click **Create Scan Profile**.

Tenable Cloud Security creates the scan profile and displays it in the **Manage scan profiles** window.

What to do next:

Initiate the scan for the scan profile. For more information, see [Run a Cloud Scan](#).



## Schedule a Scan

You can add a scan schedule to your scan profile and run scans at regular intervals. Tenable Cloud Security starts immediately after the duration since the schedule was submitted. For example, if you set the scan schedule to 6 hours now, Tenable Cloud Security starts the scan exactly after 6 hours from now. Tenable Cloud Security runs scheduled scans with the default scan profile.

**Note:** You can add only one schedule for a scan profile.

To schedule a scan for a scan profile:

1. On the home page, click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. In the row of the scan profile for which you want to schedule a scan, click **⋮ > Schedule scan**.

The **Schedule scan** window appears.

4. In the **Select interval** drop-down box, select the required schedule to run the scan: Every 6 hours, 12 hours, or 24 hours.

5. Click **Schedule Scan**.

Tenable Cloud Security schedules the scan for the selected interval and displays a confirmation message.

**Note:** To delete a scheduled scan, in the row for the project, click **⋮ > Delete scheduled scan**.



---

## Run a Cloud Scan

---

You can [create](#) a scan profile to include the resource types that you want to scan and trigger a scan for that profile.

To start a scan:

1. Click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project for the cloud scan, click **:** and do one of the following:
  - **Run default scan profile** – Select this option to run a scan on the default scan profile. If there are no other scan profiles, Tenable Cloud Security runs a scan on the system default scan profile.

**Note:** Vulnerability scan with agentless assessment is enabled by default for the default scan profile.

- **Manage cloud scan profiles** – Select this option to create a new scan profile or use a scan profile that you created earlier.

The **Manage scan profile** window appears and lists all the scan profiles.

Tenable Cloud Security runs the scan and updates the scan status column of the project on completion of the scan.

**Note:** You can view or edit other scan profiles of a project when the cloud scan is running with one of the scan profiles.

What to do next:

After running a cloud scan, you can view a summary of issues, critical security insights, remediation insights, number of cloud and IaC drifts, failing policies, and impacted resources for your project. For more information, see [View Tenable Cloud Security Dashboards and Reports](#).

---

## Manage Scan Profiles

---



Scan profiles allow you to group the scan operations of different cloud resources and schedule scans according to your needs. For a project, there are two scan profiles – one that is system default scan profile that Tenable Cloud Security creates and other is the default scan profile.

**Note:** For every project, Tenable Cloud Security creates a system default scan profile that includes scanning of common resource types. For example, an AWS project has a scan profile with the name System default AWS cloud scan profile.

To access the **Manage Scan Profiles** page:

1. On the home page, click **Projects & Connections**.

The **Projects** tab appears by default.

2. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.



# View Scan Profiles

You can view the list of scan profiles on the **Manage scan profiles** page.

To view the list of scan profiles:

1. On the home page, click **Projects & Connections**.

The **Projects** tab appears by default.

2. In the row for the project that you want to scan, click **☰ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

**Note:** Tenable Cloud Security displays the number of scan profiles above the scan profiles table on the **Manage scan profiles** window.

The **Manage scan profiles** window displays the following details:

Column name	Description
Scan profile	The name of the scan profile. The default scan profile name includes the <b>Default</b> tag next to the name.
Resource types	The number of resource types for the scan profile.
Schedule interval	The schedule configured for the scan. You can schedule only one scan at a time.
Scan status	The status of the scan. Tenable Cloud Security updates the following statuses for scan profiles: <ul style="list-style-type: none"><li>• <b>In progress</b></li><li>• <b>Successful</b></li><li>• <b>Failed</b></li><li>• <b>Completed with errors</b></li></ul>



## Actions

Click **Run Scan** to initiate the scan for that scan profile.

In this column, click the **⋮** button to display the action options:

- **Edit** – Click this option to edit the scan profile.
- **Duplicate** – Click this option to create a duplicate of the scan profile.
- **Schedule scan** – Click this option to configure a scan schedule for the profile.
- **Use as default scan** – Click this option to set the scan profile as the default.
- **Scan history** – Click this option to view the scan history details.
- **Delete** – Click this option to delete the scan profile. The **Delete** option is not available for the system default scan profile and the default scan profile.



---

## Set a Default Scan Profile

---

Tenable Cloud Security provides a default scan profile for each cloud provider. You can set any scan profile that you created as the default one. All scheduled scans run based on the default scan profile.

To set a default scan profile:

1. On the home page, click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. In the row of the scan profile that you want to set as default, click **⋮ > Use as default scan**.

Tenable Cloud Security sets the scan profile as default and indicates it with the **Default** icon.





---

## Edit a Scan Profile

---

You can edit a scan profile and change its configuration.

To edit a scan profile:

1. On the home page, click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **:** > **Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. Click the scan profile that you want to edit.

The profile details appear.

4. Click **Edit profile**.

The **Edit scan profile for cloud** window appears.

5. Modify the configuration as needed.

6. Click **Save**.

Tenable Cloud Security saves the scan profile with the modified configuration.



---

## Copy a Scan Profile

---

To create a new scan profile based on an existing scan profile, you can create a copy of the scan profile by duplicating it. You can then edit the scan profile, if required.

To duplicate a scan profile:

1. On the home page, click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. In the row of the scan profile that you want to set as default, click **⋮ > Duplicate**.

Tenable Cloud Security creates a copy of the scan profile.

4. (Optional) [Edit the scan profile](#).



---

## Delete a Scan Profile

---

You can delete a scan profile that you no longer need.

**Note:** The **Delete** option is not available for the default system scan profile and for the default scan profile.

To delete a scan profile:

1. Click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **:** > **Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. In the row of the scan profile that you want to delete, click the **:** > **Delete**.

A confirmation message appears.

4. Click **Delete** to delete the scan profile.



## View Scan History

You can view the scan history for both Misconfiguration Scans and Vulnerability Scans. Log details for failed scans or scans with errors give you the reason for the scan failure.

**Note:** The failed scan logs are available only for Vulnerability Scans.

To view the scan history details:

1. On the home page, click **Projects & Connections**.



Tenable Cloud Security displays the list of projects in the **Projects** tab.

2. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

3. In the row of the scan profile for which you want to view the scan history, click **⋮ > Scan history**.

The **Scan history** window appears with the following details:

Column name	Description
<b>Time started</b>	This is the scan start time.
<b>Scan type</b>	This shows the type of scan: <b>Misconfiguration Scan</b> or <b>Vulnerability Scan</b> .
<b>Scan status</b>	The status of the scan. Tenable Cloud Security updates the following statuses for scan profiles: <ul style="list-style-type: none"><li>• <b>In progress</b> Click  to refresh the scan status.</li><li>• <b>Successful</b></li><li>• <b>Successful</b>  – For a scan that completes, but includes errors, you</li></ul>



	<p>can download the log file by clicking the <a href="#">↓</a> button.</p> <ul style="list-style-type: none"><li>• <b>Failed</b> – For a failed scan, you can click the <a href="#">↓</a> button to download the log file.</li></ul>
<b>Scan jobs</b>	The total number of successful scans out of all the scans.
<b>Time elapsed</b>	The time elapsed since the scan started.
<b>Initiator</b>	Shows whether the scan was initiated by the scheduler or the user.

4. For vulnerability scans, click [↓](#) to download scan logs.

## Scan Logs

The scan log is a zip file containing a log file in the JSON format that you can download from the **Scan History** page. The following is an example of a scan log file:

```
{
  "cloud_scan_group_id": "",
  "cloud_scan_id": "",
  "resource_id": "",
  "instance_id": "",
  "role_arn": "",
  "external_id": "",
  "workflow_id": "",
  "last_workflow_state": "SNAPSHOT_CREATION_FAILED",
  "workflow_logs": [
    {
      "state": "SNAPSHOT_CREATION_FAILED",
      "message": "snapshot workflow failed: failed to get latest snapshotID from volumeID: failed to describe snapshots: operation error EC2: DescribeSnapshots, https response error StatusCode: 403, RequestID: 00f4c4cf-1cf7-46c1-8fff-8773ef7bc74c, api error UnauthorizedOperation: You are not authorized to perform this operation.",
      "error": ""
    }
  ]
}
```

## Scan Workflow Status

The following table shows the Agentless Assessment workflow statuses:

Workflow Status	Description
-----------------	-------------



REGION_NOT_SUPPORTED	The cloud region where this asset lives does not support Agentless Assessment scans at the moment.
WORKFLOW_INIT	A workflow is created for Agentless Assessment scan.
WORKFLOW_RESCCHEDULED	A failure occurred during scanning and the system is automatically retrying the scan.
SNAPSHOT_REQUESTED	The system is preparing to perform a scan.
SNAPSHOT_REQUEST_QUEUED	The scan is in queue.
SNAPSHOT_CREATION_INITIATED	The scan is being processed.
SNAPSHOT_CREATION_FAILED	An issue occurred while attempting to read installed packages from the snapshot. See message in logs for details.
SNAPSHOT_CREATION_SUCCESS	The data necessary to generate a package inventory has been collected successfully.
CLUSTER_CREATION_INITIATED	The system is generating an inventory of installed packages.
SCANJOB_SUCCESS	The scan job completed successfully.
SCANJOB_FAILED	The scan job failed.
<ul style="list-style-type: none"><li>• SNAPSHOT_CLEANUP_INITIATED</li><li>• SNAPSHOT_CLEANUP_SUCCESS</li><li>• SNAPSHOT_</li></ul>	The scan job completed successfully and internal metadata generated during the scan is being cleaned up from the system.



CLEANUP\_  
FAILED



# Agentless Assessment

Agentless Assessment allows you to scan and analyze short-lived cloud instances on your cloud environments. You can scan both online and offline systems with Agentless Assessment. Agentless Assessment relies on API data and snapshots and does not depend on data from Tenable or other cloud-vendor agents.

Agentless Assessment supports the following:

- AWS EC2 Instances.
- Azure Virtual Machines.

The following are the key benefits of vulnerability scanning using Agentless Assessment:

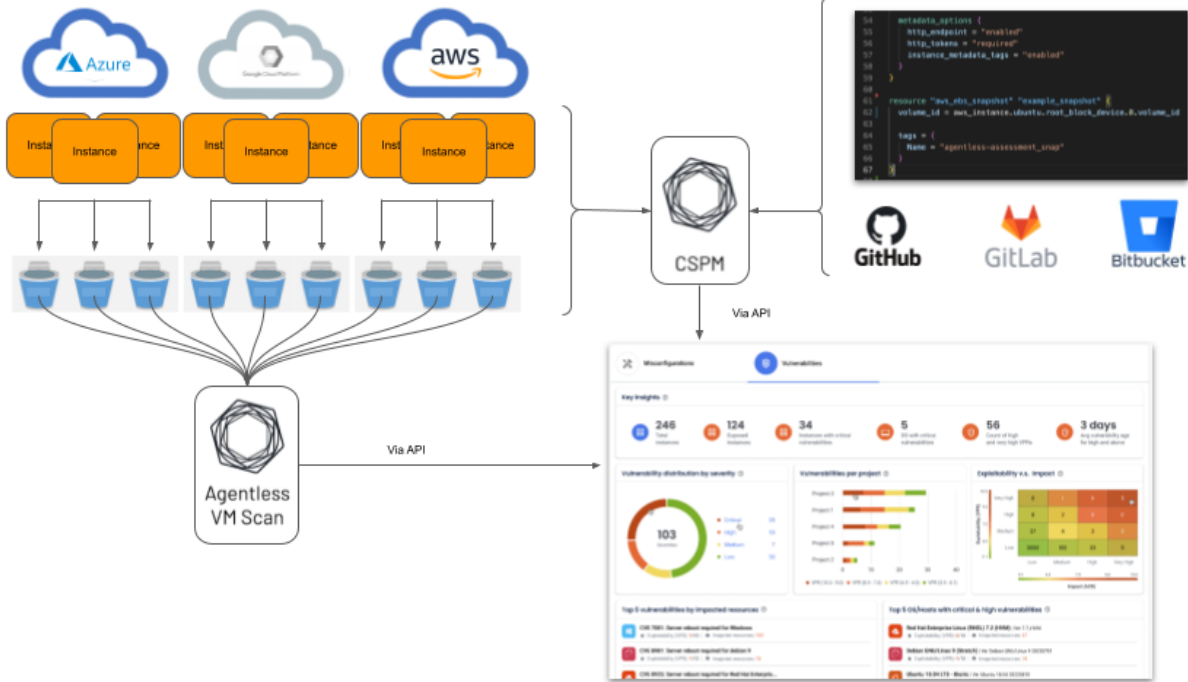
- No need for any software installation on scan targets.
- No impact on system resources.
- No need for any system credentials to perform the scans. Agentless Assessment requires read-only access to your AWS EBS.
- [Live Results](#) feature that always give you the latest Tenable threat updates.

Agentless Assessment is based on Amazon EBS snapshots of your workload EC2 instances. For Azure, Agentless assessment is based on snapshots of your virtual machines. When you trigger a cloud scan in Tenable Cloud Security, along with detecting your cloud resources and mis-configurations, Tenable Cloud Security also detects vulnerabilities in your AWS EC2 workload instances and Azure virtual machines. You can view these vulnerabilities on the [Vulnerabilities](#) page in Tenable Cloud Security and the [Findings](#) page in Tenable Vulnerability Management.

**Note:** Agentless Assessment scans AWS Instance snapshots, and not AWS volume snapshots.

The following image shows a high-level overview of Agentless Assessment:





**Note:** Agentless Assessment supports only root volume scanning and scans software installed at the operating system level.



# Live Results for Agentless Assessment

Agentless Assessment updates with new plugins automatically to allow you to assess your resources for new vulnerabilities. However, if your scan runs on an infrequent schedule, it may not apply new plugins until several days after the plugin update. This gap could leave your resources exposed to unknown vulnerabilities. When a new vulnerability detection is published to the Tenable vulnerability research feed, Tenable Cloud Security live results allows security teams to identify potential vulnerabilities within their existing collected inventory without needing to execute a new scan.

In Agentless Assessment, you can use live results to view scan results for new plugins based on the most recently collected snapshot data, without running a new scan. Live results show you potential new threats and let you determine if you need to launch a scan manually to confirm the findings. Live results are not results from an active scan – they are an assessment based on already-collected data. Live results do not produce results for new plugins that require either active detection, such as an exploit, or previously uncollected data.

Live results appear in the [Vulnerabilities](#) tab in Tenable Vulnerability Management.

**Findings** 🔍

Vulnerabilities | Cloud Misconfigurations | Host Audits | Web Application Findings

Advanced Search by Assets

Group By: None | Asset | Plugin

61 Vulnerabilities | 1 to 50 of 61 | Page 1 of 2

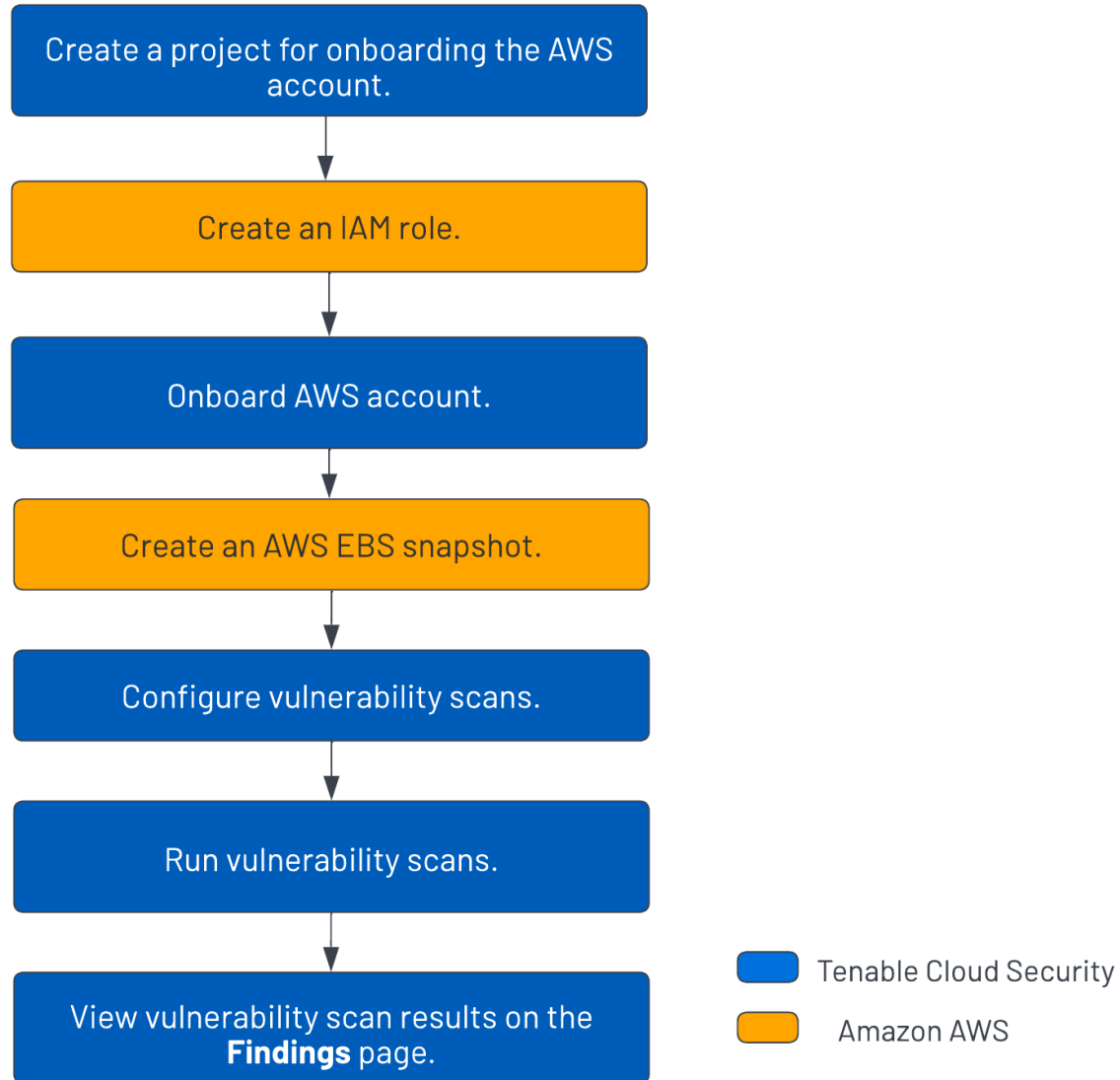
ASSET NAME	SEVERITY	PLUGIN NAME	VPR	CVSS2 BASE SC...	SCAN ORIGIN	LIVE RESULT	LAST SEEN	ACTIONS
nm-test-vm1	Critical	Ubuntu 16.04 ESM : Git vulnerabilities (USN-5810-3)	8.4	10	Agentless Assessment	Yes	02/08/2023	⋮
ubuntu_central...	High	Ubuntu 18.04 LTS / 20.04 LTS / 22.04 LTS / 22.10 : curl v...	6	7.1	Agentless Assessment	Yes	02/28/2023	⋮
ubuntu-australi...	High	Ubuntu 18.04 LTS / 20.04 LTS / 22.04 LTS / 22.10 : NSS v...	5.9	7.8	Agentless Assessment	Yes	02/28/2023	⋮
ubuntu2004-us...	High	Ubuntu 16.04 ESM / 18.04 LTS / 20.04 LTS / 22.04 LTS / ...	6.7	7.2	Agentless Assessment	Yes	03/01/2023	⋮
ubuntu-australi...	High	Ubuntu 16.04 ESM / 18.04 LTS / 20.04 LTS : Heimdal vul...		9.4	Agentless Assessment	Yes	02/09/2023	⋮
nm-test-vm1	High	Ubuntu 16.04 ESM / 18.04 LTS / 20.04 LTS : Heimdal vul...		9.4	Agentless Assessment	Yes	02/09/2023	⋮
ubuntu2004-us...	High	Ubuntu 16.04 ESM / 18.04 LTS / 20.04 LTS : Heimdal vul...		9.4	Agentless Assessment	Yes	02/09/2023	⋮

---

# AWS Agentless Assessment Workflow

---

The following workflow shows the process to set up Agentless Assessment and view the results:



To set up Agentless Assessment for AWS:

1. [Create a project](#) for onboarding the AWS account.
2. [Create an IAM role](#).
3. [Onboard AWS account](#).



4. [Create an AWS EBS snapshot.](#)
5. [Configure vulnerability scans using Agentless Assessment.](#)
6. [Run cloud scans.](#)
7. View cloud scan results on the Tenable Cloud Security [Findings > Vulnerabilities](#) page and the [Findings](#) page on Tenable Vulnerability Management.



---

# Agentless Assessment Requirements for AWS

---

The following requirements must be met for performing Agentless Assessment:

- [IAM Role for Tenable Cloud Security](#)
- [AWS Snapshots](#)
- [Supported Operating Systems for AWS](#)
- [Supported File Systems](#)
- [Supported Regions for AWS](#)

## AWS IAM Role

This is a prerequisite before setting up Agentless Assessment. Agentless Assessments of EC2 instances require an IAM role that grants Tenable Cloud Security permissions to read block data from EBS volumes. The role must provide Tenable Cloud Security the following Elastic Block Store permissions:

- ebs:ListSnapshotBlocks
- ebs:ListChangedBlocks
- ebs:GetSnapshotBlock

Follow the instructions on the [Set Up Read-Only Access to the AWS Account](#) page to configure your IAM role with the appropriate permissions for Agentless Assessments.

For snapshots encrypted with Key Management Service (KMS), you must grant the IAM role with access to the KMS key. For snapshots encrypted with KMS, you must grant the IAM role used by Tenable Cloud Security with access to the KMS key used to encrypt the snapshot. To do this, modify the KMS key's resource policy to include the following permissions:

- kms:Decrypt
- kms:DescribeKey

For more information, see [Required AWS KMS key policy for use with encrypted volumes](#) in AWS documentation.

## AWS Snapshots



---

Agentless Assessment is based on Amazon EBS snapshots of your workload EC2 instances. To configure an Agentless Assessment, you must first create a snapshot. For more information, see [Create AWS Snapshot](#).

## Supported Operating Systems for AWS

- Amazon Linux 2023
- Amazon Linux 2
- CentOS 7
- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES) 11.4 to 15.2
- Ubuntu
- Debian

## Supported File Systems

- XFS
- ext4

## Supported Regions for AWS

You can perform Agentless scans on the following AWS regions:

- us-east-1
- us-west-1
- us-east-2
- us-west-2
- ap-southeast-1
- ap-southeast-2
- ap-northeast-1



- ap-northeast-2
- ap-northeast-3
- ap-south-1
- eu-central-1
- eu-north-1
- ca-central-1
- eu-west-1
- eu-west-2
- eu-west-3
- sa-east-1



## AWS IAM Role for Agentless Assessment

This is a prerequisite before setting up Agentless Assessment. Agentless Assessment of EC2 instances requires an IAM Role that grants the Tenable Cloud Security role access to the AWS-Managed Policy **ReadOnlyAccess** as well as permissions to read block data from Elastic Block Store (EBS) volumes.

The role must provide Tenable Cloud Security the following permissions:

- [ReadOnlyAccess](#) (AWS-Managed Policy)
- ebs:ListSnapshotBlocks
- ebs:ListChangedBlocks
- ebs:GetSnapshotBlock

For the EBS requirement with Agentless Assessment, create an inline policy with the following JSON to provide EBS permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ebs:List*",
        "ebs:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```

For additional instructions on configuring the AWS IAM Role, see [Set Up Read-Only Access to the AWS Account](#).

For snapshots encrypted with Key Management Service (KMS), you must grant the IAM Role used by Tenable Cloud Security with access to the KMS key used to encrypt the snapshot or assign the Tenable Cloud Security role as a Key User in the KMS portal.

To grant the IAM Role access to the KMS Key, create an inline policy for the Tenable Cloud Security IAM Role that includes the following permissions:





- kms:Decrypt
- kms:DescribeKey

The following example shows a custom inline policy that is assigned to the Tenable Cloud Security IAM Role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:[REGION]:[ACCOUNT-ID]:key/[KEY]"
    }
  ]
}
```

**Note:** In the JSON, replace the **Resource:** value with either \* or with a list of the KMS keys used to encrypt volumes or snapshots for each region in the AWS account.

If preferred, you can add the Tenable Cloud Security IAM Role as a Key User instead of creating a custom KMS inline IAM policy. Navigate to the AWS KMS Service, find the KMS key used to encrypt the EBS Volumes and Snapshots, and add the Tenable Cloud Security IAM Role as a Key User.

## Create AWS Snapshot

This is a prerequisite before you set up an Agentless Assessment. Create snapshots for EC2 instances that you want to scan. Create snapshots for EC2 instances that you want to scan because the Agentless Assessment process requires them to read installed package data.

**Note:** Agentless Assessment scans AWS Instance snapshots, and not AWS volume snapshots.

You can create snapshots manually or you can automate the process using AWS Data Lifecycle Manager (DLM). Tenable recommends that you automate this process.

- [Create a snapshot manually](#)
- [Automate snapshot creation with AWS DLM](#)



**Note:** AWS Backup's snapshot automation feature is not currently compatible with Elastic Block Storage (EBS) service's list and describe APIs. Therefore, it is not possible to create automated EBS snapshots that are readable by Agentless Assessment using AWS Backup.

Tenable recommends that you follow these best practices for snapshots:

- Take snapshots frequently.
- Do not share snapshots between accounts.
- Ensure snapshots are not visible publicly.
- Ensure snapshots have appropriate life-cycle management for creation, archiving, and deletion.
- Encrypt all snapshots.



---

## Create AWS Snapshot Manually

---

To create a snapshot manually:

1. Log in to the AWS console.
2. In the left navigation bar, select **EC2 Service** dashboard.

The **EC2 Service Dashboard** page appears.

3. In the left navigation bar, click **Elastic Block Store > Snapshots**.



The **Create Snapshot** page appears.

4. In the **Snapshot Settings** section, under **Resource Type**, select **Instance**.
5. In the **Instance ID** box, select the EC2 Instance ID for which you want to create a snapshot.
6. Click **Create snapshot**.

AWS creates the snapshot, which takes around 10 minutes to complete.





---

## Automate Snapshot Creation with AWS Data Lifecycle Manager (DLM)

---

You can use the Data Lifecycle Manager (DLM) service to automate the creation of snapshots from EC2 instances according to a schedule. For more information, see [Amazon Data Lifecycle Manager](#).

To get you started, an example is provided to deploy DLM automatically on [Tenable GitHub](#).



---

# Configure Vulnerability Scan using Agentless Assessment for AWS

---

Workload vulnerability scans are triggered as part of the cloud scan process in Tenable Cloud Security. Tenable Cloud Security supports agentless workload scanning for AWS EC2 instances.

Before you Begin:

- Onboard cloud accounts in Tenable Cloud Security. For more information about onboarding your AWS accounts, see [Onboard AWS Accounts](#).
- [Create an IAM role](#) that provides Tenable Cloud Security the following permissions:

- Elastic Block Store:
  - ebs:ListSnapshotBlocks
  - ebs:ListChangedBlocks
  - ebs:GetSnapshotBlock
- Key Management Service (KMS):

For snapshots encrypted with KMS, you must grant the IAM role used by Tenable Cloud Security with access to the KMS key used to encrypt the snapshot. To do this, modify the KMS key's resource policy to include the following permissions:

- kms:Decrypt
  - kms:DescribeKey
- [Create snapshots](#) in AWS console.

To set up Agentless Assessment:

1. In Tenable Cloud Security, initiate a cloud scan:
  - a. On the home page, click **Projects & Connections**.  
Tenable Cloud Security displays the list of projects in the **Projects** tab.
  - b. In the row for the project that you want to scan, click **⋮ > Manage cloud scan profiles**.



The **Manage scan profiles** window appears.

- c. Click **New Scan Profile**.

The **Create new scan profile for cloud** window appears.

**Note:** You can also use the default scan profile. Vulnerability scan with agentless assessment is enabled by default for the default scan profile.

- d. In the **Scan profile name** box, type a name for the scan profile or retain the default name.
- e. In **Step 1 Cloud config assessment options**, retain the default selections or do one of the following:
  - Select the check box next to the option to select all the options within a category.
  - Click the drop-down arrow  $\vee$  to show all the available options in the category.

Select the check boxes as needed.

**Note:** The count next to the drop-down arrow  $\vee$  shows: Number of options available / Number of options selected.

- f. In **Step 2**, click the **Enable Vulnerability Scan (optional)** toggle to enable vulnerability scan.

**Note:** Tenable Cloud Security scans EC2 instances for vulnerabilities after it completes the Misconfiguration Scan. The EC2 resources are available under the **Compute** category.

- g. (Optional) Click **Preview** to view all the selected assessment options.
- h. Click **Create Scan Profile**.

Tenable Cloud Security creates the scan profile and the newly created scan profile appears on the **Configure cloud scan** window.

- i. In the row of the scan profile that you created for a vulnerability scan, click **Run Scan**.

Tenable Cloud Security runs the vulnerability scan and you can view the vulnerability scan results on the Tenable Cloud Security [Vulnerabilities](#) page and also on the Tenable Vulnerability Management [Findings](#) page.

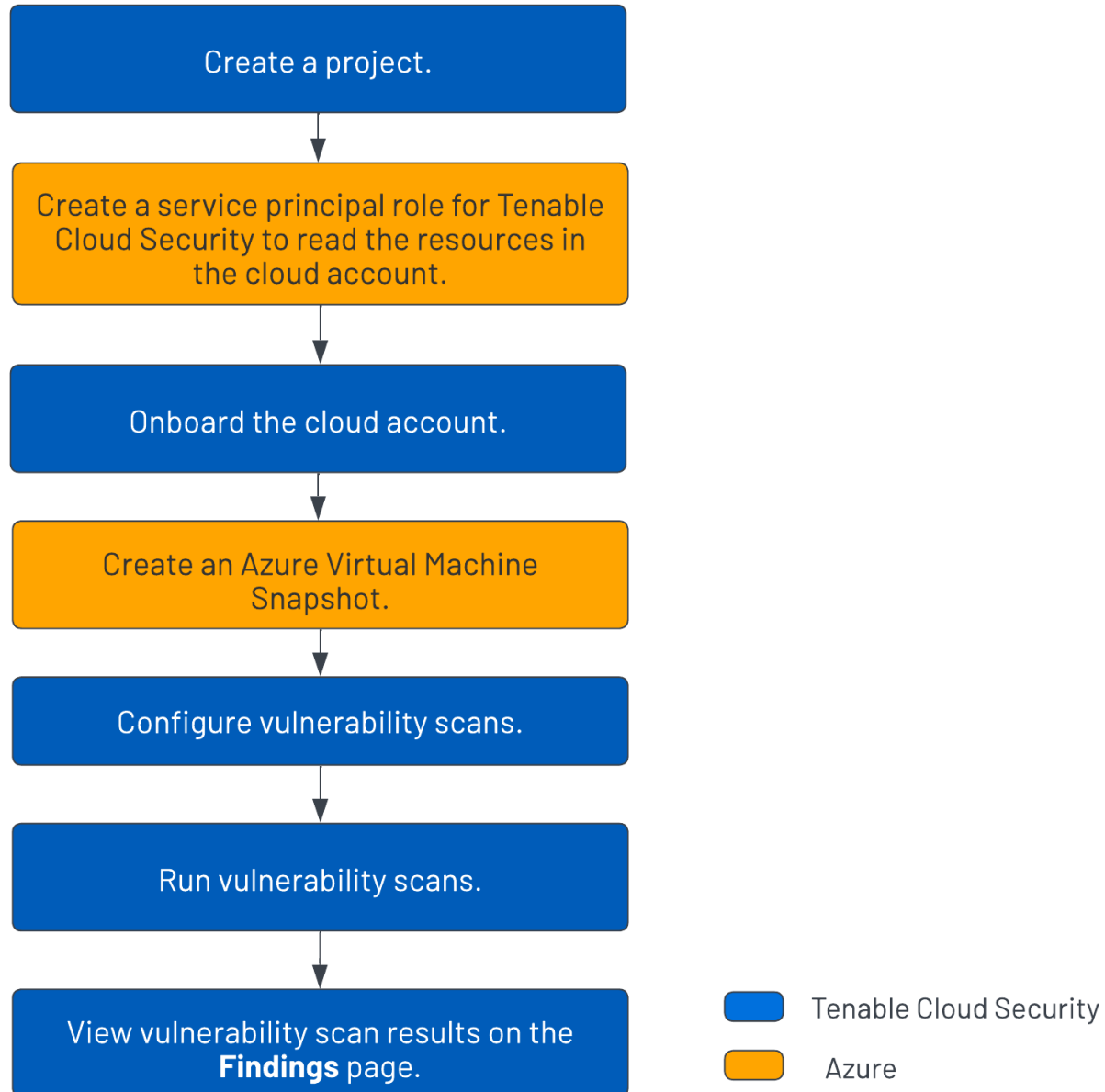


---

# Azure Agentless Assessment Workflow

---

The following workflow shows the process to set up Agentless Assessment and view the results:



To set up Agentless Assessment for Azure virtual machines:

1. [Create a project](#) for onboarding the cloud account.
2. [Create a service principal role for Tenable Cloud Security.](#)



3. [Onboard the Azure cloud account.](#)
4. [Create an Azure Virtual Machine snapshot.](#)
5. [Configure vulnerability scans using Agentless Assessment.](#)
6. [Run cloud scan.](#)
7. View cloud scan results on the Tenable Cloud Security [Findings > Vulnerabilities](#) page and the [Findings](#) page on Tenable Vulnerability Management.



---

# Agentless Assessment Requirements for Azure

---

The following requirements must be met for performing Agentless Assessment:

- [Azure Role](#)
- [Azure Snapshots](#)
- [Supported Operating Systems for Azure](#)
- [Supported File Systems](#)
- [Supported Regions for Azure](#)

## Azure Service Principal Role

This is a prerequisite before setting up Agentless Assessment. Agentless Assessments requires a role that grants Tenable Cloud Security permissions to read data from Azure virtual machine snapshots.

The following permissions are required for a vulnerability scan of Azure VMs:

- Reader
- Disk Snapshot Contributor

Follow the instructions on the [Create an Azure Service Principal Role](#) page to create a role for Tenable Cloud Security.

## Azure Snapshots

Agentless assessment for Azure is based on snapshots of your virtual machines. To configure an Agentless Assessment, you must first create a snapshot. For more information, see [Create an Azure Virtual Machine Snapshot](#).

## Supported Operating Systems for Azure

- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES) 11.4 to 15.2



- 
- Ubuntu
  - Debian

## Supported File Systems

- XFS
- ext4

## Supported Regions for Azure

- australiacentral
- australiacentral2
- australiaeast
- australiasoutheast
- brazilsouth
- brazilsoutheast
- canadacentral
- canadaeast
- centralindia
- centralus
- eastus
- eastus2
- francecentral
- francesouth
- germanynorth
- germanywestcentral
- japaneast



- 
- northcentralus
  - northeurope
  - norwayeast
  - norwaywest
  - southcentralus
  - southeastasia
  - southindia
  - swedencentral
  - sweden south
  - uksouth
  - ukwest
  - westcentralus
  - westeurope
  - westus
  - westus2
  - westus3



---

## Azure Service Principal Role for Agentless Assessment

---

This is a prerequisite before setting up [Agentless Assessment](#). Agentless Assessments requires a role that grants Tenable Cloud Security permissions to read data from Azure virtual machine snapshots.

The following permissions are required for a vulnerability scan of Azure VMs:

- Reader
- Disk Snapshot Contributor

Follow the instructions on the [Create an Azure Service Principal Role](#) page to create a role for Tenable Cloud Security.

## Create an Azure Virtual Machine Snapshot

---

Tenable Cloud Security Agentless Assessment performs scans on Azure Virtual Machines through the assessment of virtual hard disk snapshots. Snapshots can be created manually or automatically through the use of Azure Backup Vault. Tenable recommends that you automate this process.

- [Create a snapshot manually](#)
- [Automate Azure Virtual Machine Snapshot Creation](#)



---

# Create Azure Virtual Machine Snapshot Manually

---

To create a snapshot manually:

1. In the [Azure portal](#), select **Create a resource**.
2. Search for and select **Snapshot**.

The **Snapshot** window appears.

3. Click **Create**.

The **Create snapshot** window appears.

4. In the **Basics** tab, do the following:
  - a. For **Resource group**, select an existing resource group or enter the name of a new one.
  - b. In the **Instance details** section, provide the following information:
    - **Name** – Name of the snapshot.
    - **Region** – The Azure region into which the resource should be deployed. For the list of supported regions, see [Agentless Assessment Requirements for Azure](#).
    - **Snapshot type** – The type of snapshot determines its pricing and functionality.
      - Full: Make a complete read-only copy of the selected disk.
      - Incremental: Save on storage costs by making a partial copy of the disk based on the difference between the last snapshot.
    - **Source subscription** – The subscription that contains the managed disk to be backed up.
    - **Source disk** – The disk to use as the source of this new snapshot.
    - **Storage type** – Select **Standard HDD**, unless you require zone-redundant storage or high-performance storage (Premium HDD) for your snapshot.

5. Click the **Encryption** tab and ensure that Key management is set to **Platform-managed key**.

Platform-managed keys (PMKs) are key encryption keys that are generated, stored, and managed entirely by Azure.



6. Click the **Networking** tab and ensure that **Network access** is set to **Enable public access from all networks**.
7. Click the **Advanced** tab and ensure that the **Enable data access authentication mode** is disabled.
8. (Optional) Configure the **Tags** tab by providing name/value pairs for your resources.
9. Click **Review + create**.

Azure validates the snapshot and shows a summary of the snapshot.

10. Click **Create** to create the snapshot.





---

# Automate Azure Virtual Machine Snapshot Creation

---

To get you started, an automated solution is provided on [Tenable GitHub](#).



---

# Configure Vulnerability Scan using Agentless Assessment for Azure

---

Workload vulnerability scans are triggered as part of the cloud scan process in Tenable Cloud Security. Tenable Cloud Security supports agentless workload scanning for Azure Virtual Machines.

Before you Begin:

- Onboard cloud accounts in Tenable Cloud Security. For more information about onboarding your cloud accounts, see [Onboard an Azure Account](#).
- [Create an Azure service principal role](#) that provides Tenable Cloud Security the following permissions:
  - Reader
  - Disk Snapshot Contributor
- [Create an Azure Virtual Machine Snapshot](#).

To set up Agentless Assessment:

1. In Tenable Cloud Security, initiate a cloud scan:

a. On the home page, click **Projects & Connections**.

Tenable Cloud Security displays the list of projects in the **Projects** tab.

b. In the row for the project that you want to scan, click **> Manage cloud scan profiles**.

The **Manage scan profiles** window appears.

c. Click **New Scan Profile**.

The **Create new scan profile for cloud** window appears.

**Note:** You can also use the default scan profile. Vulnerability scan with agentless assessment is enabled by default for the default scan profile.

d. In the **Scan profile name** box, type a name for the scan profile or retain the default name.



e. In **Step 1 Cloud config assessment options**, retain the default selections or do one of the following:

- Select the check box next to the option to select all the options within a category.
- Click the drop-down arrow  $\vee$  to show all the available options in the category.

Select the check boxes as needed.

**Note:** The count next to the drop-down arrow  $\vee$  shows: Number of options available / Number of options selected.

f. In **Step 2**, click the **Enable Vulnerability Scan (optional)** toggle to enable vulnerability scan.

**Note:** Tenable Cloud Security scans Azure Virtual Machines for vulnerabilities after it completes the Misconfiguration Scan. These resources are available under the **Compute** category.

g. (Optional) Click **Preview** to view all the selected assessment options.

h. Click **Create Scan Profile**.

Tenable Cloud Security creates the scan profile and the newly created scan profile appears on the **Configure cloud scan** window.

i. In the row of the scan profile that you created for a vulnerability scan, click **Run Scan**.

Tenable Cloud Security runs the vulnerability scan and you can view the vulnerability scan results on the Tenable Cloud Security [Vulnerabilities](#) page and also on the Tenable Vulnerability Management [Findings](#) page.



---

# Agentless Assessment FAQ

---

The following are some of the FAQs about Agentless Assessment:

## What are the supported operating systems for EC2 workload VM?

- Amazon Linux 2023
- Amazon Linux 2
- CentOS 7
- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES) 11.4 to 15.2
- Ubuntu
- Debian

## What are the supported operating systems for Azure virtual machines?

- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES) 11.4 to 15.2
- Ubuntu
- Debian

## Why are my scans not updating?

Make sure that a newly created snapshot is scanned. For more information, see [Create AWS Snapshot](#) and [Create an Azure Virtual Machine Snapshot](#).

## Do cloud instances need to be running for Tenable Cloud Security Agentless Assessment scans to work?

Cloud instances do not need to be running at the time of a Tenable Cloud Security Agentless Assessment cloud scan, but you must have at least one snapshot of an instance's primary volume for Agentless Assessment to see data.



## What if my volumes are encrypted?

For AWS, you can use encrypted EBS snapshots with Agentless Assessment. In AWS, you have access to the default encryption keys unless you have an IAM policy that explicitly denies it. You can use your own KMS Key or the default EBS Key. For example, if you are using a KMS Customer Managed Key (CMK), add the read-only role as a “Key User” under the Key Policy, or add the necessary KMS permissions to the role for which the key would be used. If you are using the default EBS key to handle encryption, Agentless Assessment uses that key for decryption prior to gathering the EBS data.

For Azure, the virtual disk snapshots must be encrypted with the Platform-managed key.



---

# Troubleshooting Issues with Agentless Assessment

---

The following are some of the setup issues while configuring Agentless Assessment and their resolutions:

## No Snapshot is Created

### Solution:

Agentless scanning requires a snapshot for AWS instances or Azure virtual machines. You can create snapshots manually or you can automate the process. Tenable recommends that you automate the process. For more information, see [Create AWS Snapshot](#) and [Create an Azure Virtual Machine Snapshot](#).

## Permission Errors

### Solution:

- **AWS:** Agentless Assessment uses the same IAM role that you create when you [onboard](#) the Tenable Cloud Security connector. This role must have access to the `ebs:GetSnapshotBlock` and `ebs:ListSnapshotBlocks` APIs in its AWS IAM policy. For more information, see [Create IAM Role](#).
- **Azure:** Agentless Assessments requires a role that grants Tenable Cloud Security permissions to read data from Azure virtual machine snapshots with the following permissions:
  - Reader
  - Disk Snapshot Contributor

For more information, see [Create an Azure Service Principal Role](#).

## IAM Permission Errors due to KMS

### Solution:

For snapshots encrypted with AWS KMS keys, the IAM role used by Tenable Cloud Security must be granted access to the KMS key used to encrypt the snapshot. To do this, modify the KMS key's resource policy to include the following permissions:

- `kms:Decrypt`
- `kms:DescribeKey`



For more information about the IAM requirements for encrypted volumes, see the [AWS documentation](#).



# Connect Repositories

**Required Tenable Cloud Security User Role:** Administrator.

Before Tenable Cloud Security starts monitoring the code in your repositories, you must connect your repositories to Tenable Cloud Security Console. You can connect using one of the following methods:

- [Connect to a Repository Using Version Control](#)

Connect your repository using Azure DevOps, AWS CodeCommit, Bitbucket, GitHub, or GitLab.

**Note:** To set up an SCM integration, Tenable Cloud Security requires an admin-level account. This allows Tenable Cloud Security to grant itself as an authorized OAuth application to discover and scan all Infrastructure as Code (IaC) projects across all repositories within your SCM account. The admin-level privileges also allow Tenable Cloud Security to create a webhook for auto-remediation and inline reviews to automate pull requests with remediation details.

- [Connect to a Repository Using the CLI](#)

Download and install command-line interface (CLI) on your system to scan your IaC repositories.

**Note:** Make sure that the repository names do not have any special characters.

## Connect to a Repository Using Version Control

Tenable recommends connecting a repository using version control when you want to:

- Connect to your version control provider, for example, GitHub.
- Scan your infrastructure as code (IaC).

To connect a repository using version control:

1. In the left navigation bar, click  > **Connection** > **Repository**.

The **Connect to repository** page appears.

2. In the **Choose a workflow to discover repo(s)** section, click **Version control (recommended)**.





3. Click **Continue**.
4. In the **Connect to a version control provider** section, select one of the following version control system providers:
  - [Bitbucket](#)
  - [GitHub](#)
  - [GitLab](#)
  - [Azure DevOps](#)
  - [AWS Code Commit](#)

A new window appears.

5. Follow the on-screen instructions to grant Tenable Cloud Security Console access to your repository.
6. Click **Continue**.
7. In the **Choose onboarding repositories** section, connect to your repositories in one of the following ways:

To connect to all your repositories automatically:

- a. Select the **Onboard all repositories automatically** check box.
- b. Click **Onboard All**.

The **Projects & Connections** page appears. Tenable Cloud Security creates a separate project for each repository type. For example, the *Default Gitlab Repositories* contains all GitHub repositories.

Tenable Cloud Security automatically starts the scan for the onboarded repositories.

- c. Click  to refresh and view the status of the scan for each project.

To connect your repositories manually:



- a. In the list of repositories, select the required repositories.

**Tip:** You can search for repositories by their name.

- b. If you want to scan only a particular branch or folder of a repository, click the > button next to the repository name.

The **Select branch** drop-down box appears.

- c. Select the branch you want to scan.
- d. From the **Select Folder** check box, select the folders to scan.

Choose onboarding repositories:

Onboard all repositories automatically

Search repos Add Custom / Public Repository

1-5 of 528 < >


<input type="checkbox"/> Repos	Source Type	
<input checked="" type="checkbox"/> [redacted].git	Bitbucket Cloud	<input checked="" type="checkbox"/> Folder: integration_test
Select branch <input type="text" value="master"/>		
Select folder		
<input type="checkbox"/>	cmd	
<input checked="" type="checkbox"/>	integration_test	
<input type="checkbox"/>	internal	
<input type="checkbox"/>	resource	
<input type="checkbox"/>	scripts	
<input type="checkbox"/>	testdata	

**Note:** If you do not select the branch for a repository, Tenable Cloud Security uses the default branch with the root folder.

**Important:** If you have selected plan-based setup, ensure that there are Terraform (.tf) files in the selected branch; otherwise, the IaC scan fails.

To add a custom or public repository:



- a. Click **Add Custom / Public Repository** .
- b. Type the name and folder path of the repository you want to add.
- c. Click **Add**.

**Note:** The file and folder hierarchy structure of the repository depends on the version control provider. For example, Bitbucket and GitLab list the folders first and then the files, whereas GitHub lists the files and folders alphabetically.

Tenable Cloud Security connects to the repository.

8. (Optional) To configure advanced settings for a repository:

- a. Select a repository.
- b. In the **Advanced settings** field, click  for the selected repository.

A window appears.

c. In the **laC engine type** drop-down box, select one of the following engine types:

- Terraform
- CloudFormation
- Kubernetes YAML
- Helm Chart
- Kustomize YAML
- Terragrunt
- Azure Resource Manager

For more information about laC engine types, see [laC Engine Types](#).

- d. In the **Select version** drop-down box, select the engine version.
- e. (Optional) Click the **Enable Webhook** toggle to allow Tenable Cloud Security to monitor your repository continuously for any changes.




- f. For Terraform and TerragruntIaC types, in the **Auto-remediate settings** drop-down box, select an option to indicate how to handle found violations:
  - **Auto-remediate**: Tenable Cloud Security automatically fixes any violations. For more information, see [Set up Auto-Remediation](#).
  - **Inline reviews**: Tenable Cloud Security automatically creates an issue for the violation. For more information, see [Set up Inline Reviews](#).
  - **None**: Tenable Cloud Security takes no action.
- g. To add custom parameters to the repository configuration for Terraform and TerragruntIaC types:
  - i. (Optional) For plan-based setup, click the **Plan based setup** toggle.
  - ii. In the left drop-down box, select a parameter.
  - iii. In the text box, type the value for the selected parameter.

For more information, see [Repository Configuration Parameters](#).

h. Click **Save**.

9. Click **Continue**.

10. In the **Choose projects to add the repository to** section, do one of the following:

- Add a new project:
  - a. Click **Add a project**  .
  - b. Enter the name of a project.
  - c. Click **Add**.
- Select a project from the list of existing projects.

**Tip:** You can search for projects by their name.

11. Click **Connect**.




Tenable Cloud Security adds the newly connected repository to the **Projects & Connections** page.

## Connect to a Repository Using the CLI

Tenable recommends connecting a repository using the CLI when you want to:

- Integrate a command-line interface with a continuous integration / continuous deployment (CI/CD) tool, for example, Jenkins.
- Run a command-line interface locally to discover resources and violations in an infrastructure as code (IaC) repository.

To connect a repository using CLI:

1. In the left navigation bar, click  > **Connection** > **Repository**.
2. In the **Choose a workflow to discover repo(s)** section, click **CLI driven**.
3. Click **Continue**.
4. Click **Continue**.
5. In the **CLI usage instructions** section, follow the on-screen instructions.

For more information, see [Install or Upgrade the CLI](#).

6. Click **Done**.

Tenable Cloud Security adds the newly connected repository to the **Projects & Connections** page.

What to do next:

In the row corresponding to the project to which you have added the repository, click **⋮** > **IaC scan** to run an IaC scan for the repository.



# Repository Configuration Parameters

In Tenable Cloud Security, you can configure a list of parameters for your IaC repository scan. You can provide IaC parameters to improve violation detection and IaC to cloud resource mapping.

**Note:** If the specified variables are invalid, the IaC scan might fail.

Some parameters are only available for plan-based setup, whereas general configuration parameters are available with and without plan-based setup.

## General Configuration Parameters (with and without Plan-Based Setup)

Name	Description
BRANCH_NAME	The name of a branch in the source code repository which you want to scan. If you do not specify this parameter, Tenable Cloud Security scans the default branch.

## On-premises Code Scanner Configuration Parameters

Name	Description
REPO_TYPE	Depending on the repository type to onboard, Tenable Cloud Security automatically sets this parameter to <a href="#">github</a> , <a href="#">bitbucket</a> , or <a href="#">gitlab</a> .
ON_PREM	Tenable Cloud Security automatically sets this parameter to True when scanning an on-premises repository.

## Parameters for Terraform Private Modules

**Note:** On-premises repositories do not support Terraform private module parameters.

Name	Description
TFC_HOST_NAME	The hostname of Terraform Cloud. Use <code>app.terraform.io</code> as the host-name value.



TFC_USER_TOKEN	The API token to authenticate with the Terraform Cloud.  For more information, see <a href="#">authentication</a> in Terraform Cloud documentation.
----------------	---

## Plan-Based Parameters

Tenable Cloud Security provides you with the plan-based setup for specifying run-time parameters during an IaC scan.

To view and manage repository configuration parameters:

1. On the **Repositories** page, click the  button.

The **Advanced Settings** window appears.

2. Click the **Plan based setup** toggle.

All plan-based repository configuration parameter options appear.

The following tables explain the repository configuration parameters available in the plan-based setup:

### AWS Configuration Parameters

Name	Description
TFSTATE_URL	The URL of the AWS S3 bucket that contains the state file.
TFSTATE_ASSUME_ROLE_ARN	The AWS role that has read-only access to the S3 bucket containing the state file.
TFSTATE_EXTERNAL_ID	(Optional) The external ID of the AWS role that has read-only access to the S3 bucket containing the state file.
BUCKET_REGION	The AWS region of the S3 bucket containing the state file.

### Microsoft Azure Configuration Parameters

Name	Description
AZURE_STORAGE_ACCOUNT	The storage account on Azure.



AZURE_STORAGE_ACCESS_KEY	The access key for the storage account on Azure.
TFSTATE_CONTAINER_NAME	The name of the Azure container that contains the state file.
TFSTATE_FILE_NAME	The name of the state file located on Azure.

## Terraform Plan File Parameters

Name	Description
CONSOLE_FILE	The repository path to the console file generated by the Terraform plan file. This parameter is only applicable for Terraform v11. If you do not specify this parameter, Tenable Cloud Security scans the repository path.
PLAN_FILE	The repository path to the Terraform plan file. This parameter is only applicable for Terraform v11 and v12. This is a binary file and must be from the Linux operating plan output. If you do not specify this parameter, Tenable Cloud Security scans the repository path.

## Terraform Workspace Parameters

Name	Description
TERRAFORM_WORKSPACE	(Optional) The name of the Terraform workspace. While running the Terraform plan, Tenable Cloud Security replaces any Terraform code that uses the Terraform workspace value with this value. If you do not specify this parameter, Tenable Cloud Security scans the default workspace.
TF_ASSUME_ROLE_ARN	The name of the role that has read-only access to run the Terraform plan. The role is assumed/used before calling the Terraform plan to ensure that the Terraform plan avoids any access denial errors.

## Terraform Module Parameters

Name	Description
MODULE	The name of the module to scan in the code file. Tenable Cloud Security only scans the specified module.
SUBMODULE	The name of the submodule to scan if using a public module. Specify the





	SUBMODULE_HTTP parameter along with this parameter.
SUBMODULE_HTTP	The URL of the submodule if using a public module.

## Custom Parameters

Name	Description
var-file	If the variable file is used within the Terraform plan, the relative path to the file.
<custom_variable>	Specify a custom parameter and provide a value (<value1>) for it. The custom parameter is processed using the following syntax: -var key1-1=value1



## IaC Engine Types

The following table provides the applicable values for each IaC Engine/ repository type. Based on the selected repository type, configure the applicable settings.

Repo Type	IaC Engine Type	Select Version	Auto Remediate Settings	Plan Based Setup
Terraform	Terraform	Applicable to change/-configure	Applicable to change/-configure	Applicable to change/-configure
Terragrunt	Terragrunt	Applicable to change/-configure	Not applicable	Applicable to change/-configure
CloudFormation	CloudFormation	Not applicable	Not applicable	Not applicable
Kubernetes YAML	Kubernetes YAML	Not applicable	Not applicable	Not applicable
Kustomize YAML	Kustomize YAML	Not applicable	Not applicable	Not applicable
Helm Chart	Helm Chart	Not applicable	Not applicable	Not applicable
Azure Resource Manager (ARM)	Azure Resource Manager (ARM)	Not applicable	Not applicable	Not applicable




# Integrate with GitHub

Before you begin:


Obtain access to a GitHub source code provider account to connect to the repositories.

To integrate Tenable Cloud Security with GitHub:

1. Navigate to the [Connect to repository](#) page.
2. In the **Connect to a version control provider** section, click **GitHub**.


 **Connect to repository**








---

Choose a workflow to discover repo(s): **Version control** 

|

- **Connect to a version control provider**

 For Auto-Remediation features to work, integrate using SCM credentials with admin equivalent privileges on target repositories.

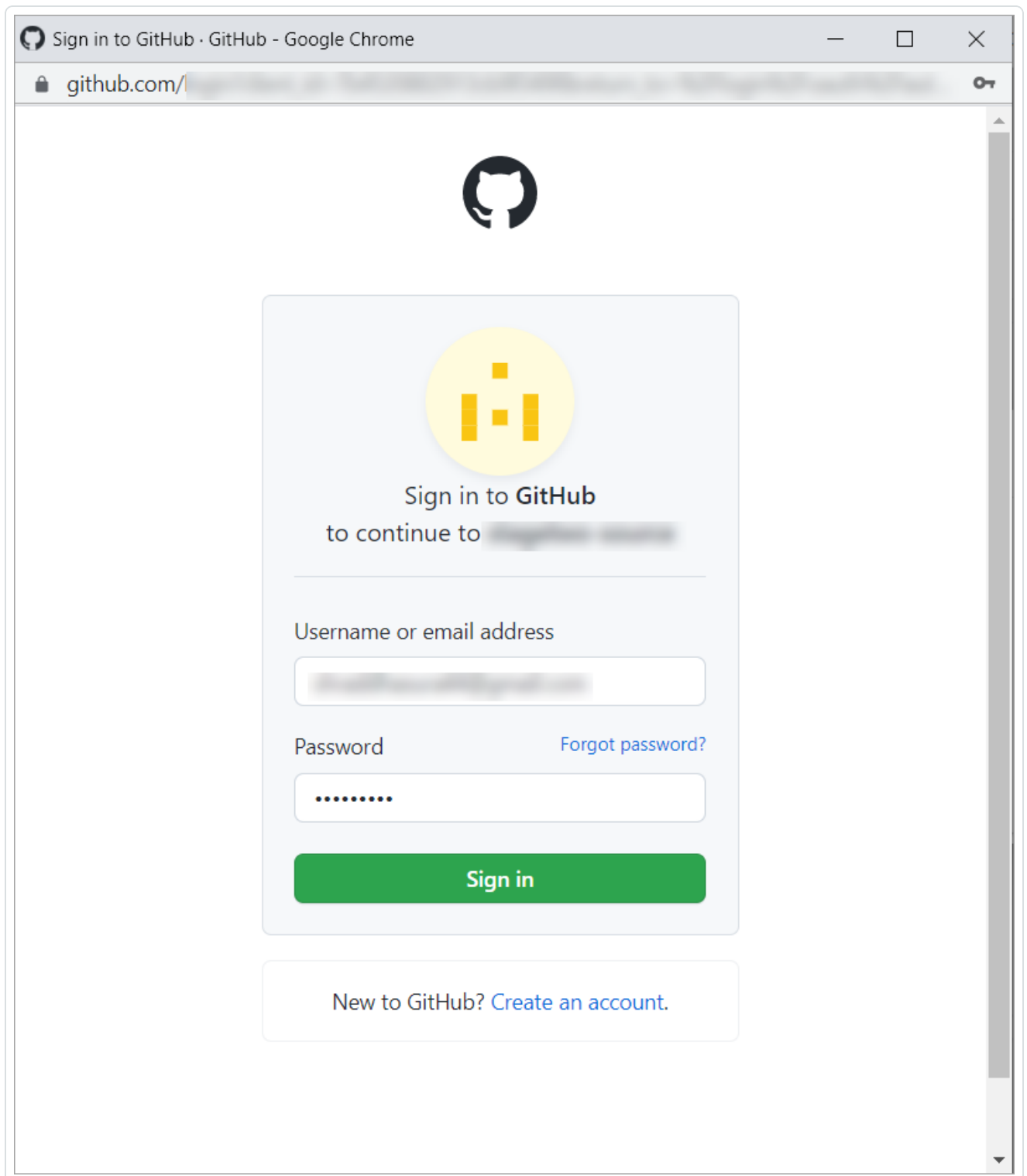
 <b>BITBUCKET</b> Not setup	 <b>GITHUB</b>  Reset	 <b>GITLAB</b> Not setup	 <b>AZURE DEVOPS</b> Not setup
 <b>AWS CODE COMMIT</b> Not setup	 <b>ON-PREMISE CODE SCANNER</b>		

**CONTINUE**      PREVIOUS

Tenable Cloud Security Console redirects you to the log in page of the GitHub source code provider.





3. In the **Sign in to GitHub** window, type your credentials.





4. Click **Sign in**.

Tenable Cloud Security connects to the source code provider. Once the connection succeeds a  **Connected** icon appear next to the source code provider.

5. (Optional) To disconnect the source code provider, click  .

A dialog appears asking you to confirm whether you want to disconnect. Click **Yes** to disconnect.



# Integrate with Bitbucket

Before you begin

Obtain access to a Bitbucket source code provider account to connect to the repositories.

To integrate Tenable Cloud Security with Bitbucket:

1. Navigate to the [Connect to repository](#) page.
2. In the **Connect to a version control provider** section, click **Bitbucket**.

**Connect to repository**

✓ Choose a workflow to discover repo(s): **Version control** ✎

• Connect to a version control provider

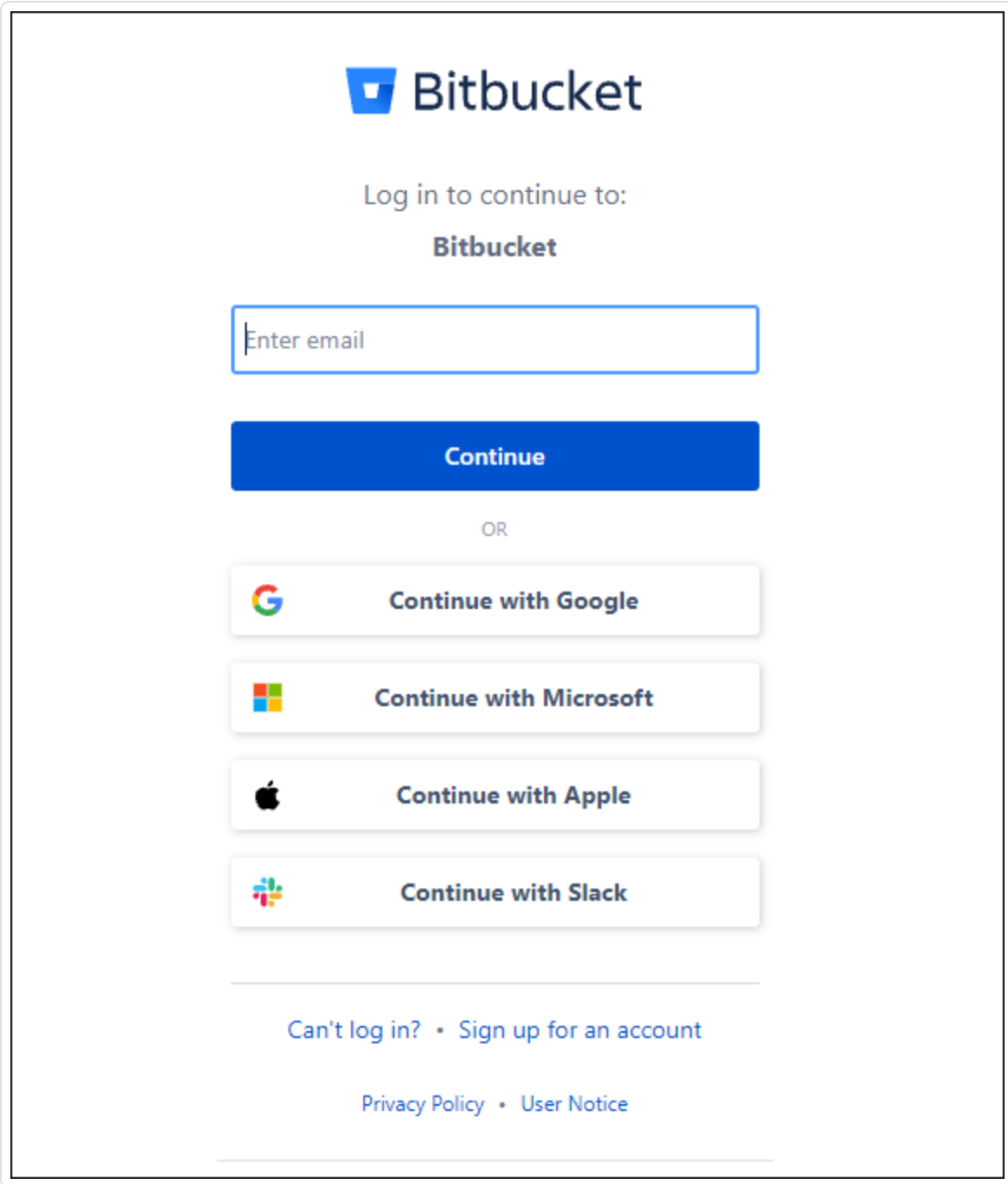
**i** For Auto-Remediation features to work, integrate using SCM credentials with admin equivalent privileges on target repositories.

<b>BITBUCKET</b> Not setup	<b>GITHUB</b> ✓ Reset	<b>GITLAB</b> Not setup	<b>AZURE DEVOPS</b> Not setup
<b>AWS CODE COMMIT</b> Not setup	<b>ON-PREMISE CODE SCANNER</b>		

**CONTINUE**      PREVIOUS

3. Click **Connect to Bitbucket**.

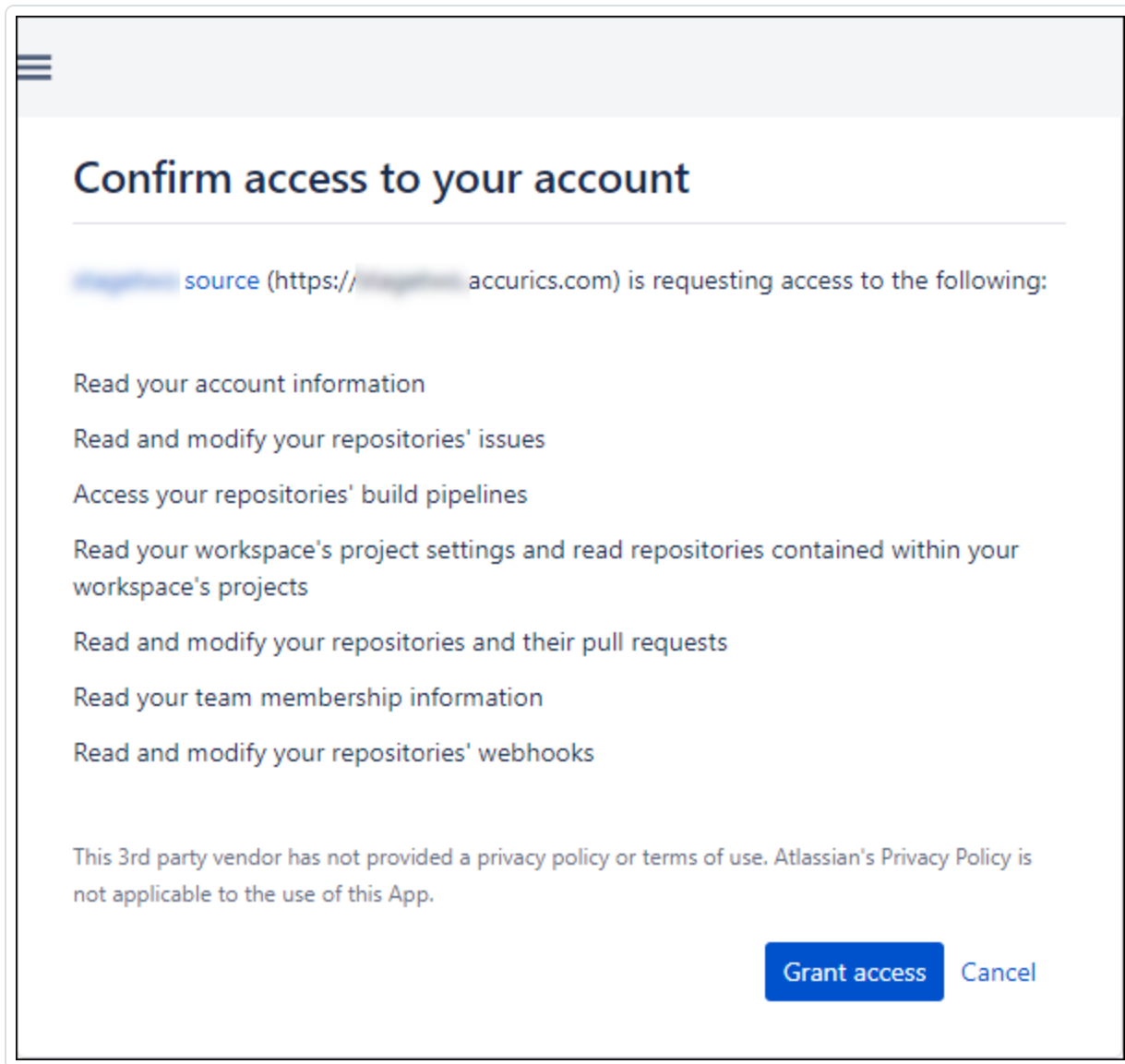
Tenable Cloud Security redirects you to the log in page of the Bitbucket source code provider.



4. In the Bitbucket login window, type the email address of your Bitbucket account.
5. Click **Continue**.
6. Type the password for the specified email address.
7. Click **Log in**.





If you are connecting to Bitbucket for the first time, Tenable Cloud Security requires access permissions to your Bitbucket account.



8. Read the access permissions required.

9. Click **Grant access**.

A message confirms that Tenable Cloud Security connected to Bitbucket using the specified credentials. Once the connection succeeds a  **Connected** icon appear next to the source code provider.

10. (Optional) To disconnect the source code provider, click .





A dialog appears asking you to confirm whether you want to disconnect. Click **Yes** to disconnect.



---

## Integrate with GitLab

---

Before you begin:

Obtain access to a GitLab source code provider account to connect to the repositories.

To integrate Tenable Cloud Security with GitLab:

1. Navigate to the [Connect to repository](#) page.
2. In the **Connect to a version control provider** section, click **GitLab**.

Tenable Cloud Security redirects you to the log in page of the GitLab source code provider.



**GitLab.com**

GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab Community Forum](#)
- [GitLab Homepage](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms.](#)

**Username or email**

**Password**

Remember me

[Forgot your password?](#)

**Sign in**

Don't have an account yet? [Register now](#)

**Sign in with**

Google


GitHub

3. In the **Username** and **Password** fields, type your GitLab credentials.

4. Click **Sign in**.

Tenable Cloud Security connects to the source code provider. Once the connection succeeds a **Connected** icon appear next to the source code provider.



5. (Optional) To disconnect the source code provider, click  .

A dialog appears asking you to confirm whether you want to disconnect. Click **Yes** to disconnect.



# Integrate with Azure DevOps

Before you begin:

Obtain access to a Microsoft Azure DevOps source code provider account to connect to the repositories.

To integrate Tenable Cloud Security with Azure DevOps:

1. Navigate to the [Connect to repository](#) page.
2. In the **Connect to a version control provider** section, click **Azure DevOps**.

**Connect to repository**

✓ Choose a workflow to discover repo(s): [Version control](#)

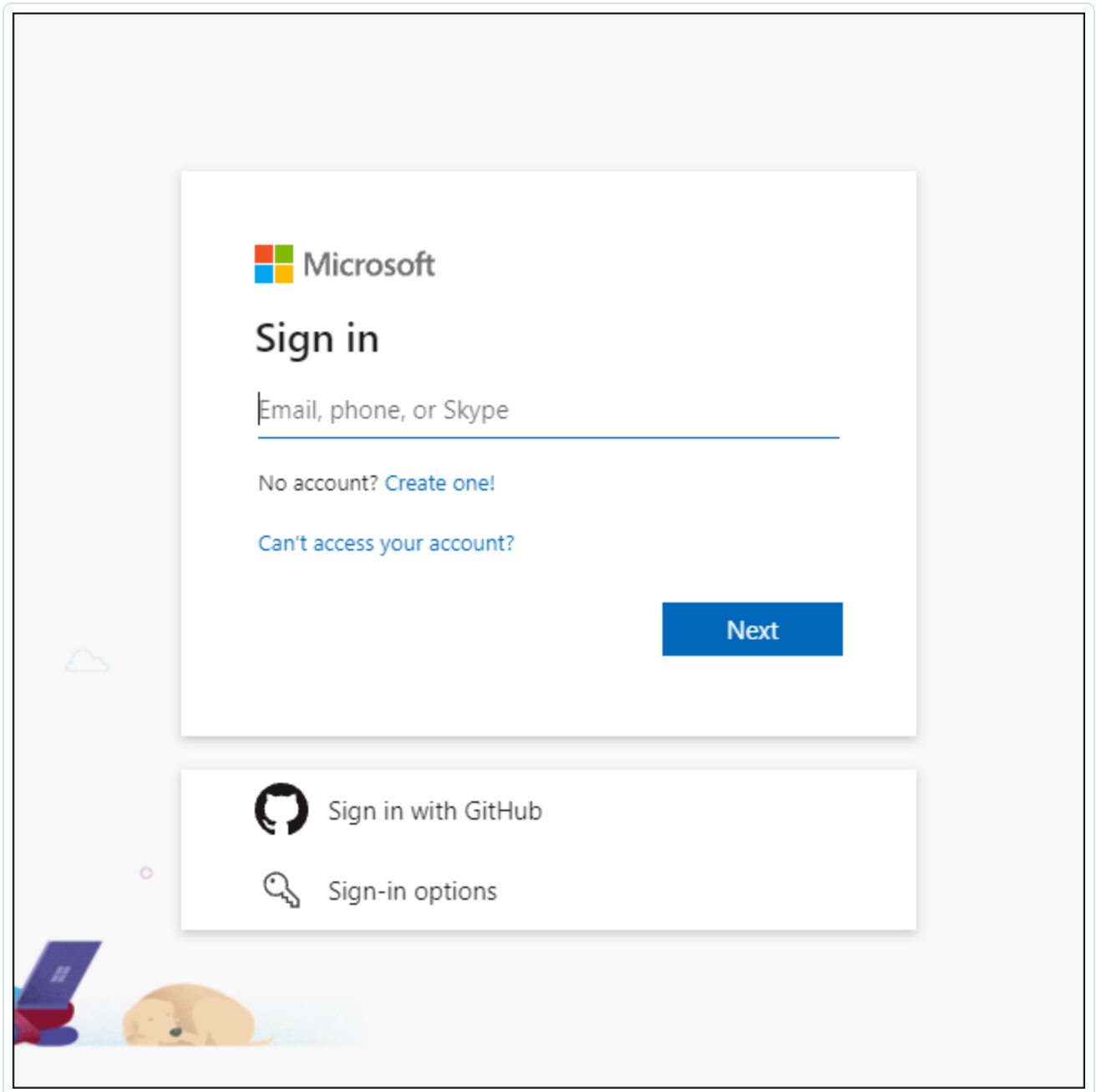
• Connect to a version control provider

**i** For Auto-Remediation features to work, integrate using SCM credentials with admin equivalent privileges on target repositories.

BITBUCKET Not setup	GITHUB Reset	GITLAB Not setup	AZURE DEVOPS Not setup
AWS CODE COMMIT Not setup	ON-PREMISE CODE SCANNER		

**CONTINUE**      PREVIOUS


Tenable Cloud Security Console redirects you to the Microsoft **Sign in** page of the Azure DevOps source code provider.



3. Type your Microsoft email address for Azure DevOps.
4. Click **Next**.
5. Select the **Work or school account** option to sign in with your work account.
6. Type the password associated with the email address.
7. Click **Sign in**.



If this is your first time connecting to Microsoft Azure DevOps, Tenable Cloud Security requires access permissions to your Microsoft account.

 by Accurics

App requests the following permissions from: [\[blurred email\]](#)  
(Tenable Security Solutions)

**Identity (read)**  
Grants the ability to read identities and groups.

**Code (read and write)**  
Grants the ability to read, update, and delete source code, access metadata about commits, changesets, branches, and other version control artifacts. Also grants the ability to create and manage pull requests and code reviews and to receive notifications about version control events via service hooks.

**Project and team (read)**  
Grants the ability to read projects and teams.

**Delegated Authorization Tokens**  
Grants the ability to manage delegated authorization tokens to users


[Learn more](#)


If you change your mind at any time, you can manage authorizations on your [profile page](#).

By clicking **Accept**, you allow this app to perform the above actions on your behalf and you agree to Accurics Terms of Use and Privacy Statement.

8. Read the access permissions required and click **Accept**.



Tenable Cloud Security connects to the source code provider. Once the connection succeeds a  **Connected** icon appear next to the source code provider.

9. (Optional) To disconnect the source code provider, click  .

A dialog appears asking you to confirm whether you want to disconnect. Click **Yes** to disconnect.





# Integrate with AWS CodeCommit

Tenable Cloud Security integrates with AWS CodeCommit and scans the repositories for any violations.

**Note:** The following features are not currently available with AWS CodeCommit:

- Scanning a particular branch or folder of a repository. Tenable Cloud Security scans only the main or master branch.
- **Auto-remediate settings** option during repository scan.
- Creating pull requests when remediating violations.

Before you begin:


- Obtain access to an AWS CodeCommit source code provider account to connect to the repositories.
- You must have the ARN of the role with access to AWS CodeCommit.


For more information, see [Set Up Write Access for AWS CodeCommit](#).

To integrate Tenable Cloud Security with AWS CodeCommit:

1. Navigate to the [Connect to repository](#) page.
2. In the **Connect to a version control provider** section, click **AWS CodeCommit**.  
Tenable Cloud Security redirects you to the log in page of the AWS CodeCommit source code provider.
3. In the **Role ARN for Code Commit** box, type the role ARN.  
For more information about getting the Role ARN for code commit, see [Setting up write access for AWS CodeCommit](#).
4. Click the **Select a region** box.
5. Select the appropriate AWS region.
6. Click **Connect to AWS Code Commit**.



Tenable Cloud Security connects to the source code provider. Once the connection succeeds a  **Connected** icon appear next to the source code provider.

7. (Optional) To disconnect the source code provider, click  .

A dialog appears asking you to confirm whether you want to disconnect. Click **Yes** to disconnect.



# Set Up Write Access for AWS CodeCommit

To onboard your AWS CodeCommit repositories, you must provision an IAM (Identity and Access Management) role in the target AWS cloud account and configure it for Tenable Cloud Security to access the resources in that AWS account. Attach the following AWS policy to provide sufficient permissions to Tenable Cloud Security:

- **AWSCodeCommitFullAccess**: Provides full access to AWS CodeCommit via the AWS Management Console.

Before you begin:

- Log in to the AWS web console with a user account with permission to create IAM roles.

For more information about IAM roles, see Amazon's [AWS Identity and Access Management User Guide](#).

To create a read-only role:

1. In the AWS web console, go to **Identity and Access Management (IAM)**.
2. On the left navigation pane, click **Roles**.

The **Roles** page appears.

3. Click **Create Role**.

The Create Role wizard appears.

4. In the **Select trusted entity** page, do the following:

- a. In the **Trusted entity type** section, select **AWS Account**.
- b. In the **An AWS Account** section, select **Another AWS Account**.
- c. In the **Account ID** box, type **012615275169**.

**Note:** 012615275169 is the account ID of the Tenable AWS account that you will be establishing a trust relationship with to support AWS role delegation.

- d. Under **Options**, click the **Require External ID** check box and type your Tenable Vulnerability Management Container UUID in the External ID box.



**Note:** In Tenable Vulnerability Management, navigate to **Settings > License** to get your container UUID. For more information, see [View your License Information](#) in Tenable Vulnerability Management.

- e. Click **Next**.
5. On the **Add permission policies** page, perform the following:
    - a. Search for and select the **AWSCodeCommitFullAccess** policy.
    - b. Click **Next**.
  6. In the **Name, review, and create** page, do the following:
    - a. In the **Role Details** section, type a **Role Name** for the role.
    - b. (Optional) Add a role description in the **Description** box.
    - c. (Optional) Click **Add Tags** to add key-value pairs to AWS resources.
    - d. Click **Create Role**.

The role is created and the role summary appears. In the **Summary** section, note the **Role ARN** value. You need the role ARN when onboarding AWS CodeCommit repositories.

**Summary** Edit

Creation date September 24, 2021, 11:49 (UTC-07:00)	ARN <a href="#">arn:aws:iam::333567660568:role/codeCommitReadRole</a>	Link to switch roles in console <a href="https://signin.aws.amazon.com/switchrole?roleName=codeCommitReadRole&amp;account=tenable-accurics-devqa">https://signin.aws.amazon.com/switchrole?roleName=codeCommitReadRole&amp;account=tenable-accurics-devqa</a>
Last activity <span>4 days ago</span>	Maximum session duration 1 hour	

**Permissions** | Trust relationships | Tags (4) | Access Advisor | Revoke sessions

**Permissions policies** (1)  
You can attach up to 10 managed policies.

< 1 > ⚙️

<input type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input type="checkbox"/>	<a href="#">AWSCodeCommitFullAccess</a>	AWS managed	Provides full access to AWS CodeCom...



# Scan Kubernetes Cluster Environments

In Tenable Cloud Security, you can assign policies to Kubernetes cluster environments and perform cloud scans on these environments to check if they comply with the assigned policies. You can initiate a scan from the Tenable Cloud Security Console, the command line (CLI), or using Helm Chart.

**Note:** You might see the resource count in a Kubernetes cluster changing during the scan. This is due to the frequently changing run-time state of the Kubernetes cluster. For example, pods and other resources in the cluster might go through different phases in their life cycle.

- [To run a scan from the Tenable Cloud Security Console](#)
- [To run a scan from the CLI](#)
- [To run a scan using Helm Chart](#)

Before you begin:

- Download and install the Tenable Cloud Security CLI. For more information, see [Set up Code Analysis through CLI](#).
- Ensure that you have the following access:
  - Read access to the kube-system namespace resource (excluding the resources within the kube-system namespace).
  - Read access to the list of namespaces present in the cluster.
  - (Only for Azure) Read access to query a configmap named container-azm-ms-aks-k8scluster within the kube-system namespace.

## To run a scan from the Tenable Cloud Security Console:

1. [Access Tenable Cloud Security](#).

The **Dashboard** page appears.

2. Click the **Projects & Connections** tab.

The **Projects & Connections** page appears.



3. Hover over the project that you want to scan and click **Run Scan > Configure Cloud Scan**.

The **Scan Options** window appears.

4. Select one of the following options as required by your cloud provider:

- AWS – Elastic Kubernetes Service (EKS)
- Azure– Kubernetes Cluster
- Google Cloud Platform – Google Kubernetes Engine

5. Click **Run Scan**.

A message confirms that Tenable Cloud Security initiated the cloud scan.

### To run a scan from the CLI:

In the CLI, you can use the pipeline mode or the configuration file mode to scan cluster environments. Tenable Cloud Security scans clusters as part of a regular cloud scan.

To run a scan using the pipeline mode:

1. In the Tenable Cloud Security CLI, run the following command to scan Kubernetes cluster environments, where:

- `cluster` triggers the scan for the Kubernetes cluster.

**Note:** Add the `cluster` command to scan Kubernetes clusters.

- `provider` is the cloud provider: AWS, Azure, or Google Cloud Platform.
- `token` is the API token.

```
accurics scan k8s -cluster -mode=pipeline -provider=<aws/azure/gcp> -  
appurl=https://cloud.tenable.com/cns -token=<token>
```



**Tip:** Some commonly used flags include:

- `wait` – Lists the violation reports on the terminal.
- `fail` – Returns an exit code of 1.
- `verbose` – Lists violation details.

Run `accurics -h` to access Help. For more information about flags, see [Tenable Cloud Security Commands and Options](#).

Once the scan completes, Tenable Cloud Security displays the violation reports from the scan.

**Note:** Tenable Cloud Security displays the resources and misconfigurations from this scan under **Cloud** (not IaC) in the **Projects** tab.

To run a scan using the configuration file:

1. [Access Tenable Cloud Security](#)

2. In the left navigation bar, click  HOME .

The **Dashboard** page appears.

3. Click the **Projects & Connections** tab.

The **Projects & Connections** page appears.

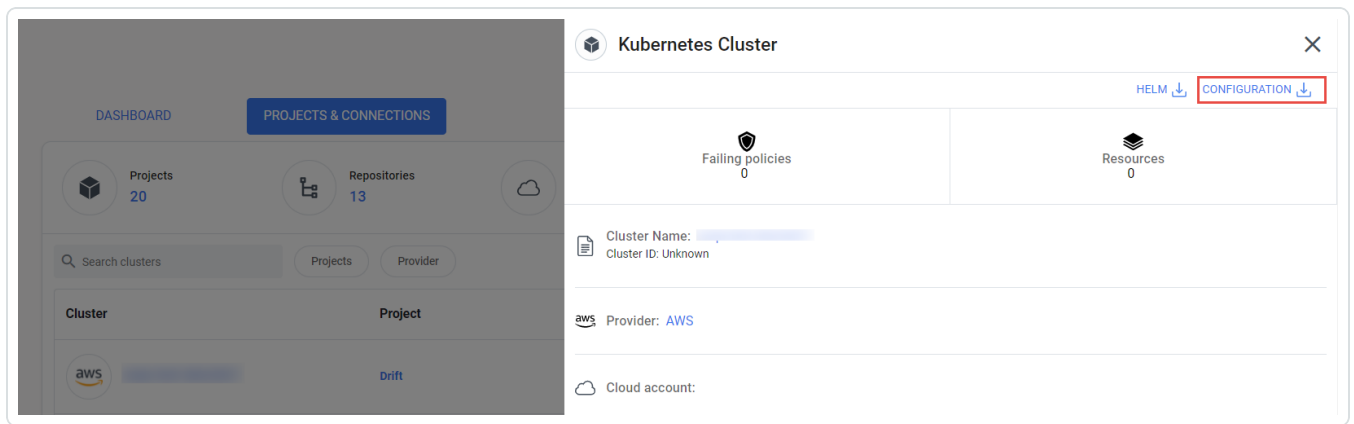
4. Click **K8s Clusters**.

The **K8s Clusters** page appears.

5. Select the Kubernetes cluster project you want to scan.

The **Kubernetes Cluster** pane appears.

6. Click **Configuration** to download the configuration file.



7. In the Tenable Cloud Security CLI, run the following command to scan the Kubernetes cluster project, where configuration file path is the location of the configuration file:

```
accurics scan k8s -cluster -config=<configuration file path>
```

**Tip:** Some commonly used flags include:

- wait – Lists the violation reports on the terminal.
- fail – Returns an exit code of 1.
- verbose – Lists violation details.

Run `accurics -h` to access Help. For more information about flags, see [Tenable Cloud Security Commands and Options](#).

Once the scan completes, Tenable Cloud Security shows the violation reports from the scan on the **Projects and Connections** tab.

**Note:** Tenable Cloud Security displays the resources and misconfigurations from this scan under **Cloud** (not IaC) in the **Projects** tab.

### To run a scan using Helm Chart:

1. [Access Tenable Cloud Security](#)

2. In the left navigation bar, click  HOME .

The **Dashboard** page appears.





3. Click the **Projects & Connections** tab.

The **Projects & Connections** page appears.

4. Click **K8s Clusters**.

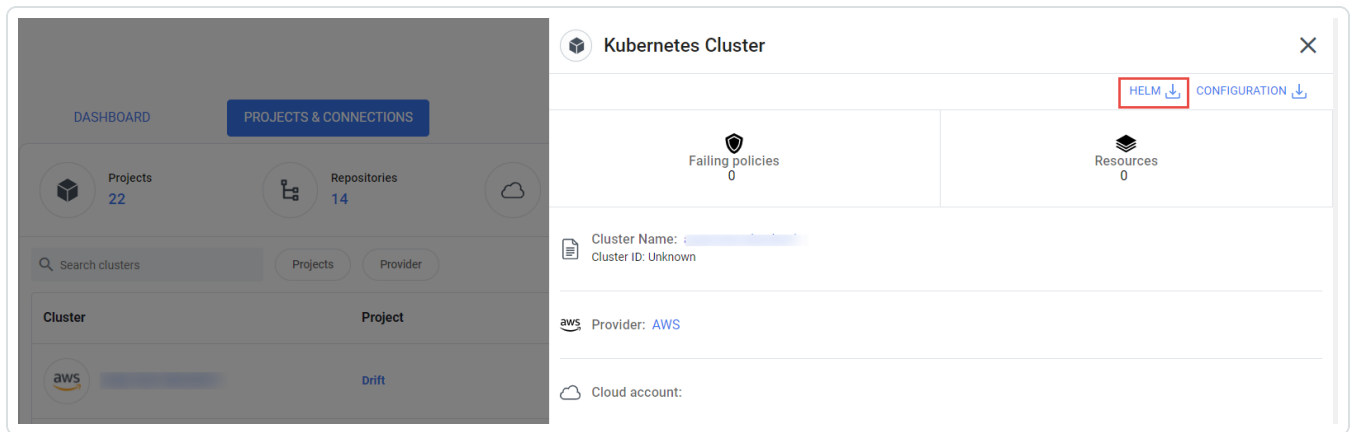
The **K8s Clusters** page appears.

5. Select the Kubernetes cluster project that you want to scan.

The **Kubernetes Cluster** pane appears.

6. Click **Helm** to download the Helm Chart file.

Tenable Cloud Security downloads the `accurics-kubescan-helm.zip`.



7. Extract the zip file and follow the instructions in the `instructions.txt` file to deploy the Helm Chart resources.

Once the scan completes, Tenable Cloud Security shows the violation reports from the scan.



---

## Set up Code Analysis Using CLI

---

You can use the Tenable Cloud Security (formerly known as Tenable.cs) command-line interface (CLI) to scan code on your local machine. Tenable Cloud Security provides security for CI/CD pipelines. You can integrate Tenable Cloud Security CLI into the CI/CD jobs to detect violations and block risky builds and view scan results in the Tenable Cloud Security Console.

**Note:** All instances of Tenable Cloud Security CLI refer to Accurics CLI.

There are two ways to scan your IaC code through the CLI:

- **Plan-based analysis** (`accurics plan`): The `accurics plan` command supports only Terraform files.
- **Static analysis** (`accurics scan`): The `accurics scan` command supports Terraform, CloudFormation templates, Azure Resource Manager template, Kubernetes, Kustomize, and Helm Chart. You must install Terrascan in your environment to perform static analysis.



## Download Configuration File

The configuration file of a project contains target, environment, application, and repository details. Tenable Cloud Security CLI uses the configuration file to run a scan.

1. [Access Tenable Cloud Security](#).

2. In the left navigation bar, click  HOME .

The **Projects & Connections** page appears.

3. Click the **Projects & Connections** tab.

Tenable Cloud Security displays the list of projects.

4. Hover over the required project and click the project name. For information about creating projects, see [Create a Project](#).

**Tip:** Use the **Search** box to search for a specific project.

The **Project** details pane appears.

5. Click the **Configuration**  button to download the configuration file.

Tenable Cloud Security downloads a .zip file that contains the configuration files for each repository in the project. The configuration file includes the following fields:

- target: Tenable Cloud Security cloud environment.
- env: Project ID in the Tenable Cloud Security Console.
- app and api-token: Authentication token.
- repoName: Repository URL.



---

## Install or Upgrade the CLI

---

You must download the Tenable Cloud Security CLI before you can use it in a CI/CD job. The procedure to install the Tenable Cloud Security CLI varies for each operating system. You can also upgrade to the latest version of the Tenable Cloud Security CLI on macOS and Linux.

Before you begin:


- You must have a Tenable Cloud Security user account with an **Operator** role.
- [Create](#) a project in the Tenable Cloud Security Console to scan the IaC repository to use for the CI/CD builds.
- [Download](#) the configuration file.

## Install the Tenable Cloud Security from the Web Console

To download and install the Tenable Cloud Security CLI:

1. [Access Tenable Cloud Security](#).
2. On the **Home** page, click the project for which you want to download the Tenable Cloud Security CLI.
3. On the Project details panel, click the **CLI** download link.

The **CLI usage instructions** window appears.

4. Select the target operating system you are planning to execute the CLI on from the **OS** dropdown list and click .

The following CLI installation executable file is downloaded depending on your selection of operating system:

- Windows: `accurics`
- macOS: `accurics_mac`
- Linux: `accurics_linux`



**Note:** Alternatively, you can download the latest version of the CLI directly from the following URLs:

- Mac: [https://downloads accurics.com/cli/latest/accurics\\_mac](https://downloads accurics.com/cli/latest/accurics_mac)
- Linux: [https://downloads accurics.com/cli/latest/accurics\\_linux](https://downloads accurics.com/cli/latest/accurics_linux)
- Windows: <https://downloads accurics.com/cli/latest/accurics.exe>

5. Give execute permission to the downloaded CLI file. For example, `chmod +x accurics_linux`.

## Install or Upgrade the Tenable Cloud Security CLI from Homebrew

Tenable Cloud Security CLI is also available on [Homebrew](#). You can upgrade the CLI only on Linux and macOS.

To install or upgrade the Tenable Cloud Security CLI:

1. Run one of the following command on a macOS or Linux system:
  - `brew install accurics`
  - `brew upgrade accurics`

What to do next:

Place the Tenable Cloud Security executable and the Tenable Cloud Security configuration file in the CI/CD build repository.



# Scan IaC Files Using CLI

You can use Tenable Cloud Security CLI to scan and list the vulnerabilities in the IaC code. There are two ways to scan your IaC code:

- [Plan-based analysis \(accurics plan\)](#)
- [Static analysis \(accurics scan\)](#)

You can run the Tenable Cloud Security CLI in the following modes:

- **Pipeline mode** – In this mode, specify all the required parameters with the `accurics plan` or `accurics scan` command to run the scan.
- **With configuration file** – In this mode, specify the configuration file. Tenable Cloud Security uses the configuration file parameters and automatically runs the scan.

## Plan-based Analysis

You can run a plan-based analysis using the `accurics plan` command. Plan-based analysis supports only Terraform files. You can view the scan results in the Tenable Cloud Security Console.

Before you begin:

- [Download the configuration file.](#)
- [Install Terraform.](#)

To run a plan-based analysis using the Tenable Cloud Security CLI:

1. In the command terminal, initialize Terraform configuration files:

```
accurics init
```

2. Run the `accurics plan` command in the following ways:



- **Pipeline mode**

```
accurics plan -mode=pipeline -appurl=<application_url> -token=<API_token> [-pro-  
ject=<project_ID>]
```

Where:

- `application_url`: URL of the Tenable Cloud Security Console, which is `https://cloud.tenable.com/cns`.
- `API_token`: API authentication token you generate from Tenable Cloud Security. For more information, see [Generate API Tokens](#).
- `project_ID`: (Optional) Project in Tenable Cloud Security. If you specify the project, Tenable Cloud Security sends the scan results to this project. If you do not specify the project, Tenable Cloud Security creates a default project for displaying the scan results.

- **With configuration file**

```
accurics plan -config=<config_file_path>
```

Where:

- `config_file_path`: Relative or absolute path of the configuration file that you download from the Tenable Cloud Security Console.

## Example



```
MacBook-Pro acqa-repo1-aws-tf12-part1 % accurics plan -config=config_acqa-repo1-aws-tf12-part1
2021/01/15 18:33:44 runPlan...
2021/01/15 18:33:44 [plan -out=1610715824491.out]

2021/01/15 18:34:40 Running Accurics analysis...
MacBook-Pro acqa-repo1-aws-tf12-part1 /acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 mapping terraform resources to source code...
2021/01/15 18:34:40 Repo Root Path... /acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 Current working directory ... /acqa-repo1-aws-tf12-part1
2021/01/15 18:34:40 getting source code for all the resources present in '...' /acqa
2021/01/15 18:34:40 getting source code for all the resources present in '...' /acqa
m/lgallard/terraform-aws-codebuild.git'
2021/01/15 18:34:40 resources to source code mapping done!
2021/01/15 18:34:40 Creating dependency graph...
2021/01/15 18:34:40 GetDotFileUsingGraph Directory: /acqa-repo1-aws-tf12-part1
2021/01/15 18:34:43 Using configuration file:- config_acqa-repo1-aws-tf12-part1

-----

Accurics successfully scanned the repository! Following is the summary - for details visit Accurics Web Console.

{
  "resources": 48,
  "violation": 4,
  "low": 0,
  "medium": 0,
  "high": 4,
  "native": 1,
  "inherit": 3,
  "drift": 0,
  "iacdrift": 0,
  "clouddrift": 0
}

-----

MacBook-Pro acqa-repo1-aws-tf12-part1 % echo $?
1
MacBook-Pro acqa-repo1-aws-tf12-part1 % █
```

## Static Analysis

You can run a static analysis with the `accurics scan` command. The `accurics scan` command Terraform, CloudFormation templates, Azure Resource Manager template, Kubernetes, Kustomize, and Helm Chart.

Before you begin:

- [Download the configuration file.](#)
- [Install Terrascan](#)

To run a static analysis using the Tenable Cloud Security CLI:





1. Run the `accurics scan` command in the following ways:

- **Pipeline mode**

```
accurics scan -mode=pipeline -appurl=<application_url> -token=<API_token>
```

- **With configuration file**

```
accurics scan -config=<config_file_path>
```

Where:

- `application_url`: URL of the Tenable Cloud Security Console, which is `https://cloud.tenable.com/cns`.
- `API_token`: API authentication token you generate from Tenable Cloud Security. For more information, see [Generate API Tokens](#).
- `config_file_path`: Relative or absolute path of the configuration file that you downloaded from the Tenable Cloud Security Console.

For detailed information about the commands and parameters in Tenable Cloud Security CLI, see [Tenable Cloud Security Commands and Options](#).



## Scan IaC Files in the CLI Local Mode

You can use Tenable Cloud Security CLI to view scan results locally without publishing them to the cloud with the local mode. In this mode, the scan results are displayed in the console and the CLI does not push the scan results to the Tenable Cloud Security Console. You can use this feature to scan your test repository branches for any violations. Local mode is supported only for IaC scans with both plan-based and static analysis.

**Note:** Kubernetes scan is not supported in the local mode.

Before you begin:

You must have the following:

- Project ID

The policy attached to the selected project is used for the assessment. For more information, see [Create a Project](#) and [Associate Policies with a Project](#).

- Configuration file. For more information, see [Download the configuration file](#).
- Terraform. For more information, see [Install Terraform](#).
- CLI. For more information, see [Install or Upgrade the CLI](#).

Ensure that the CLI version is 1.0.42 and higher.

To run an IaC scan using the Tenable Cloud Security CLI:

1. In the command terminal, initialize Terraform configuration files:

```
accurics init
```

2. Run the `accurics plan` or `accurics scan` command in the following ways:

- **Pipeline mode**

```
accurics plan -mode=pipeline -appurl=<application_url> -token=<API_token> -project=<project_ID> -test
```



```
accurics scan -mode=pipeline -appurl=<application_url> -token=<API_token> -project=<project_ID> -test
```

Where:

- `application_url`: URL of the Tenable Cloud Security Console, which is `https://cloud.tenable.com/cns`.
  - `API_token`: API authentication token you generate from Tenable Cloud Security. For more information, see [Generate API Tokens](#).
  - `project_ID`: Project in Tenable Cloud Security. Specify the project ID for running a scan in the local mode.
  - `-test`: Specifies that the repository and scan results are not pushed to the Tenable Cloud Security Console.
- **With configuration file**

```
accurics plan -config=<config_file_path> -test
```

```
accurics scan -config=<config_file_path> -test
```

Where:

- `config_file_path`: Relative or absolute path of the configuration file that you download from the Tenable Cloud Security Console.

For detailed information about the commands and parameters in Tenable Cloud Security CLI, see [Tenable Cloud Security Commands and Options](#).



# Tenable Cloud Security Commands and Options

This section lists the following Tenable Cloud Security commands and parameters:

- [General Commands](#)
- [Scan Commands](#)
- [Command Options](#)

## Commands

Tenable Cloud Security CLI supports the following commands:

### General Commands

Command	Description
<code>init</code>	This command is a wrapper over the terraform <code>init</code> command.  <code>accurics init</code>
<code>configure</code>	This command prompts you to provide the endpoint of the Tenable Cloud Security Console and creates a configuration file that you can use while running the <code>accurics plan</code> command.  <code>accurics configure</code>
<code>workspace</code>	This command is a wrapper over the terraform <code>workspace</code> command.
<code>version</code>	Shows the Tenable Cloud Security CLI version.

### Scan Commands

Command	Description
<code>plan</code>	Use this command for plan-based analysis. This command supports only Terraform files. This command detects viol-



	<p>ations in the Terraform files located in the current directory.</p> <p>Syntax:</p> <ul style="list-style-type: none"><li>• <code>accurics plan -mode=pipeline -appurl=&lt;application_url&gt; -token=&lt;API_token&gt;</code></li><li>• <code>accurics plan -config=&lt;configfile_path&gt;</code></li></ul>
<code>tgplanall</code> or <code>plan-all</code>	<p>This command detects violations in the Terragrunt/Terraform files that are in the current directory and within each subfolder.</p> <p>Syntax:</p> <ul style="list-style-type: none"><li>• <code>accurics tgplanall -config=&lt;configfile_path&gt;</code></li><li>• <code>accurics plan-all -config=&lt;configfile_path&gt;</code></li></ul>
<code>tgplan</code>	<p>This command detects violations in the Terragrunt/Terraform files in the current directory. Use this command if you do not want to run the <code>terragrunt plan-all</code> command and want to scan individual folders under the main Terragrunt folder. In the following example, <b>topfolder</b> is the top-level folder and <b>folder1</b> and <b>folder2</b> are subfolders. You can run the <code>accurics tgplan</code> command on one folder at a time.</p> <p>Syntax:</p> <ul style="list-style-type: none"><li>• <code>topfolder&gt;folder1&gt; accurics tgplan -config=&lt;configfile_path&gt;</code></li><li>• <code>topfolder&gt;folder2&gt; accurics tgplan -config=&lt;configfile_path&gt;</code></li></ul>
<code>scan</code>	<p>This command is for static analysis and uses Terrascan (<a href="https://github.com/accurics/terrascan">github.com/accurics/terrascan</a>) to scan different IaC types. Supports the following IaC types:</p>



- Terraform
- Kubernetes
- Helm Chart
- Kustomize
- CloudFormation template

Syntax:

- `accurics scan -mode=pipeline -appurl=<application_url> -token=<API_token>`
- `accurics scan -config=<configfile_path>`

## Command Options

Tenable Cloud Security CLI supports the following options with the `accurics plan` and `accurics scan` commands:

Option	Description	Required/Optional
<code>-config=&lt;configfile_path&gt;</code>	Specify the configuration file location that you downloaded from Tenable Cloud Security. This option accepts absolute or relative file paths (defaults to <code>./config</code> , then checks <code>&lt;HOMEDIR&gt;/.accurics/config</code> ). <a href="#">Download Configuration File</a>	Required if not running the pipeline mode
<code>-fail</code>	Returns exit code 1 when Tenable Cloud Security detects high severity violations.	Optional
<code>-verbose</code>	Print detailed logs along with the output.	Optional
<code>-pulltfstate</code>	Pull the Terraform state file from a	Optional. Only applic-



	remote data store (S3 buckets on AWS). This command downloads the state file and also triggers a cloud scan.	able for the <code>accurics plan</code> command.
<code>-tfstate=&lt;statefile_path&gt;</code>	Specify the file path of the locally stored state file. For example:  <code>accurics plan -config=&lt;config file&gt; -tfstate=&lt;statefile_path&gt;</code> This command uses the provided state file and triggers a cloud scan.	Optional
<code>-cloudscan</code>	Trigger a cloud scan from the CLI. Tenable Cloud Security downloads the file from the S3 bucket if you provide the S3 bucket details during repository configuration on the Tenable Cloud Security Console.	Optional
<code>-planjson=&lt;file&gt;</code>	Specify the Terraform plan JSON output file with the <code>accurics plan</code> command to use that file for scanning.	Optional. Only applicable for the <code>accurics plan</code> command.
<code>-mode=pipeline</code>	Set the mode to pipeline. Optional if you specify the configuration file.	Required for pipeline mode. Optional if you specify the configuration file.
<code>-token=&lt;token&gt;</code>	Specify the authentication token.	Required for pipeline mode. Optional if you specify the configuration file.
<code>-appurl=&lt;application_url&gt;</code>	Specify the URL of the Tenable Cloud Security console.	Required for pipeline mode. Optional if you



		specify the configuration file.
<code>-project=&lt;project_ID&gt;</code> or <code>-env=&lt;environment ID&gt;</code>	Specify the project in Tenable Cloud Security.	Optional
<code>-test</code>	Results of the IaC scan are not pushed to the Tenable Cloud Security Console.	Optional. Supported with CLI version 1.0.42 and higher.
<code>var-file</code>	If a variable file is used with Terraform plan, specify the relative path to the file. For example, <code>-var-file-e=/varDefs/values.tfvars</code>	Optional
<code>&lt;custom_variable&gt;</code>	Specify a custom parameter name and provide a value for it. For example, <code>var="foo=bar"</code>	Optional

## Environment Variables

Option	Description	Required/Optional
<code>ACCURICS_APP_ID</code>	Specify the application ID.	Required
<code>ACCURICS_ENV_ID</code>	Specify the project ID.	Required
<code>ACCURICS_REPO_NAME</code>	Specify the repository name.	Required
<code>ACCURICS_URL</code>	Specify the URL endpoint.	Optional





---

# Container Security with Tenable Cloud Security

---

Tenable Cloud Security scans your container images and container registries to assess for vulnerabilities. Tenable Cloud Security allows you to scan container images securely without sending the images outside your organization's network. After your scan completes, you can view the scan results in the Tenable Cloud Security Console.

Tenable Cloud Security allows you to scan the following:

- A local image from Docker daemon.
- An image in a build pipeline.
- All images hosted in a specific registry (for example, a Docker registry).

Before you begin:

- [Create a project](#) in the Tenable Cloud Security Console to use for the container scan.
- Ensure that the container image is available in the docker daemon.

To configure container scans with Tenable Cloud Security:

1. Create custom policies and policy group for your image. For more information, see [Create a Custom Policy](#) and [Create a Custom Policy Group](#).
2. [Associate Container Security policies to the project](#).
3. (Optional) [Download the configuration file for the project](#).
4. [Download and install the CLI](#).

**Note:** You can install the CLI locally on your system, integrate the CLI in your CI/CD pipeline, or run the CLI as a Docker image.

5. Scan the container image or container registry.
  - [Scan a Container Image](#)
  - [Scan a Container Registry](#)
6. On the Tenable Cloud Security Console, view the scan results on the [Vulnerabilities](#) tab on



the **Findings** page.

7. Get container security insights from the [Containers dashboard](#).



---

# Install Tenable Cloud Security CLI for Tenable Container Security

---

Use Tenable Cloud Security CLI version 2.0 to scan container images. You can install Tenable Cloud Security CLI on Linux and macOS operating systems.

## Supported Operating Systems

- macOS
- Linux

## Install Tenable Cloud Security on a Local System

To download and install the Tenable Cloud Security CLI:

1. Log in to the [Tenable Cloud Security Downloads page](#).
2. On the row for Tenable Cloud Security, click **View Downloads**.
3. Download the [latest installation file](#) for your operating system.

Tenable Cloud Security CLI is available for macOS and Linux operating systems. Use the following links for the download URLs:

- Linux (arm64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_Linux\\_arm64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_Linux_arm64.tar.gz)
- Linux (x86\_64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_Linux\\_x86\\_64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_Linux_x86_64.tar.gz)
- MacOS (arm64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_MacOs\\_arm64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_MacOs_arm64.tar.gz)
- MacOS (x86\_64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_MacOs\\_x86\\_64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_MacOs_x86_64.tar.gz)

4. Untar the Tenable Cloud Security CLI.
5. Allow executable permissions to the Tenable Cloud Security CLI binary file:



```
chmod +x tcs
```

6. From a command-line terminal, navigate to the download location and type the `tcs` command to verify that the installation is successful.

```
./tcs
Tenable.CS

Discover vulnerabilities and misconfigurations in container images.

Usage:
  tcs [command]

Examples:
tcs version

Available Commands:
  consec      Container Security
  env         Display the Tenable.cs CLI environment variables
  version     Display the Tenable.cs CLI version

Flags:
  -c, --config string      Specify the configuration file location.
  --fail                   Returns an exit code of 1 when a high severity violation is detected
  --log-dir string        Directory path to write the log file. Works only with '--log-level-
l=debug' (default "log")
  -l, --log-level string   Log level (Values: debug, info, warn, error, panic, fatal) (default
"info")
  -x, --log-type string    Log output type (Values: console, json) (default "console")
  -p, --project string     Project to associate the results in the Tenable Cloud Security web
console.
                           Use 'TCS_PROJECT_ID' to pass the project ID using an environment
variable
  --token string          API token from the configuration file (same as App Token in older
config files).
                           Use 'TCS_TOKEN' to pass the token using an environment variable

Use "tcs [command] --help" for more information about a command
```



# Scan a Container Image

Use the Tenable Cloud Security CLI to scan a container image. After Tenable Cloud Security scans your container image, you can view the detailed scan results on the Tenable Cloud Security Console.

Before you begin:

- [Create a project](#) in the Tenable Cloud Security Console to scan the container image.

**Note:** For accurate results in scan reports and dashboards, Tenable recommends to avoid scanning the same image from multiple projects.

- Ensure that the container image is available in the docker daemon.

To scan a container image from the Tenable Cloud Security CLI:

1. Run the `tcs consec image` command in one of the following ways:

- Without the configuration file

```
tcs consec image <image_name>:<tag> --project=<project_ID> --token=<API_token> --wait --retryInterval <poll_interval> --timeout <timeout_sec>
```

- With the configuration file:

```
tcs consec image <image_name>:<tag> --config=<config_file_path> --wait --retryInterval <poll_interval> --timeout <timeout_sec>
```

Where:

- `<image_name>:<tag>`: Image name with its tag. For example, `alpine:latest`.
- `<project_ID>`: Project ID in Tenable Cloud Security. Use `TCS_PROJECT_ID` to set the project ID with an environment variable.
- `<API_token>`: API authentication token you generate from Tenable Cloud Security. Use `TCS_TOKEN` to set the API token with an environment variable. For more information, see [Generate API Tokens](#).



- `<config_file_path>`: Specify the configuration file location that you downloaded from Tenable Cloud Security. This option accepts absolute or relative file paths (defaults to `./config`, then checks `<HOMEDIR>/.accurics/config`). For more information, see [Download Configuration File](#).
- Use the following optional parameters to wait for the scan results:
  - `--wait`: If you specify this option, Tenable Cloud Security waits for the duration specified with the `--timeout` parameter for the scan to complete. If the scan completes within the specified duration, Tenable Cloud Security generates two types of CLI outputs:
    - **Scan summary on the console**: Includes the summary of total misconfigurations (violations) and total vulnerabilities.
    - **JSON report**: Detailed scan report that indicates the details about the misconfigurations and vulnerabilities.

For more information about these CLI outputs, see [CLI Outputs for Container Image Scans](#).

- `--timeout <timeout_sec>`: The maximum time (in seconds) to wait for the violation report of the scan. The default value is 300 seconds (5 minutes). To change the default, use this option with the `--wait` option.
- `--retryInterval <poll_interval>`: The polling time interval (in seconds) while polling for the violation report of the scan. The default value is 5 seconds. Tenable Cloud Security checks whether the violation report is ready after every polling interval.

## CLI Outputs for Container Image Scans

In addition to displaying the scan results on the Tenable Cloud Security user interface, Tenable Cloud Security generates a scan summary on the console and a JSON report when you scan container images. To generate these two CLI outputs, you must use the `tcs consec image` command with the `--wait` option. The JSON report can be additionally used as an artifact of a successful CI/CD pipeline run or as raw data for post-processing of the scan results.



**Note:** If the `--wait` option is not specified with the `tcs consec image` command, the console summary and JSON report are not generated.

Tenable Cloud Security generates the following two CLI outputs:

- **Scan summary on the console:** Includes the summary of total misconfigurations (violations) and total vulnerabilities, categorized by severity.

```
Violation Summary:
  Policy Status      : MONITOR_FAIL
  Total Violations   : 1
  Enforced Violations : 0

  More details : https://cloud.tenable.com/cns/issues/vulnerabilities?project=<project_id>
  Total Count   : 5
  Distinct CVEs : 4
  Highest CVSSv2 Score : 6.7
  Highest CVSSv3 Score : 6.7
  Highest VPR Score  : 6.7

  CRITICAL : 0
  HIGH     : 4
  MEDIUM  : 0
  LOW      : 0

  More details : https://cloud.tenable.com/cns/issues/vulnerabilities?project=<project_id>
```

- **JSON report:** Detailed scan report that indicates the details about the misconfigurations and vulnerabilities.

**Note:** Tenable Cloud Security generates the JSON report in the `$(pwd)/report` folder with the name `tcs_image_scan_<project_id>.json`.

The following is a sample JSON report:

```
{
  "schema": "application/vnd+tenable.consec.report.v1.0+json",
  "scan_status": "FINISHED",
  "scan": {
    "asset_type": "image",
    "asset_id": "9ab82761-51f5-5fc4-ae33-7a052905f439",
    "scan_id": "d121c6de-ab7a-4929-ac33-72695ed9fb3c",
    "project_id": "5edaba47-4185-4b2e-abf1-c97803df5928",
    "asset": {
```



```
"name": "docker.io/library/influxdb:alpine",
"tag": "alpine",
"os": "linux",
"architecture": "arm64",
"built_at": "2023-06-15T03:03:03.448Z",
"last_updated": "2023-07-03T08:18:34.342021604Z",
"observation_source": "PIPELINE_IMAGE",
"exposed_ports": [
  "8086/tcp"
],
"environment_variables": [
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
  "INFLUXDB_VERSION=2.7.1",
  "INFLUX_CLI_VERSION=2.7.3",
  "INFLUX_CONFIGS_PATH=/etc/influxdb2/influx-configs",
  "INFLUXD_INIT_PORT=9999",
  "INFLUXD_INIT_PING_ATTEMPTS=600",
  "DOCKER_INFLUXDB_INIT_CLI_CONFIG_NAME=default"
],
"cmd": [
  "influxd"
],
"entry_point": null,
"image_labels": [],
"imported_at": "2023-06-29T17:32:38.258Z",
"image_id": "sha256:c0bc4371bc3a1e0c5f6c6e27e356724cf765e022bfe3984572e6960e4c55dbf5",
"digest": "sha256:c0bc4371bc3a1e0c5f6c6e27e356724cf765e022bfe3984572e6960e4c55dbf5",
"registry_url": ""
}
},
"policies": {
  "summary": {
    "status": "ENFORCE_FAIL",
    "enforce_failed_count": 1,
    "total_failed_count": 1
  },
  "violations": [
    {
      "id": "",
      "name": "test-custom-policy-group",
      "remediation": "test",
      "policy_mode": "ENFORCE",
      "severity": "HIGH"
    }
  ]
},
"vulnerabilities": {
  "summary": {
    "total_count": 5,
    "distinct_cve_count": 4,
    "max_cvss_v2_score": 7.8,
    "max_cvss_v3_score": 7.8,
    "max_vpr_score": 6.7,
    "severity_breakdown": {
      "critical": 0,
      "high": 4,

```



```

    "medium": 0,
    "low": 0
  }
},
"findings": [
  {
    "plugin_id": 400061,
    "vpr": {
      "score": 2.2
    },
    "cvss_v2": {
      "base_score": 5,
      "base_vector": "CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N"
    },
    "cvss_v3": {
      "base_score": 5.3,
      "base_vector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N"
    },
    "description": "There are packages installed that are affected by a vulnerability referenced in the following CVE:\n\n - Applications that use a non-default option when verifying certificates may be vulnerable to an attack from\n a malicious CA to circumvent certain checks. Invalid certificate policies in leaf certificates are\n silently ignored by OpenSSL and other certificate policy checks are skipped for that certificate. A\n malicious CA could use this to deliberately assert invalid certificate policies in order to circumvent\n policy checking on the certificate altogether. Policy processing is disabled by default but can be enabled\n by passing the '-policy' argument to the command line utilities or by calling the\n `X509_VERIFY_PARAM_set1_policies()' function. (CVE-2023-0465)",
    "family": "Alpine Linux Local Security Checks",
    "severity": "HIGH",
    "cve_ids": [
      "CVE-2023-0465"
    ],
    "published_date": "2023-03-21T00:00:00Z",
    "affected_packages": [
      {
        "name": "libcrypto3",
        "version": "3.0.9-r1"
      },
      {
        "name": "libssl3",
        "version": "3.0.9-r1"
      }
    ],
    "remediation": "Upgrade the affected packages."
  },
  {
    "plugin_id": 400067,
    "vpr": {
      "score": 3.6
    },
    "cvss_v2": {
      "base_score": 5.4,
      "base_vector": "CVSS2#AV:N/AC:H/Au:N/C:N/I:N/A:C"
    },
    "cvss_v3": {
      "base_score": 5.9,
      "base_vector": "CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H"
    },
  },

```



```
"description": "There are packages installed that are affected by a vulnerability referenced in the following CVE:\n\n - Issue summary: The AES-XTS cipher decryption implementation for 64 bit ARM platform contains a bug that\n could cause it to read past the input buffer, leading to a crash. Impact summary: Applications that use\n the AES-XTS algorithm on the 64 bit ARM platform can crash in rare circumstances. The AES-XTS algorithm is\n usually used for disk encryption. The AES-XTS cipher decryption implementation for 64 bit ARM platform\n will read past the end of the ciphertext buffer if the ciphertext size is 4 mod 5 in 16 byte blocks, e.g.\n 144 bytes or 1024 bytes. If the memory after the ciphertext buffer is unmapped, this will trigger a crash\n which results in a denial of service. If an attacker can control the size and location of the ciphertext\n buffer being decrypted by an application using AES-XTS on 64 bit ARM, the application is affected. This is\n fairly unlikely making this issue a Low severity one. (CVE-2023-1255)",
```

```
  "family": "Alpine Linux Local Security Checks",  
  "severity": "HIGH",  
  "cve_ids": [  
    "CVE-2023-1255"  
  ],  
  "published_date": "2023-04-20T00:00:00Z",  
  "affected_packages": [  
    {  
      "name": "libcrypto3",  
      "version": "3.0.9-r1"  
    },  
    {  
      "name": "libssl3",  
      "version": "3.0.9-r1"  
    }  
  ],  
  "remediation": "Upgrade the affected packages."  
},
```

```
{  
  "plugin_id": 400058,  
  "vpr": {  
    "score": 4.4  
  },  
  "cvss_v2": {  
    "base_score": 7.8,  
    "base_vector": "CVSS2#AV:N/AC:L/Au:N/C:N/I:N/A:C"  
  },  
  "cvss_v3": {  
    "base_score": 7.5,  
    "base_vector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H"  
  },  
  "description": "There are packages installed that are affected by a vulnerability referenced in the following CVE:\n\n - A security vulnerability has been identified in all supported versions of OpenSSL related to the\n verification of X.509 certificate chains that include policy constraints. Attackers may be able to exploit\n this vulnerability by creating a malicious certificate chain that triggers exponential use of\n computational resources, leading to a denial-of-service (DoS) attack on affected systems. Policy\n processing is disabled by default but can be enabled by passing the '-policy' argument to the command line\n utilities or by calling the 'X509_VERIFY_PARAM_set1_policies()' function. (CVE-2023-0464)",
```

```
  "family": "Alpine Linux Local Security Checks",  
  "severity": "HIGH",  
  "cve_ids": [  
    "CVE-2023-0464"  
  ],  
  "published_date": "2023-03-22T00:00:00Z",  
  "affected_packages": [  
    {  
      "name": "openssl",  
      "version": "3.0.9-r1"  
    }  
  ],  
  "remediation": "Upgrade the affected packages."  
},
```



```
    {
      "name": "libcrypto3",
      "version": "3.0.9-r1"
    },
    {
      "name": "libssl3",
      "version": "3.0.9-r1"
    }
  ],
  "remediation": "Upgrade the affected packages."
},
{
  "plugin_id": 400069,
  "vpr": {
    "score": 6.7
  },
  "cvss_v2": {
    "base_score": 6.8,
    "base_vector": "CVSS2#AV:L/AC:L/Au:S/C:C/I:C/A:C"
  },
  "cvss_v3": {
    "base_score": 7.8,
    "base_vector": "CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H"
  },
  "description": "There are packages installed that are affected by a vulnerability referenced in the following CVE:\n\n - ncurses before 6.4 20230408, when used by a setuid application, allows local users to trigger security-\n relevant memory corruption via malformed data in a terminfo database file that is found in $HOME/.terminfo\n or reached via the TERMINFO or TERM environment variable. (CVE-2023-29491)",
  "family": "Alpine Linux Local Security Checks",
  "severity": "HIGH",
  "cve_ids": [
    "CVE-2023-29491"
  ],
  "published_date": "2023-04-14T00:00:00Z",
  "affected_packages": [
    {
      "name": "ncurses-terminfo-base",
      "version": "6.3_p20221119-r1"
    },
    {
      "name": "ncurses-libs",
      "version": "6.3_p20221119-r1"
    }
  ],
  "remediation": "Upgrade the affected packages."
},
{
  "plugin_id": 144938,
  "vpr": {
    "score": 0
  },
  "cvss_v2": {
    "base_score": 0,
    "base_vector": ""
  },
  "cvss_v3": {
```



```
    "base_score": 0,
    "base_vector": ""
  },
  "description": "This plugin returns information about a Frictionless Assessment scan. This plugin
is only available to Frictionless Assessment.",
  "family": "Misc.",
  "severity": "NONE",
  "cve_ids": [],
  "published_date": "1970-01-01T00:00:00Z",
  "affected_packages": [
    {
      "name": "",
      "version": ""
    }
  ],
  "remediation": "Upgrade the affected packages."
}
]
```

### What to do next:

On the Tenable Cloud Security Console, go to the **Findings** page. Tenable Cloud Security shows the vulnerabilities detected for the scanned image. For more information, see [View Vulnerabilities](#).



# Integrate Tenable Cloud Security CLI with SCM and CI/CD Pipelines

Tenable Cloud Security integrates with source code management (SCM) and CI/CD pipelines to scan any container image for vulnerabilities and misconfigurations. Tenable recommends using single image scan with the Tenable Cloud Security CLI binary for integrating with SCM and CI/CD pipelines.

Integrate Tenable Cloud Security CLI with the following SCM and CI/CD pipelines:

- [GitHub](#)
- [Jenkins](#)
- [CircleCI](#)
- [Azure DevOps](#)

Before you begin:

- For scanning a container image, ensure that the container image is available in the docker daemon.

## GitHub

The following sample code shows how to integrate a single image scan in a GitHub pipeline:

```
name: consec_tcs_cli_build_and_scan_single_image
on:
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Build Image
        run: |
          echo "building docker image"
          docker build -t getting-started:new .

      - name: TCS CLI Scan
        env:
          TCS_CLI_DOWNLOAD_URL: ${ secrets.TCS_CLI_DOWNLOAD_URL }
          TCS_TOKEN: ${ secrets.TCS_TOKEN }
```



```
TCS_PROJECT_ID: ${vars.TCS_PROJECT_ID}

run: |
  echo ~~~~Installation of TCS CLI..
  wget $TCS_CLI_DOWNLOAD_URL

  file_name=`echo $TCS_CLI_DOWNLOAD_URL | cut -f10 -d "/"`
  tar -xf $file_name && chmod +x tcs

  echo ~~~~Starting TCS CLI Image scan ..
  ./tcs consec image getting-started:new --wait
```

## Jenkins

The following sample code shows integrating a single image scan in Jenkins pipeline:

```
pipeline {
  agent any
  stages {
    stage("Build Image...") {
      steps {
        sh "docker build -t getting-started:new ."
      }
    }
    stage("Install TCS CLI...") {
      steps {
        sh "wget ${TCS_CLI_DOWNLOAD_URL}"
        script{
          def fileName = sh(returnStdout:true, script: 'echo "${TCS_CLI_DOWNLOAD_URL}" | cut -
f10 -d "/"')
          sh "tar -xvf $fileName "
          sh "chmod +x tcs"
        }
      }
    }
    stage("Scanning Image...") {
      steps {
        sh "./tcs consec image getting-started:new --wait"
      }
    }
  }
}
```

## CircleCI

The following sample code shows how to integrate a single image scan in a CircleCI pipeline:

```
version: 2.1

jobs:
  consec-image:
```



```
machine:
  image: ubuntu-2004:202010-01
steps:
  - checkout
  - run:
    name: "Build Image"
    command: "docker build -t getting-started:new ."
  - run:
    name: "Download TCS CLI"
    command: >-
      wget $TCS_CLI_DOWNLOAD_URL &&
      file_name=`echo $TCS_CLI_DOWNLOAD_URL | cut -f10 -d "/"` &&
      tar -xf $file_name && chmod 777 tcs
  - run:
    name: "TCS Image Scan"
    command: "./tcs consec image getting-started:new --wait"

workflows:
  consec-workflow:
    jobs:
      - consec-image
```

## Azure DevOps

The following sample code shows how to integrate a single image scan in an Azure DevOps pipeline:

```
trigger:
  branches:
    include:
      - refs/heads/main
  paths:
    exclude:
      - tcs.yml

variables:
  vmImageName: 'ubuntu-latest'
  TCS_CLI_DOWNLOAD_URL: https://www.tenable.com/downloads/api/v2/pages/tenable-cs/files/tenable.cs-cli_latest_Linux_x86_64.tar.gz
jobs:
  - job: consec_scan
    pool:
      vmImage: $(vmImageName)
    steps:
      - script: |
          echo "building docker image"
          docker build -t getting-started:new .
          displayName: 'Build Image'
      - script: |
          echo "installing TCS CLI"
          wget $(TCS_CLI_DOWNLOAD_URL)
          file_name=`echo $(TCS_CLI_DOWNLOAD_URL) | cut -f10 -d "/"`
          tar -xf $file_name && chmod 777 tcs
          ./tcs version
```



```
echo "initiate TCS scan"  
./tcs consec image getting-started:new --wait -l debug  
displayName: 'TCS Scan'  
continueOnError: true
```

Where:

- **TCS\_CLI\_DOWNLOAD\_URL**: Tenable Cloud Security CLI download location. Use the following links for the download URLs:
  - Linux (arm64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_Linux\\_arm64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_Linux_arm64.tar.gz)
  - Linux (x86\_64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_Linux\\_x86\\_64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_Linux_x86_64.tar.gz)
  - MacOs (arm64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_MacOs\\_arm64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_MacOs_arm64.tar.gz)
  - MacOs (x86\_64): [https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli\\_latest\\_MacOs\\_x86\\_64.tar.gz](https://www.tenable.com/downloads/api/v2/pages/tenable-cs/-files/tenable.cs-cli_latest_MacOs_x86_64.tar.gz)
- **TCS\_PROJECT\_ID**: Project ID in Tenable Cloud Security. Use **TCS\_PROJECT\_ID** to set the project ID with an environment variable.
- **TCS\_TOKEN**: API authentication token you generate from Tenable Cloud Security. Use **TCS\_TOKEN** to set the API token with an environment variable. For more information, see [Generate API Tokens](#).

## Scan a Container Registry

Use the Tenable Cloud Security CLI to scan a container registry for vulnerabilities. After Tenable Cloud Security scans your container registry, you can view the detailed scan results on the Tenable Cloud Security Console. Tenable Cloud Security supports the following registries for scanning:

- Amazon Elastic Container Registry (ECR)
- Docker Hub
- Docker Registry V2





- Nexus
- Harbor
- Quay
- JFrog
- Azure Container Registry

You can run the scan in two ways:

- [Container Registry Scan using CLI](#)
- [Container Registry Scan using Tenable Cloud Security Docker Image](#)



# Container Registry Scan using CLI

For running a container registry scan, you must provide the registry username and password with the `tcs consec` command. This section provides the steps to run a container registry scan using CLI on an Amazon EC2 instance.

Before you begin:

- [Download and install the Tenable Cloud Security CLI.](#)
- [Create](#) a project in the Tenable Cloud Security Console to use for the container registry scan.

**Note:** For accurate results in scan reports and dashboards, Tenable recommends to avoid scanning the same image from multiple projects.

- To use an EC2 machine for registry scanning, do the following:
  - a. Create an EC2 Linux machine.
  - b. Open firewall for Tenable Vulnerability Management, if not open already.
  - c. Check that the EC2 machine has internet access.

To scan a container registry with registry credentials:

1. From the CLI, run the `tcs consec` command in one of the following ways:
  - Without the configuration file

```
tcs consec registry <registry_url> \  
  --username=<registry_username> \  
  --password=<registry_password> \  
  --project=<project_ID> \  
  --token=<API_token> \  
  --allowList=<images_to_scan> \  
  --denyList=<images_to_skip> \  
  --mode=[scan | dry-run]
```



- With the configuration file:

```
tcs consec registry <registry_url> \  
  --username=<registry_username> \  
  --password=<registry_password> \  
  --config=<config_file_path> \  
  --allowList=<images_to_scan> \  
  --denyList=<images_to_skip> \  
  --mode=[scan | dry-run] \  

```

Where:

- `<registry_url>`: URL of the container registry. For example, `http://localhost:5000`.
- `<registry_username>`: Registry username. Use `TCS_REGISTRY_USERNAME` to set the username with an environment variable.
- `<registry_password>`: Registry password. If you do not want to enter the password in plain text, use `TCS_REGISTRY_PASSWORD` to set the password with an environment variable.
- `<project_ID>`: Project ID in Tenable Cloud Security. Use `TCS_PROJECT_ID` to set the project ID with an environment variable.
- `<API_token>`: API authentication token you generate from Tenable Cloud Security. Use `TCS_TOKEN` to set the API token with an environment variable. For more information, see [Generate API Tokens](#).
- `<images_to_scan>`: Specify a comma-separated list of images that you want to scan. You can provide a pattern and only those images that match the pattern are scanned. This parameter supports wildcard characters. For example:
  - `"*"` : Scans all images
  - `"foo:*"` or `"foo"` : Scans images with the repository name as foo with any tag.
  - `"*:bar"`: Scans images with the tag name as bar.



- `"*/foo:bar"` : Scans all repositories with names that end with `foo` and have a tag named `bar`.
- `"foo*/bar:baz"` : Scans all repositories with names that start with `foo`, end with `bar` and have a tag `baz`.
- `"*/foo/*:*"` : Scans all repositories with names that have `foo` in the middle.

**Note:** The CLI supports only complete string patterns, and not substrings.

- `<images_to_skip>`: Specify a comma-separated list of images that you want to skip during a scan. You can provide a pattern and the images that match the pattern are skipped.

**Note:** If you specify both the `--allowlist` and `--denylist` parameters, the `--denylist` parameter takes precedence.

- `--mode`: (Optional) Specifies the mode of the scan. This parameter can take one of the following two values:
  - `scan` – Scans the registry for vulnerabilities. This is the default value.
  - `dry-run` – Creates a CSV report listing all the repositories and tags in the registry. For more information, see [Generate a Report of Images in a Container Registry](#).
- `<config_file_path>`: Specify the configuration file location that you downloaded from Tenable Cloud Security, which contains the project ID and token. This option accepts absolute or relative file paths (defaults to `./config`, then checks `<HOMEDIR>/.ac-curics/config`). For more information, see [Download Configuration File](#).

## Examples



- Docker Hub

```
tcs consec registry https://hub.docker.com --username=<registry_username> --password=<registry_password> --project=<project_ID> --token=<API_token>
```

- Harbor

```
tcs consec registry https://harbor-registry.service.example.com --username=<registry_username> --password=<registry_password> --project=<project_ID> --token=<API_token>
```

- Nexus

```
tcs consec registry https://nexus.example.com:8483 --username=<registry_username> --password=<registry_password> --project=<project_ID> --token=<API_token>
```

After Tenable Cloud Security completes the registry scan, the CLI output shows a summary with number of images discovered, images newly added after previous scan with links to misconfigurations (violations) and vulnerabilities on the Tenable Cloud Security Console.

```
Registry Summary:  
  Total images found      : 1  
  New images found       : 1  
  
  Violation details       : https://cloud.tenable.com/cns/issues/violations?project=<project_id>  
  Vulnerability details   : https://cloud.tenable.com/cns/issues/vulnerabilities?project=<project_id>
```



## Scan an Amazon Elastic Container Registry (ECR)

For an Amazon ECR, you can run a registry scan with the AWS ECR access keys instead of providing the registry username and password.

Before you begin:

- If you are using an EC2 machine for scanning, add the AmazonEC2ContainerRegistryReadOnly policy to the IAM role used by the Amazon ECR instance.

To scan an Amazon ECR with ECR security credentials:

1. Set up the environment variables for connecting to the Amazon ECR:

```
export AWS_ACCESS_KEY_ID=<key_id>
```

```
export AWS_SECRET_ACCESS_KEY=<access_key>
```

```
export AWS_DEFAULT_REGION=<region>
```

**Note:** You need not export the `AWS_ACCESS_KEY_ID` or `AWS_SECRET_ACCESS_KEY` when running in an EC2 instance with the `AmazonEC2ContainerRegistryReadOnly` policy attached.

2. Run the container registry scan with the following command:

```
tcs consec registry <registry_url> --project=<project_ID> --token=<API_token>
```

Where:

- `<registry_url>`: URL of the container registry. For example, `https://<aws_account_id>.dkr.ecr.<region>.amazonaws.com`.
- `<project_ID>`: Project ID in Tenable Cloud Security. Use `TCS_PROJECT_ID` to set the project ID with an environment variable.



- <API\_token>: API authentication token you generate from Tenable Cloud Security. Use **TCS\_TOKEN** to set the API token with an environment variable. For more information, see [Generate API Tokens](#).



---

# Scan a Quay Container Registry

---

Before you begin:

- Set up a robot account with read permissions to the registries that you want to scan.

Use the credentials of this robot account for authenticating and scanning the Quay registry.

For more information, see [Robot Accounts](#) in Red Hat Quay.io documentation.

To scan a Quay registry:

1. Scan the container registry with the `tcs consec registry` command.

```
tcs consec registry https://quay.io --username=<Quay_username> --password=<Quay_password> --project=<project_id> --token=<API_token>
```

Where:

- `<Quay_username>`: Username of the robot account
- `<Quay_password>`: Robot token





---

## Scan a JFrog Container Registry

---

Before you begin:

- Create an access token for JFrog container registry.

For more information, see [Access Tokens](#) in JFrog Platform Administration Documentation.

To scan a JFrog container registry:

1. Scan the container registry with the `tcs consec` command.

```
tcs consec registry https://test.jfrog.io/docker --username=<JFrog_username> --password=<JFrog_Password> --project=<project_id> --token=<API_token>
```

**Note:** The registry URL format is `<jfrog_registry>/docker`.

Where:

- `<JFrog_username>`: JFrog username
- `<JFrog_password>`: JFrog access token

---

## Scan an Azure Container Registry

---

You can scan an Azure Container Registry either using a [service principal](#) or a [managed entity](#).

### Scan an Azure Container Registry Using Service Principal

Before you begin:

- Create a service principal for your Azure registry and assign the **AcrPull** role to the service principal. For more information, see [Azure Container Registry roles and permissions](#) in Azure documentation.

The following JSON shows the permissions for the **AcrPull** role.



```
{
  "id": "/providers/Microsoft.Authorization/roleDefinitions/<ROLE_DEFINITION_ID>",
  "properties": {
    "roleName": "AcrPull",
    "description": "acr pull",
    "assignableScopes": [
      "/"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.ContainerRegistry/registries/pull/read"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```

To scan an Azure Container Registry using service principal:

1. Scan the container registry with the `tcs consec` command.

```
./tcs consec registry https://<REGISTRY_NAME>.azurecr.io \
--project "<PROJECT_ID>" \
--token "<TCS_TOKEN>" \
--username ${USER_NAME} \
--password ${PASSWORD}
```

Where:

- `<USER_NAME>`: Azure Service Principal username
- `<PASSWORD>`: Azure Service Principal password

## Scan an Azure Container Registry Using Managed Entity

Before you begin:

- Create a managed identity for your Azure registry and assign the [AcrPull](#) role to the managed identity. For more information, see [Use an Azure managed identity to authenticate to an Azure container registry](#) in Azure documentation.

To scan an Azure Container Registry using managed entity:



1. Assign the managed identity to an Azure virtual machine or authenticate the Azure CLI with the managed identity.
2. Scan the container registry with the `tcs consec` command without username or password.

```
./tcs consec registry https://<REGISTRY NAME>.azurecr.io \  
--project "<PROJECT_ID>" \  
--token "<TCS_TOKEN>"
```



# Generate a Report of Images in a Container Registry

To list the repositories and tags in a container registry and generate a CSV report:

1. Scan the container registry with the `tcs consec` command.

```
./tcs consec registry https://<REGISTRY_NAME>.azurecr.io \  
--project <PROJECT_ID> \  
--token <TOKEN> \  
--username <USERNAME> \  
--password <PASSWORD> \  
--mode=dry-run
```

The following sample shows the console output :

```
2023-08-15T13:01:57.950+0200 info Identified registry as: DEFAULT_V2  
2023-08-15T13:01:57.951+0200 info Beginning discovery of registry: <https://test.azurecr.io>  
2023-08-15T13:01:57.952+0200 info Beginning image discovery of registry registry-  
y=<https://test.azurecr.io>  
2023-08-15T13:02:00.244+0200 info Fetched image discovery details image=test.azurecr.io/hello-  
world:v1  
2023-08-15T13:02:04.991+0200 info Fetched image discovery details image=test.azurecr.io/hello-  
world:v2  
2023-08-15T13:02:05.672+0200 info Fetched image discovery details image=test.azurecr.io/hello-  
world:latest  
2023-08-15T13:02:07.860+0200 info Completed fetching image discoveries from registry-  
y=<https://test.azurecr.io>  
2023-08-15T13:02:07.860+0200 info Output report: test-azurecr-io-report.csv
```

The **Output report** line in the console output shows the name of the CSV report. The CSV file contains the following information:

- `repository` – The repository and image name in the format `<repository_name>/<image_name>`.
- `tag` – Image tag.
- `build_time` – Build timestamp of the image.
- `skipped` – Shows the status as `false` or `true` to indicate whether the image will be skipped during a vulnerability scan. Images are skipped based on the `--allowlist` or `--denylist` parameter as well as licensing limits.

The following example shows the content of a CSV file:



```
# test-azurecr-io-report.csv
repository,tag,build_time,skipped
test.azurecr.io/hello-world,v1,2019-01-01T01:29:27Z,false
test.azurecr.io/hello-world,v2,2019-01-01T01:29:27Z,false
test.azurecr.io/example/hello-world,latest,2019-01-01T01:29:27Z,true
```



# Scan a Container Registry using Tenable Cloud Security Docker Image

To scan a container registry using the Tenable Cloud Security Docker image:

1. Verify that Docker Hub is accessible.
2. Pull the latest Tenable Cloud Security Tenable Cloud Security CLI image from Docker.

The location of the image is <https://hub.docker.com/r/tenable/tcs>.

3. Scan the container registry using the Tenable Cloud Security Docker image:

```
docker run --rm -t -u root -v /var/run/docker.sock:/var/run/docker.sock -v <report_file_directory> tenable/tcs:latest consec registry <registry_url> --project=<project_ID> --token=<API_token> --username=<registry_username> --password=<registry_password>
```

Where:

- `<report_file_directory>`: Directory to save the Tenable Cloud Security scan reports.
- `<registry_url>`: URL of the container registry. For example, `http://localhost:5000`.
- `<registry_username>`: Registry username. Use `TCS_REGISTRY_USERNAME` to set the username with an environment variable.
- `<registry_password>`: Registry password. If you do not want to enter the password in plain text, use `TCS_REGISTRY_PASSWORD` to set the password with an environment variable.
- `<project_ID>`: Project ID in Tenable Cloud Security. Use `TCS_PROJECT_ID` to set the project ID with an environment variable.
- `<API_token>`: API authentication token you generate from Tenable Cloud Security. Use `TCS_TOKEN` to set the API token with an environment variable. For more information, see [Generate API Tokens](#).

**Note:** Container registry scan might take a long time to complete. The duration to complete a registry scan depends on the number of images in the repository.

What to do next:



---

Go to the link in the CLI output to view the misconfigurations and vulnerabilities on the **Findings** page. Tenable Cloud Security shows the vulnerabilities detected for the scanned container. For more information, see [View Vulnerabilities](#).

## Tenable Cloud Security Container Security Commands and Options

---

This section lists the commands and options to use with the `tcs` command.



## Commands

Command	Description
<code>tcs consec</code>	Scan a container image or registry for vulnerabilities and misconfigurations.
<code>tcs env</code>	Display the Tenable Cloud Security CLI environment variables.
<code>tcs version</code>	Display the Tenable Cloud Security CLI version.

**Note:** Use the `tcs [command] --help` for more information about a command.





## Global Scan Options for Image and Registry Scans (`tcs consec` command)

Use the following options with the `tcs consec` command for both container image and registry scans:

Option	Description	Required/Optional
<code>-c</code> or <code>--config=&lt;configfile_path&gt;</code>	The location of the configuration file that you downloaded. This option accepts absolute or relative file paths (defaults to <code>./config</code> , then checks <code>&lt;HOMEDIR&gt;/.accurics/config</code> ). <a href="#">Download Configuration File</a>	Required if you do not specify the project ID and API token.
<code>--token=&lt;API_token&gt;</code>	The API authentication token. Use <code>TCS_TOKEN</code> to pass the token using an environment variable	Optional if you specify the configuration file.
<code>-p=&lt;project_ID&gt;</code> or <code>--project=&lt;project_ID&gt;</code>	The project in Tenable Cloud Security. Use <code>TCS_PROJECT_ID</code> to set the project ID with an environment variable.	Required
<code>--fail</code>	Returns exit code 1 when Tenable Cloud Security detects high severity violations.	Optional
<code>-l</code> or <code>--log-level</code>	Specify one of the following log levels to show in the CLI output: <ul style="list-style-type: none"><li>• debug</li><li>• info</li><li>• warn</li><li>• error</li></ul>	Optional



	<ul style="list-style-type: none"><li>• panic</li><li>• fatal</li></ul> <p>The default value is <b>info</b>.</p>	
-x or --log-type	<p>Specify one of the following log output type:</p> <ul style="list-style-type: none"><li>• console</li><li>• json</li></ul> <p>The default value is <b>console</b>.</p>	Optional
--log-dir <directory_name>	<p>Specify a directory for the logs other than the default directory when running the scan in debug mode.</p> <div style="border: 1px solid blue; padding: 5px;"><p><b>Note:</b> Tenable Cloud Security generates a log file if the scan is run in debug mode (<code>--log-level=debug</code>). By default, the log directory is <code>\${pwd}/log</code>.</p></div>	Optional



## Scan Options for Container Images (tcs consec image command)

The following command syntax shows how to run a scan for container images without a configuration file:

```
tcs consec image <image_name>:<tag> --token=<API_token> --project=<project_id> [--wait] [--retryInterval <interval>] [--timeout <seconds>]
```

The following command syntax shows how to run a scan for container images with a configuration file:

```
tcs consec image <image_name>:<tag> --config=<config_file_path> [--wait] [--retryInterval <interval>] [--timeout <seconds>]
```

Option	Description	Required/Optional
<image_name>:<tag>	Image name with its tag. For example, alpine:latest.	Required
--wait	If you specify this option, Tenable Cloud Security waits for the duration specified with the --timeout parameter for the scan to complete. If the scan completes within the specified duration, Tenable Cloud Security generates two types of CLI outputs: <ul style="list-style-type: none"><li>• <b>Scan summary on the console:</b> Includes the summary of total misconfigurations (violations) and total vulnerabilities.</li><li>• <b>JSON report:</b> Detailed scan report that indicates the details about the misconfigurations and vulnerabilities.</li></ul>	Optional



	<p>For more information about these CLI outputs, see <a href="#">CLI Outputs for Container Image Scans</a>.</p> <div style="border: 1px solid blue; padding: 5px;"><p><b>Note:</b> If the <code>--wait</code> option is not specified with the <code>tcs consec image</code> command, the console summary and JSON report are not generated.</p></div>	
<code>--timeout &lt;timeout_sec&gt;</code>	<p>The maximum time (in seconds) to wait for the violation report of the scan. The default value is 300 seconds (5 minutes). To change the default, use this option with the <code>--wait</code> option.</p>	Optional
<code>--retryInterval &lt;poll_interval&gt;</code>	<p>The polling time interval (in seconds) while polling for the violation report of the scan. The default value is 5 seconds. Tenable Cloud Security checks whether the violation report is ready after every polling interval.</p>	Optional



## Scan Options for Container Registries (tcs consec registry command)

The following command syntax shows how to run the `tcs consec` command for scanning container registries without a configuration file:

```
tcs consec registry <registry_url> --username=<registry_username> --password=<registry_password> --project=<project ID> --token=<API_token> --allowList=<images_to_scan> --denyList=<images_to_skip> [--builtAfter=<duration> | --builtBefore=<duration>] --mode=[scan | dry-run]
```

The following command syntax shows how to run the `tcs consec` command for scanning container registries with a configuration file:

```
tcs consec registry <registry_url> --username=<registry_username> --password=<registry_password> --allowList=<list_of_allowed_images> --denyList=<list_of_skipped_images> [--builtAfter=<duration> | --builtBefore=<duration>] --config=<config_file_path>
```

Option	Description	Required/Optional
<registry_url>	URL of the container registry. For example, <code>http://localhost:5000</code> .	Required
--username=<registry_username>	Container registry username. Use <b>TCS_REGISTRY_USERNAME</b> to set username with an environment variable.	Required
--password=<registry_password>	Container registry password. If you do not want to enter the password in plain text, use <b>TCS_REGISTRY_PASSWORD</b> to set the password with an environment variable.	Required
--allowList=<images_to_scan>	Specify a comma-separated list of images that you want to scan.	Optional



	<p>You can provide a pattern and only those images that match the pattern are scanned. This parameter supports wildcard characters. For examples, see <a href="#">Scan a Container Registry</a>.</p>	
<code>--denyList=&lt;images_to_skip&gt;</code>	<p>Specify a comma-separated list of images that you want to skip during a scan. You can provide a pattern and the images that match the pattern are skipped.</p> <div style="border: 1px solid blue; padding: 5px;"><p><b>Note:</b> If you specify both the <code>--allowlist</code> and <code>--denylist</code> parameters, the <code>--denylist</code> parameter takes precedence.</p></div>	Optional
<code>--builtAfter=&lt;duration&gt;</code>	<p>Scans only images that are built after the specified duration. Any images built before this duration are not considered for the scan. Specify the duration as d (day), w (week), m (month), or y year). For example, 1d, 2w, 3m, or 4y.</p> <div style="border: 1px solid blue; padding: 5px;"><p><b>Note:</b> Specify only one value for the duration parameter. You cannot use a combination of values.</p></div>	Optional
<code>--builtBefore=&lt;duration&gt;</code>	<p>Scans only images that are built before the specified duration. Any images built after this duration are not considered for the scan. Specify the duration as d (day), w (week), m month), or y (year).</p>	Optional



<code>--mode=[scan   dry-run]</code>	<p>Specifies the mode of the scan. This parameter can take one of the following two values:</p> <ul style="list-style-type: none"><li>• <code>scan</code> – Scans the registry for vulnerabilities. This is the default value.</li><li>• <code>dry-run</code> – Creates a CSV report listing all the repositories and tags in the registry. For more information, see <a href="#">Generate a Report of Images in a Container Registry</a>.</li></ul>	Optional
--------------------------------------	--	----------



## Scan with Environment Variables

Use the `tcs env` command to view the environment variables.

Option	Description
TCS_PROJECT_ID	The project ID in Tenable Cloud Security.
TCS_TOKEN	The API authentication token.
TCS_REGISTRY_USERNAME	The container registry username.
TCS_REGISTRY_PASSWORD	The container registry password.
HTTP_PROXY	HTTP proxy for all communications with the Tenable Cloud Security CLI.
HTTPS_PROXY	HTTPS proxy for all communications with the Tenable Cloud Security CLI.
NO_PROXY	<p>List of domains that do not need to go through the HTTPS_PROXY or HTTP_PROXY.</p> <p><b>Example</b></p> <p>If you have a local registry and need proxy for Tenable Cloud Security, set up the following environment variables:</p> <ul style="list-style-type: none"><li>• HTTPS_PROXY – Proxy to communicate with Tenable</li><li>• NO_PROXY – List with the registry domain (to skip proxy)</li></ul>





---

## Script Options

---

Use the `tcs completion` command to generate the autocompletion script for the following shells:

Option	Description
<code>bash</code>	Generate a Bash script.
<code>fish</code>	Generate a fish shell script.
<code>powershell</code>	Generate a PowerShell script.
<code>zsh</code>	Generate a ZSH shell script.



# View the Containers Dashboard

The Tenable Cloud Security **Containers** dashboard shows the vulnerabilities detected during a container image and registry scan.

To view the **Containers** dashboard:

1. [Access Tenable Cloud Security.](#)

The **Dashboards** page appears. The **Misconfigurations** tab is selected by default.

2. Click the **Containers** tab.

The **Containers** dashboard appears with several widgets showing key insights about the vulnerabilities detected in the container images and container registry scans.

3. Click a widget to view more details on the [Vulnerabilities](#) page.

The following table describes the widgets on the **Containers** dashboard:

Widget	Description
Key Insights	<p>Provides a quick overview of the following metrics:</p> <ul style="list-style-type: none"><li>• Total public Kubernetes clusters</li><li>• Total private Kubernetes clusters</li></ul> <div style="border: 1px solid #0070C0; padding: 5px;"><p><b>Note:</b> Public clusters added through the CLI scan are listed as private clusters. If a cloud scan is run on such a cluster, it switches over as a public cluster. After that, the cluster remains as a public cluster irrespective of how the scan is run.</p></div> <ul style="list-style-type: none"><li>• Total number of registries scanned</li><li>• Total number of images scanned</li><li>• Number of images with critical vulnerabilities</li></ul>
<b>Kubernetes (K8s) Summary</b>	
K8s environment summary	<p>Provides an overview of your Kubernetes environment:</p> <ul style="list-style-type: none"><li>• Total namespaces</li></ul>



	<ul style="list-style-type: none"><li>• Total deployments</li><li>• Total services</li><li>• Total pods</li><li>• Total jobs</li></ul> <p>By default, the data is shown for all clusters. Use the <b>All clusters</b> filter drop-down to select a specific cluster.</p>
<b>K8s misconfiguration summary</b>	Provides a summary of misconfigurations for Kubernetes clusters organized by severity.
<b>Top 5 mis-configurations by impacted resources</b>	Lists the top five misconfigurations along with the number of impacted resources.
<b>Image Summary</b>	
<b>Top 5 image tags by VPR</b>	Lists the top five image tags, organized by Vulnerability Priority Rating (VPR).
<b>Top 5 images by vulnerability</b>	Lists the top five images organized by the number of vulnerabilities.



---

## Configure CI/CD Integrations

---

Tenable Cloud Security can integrate with your CI/CD provider to scan your IaC files for violations in your build pipeline and fail the builds if Tenable Cloud Security finds severe vulnerabilities in the code. By integrating with your CI/CD provider, Tenable Cloud Security helps you track violations or drifts, and set up alerts and escalations in your applications.

For some CI/CD integrations, you must use the Tenable Cloud Security CLI to scan code in your CI/CD pipeline. After installing Tenable Cloud Security CLI on the build machine, you must add the necessary instructions to the pipeline script to run the tool against the files present in the repository.

Tenable Cloud Security supports integrating with following CI/CD applications or components:

- [Integrate with Terraform Cloud](#)
- [Integrate with Jenkins Pipeline](#)
- [Integrate with GitHub Action](#)
- [Integrate with Azure DevOps Pipeline](#)
- [Set Up Policy Guardrails \(CI/CD\)](#)



# Generate API Tokens

You can generate API Tokens, also known as bearer tokens, to authenticate any application with Tenable Cloud Security.

To generate API tokens:



1. In the left navigation bar, click **Integrations**.

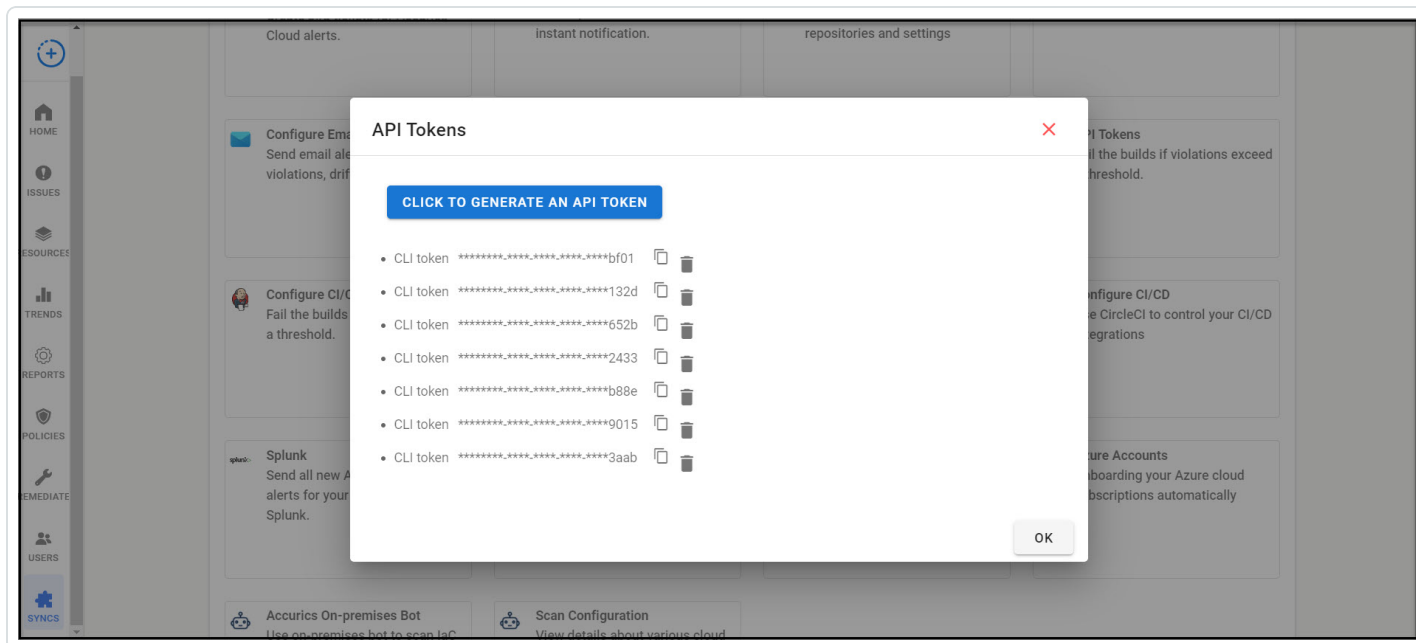
The **Integrations** window appears.

2. Click **API Tokens**.

The **API Tokens** window appears.

3. Do one of the following:

- To generate a new API token:
  - a. Click **Click to generate an API token**.
  - b. Click the  button to copy the corresponding API token.
- To copy an existing API token, click the  button to copy the API token.



4. Click **OK** to close the **API Tokens** window.



You can use this token to authenticate applications or integrate Tenable Cloud Security with different repositories.



## Integrate with Terraform Cloud

You can integrate Tenable Cloud Security with Terraform Cloud to scan your Terraform IaC files. For this integration, you must create a Terraform *Run Task* for Tenable Cloud Security in Terraform Cloud. A Terraform run task for Tenable Cloud Security allows you to scan your workspace within a Terraform run, specifically between the `plan` and `apply` stages of the Terraform Cloud workflow.

**Note:** Tenable Cloud Security supports only Terraform Cloud workspaces that are linked to a version control system (VCS) repository.

In Terraform Cloud, you must first create a run task in the settings of your organization by providing the Tenable Cloud Security URL as the endpoint. Then, you must add this run task to the required Terraform workspaces. When the Terraform Cloud workflow triggers the run task, Tenable Cloud Security scans and returns a passed or failed response back to Terraform Cloud. The status response along with the enforcement setting of the run task determine whether a Terraform run proceeds to the next stage of the workflow. For more information about run tasks, see [Run Tasks](#) in the Terraform documentation.

**Note:** If there is no Terraform Cloud repository onboarded in Tenable Cloud Security when you create run task in Terraform Cloud, Tenable Cloud Security creates a default project automatically for the Terraform Cloud repository.

Before you begin:

- Ensure the Terraform workspace uses Terraform version 0.12 or later.
- Ensure you have the correct permissions within Terraform:
  - To create a run task, you must have a user account with organization owner permissions.
  - To associate run tasks to a workspace, you must be at least a workspace administrator.

For more information, see [Permissions](#) in Terraform documentation.

To integrate Terraform Cloud with Tenable Cloud Security:

1. In the integrations list, click **Terraform Cloud**.

The **Terraform cloud** page appears.



## Terraform cloud

- **Configure run task**

1. On terraform cloud navigate to the settings page of the organization you wish to integrate with tenable.cs.
2. Navigate to the "Run Task" page and click on "Create run task"
3. Enter "tenable\_cs" as the name for the "Run Task"
4. Copy the values below for the Endpoint URL and HMAC

### Endpoint URL

`https://cloud.tenable.com/cns/external/api/application/trigger/terra`

Copy

### HMAC

`11a62a5a-2f18-4f74-9a98-7ae4d281cb83`

Copy

5. Enable the tenable.cs "Run Task" by navigating to the desired workspaces within terraform cloud.

Close

**Tip:** You can copy the **Endpoint URL** and **HMAC key** values from this page when configuring the run task in Terraform Cloud.

2. Log in to [Terraform Cloud](#).
3. In the Terraform Cloud user interface, navigate to the workspace that you want to integrate with Tenable Cloud Security.
4. [Create a run task](#) to scan the Terraform cloud using Tenable Cloud Security by specifying the following options:

Option	Description
Enabled	This option when selected triggers the run task across all associated workspaces. This option is enabled by default for new run tasks.
Name	The name of the run task. Tenable recommends





Option	Description
	entering <b>tenable_cs</b> as the name of the run task for easy identification.
<b>Endpoint URL</b>	The Tenable Cloud Security URL. You can copy the URL from the <b>Terraform cloud</b> page in Tenable Cloud Security.
<b>HMAC key</b>	A secret key that Tenable Cloud Security uses to authenticate the request. You can copy the HMAC key from the <b>Terraform cloud</b> page in Tenable Cloud Security.

For more information, see [Creating a Run Task](#) in the Terraform documentation.

5. [Add the run task](#) created in the previous step to the required workspaces in the Terraform Cloud.
  - a. When adding a run task to a workspace, select the **Enforcement Level**. Enforcement levels control how a run task behaves in a Terraform run. The following enforcement levels are available:
    - **Advisory** – Does not interrupt the run, and only informs about the failure of the run task.
    - **Mandatory** – Requires that the run task passes for the run to continue. If a run task fails, the run halts and cannot be applied until you resolve the failure.

For more information, see [Adding Run Tasks to a Workspace](#) in the Terraform documentation.

Terraform executes the run task after the `plan` stage during a Terraform run.

## Examples

The following example shows a run task with **Mandatory** enforcement level. The Terraform run fails because of the scan violations.



**Errored Demo** CURRENT

triggered a run from UI a few seconds ago Run Details

**Plan finished** a few seconds ago Resources: 2 to add, 0 to change, 0 to destroy

**Tasks failed** a few seconds ago Tasks: 0 passed, 1 failed (mandatory) **Beta**

Running a few seconds ago > Failed a few seconds ago

> **tenable\_cs** failed (mandatory) Details

Apply will not run

The following example shows a run task with the **Advisory** enforcement level. Although there are violations reported in the scan, the run does not fail.

Overview **Runs** States Variables Settings Running Actions

**Needs Confirmation Test11** CURRENT

triggered a run from UI a few seconds ago Run Details

**Plan finished** 18 minutes ago Resources: 42 to add, 0 to change, 0 to destroy

**Tasks passed** 16 minutes ago Tasks: 0 passed, 1 failed (advisory) **Beta**

Running 5 minutes ago > Passed 2 minutes ago

> **tenable\_cs** failed (advisory) Details

tenable.cs scan completed and found 50 violations across 42 resources classified as: 19 high, 29 medium, 15 low

Apply pending

**Note:** Click the **Details** link to view the scan summary and results in Tenable Cloud Security.



# Integrate with Jenkins Pipeline

Tenable Cloud Security integrates with Jenkins and scans your Jenkins pipeline for violations.

Before you begin:

- Ensure you have access to working Jenkins instance.
- Ensure you have a repository to scan.

To connect to Jenkins:

1. From the root folder of the repository, open the groovy file that hosts the pipeline, usually named `jenkins-pipeline.groovy`.
2. Add the following block of statements in the groovy file under the steps before it starts deploying the infrastructure.

For scanning a repository:

```
sh 'echo downloading Tenable CS CLI'
sh 'wget https://downloads.accurics.com/cli/latest/accurics_linux -O tcs-iac-scanner'
sh 'chmod +x tcs-iac-scanner'
sh './tcs-iac-scanner init'
sh './tcs-iac-scanner scan -mode=pipeline -appurl=https://cloud.tenable.com/cns -token=<tcs_
api_token> -fail -project=<project_id>'
```

Where:

- `API_token`: API authentication token you generate from Tenable Cloud Security. For more information, see [Generate API Tokens](#).
- `-fail`: (Optional) Specify this parameter to fail the pipeline if Tenable Cloud Security finds a High severity policy failure.
- `project_ID`: (Optional) Project in Tenable Cloud Security. If you specify the project, Tenable Cloud Security sends the scan results to this project. If you do not specify the project, Tenable Cloud Security creates a default project for displaying the scan results.



# Integrate with GitHub Action

The **Accurics GitHub Action** integration scans the IaC files in your repository and can fail a pipeline build when it finds violations or errors. You can view the scan results in the pipeline results or in Tenable Cloud Security.

Before you begin:

- [Download the configuration file](#) for your repository from Tenable Cloud Security.
- Create GitHub secrets to store the Environment ID and Application Token.
  - a. Navigate to your repository and click **Settings** under your repository name.
  - b. In the left navigation bar, click **Secrets > New Repository Secret**.
  - c. Create the following two secrets:
    - **ACCURICS\_APP\_ID**: Provide the value of the app parameter in the configuration file you downloaded from Tenable Cloud Security.
    - **ACCURICS\_ENV\_ID**: Provide the value of the env parameter in the configuration file.

To set up **Accurics GitHub Action**:

1. On the [GitHub Marketplace](#), In the search box, type **Accurics**.
2. In the search results, click **Accurics GitHub Action**.

The [Accurics GitHub Actions](#) page appears.

3. Copy and paste the following code to the `action.yml` file to set up the latest version of **Accurics GitHub Action**:

```
steps:
  - name: Accurics
    uses: accurics/accurics-action@v2.0.2
    id: accurics
    with:
      app-id: ${ secrets.ACCURICS_APP_ID }
      env-id: ${ secrets.ACCURICS_ENV_ID }
      repo: "<URL of the repository>"
```



**Note:** (Optional) You can specify input parameters to customize your scan. For more information about parameters that you can specify, see [Input Parameters for GitHub Action](#).

## Input Parameters for GitHub Action

Specify the following required and optional parameters to customize the scan and view results from GitHub Action:

Name	Description	Required/Optional	Default Value
app-id	The application token ID.	Required	
env-id	The environment ID.	Required	
repo	The repository location URL.	Required	
terraform-version	The Terraform version used to process the files in this repository.	Optional	latest
plan-args	The Terraform variables along with other required command-line parameters when running terraform plan.	Optional	
directories	A directory to scan within this repository.	Optional	./
fail-on-violations	When true, Tenable Cloud Security fails the build if violations are found.	Optional	true
fail-on-all-errors	When true, Tenable Cloud Security fails the build if it encounters any errors.	Optional	true
scan-mode	Specify the scan mode to	Optional	plan



	either Terraform (plan) or Terrascan (scan) for scanning.		
url	The URL of the target endpoint. For example, <a href="https://cloud.tenable.com/cns">https://cloud.tenable.com/cns</a>	Optional	<a href="https://app accurics.com">https://app accurics.com</a>
pipeline	Set this parameter to true if the mode is pipeline.	Optional	

## AWS Parameters for Terraform Plan-Based Scan

Specify the following environment parameters under the env section for your AWS Terraform files.

Name	Description	Required/Optional
AWS_ACCESS_KEY_ID	An AWS access key associated with the IAM user or role.	Required
AWS_SECRET_ACCESS_KEY	The secret key associated with the access key. This is essentially the "password" for the access key.	Required
REPO_URL	The GitHub repository location URL.	Required
GIT_BRANCH	The name of the current GitHub branch.	Required
GIT_COMMIT	The GitHub commit ID (SHA).	Required
TF_CLI_CONFIG_FILE	Name of the file that contains the API token of the Terraform Cloud in the following format:	Required if repository uses modules from Terraform Cloud.



```
credentials "app.terraform.io" {  
  # valid user API token:  
  token = "API Token from Terraform Cloud site"}
```

**Note:** This file must be in the GitHub repository.

## Output Parameters for GitHub Action

Specify the following output parameters to customize your scan results.

Parameter	Description
<code>\$env_name</code>	Environment name
<code>\$num_violations</code>	Violation count
<code>\$num_resources</code>	Resource count
<code>\$high</code>	High-severity violations
<code>\$medium</code>	Medium-severity violations
<code>\$low</code>	Low-severity violations
<code>\$native</code>	Native resources
<code>\$has_errors</code>	Scan has errors

## Example

The following example shows an IaC scan configuration using the latest Terraform version, custom variables, and output scan status:

```
steps:  
  - name: Checkout  
    uses: actions/checkout@v2  
  - name: Accurics  
    uses: accurics/accurics-action@v2.0.2  
    id: accurics  
  env:  
    # Required by Terraform  
    AWS_ACCESS_KEY_ID: ${ secrets.AWS_ACCESS_KEY_ID }
```



```
AWS_SECRET_ACCESS_KEY: ${ secrets.AWS_SECRET_ACCESS_KEY }}
REPO_URL: ${ github.repositoryUrl }}
GIT_BRANCH: ${ github.ref_name }}
GIT_COMMIT: ${ github.sha }}
TF_CLI_CONFIG_FILE: fileNameWithHostAndToken
with:
  # Required by Accurics
  app-id: ${ secrets.ACCURICS_APP_ID }}
  env-id: ${ secrets.ACCURICS_ENV_ID }}
  repo: "https://bitbucket.org/myrepo/reponame.git"
  # Optional args
  plan-args: '-var myvar1=val1 -var myvar2=val2'
  fail-on-violations: true
  url: "https://cloud.tenable.com/cns"
  scan-mode: "scan"
  pipeline: true
- name: Display statistics
  run: '
    echo ""
    echo "Environment Name           : ${ steps accurics.outputs.env-name }}";
    echo "Repository                   : ${ steps accurics.outputs.repo }}";
    echo "Violation Count                 : ${ steps accurics.outputs.num-violations }}";
    echo "Resource Count                  : ${ steps accurics.outputs.num-resources }}";
    echo ""
    echo "Native Resources                : ${ steps accurics.outputs.native }}";
    echo "Inherited Resources             : ${ steps accurics.outputs.inherited }}";
    echo ""
    echo "High-Severity Violations        : ${ steps accurics.outputs.high }}";
    echo "Medium-Severity Violations      : ${ steps accurics.outputs.medium }}";
    echo "Low-Severity Violations         : ${ steps accurics.outputs.low }}";
    echo ""
    echo "Drift                           : ${ steps accurics.outputs.drift }}";
    echo "IaC Drift                       : ${ steps accurics.outputs.iacdriфт }}";
    echo "Cloud Drift                     : ${ steps accurics.outputs.cloudriфт }}";
    echo ""
  '
```

For more examples, see [Accurics GitHub Action](#) in the GitHub marketplace.





# Integrate with Azure DevOps Pipeline

You can integrate an Azure DevOps pipeline with Tenable Cloud Security to scan for violations and to break the pipeline if Tenable Cloud Security finds high severity violations in the code.

Before you begin:

Configure the following:

- **Azure Account** to host the infrastructure provisioned by IaC
- **Azure DevOps Organization and Project** to host one or more Azure DevOps pipelines and IaC repositories
- [Azure DevOps Pipeline](#)

To integrate Azure DevOps Pipeline with Tenable Cloud Security:

1. Log in to the Azure DevOps console.
2. Open the Azure DevOps project and the IaC repository that you want to integrate with Tenable Cloud Security.
3. From the root folder of the repository, open the YAML file for the pipeline, usually named `azure-pipelines.yml`, and add the following block of code in the `script` block under the `steps` parameter.

With the `plan` command:

```
script: |
  export ARM_SUBSCRIPTION_ID=$(azSubID)
  export ARM_TENANT_ID=$(azTenantID)
  export ARM_CLIENT_ID=$(azClientID)
  export ARM_CLIENT_SECRET=$(azClientSecret)
  # Download Tenable Cloud Security CLI
  wget https://downloads.accurics.com/cli/latest/accurics_linux -O tcs-iac-scanner
  chmod +x tcs-iac-scanner
  ./tcs-iac-scanner init
  ./tcs-iac-scanner plan -mode=pipeline -appurl=https://cloud.tenable.com/cns -token=<tcs_api_token> -fail -project=<project_id>
```

With the `scan` command:



```
script: |
  export ARM_SUBSCRIPTION_ID=$(azSubID)
  export ARM_TENANT_ID=$(azTenantID)
  export ARM_CLIENT_ID=$(azClientID)
  export ARM_CLIENT_SECRET=$(azClientSecret)
  # Download Tenable CS CLI
  wget https://downloads.accurics.com/cli/latest/accurics_linux -O tcs-iac-scanner
  chmod +x tcs-iac-scanner
  ./tcs-iac-scanner init
  ./tcs-iac-scanner scan -mode=pipeline -appurl=https://cloud.tenable.com/cns -token=<tcs_api_token> -fail -project=<project_id>
```

where:

- `tcs_api_token`: Specify the API token to authenticate with Tenable Cloud Security. For more information, see [Generate API Tokens](#).
- `-fail`: (Optional) Specify this parameter to fail the pipeline if Tenable Cloud Security finds a **High** severity policy failure.
- `project_id`: (Optional) Specify the Tenable Cloud Security project ID to which you want to add the Azure DevOps pipeline repository.

**Note:** If you do not specify the project, Tenable Cloud Security creates a default project called **DEFAULT\_AZURE** for the repository.

**Note:** To use the CLI in the plan mode, ensure the required Azure credentials are available.

## Example

```
trigger:
  - main

pool:
  vmImage: ubuntu-latest

variables:
  - name: azSubID
    value: 5XXXXXXXX-XXX4-1XXX-XXX6-9XXXXXXXXXXXXX
  - name: azTenantID
    value: 5XXXXXXXX-XXX4-1XXX-XXX6-9XXXXXXXXXXXXX
  - name: azClientID
    value: 5XXXXXXXX-XXX4-1XXX-XXX6-9XXXXXXXXXXXXX
  - name: azClientSecret
    value: 5XXXXXXXX-XXX4-1XXX-XXX6-9XXXXXXXXXXXXX
  - name: tcsCLIVersion
    value: latest
```



```
- name: tfVersion
  value: 1.0.11
- name: tfPlanOutFilePrefix
  value: tfplan
- name: tcsURL
  value: https://cloud.tenable.com/cns
- name: apiToken
  value: bd91db85-f431-4b3e-93c4-ae3249047399
- name: do_plan_or_scan
  value: plan

steps:
- task: CmdLine@2
  inputs:
    script: |
      export ARM_SUBSCRIPTION_ID=$(azSubID)
      export ARM_TENANT_ID=$(azTenantID)
      export ARM_CLIENT_ID=$(azClientID)
      export ARM_CLIENT_SECRET=$(azClientSecret)
      if [ $(do_plan_or_scan) == plan ]; then
        echo Installing terraform..
        sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl
        curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
        sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -
cs) main"
        sudo apt-get update && sudo apt-get install terraform=$(tfVersion)
        curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
        terraform init
        echo ~~~~GENERATING PLAN OUTPUT..
        terraform plan -out $(tfPlanOutFilePrefix)_$(Build.BuildNumber).out
        echo ~~~~GENERATING PLAN JSON..
        terraform show -json $(tfPlanOutFilePrefix)_$(Build.BuildNumber).out > $(t-
fPlanOutFilePrefix)_$(Build.BuildNumber).json
        elif [ $(do_plan_or_scan) == scan ]; then
          echo Installing terrascan..
          curl -L "$(curl -s https://api.github.com/repos/tenable/terrascan/releases/latest | grep -o -
E "https://.+?_linux_x86_64.tar.gz")" > terrascan.tar.gz
          tar -xf terrascan.tar.gz terrascan && rm terrascan.tar.gz
          install terrascan /usr/local/bin && rm terrascan
        fi
        echo ~~~~Downloading Tenable CS cli..
        wget https://downloads.accurics.com/cli/$(tcsCLIVersion)/accurics_linux -O tcs-iac-scanner
        chmod +x tcs-iac-scanner
        echo ~~~~Getting Tenable CS cli verison..
        ./tcs-iac-scanner version
      displayName: 'Install T.CS dependencies'

- task: CmdLine@2
  inputs:
    script: |
      if [ $(do_plan_or_scan) == plan ]; then
        echo ~~~~RUNNING Tenable CS assessment with pre-cooked plan..
        ./tcs-iac-scanner plan -mode=pipeline -appurl=$(tcsURL) -token=$(apiToken) -planjson-
n=$(tfPlanOutFilePrefix)_$(Build.BuildNumber).json
        elif [ $(do_plan_or_scan) == scan ]; then
          echo ~~~~RUNNING Tenable CS non plan based assessment..
          ./tcs-iac-scanner scan -mode=pipeline -appurl=$(tcsURL) -token=$(apiToken)
        fi
      displayName: 'Tenable CS Assessment - CLI'
    env:
```



```
REPO_URL: $(Build.Repository.Uri)
GIT_BRANCH: $(Build.SourceBranchName)
GIT_COMMIT: $(Build.SourceVersion)

- task: CopyFiles@2
  inputs:
    Contents: |
      **/*.json
      **/*.html
      **/*.out
      **/*.tfstate
    TargetFolder: '$(Build.ArtifactStagingDirectory)'
    condition: always()

- task: PublishBuildArtifacts@1
  inputs:
    pathToPublish: $(Build.ArtifactStagingDirectory)
    artifactName: drop
    condition: always()

- task: DownloadBuildArtifacts@0
  inputs:
    buildType: 'current'
    downloadType: 'single'
    artifactName: 'drop'
    downloadPath: '$(System.ArtifactsDirectory)'
    condition: always()
```

In this example, Tenable Cloud Security creates a new project called **DEFAULT\_AZURE** and publishes the scan results in that project.



## Set Up Policy Guardrails (CI/CD)

You can use the Tenable Cloud Security CLI to scan code in your CI/CD pipeline and fail the builds if Tenable Cloud Security finds severe vulnerabilities in the code. After installing Tenable Cloud Security CLI on the build machine, you must add the necessary instructions to the pipeline script to run the tool against the files present in the repository.

Following are some examples:

- [Azure DevOps \(on MAC\)](#)
- [AWS Code Pipeline \(on Linux\)](#)
- [Bamboo \(on Linux\)](#)
- [GitLab](#)

### Azure DevOps (on MAC)

Add the following commands to the YAML file:

```
trigger:
  -master

pool:
  vmImage: 'macOS-latest'

steps

task: CmdLine@2
  inputs:
    script: |
      brew install terraform
      brew install accurics
      export ARM_SUBSCRIPTION_ID= subscription id
      export ARM_TENANT_ID= tenant id
      export ARM_CLIENT_ID= client id
      export ARM_CLIENT_SECRET= client secret
      accurics init
      accurics plan
```

See [Integration with Azure DevOps Pipeline](#).

### AWS Code Pipeline (On Linux)

Add the following commands to the buildspec.YAML file:



```
version: 0.2

phases:
  install:
    commands:
      curl -s -qL -o terraform_install.zip https://releases.hashicorp.com/terraform/0.13.5/terraform_0.13.5_linux_amd64.zip
      unzip terraform_install.zip -d /usr/bin/

      chmod +x /usr/bin/terraform

    finally:
      terraform --version

  build:
    commands:
      export ARM_SUBSCRIPTION_ID=subscription ID
      export ARM_TENANT_ID=tenant ID
      export ARM_CLIENT_ID=client ID
      export ARM_CLIENT_SECRET=client secret
      ./accurics init
      ./accurics plan
```

## Bamboo (on Linux)

Add the following commands in the **Script body** of a **Script Configuration** in a **Bamboo Task**.

```
cp /home/user/AccuricsCLI/* ./
export ARM_SUBSCRIPTION_ID=<SUBSCRIPTION ID>
export ARM_TENANT_ID=<TENANT ID>
export ARM_CLIENT_ID=<CLIENT ID>
export ARM_CLIENT_SECRET=<CLIENT SECRET>
./accurics init
./accurics plan
if [ $? -eq 0 ]; then exit 0; else exit 1; fi
```

**Note:** Make sure to replace the Azure credential placeholder values with valid Azure credentials (required for Terraform):

- SUBSCRIPTION ID
- TENANT ID
- CLIENT ID
- CLIENT SECRET

## GitLab

The following example shows a GitLab pipeline.



```
variables:
  awsAccessKey: 5XXXXXXXXXXXXXXXXXXXX5
  awsSecretAccessKey: 5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX5
  tcsCLIVersion: latest
  tfVersion: 1.0.11
  tfPlanOutFilePrefix: tfplan
  tcsURL: https://cloud.tenable.com/cns
  tcsProjectID: 6xxxxxyy6-4XX4-4XX4-8XX8-0XXXXYYYYY0
  apiToken: bXXXXYYY5-fBB1-4RRe-9ZZ4-aXXXHHUUUV9
  do_plan_or_scan: plan
  GIT_BRANCH: $CI_COMMIT_BRANCH

Tcs-iac-assessment:
  script: |
    if [ $do_plan_or_scan == plan ]; then
      export AWS_ACCESS_KEY=$awsAccessKey
      export AWS_SECRET_ACCESS_KEY=$awsSecretAccessKey
      echo Installing terraform..
      apt-get update && apt-get install -y gnupg software-properties-common curl
      curl -fsSL https://apt.releases.hashicorp.com/gpg | apt-key add -
      apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs)
main"
      apt-get update && apt-get install terraform=$tfVersion
      curl -sL https://aka.ms/InstallAzureCLIDeb | bash
      terraform init
      echo ~~~~GENERATING PLAN OUTPUT..
      terraform plan -out $tfPlanOutFilePrefix.out
      echo ~~~~GENERATING PLAN JSON..
      terraform show -json $tfPlanOutFilePrefix.out > $tfPlanOutFilePrefix.json
    elif [ $do_plan_or_scan == scan ]; then
      echo Installing terrascan..
      curl -L "$(curl -s https://api.github.com/repos/tenable/terrascan/releases/latest | grep -o -
E "https://.+?_linux_x86_64.tar.gz")" > terrascan.tar.gz
      tar -xf terrascan.tar.gz terrascan && rm terrascan.tar.gz
      install terrascan /usr/local/bin && rm terrascan
    fi

    echo ~~~~Downloading Tenable CS cli..
    wget https://downloads.accurics.com/cli/$tcsCLIVersion/accurics_linux -O tcs-iac-scanner
    chmod +x tcs-iac-scanner

    echo ~~~~Getting Tenable CS cli verison..
    ./tcs-iac-scanner version

    echo ~~~~Running IaC assessment..
    if [ $do_plan_or_scan == plan ]; then
      echo ~~~~RUNNING Tenable CS assessment with pre-cooked plan..
      ./tcs-iac-scanner plan -mode=pipeline -project=$tcsProjectID -appurl=$tcsURL -token=$apiToken -
planjson=$tfPlanOutFilePrefix.json
    elif [ $do_plan_or_scan == scan ]; then
      echo ~~~~RUNNING Tenable CS non plan based assessment..
      ./tcs-iac-scanner scan -mode=pipeline -project=$tcsProjectID -appurl=$tcsURL -token=$apiToken
    fi
```

**Note:** Add the following command in your pipeline before running the `accurics init` command to specify the commit branch:

```
export GIT_BRANCH=${CI_COMMIT_BRANCH}
```



**Caution:** If the IaC scan fails with the "panic: runtime error: invalid memory address or nil pointer dereference" error, add the following command to the variables section of the pipeline:

```
GIT_BRANCH: $CI_COMMIT_BRANCH
```





---

## Use an On-Premises Code Scanner

---

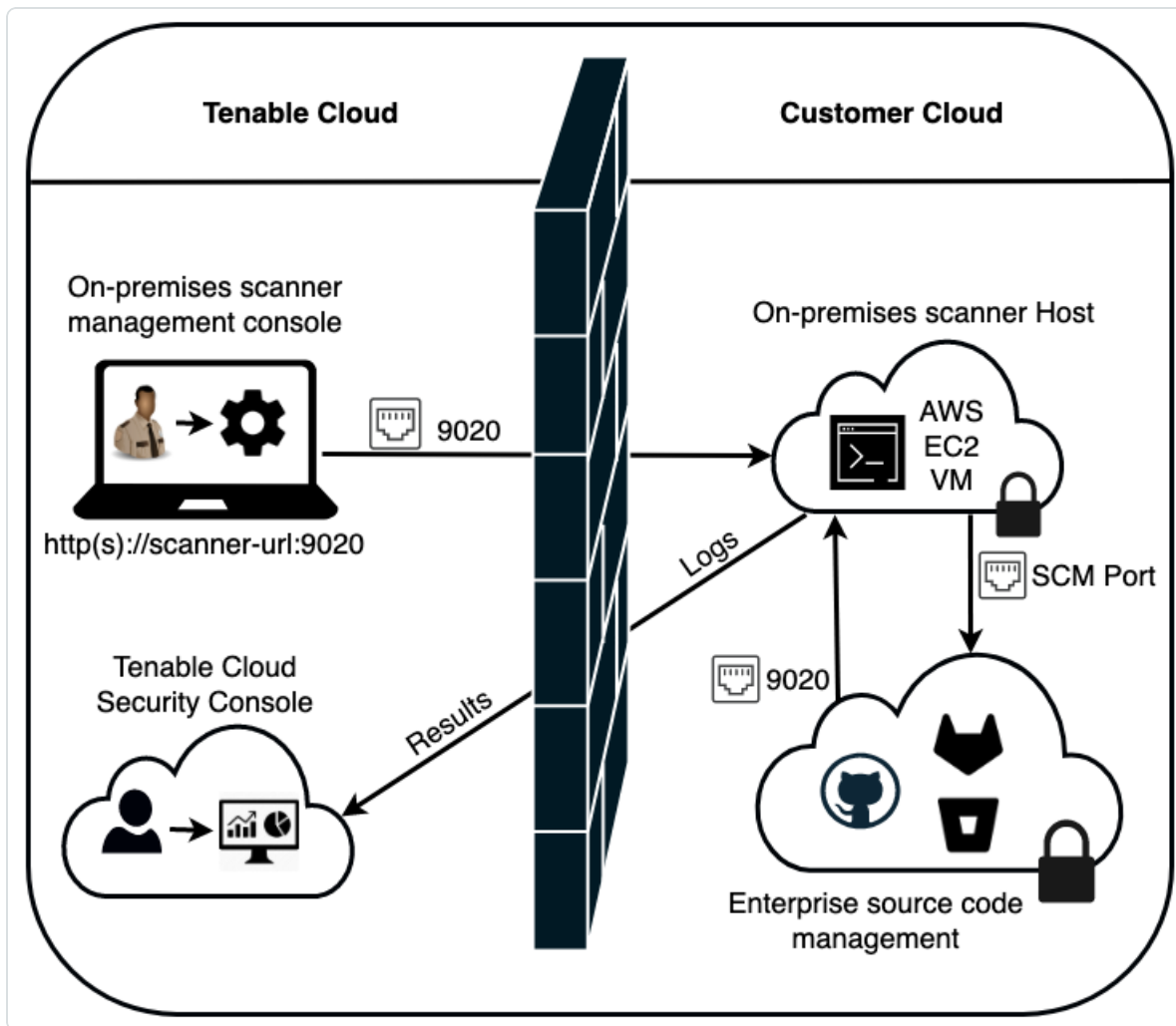
You can use Tenable Cloud Security on-premises code scanner to connect any repositories deployed behind a firewall. The Tenable Cloud Security code scanner scans the repository within the firewall-bound network and sends the processed data to Tenable Cloud Security services for reporting in Tenable Cloud Security.

### What data does the on-premises code scanner send to Tenable Cloud Security Cloud?

Tenable Cloud Security collects the metadata on cloud and IaC resources and normalizes it into native format before sending it to the cloud. When Tenable Cloud Security analyzes the IaC or cloud resources, secrets embedded in the configurations are redacted before the Tenable Cloud Security platform stores them. Those secrets remain on the on-premise scanner in terms of roles inside terraform files since the code never leaves the customer boundary.

**Note:** (Optional) If the state file location is provided during repository configuration, Tenable Cloud Security sends this as well. The content of the state file is only used for improving the accuracy of the mapping algorithm between IAC and cloud.

The following image explains the functionality of an **On-premise code scanner**.



You can deploy an on-premise scanner on the following SCMs:

- [GitHub Enterprise](#)
- [Bitbucket Server](#)
- [GitLab Server](#)



# Deploy an On-Premises Code Scanner

To deploy an on-premises code scanner, you must first download the deployment package for Ubuntu Linux from Tenable Cloud Security and then deploy the package on a virtual machine.

- [Download the On-premises Code Scanner Package](#)
- [Deploy the On-premises Code Scanner on a Virtual Machine](#)

To download the on-premises code scanner package:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Integrations**.  
The **All Integrations** page appears.
3. Click **On-premise code scanner**.  
The **On-premise code scanner** window appears.
4. In the upper-right corner, click **Download new**.  
The **New On-premise code scanner** window appears.
5. In the **Select deployment option** section, select **Ubuntu Linux**.
6. Click **Continue**.

Tenable Cloud Security displays the setup instructions for Ubuntu Linux.

**Note:** Depending on the number of enterprise repository servers, you can deploy multiple on-premises code scanners. You must have one code scanner per virtual machine instance.

7. Click **Download**.

Tenable Cloud Security downloads the `tenable-code-scanner-docker.zip` file.

8. Extract the on-premises code scanner deployment zip file.

**Note:** Do not alter the extracted contents.

To configure your on-premises code scanner to work with a self-signed certificate, see [Configure an On-Premise Code Scanner to Use Self-Signed Certificate](#).

What to do next:



## Deploy the On-premises Code Scanner on a Virtual Machine

Before you begin:

- You must have a virtual machine or system with the following minimum requirements:
  - A virtual machine with 4 GB RAM
  - 20 GB Solid State Drive (SSD)
  - Ubuntu 18 or later

Examples of virtual machine include Amazon Elastic Compute Cloud (Amazon EC2) instance, Azure virtual machine, VMware, and so on.

- Install Docker Engine. For more information, see [Install Docker Engine on Ubuntu](#).

Tenable recommends the following installation methods:

- [Install using the convenience script](#)
- [Install from a package](#)

(Optional) Perform the post-installation steps for Docker. For more information, see [Post-installation steps for Linux](#).

**Note:** The latest version installs Compose V2, which uses the `docker compose` command. For more information, see [Compose V2 Overview](#).

- Add the [Terraform versions](#) to your firewall whitelist. To test that the on-premises scanner works for Terraform, do the following:

1. Run cURL on the [Terraform version URL](#).

```
cURL https://releases.hashicorp.com/terraform/
```

2. Clone a repository.
3. Run the `terraform init` command on the repository.

To deploy the on-premises code scanner on a virtual machine:



1. Copy the on-premises code scanner configuration files that you extracted in [Deploy an On-Premises Code Scanner](#).
2. Open a terminal on the virtual machine created for the on-premises scanner and run the following commands:

```
cd <path_configuration_files_are_located>
chmod +x tenable-cs-code-scanner
sudo./tenable-cs-code-scanner
```

**Caution:** Tenable Cloud Security uses the `docker-compose` command that is supported with Compose V1. If you have Docker Compose V2, run the following command after executing the commands in [Step 2](#) to deploy the on-premises code scanner:

```
sudo docker compose up -d
```

The following is a sample output after a deployment:

```
o upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
[Tenable.cs OnPrem CodeScanner] [INFO] got the IP address of the machine as [REDACTED]
[Tenable.cs OnPrem CodeScanner] [INFO] downloading the accurics container images now, please hang on, this might take some time...
[Tenable.cs OnPrem CodeScanner] [DEBUG] Running the command 'docker-compose up -d' in the directory '[REDACTED]'.amazonaws.com
Removing login credentials for [REDACTED].amazonaws.com
[Tenable.cs OnPrem CodeScanner] [INFO] To view the realtime logs, please execute 'docker-compose logs -f'
[Tenable.cs OnPrem CodeScanner] [INFO] You can now visit url 'http://[REDACTED]' in the browser of your choice to complete OAuth App authorization..
```

3. In a browser, type the URL displayed in the output to launch the **On Premise Scanner Management Console**.

The **On Premise Scanner Management Console** page opens.

**Note:** If you have the IP address for the on-premises code scanner host virtual machine, you can manually launch the **On Premise Code Scanner Management Console** using the following URL:

```
https://<ip-address>/<dns-name>:9020
```

Where:


- `ip-address` is the IP address of host virtual machine.
- `dns-name` is the domain name of the host virtual machine.

Tenable Cloud Security deploys the on-premises code scanner.

To configure the on-premises scanner on your repositories, see the following topics:



- [Use an On-Premises Code Scanner to Scan GitHub Enterprise IaCs](#)
- [Use an On-Premises Code Scanner to Scan Bitbucket Server IaCs](#)
- [Use an On-Premises Code Scanner to Scan GitLab Server IaCs](#)

4. To check the status of the on-premises code scanner in Tenable Cloud Security, navigate to **Integrations > On-premise code scanner**.
  - a. Hover over the on-premises code scanner.
  - b. Click the  button to view more options:

Option	Description
Download weekly logs	Download the on-premises scanner logs for the last seven days. <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"><b>Note:</b> Enable the <b>Allow on-premise code scanner to send logs to Tenable Cloud Security</b> option when configuring the on-premises scanner.</div>
Download installer	Download the configuration file.
Edit	Modify the name of the on-premises scanner.
Delete	Delete the on-premises scanner.





# Use an On-Premises Code Scanner to Scan GitHub Enterprise IaCs

You can connect your GitHub repositories to an **on-premises code scanner** and scan your code for violations. Perform the following tasks to connect your GitHub repositories to an on-premises scanner:

1. [Create an OAuth Application in GitHub Enterprise Server.](#)
2. [Authorize the on-premises code scanner to access GitHub Enterprise Server.](#)
3. [Connect an IaC from GitHub Enterprise Server to a Tenable Cloud Security project.](#)

## To create an OAuth Application in GitHub Enterprise Server:

1. Sign in to your GitHub Enterprise Server console with an administrator account.
2. Navigate to **User Settings > Developer Settings > OAuth Apps > New Application.**

The **Register a new OAuth application** page appears.

**Note:** The on-premises code scanner requires port 9020 to authorize SCM applications. Ensure you have the correct network configuration in place for port 9020 on the on-premises code scanner machine to allow the SCM authorizer to access on-premises code scanner.

3. Create a new application by providing the following information:
  - In the **Application name** box, type a name for the application.
  - In the **Homepage URL** box, type the Tenable Cloud Security URL.
  - In the **Authorization callback URL** box, type: `http(s)://<on-premise_code_scanner_host_fqdn>.com:9020/v1/auth/oauth/github/callback`

Where:

    - `on-premise_code_scanner_host_fqdn` is the fully qualified domain name of the on-premises code scanner.
4. Click **Register application.**
5. Note the Client ID and Client Secret displayed after the creation of OAuth Application.





## To authorize the on-premises code scanner to access GitHub Enterprise Server:

1. Launch the URL displayed in the output after the on-premises code scanner deployment. For more information, see [Deploy an On-Premises Code Scanner](#).

The **On Premise Scanner Management Console** page appears. In the **On Premise Scanner Management Console** page, you can authorize the on-premises code scanner with different Source Code Management (SCM) providers.

2. In the **Configure servers** section, provide the following:
  - In the **Repository Server Address** box, type the repository server address.
  - In the **On-premise code scanner address (use port:9020)** box, type the code scanner address.

3. Click **Continue**.

The **Configure cloud (Optional)** section appears.

4. (Optional) In the **Select cloud provider** drop-down box, select one of the following options:
  - **AWS**
    1. In the **AWS Access Key** box, type the AWS access key.
    2. In the **AWS Secret Key** box, type the AWS secret key.
  - **GCP**
    - Click **Upload** to upload your service account credentials file.
  - **Azure**
    1. In the **Azure Client ID** box, type the Azure client ID.
    2. In the **Azure Tenant ID** box, type the Azure tenant ID.
    3. In the **Azure Subscription ID** box, type the Azure subscription ID.
    4. In the **Azure Client Secret** box, type the Azure client secret.

**Note:** The on-premises code scanner requires your cloud account details when you enable **Plan based setup** to scan your repositories. For more information, see [Connect Repositories](#).



5. Click **Continue**.

The **Setup authentication** section appears.


6. In the **Select repository server** drop-down box, select **GitHub**.  
Tenable Cloud Security displays an information form for GitHub.
7. Provide the following:
  - a. In the **Client ID** box, type the client ID.
  - b. In the **Client Secret** box, type the client secret.

**Note:** For information about how to obtain Client ID and Client Secret, see [Create an OAuth Application in GitHub Enterprise Server](#).

- c. Click **Submit**.
8. (Optional) In the **Other Settings** section, click the **Allow on-premise code scanner to send logs to Tenable Cloud Security** toggle.

Tenable Cloud Security redirects you to the GitHub Enterprise server to authorize the permissions on the OAuth Application. A message confirms successful authorization and GitHub redirects you to the **On-premise code scanner** page.

#### To connect an IaC from GitHub Enterprise Server to a Tenable Cloud Security project:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click the  icon.
3. Click **Connection > Repository**.  
The **Connect to repository** page appears.
4. In the **Choose a workflow to discover repo(s)** section, select **Version control**.
5. Click **Continue**.  
The **Connect to a version control provider** section appears.
6. In the **Connect to a version control provider** section, select **GitHub** and **On-Premise Code Scanner**.



7. Click **Continue**.

The **Choose onboarding repositories** section appears.

8. Select the required repository.

9. Hover over the selected repository and click  to configure the advanced settings.

For more information, see [Repository Configuration Parameters](#).

10. Click **Continue**.

The **Choose projects to add the repository to** section appears.

11. Select the project that you want to connect to the repository.

12. Click **Connect**.

A message confirms that Tenable Cloud Security connected the GitHub IaC repository to the selected project.



---

## Use an On-Premises Code Scanner to Scan Bitbucket Server IaCs

---

You can connect your Bitbucket repositories to an **On-Premise code scanner** and scan your code for violations. Perform the following tasks to connect your Bitbucket repositories to an on-premises scanner:

1. [Create a personal access token in Bitbucket Server.](#)
2. [Authorize the on-premise code scanner to access Bitbucket Server.](#)
3. [Connect an IaC from Bitbucket Server to Tenable Cloud Security project.](#)

### To create a personal access token in Bitbucket Server:

1. Sign in to the Bitbucket server with administrator level account credentials.
2. Navigate to **Profile picture > Manage account > Personal access tokens**.
3. Click **Create token**.
4. Configure the values as follows:



### Token details

Token name

### Permissions

Tokens are like another password, so their permissions will default to the level of access you have. Because of this, it is recommended that you restrict the token's permission to the level it will need.

Projects

Repositories

This personal access token will allow the supplied third-party application to:

- Perform pull request actions
- Update repository settings and permissions
- Update project settings and permissions
- Push, pull, clone, and fork repositories
- Create repositories

### Expiry

For added security, you can set this token to automatically expire. If you set an expiry date, you won't be able to edit it once you've created the token.

Automatic expiry  No  Yes

5. Click **Create**.
6. Note the personal access token provided.

To authorize the on-premise code scanner to access Bitbucket Server:



1. Launch the URL displayed in the output of the on-premise code scanner deployment. For more information, see [Deploy an On-Premises Code Scanner](#).

The **On Premise Scanner Management Console** page appears. You can now authorize the on-premise code scanner with different Source Code Management (SCM) providers.

2. In the **Configure servers** section, provide the following:
  - In the **Repository Server Address** box, type the repository server address.
  - In the **On-premise code scanner address (use port:9020)** box, type the code scanner address.
3. Click **Continue**.

The **Configure cloud (Optional)** section appears.

4. (Optional) In the **Select cloud provider** drop-down box, select one of the following options:
  - **AWS**
    1. In the **AWS Access Key** box, type the AWS access key.
    2. In the **AWS Secret Key** box, type the AWS secret key.
  - **GCP**
    - Click **Upload** to upload your service account credentials file.
  - **Azure**
    1. In the **Azure Client ID** box, type the Azure client ID.
    2. In the **Azure Tenant ID** box, type the Azure tenant ID.
    3. In the **Azure Subscription ID** box, type the Azure subscription ID.
    4. In the **Azure Client Secret** box, type the Azure client secret.

**Note:** The on-premise code scanner requires your cloud account details when you enable **Plan based setup** to scan your repositories. For more information, see [Connect Repositories](#).


5. Click **Continue**.

The **Setup authentication** section appears.



6. In the **Select repository server** drop-down box, select **Bitbucket**.  
Tenable Cloud Security displays an information form for Bitbucket.
7. Provide the following:
  - a. In the **Personal Access Token** box, type the personal access token. For more information about how to obtain the personal access token, see [To create a personal access token in Bitbucket Server](#).
  - b. Click **Submit**.
8. (Optional) In the **Other Settings** section, click the **Allow on-premise code scanner to send logs to Tenable Cloud Security** toggle.  
  
Tenable Cloud Security redirects you to the Bitbucket Enterprise server to authorize the permissions on the OAuth Application. A message confirms successful authorization and Bitbucket redirects you to the **On-premise code scanner** page.

#### To connect an IaC from Bitbucket Server to Tenable Cloud Security project:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click the  icon.
3. Click **Connection > Repository**.  
The **Connect to repository** page appears.
4. In the **Choose a workflow to discover repo(s)** section, select **Version control**.
5. Click **Continue**.  
  
The **Connect to a version control provider** section appears.
6. In the **Connect to a version control provider** section, select **Bitbucket** and **On-premise Code Scanner**.
7. Click **Continue**.  
  
The **Choose onboarding repositories** section appears.
8. Select the required repository.



9. Hover over the selected repository and click  to configure the advanced settings.

For more information, see [Repository Configuration Parameters](#).

10. Click **Continue**.

The **Choose projects to add the repository to** section appears.

11. Select the project that you want to connect to the repository.

12. Click **Connect**.

A message confirms that Tenable Cloud Security connects the Bitbucket IaC repository to the selected project.





# Use an On-Premises Code Scanner to Scan GitLab Server IaCs

You can connect your GitLab repositories to an **on-premises code scanner** and scan your code for violations. Perform the following tasks to connect your GitLab repositories to an on-premises scanner:

1. [Create an OAuth Application in GitLab Server.](#)
2. [Authorize the on-premise code scanner to access the GitLab Enterprise Server.](#)
3. [Connect an IaC from GitLab Server to Tenable Cloud Security project.](#)

## To create an OAuth Application in GitLab Server:

1. Sign in to the GitLab Server console with admin level account credentials.
2. To create an **Application** on the GitLab Server, go to **Preferences > Applications**.
3. On the **Add new application** page, create an application with the following configuration:
  - a. Specify a name for the application.
  - b. Select **Confidential** to use the application where the client secret can remain confidential.
  - c. In the **Scopes** section, select:
    - **api** - to grant read/write access to the API.
    - **read\_repository** - to grant read-only access to repositories on private projects.
  - d. Open the application that you created.
  - e. Note down the **Application ID**, **Secret**, and the **Authorization callback URL**: `http(s)://<on-premise_code_scanner_host_fqdn>.com:9020/v1/auth/oauth/gitlab/callback`

Where:

- `on-premise_code_scanner_host_fqdn` is the fully qualified domain name of the on-premise code scanner.

## To authorize the on-premise code scanner to access the GitLab Enterprise Server:



1. Launch the URL displayed in the output of the on-premise code scanner deployment. For more information, see [Deploy an On-Premises Code Scanner](#).

The **On Premise Scanner Management Console** page appears. In the **On Premise Scanner Management Console** page, you can authorize the on-premise code scanner with different Source Code Management (SCM) providers.

2. In the **Configure servers** section, provide the following:
  - In the **Repository Server Address** box, type the repository server address.
  - In the **On-premise code scanner address (use port:9020)** box, type the code scanner address.

3. Click **Continue**.

The **Configure cloud (Optional)** section appears.

4. (Optional) In the **Select cloud provider** drop-down box, select one of the following options:
  - **AWS**
    1. In the **AWS Access Key** box, type the AWS access key.
    2. In the **AWS Secret Key** box, type the AWS secret key.
  - **GCP**
    - Click **Upload** to upload your service account credentials file.
  - **Azure**
    1. In the **Azure Client ID** box, type the Azure client ID.
    2. In the **Azure Tenant ID** box, type the Azure tenant ID.
    3. In the **Azure Subscription ID** box, type the Azure subscription ID.
    4. In the **Azure Client Secret** box, type the Azure client secret.

**Note:** The on-premise code scanner requires your cloud account details when you enable **Plan based setup** to scan your repositories. For more information, see [Connect Repositories](#).



5. Click **Continue**.

The **Setup authentication** section appears.

6. In the **Select repository server** drop-down box, select GitLab.  
Tenable Cloud Security displays an information form for GitLab.
7. Provide the following:
  - a. In the **Client ID** box, type the client ID.
  - b. In the **Client Secret** box, type the client secret.

**Note:** For information about how to obtain Client ID and Client Secret, see [To create an OAuth Application in GitLab Server](#):

- c. Click **Submit**.
8. (Optional) In the **Other Settings** section, click the **Allow on-premise code scanner to send logs to Tenable Cloud Security** toggle.

Tenable Cloud Security redirects you to the GitLab Enterprise server to authorize the permissions on the OAuth Application. A message confirms successful authorization and GitLab redirects you to the **On-premise code scanner** page.

#### To connect an IaC from GitLab Server to Tenable Cloud Security project:

1. [Access Tenable Cloud Security](#).

2. In the left navigation bar, click the  icon.

3. Click **Connection > Repository**.  
The **Connect to repository** page appears.

4. In the **Choose a workflow to discover repo(s)** section, select **Version control**.

5. Click **Continue**.

The **Connect to a version control provider** section appears.

6. In the **Connect to a version control provider** section, select **GitLab** and **On-Premise Code Scanner**.



7. Click **Continue**.

The **Choose onboarding repositories** section appears.

8. Select the required repository.

9. Hover over the selected repository and click  to configure the advanced settings.

For more information, see [Repository Configuration Parameters](#).

10. Click **Continue**.

The **Choose projects to add the repository to** section appears.

11. Select the project that you want to connect to the repository.

12. Click **Connect**.

A message confirms that Tenable Cloud Security connected the GitLab IaC repository to the selected project.

---

## Configure an On-Premise Code Scanner to Use Self-Signed Certificate

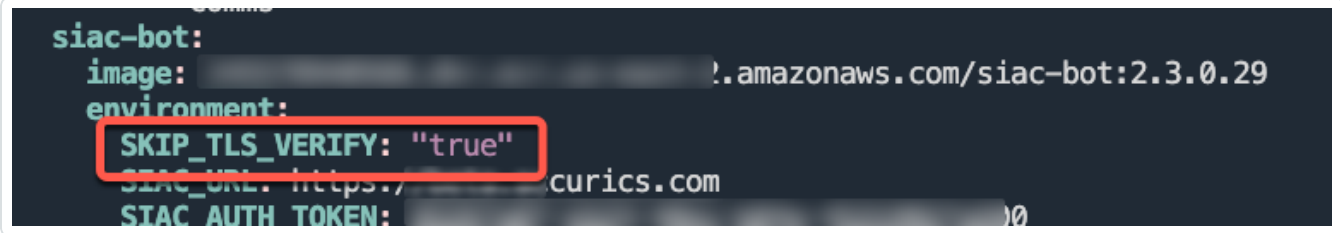
---

If you secure your repository server with a self-signed certificate not signed by a trusted certification authority, you can configure the on-premise code scanner to skip the TLS verification.

To configure an on-premise code scanner to use self-signed certificate:

1. Use Secure Shell (SSH) and access the on-premise code scanner VM.
2. Locate the `docker-compose.yaml` file.

**Note:** The `docker-compose.yaml` file is part of the `tenable-code-scanner-docker.zip`. For more information, see [Deploy an On-Premises Code Scanner](#)



```
siac-bot:
  image: !.amazonaws.com/siac-bot:2.3.0.29
  environment:
    SKIP_TLS_VERIFY: "true"
  SIAC_URL: https://curics.com
  SIAC_AUTH_TOKEN: 0
```

3. In the `siac-bot` section, add an environment variable: `SKIP_TLS_VERIFY: "true"`
4. Save the `docker-compose.yaml` file.
5. In the CLI of Tenable Cloud Security, run the following command:

```
sudo ./tenable-cs-code-scanner
```

Tenable Cloud Security uses the self-signed certificate to scan your repository.



---

## Viewing the Logs from an On-Premises Code Scanner

---

You can access the logs from the on-premises code scanner to troubleshoot any issues with the IaC scan.

Before you begin:

- Connect to the virtual machine or system where the on-premise scanner is hosted via SSH.

To view the code scanner logs:

- To view all the logs, use the following command:

```
sudo docker-compose logs | grep 'siac-bot\|etcd'
```

- To append all the logs to a text file, use the following command:

```
sudo docker-compose logs --no-color >> on-premise-scanner-logs.txt
```

- To view the last 100 lines of the logs, use the following command:

```
sudo docker-compose logs -f --tail="100" | grep 'siac-bot\|etcd'
```

- To view the last 100 lines of the logs and copy them to a text file, use the following command:

```
sudo docker-compose logs --no-color -f --tail="100" | grep 'siac-bot\|etcd' |&tee on-premise-scanner-logs.txt
```

- To view all the errors in the logs, use the following command:

```
sudo docker-compose logs | grep 'siac-bot\|etcd' | grep -i error
```



---

# Policies and Policy Groups

---

Tenable Cloud Security uses policies to identify misconfigurations and vulnerabilities present on cloud resources. Tenable Cloud Security has built-in policies for cloud and IaC resources that define the compliance standard for your cloud and IaC infrastructure. Tenable Cloud Security combines related policies in a policy group. A policy can support multiple benchmarks. Therefore, a policy group includes all the benchmarks supported by the policies in the group.

Tenable Cloud Security includes built-in policies and policy groups for all cloud providers. You can also create custom policies and policy groups.

To see a list of all policies, see [Tenable Cloud Security Policies](#).

See the following topics for more information:

## [How Policies Work in Tenable Cloud Security](#)

### [Manage Policies](#)

[Policy Modes](#)

[Create a Custom Policy](#)

[View and Download Policies](#)

[Edit a Policy](#)

[Delete a Policy](#)

### [Manage Policy Groups](#)

[Create a Custom Policy Group](#)

[View Policy Groups](#)

[Edit a Policy Group](#)

[Delete Policy Groups](#)

### [Associate Policies with a Project](#)

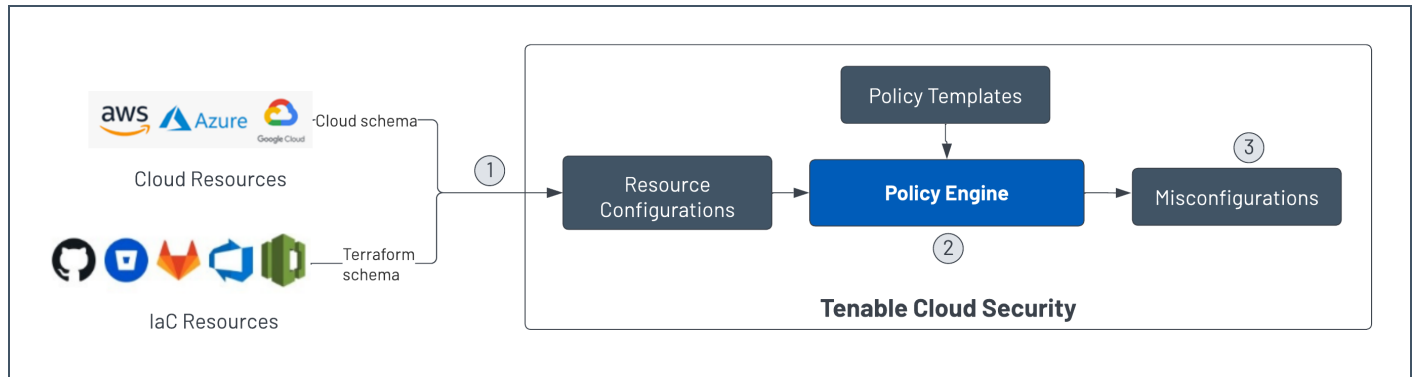
---

# How Policies Work in Tenable Cloud Security

---

Tenable Cloud Security defines policies as policy templates in the Rego policy language. Tenable Cloud Security includes the Open Policy Agent (OPA) in its policy engine that uses these policies for detecting any configuration violations in resources. Tenable Cloud Security reads the cloud and IaC resources and assesses these resources against the defined policies for those resources. Tenable Cloud Security displays the misconfigurations if any violations are detected.

The following image shows how policies work in Tenable Cloud Security:



The following process describes how Tenable Cloud Security reports misconfigurations:

1. Tenable Cloud Security reads the cloud resources in a schema specific to the cloud provider and converts it into a common resource configuration format. Similarly, Tenable Cloud Security converts the Terraform schema of IaC resources to the common resource configuration format.
2. The Tenable Cloud Security policy engine then compares these resources against the policies for that resource type.
3. If any violations are detected, Tenable Cloud Security reports these as misconfigurations.

## Benefits

Tenable Cloud Security includes a set of built-in policies for each resource type of a cloud provider. For example, Tenable Cloud Security defines a set of policies for AWS EC2 instances. Tenable Cloud Security uses the same policy to detect violations in both cloud and IaC for a particular resource type.

Tenable Cloud Security provides a vast coverage of policies to verify compliance across various resource types.





**Note:** Tenable Cloud Security provides over 1,800 policies out of the box, and is constantly adding more.

By default, Tenable Cloud Security automatically assigns the Accurics Security Best Practices policy group for the selected cloud provider to your project. You can modify the policy group for the project, if required.



---

## Manage Policies

---

Tenable Cloud Security includes built-in policies for all cloud providers. You can create custom policies, view the list of available policies, set alerts for a policy, edit a policy, or delete a policy.

- [Create a Custom Policy](#)
- [View and Download Policies](#)
- [Edit a Policy](#)
- [Delete a Policy](#)



## Policy Modes

In scenarios where you do not want your CI/CD tool to deploy cloud resources if Tenable Cloud Security detects violations in your IaC, the Tenable Cloud Security CLI provides special status codes based on the policy modes.

You can then configure your CI/CD to catch these codes and decide on failing the builds.

### Monitor

This is the default mode. Tenable Cloud Security CLI always responds with the status 0 (Success), if it detects any violation in your IaC.

Tenable Cloud Security CLI output for a policy in the **Monitor** mode:

```
-----  
Accurics successfully scanned the repository! Following is the summary - for details visit Accurics Web Console.  
{  
  "resources": 41,  
  "violation": 21,  
  "low": 5,  
  "medium": 10,  
  "high": 6,  
  "native": 15,  
  "inherit": 6,  
  "drift": 0,  
  "iacdrift": 0,  
  "clouddrift": 0  
}
```

### Enforce

In the **Enforce** policy mode, if Tenable Cloud Security CLI detects any violation in your IaC, it responds with an exit code status 1 (Failure).

Tenable Cloud Security CLI output for a policy in the **Enforce** mode:



-----  
Accurics successfully scanned the repository! Following is the summary - for details visit Accurics

```
{  
  "resources": 41,  
  "violation": 21,  
  "low": 5,  
  "medium": 10,  
  "high": 6,  
  "native": 15,  
  "inherit": 6,  
  "drift": 0,  
  "iacdrift": 0,  
  "clouddrift": 0  
}
```


-----  
**2021/05/13 14:53:41 Error: detected IaC violations** -



## Create a Custom Policy

You can create a custom policy for any resource type if the built-in policies do not meet your requirements. Tenable Cloud Security allows you to test the policy on a project before you add the custom policy.

To add or create a custom policy in Tenable Cloud Security:

1. [Access Tenable Cloud Security](#).
2. On the left navigation bar, click the  button.
3. Click **Custom policy**.
4. Click **Add policy**.  
The **Create Policy** page opens.
5. In the **Choose Resource** section, do one of the following:
  - Type a resource in the search box to bring up its name.
  - Select a resource from the list of available resources.

**Note:** You can create policies for any cloud resource or schema supported by the IaC providers. Tenable Cloud Security also supports policies for container images.

6. Click **Continue**.
7. In the **Policy Condition** section, use the query builder to select the conditions that the policy must meet. Click the arrow on the drop-down list to select a parameter, operator, value, and an AND/OR operator.

**Note:** The inputs to the query builder are dynamic and based on the resource's schema.

8. Click **Continue**.
9. In the **Test Policy** section, click the arrow on the drop-down list to select the project name.
10. Click **Test** to verify that the policy condition runs successfully. You can test policies for the projects for which you have access.
11. Click **Continue**.



12. In the **Remediation Details** section, select the parameter, type, and the required value to create the remediation for the policy.
13. (Optional) Click + to add more remediation details.
14. Click **Continue**.
15. In the **Policy Details** section, provide the following:
  - Type the policy name.
  - Select the policy category.
  - Select the severity of the policy.
  - Select the applicable benchmark for the policy.

**Note:** You can create a user-defined compliance benchmark and add the required policy to the created benchmark.

  - Select the required custom policy group.
  - Type the remediation description details.
16. Click **Create**.

Tenable Cloud Security creates a custom policy.










## View and Download Policies

You can view the list of available policies, including both built-in and custom policies, on the **Policies** page and download a CSV report.

To view and download the policies:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears and shows the **Policies** tab by default. The **Policies** tab displays the following details:

Column	Description
<b>Severity</b>	A policy can have one of the following severity levels: <ul style="list-style-type: none"><li>•  – High</li><li>•  – Medium</li><li>•  – Low</li></ul>
<b>Provider</b>	The following icons indicate the cloud provider for the policy: <ul style="list-style-type: none"><li>•  – AWS</li><li>•  – Azure</li><li>•  – GCP</li><li>•  – Kubernetes</li></ul>
<b>Name</b>	Displays the policy name.
<b>Category</b>	Displays one of the following categories of the policy: <ul style="list-style-type: none"><li>• Compliance Validation</li><li>• Configuration and Vulnerability Analysis</li><li>• Data Protection</li></ul>



	<ul style="list-style-type: none"><li>• Identity and Access Management</li><li>• Infrastructure Security</li><li>• Logging and Monitoring</li><li>• Resilience</li><li>• Security Best Practices</li></ul>
<b>Resource Type</b>	Displays the resource type of the policy.
<b>Compliance</b>	Displays the compliance status of the policy. The status can be one of the following: <ul style="list-style-type: none"><li>• Not Assessed – Not scanned.</li><li>• Compliant – Resource type is compliant with the policy after scan.</li><li>• Non-Compliant – Resource type is not compliant with the policy after the scan.</li><li>• Ignored – The policy violation has been ignored.</li></ul>
<b>Last Assessed</b>	Displays the date and time when the resource type was last assessed for that policy.

2. To search and filter the policies, do one the following:

- Use the **Search Policy** box to search for specific policies.
- Click **Filters** to view the filters. You can filter the policies by:
  - **Cloud Providers** – Filters by the cloud provider.
  - **Benchmarks** – Filters by policy benchmarks.
  - **Categories** – Filters by policy categories.
  - **Policy Groups** – Filters by the policy group.
  - **Policy Status** – Filters by the compliance status of the policy for that resource type.





- **Resource Type** – Filters by the resource type.
- **Severity** – Filters by the severity of the policy violation.

3. Click a policy to view its details.

The **Policy** plane appears. This plane has two tabs:

- **Policy Details** – Displays policy name, policy violation details, policy remediation, severity, cloud provider, benchmarks supported by the policy, and policy ID.
- **Policy Template** – Displays the Rego policy template.

4. Click **Export** to download a CSV report of policies. The CSV report contains the following fields:

- Policy Group
- Cloud Provider
- Severity
- Category
- Policy ID
- Policy Status
- Total Evaluated Resources
- Count of Failed Resources
- Count of Passed Resources
- Last Assessed

**Note:** The CSV report contains the filtered data if any filters are applied.



## Edit a Policy

You can only edit custom policies, and not built-in policies.

To edit a policy:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears.

2. In the row of the policy that you want to edit, click **⋮ > Edit**.

The **Edit Policy** window appears.

3. In the **Choose Resource** section, edit the resource, if needed.

4. Click **Continue**.

5. In the **Policy Condition** section, use the query builder to select the conditions that the policy must meet. Click the arrow on the drop-down list to select a parameter, operator, value, and an AND/OR operator.

**Note:** The inputs to the query builder are dynamic and based on the resource's schema.

6. Click **Continue**.

7. In the **Test Policy** section, click the arrow on the drop-down list to select the project name.

8. Click **Test** to verify that the policy condition runs successfully. You can test policies for the projects for which you have access.

9. Click **Continue**.

10. In the **Remediation Details** section, select the parameter, type, and the required value to create the remediation for the policy.

11. (Optional) Click **+** to add more remediation details.

12. Click **Continue**.

13. In the **Policy Details** section, provide the following:



- Type the policy name.
- Select the policy category.
- Select the severity of the policy.
- Select the applicable benchmark for the policy.

**Note:** You can create a user-defined compliance benchmark and add the required policy to the created benchmark.

- Select the required custom policy group.
- Type the remediation description details.

14. Click **Update**.



---

## Delete a Policy

---

You can only delete custom policies, and not built-in policies.

To delete a policy:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears.

2. In the row of the policy that you want to delete, click **⋮ > Delete**.

The **Delete Policy** window appears.

3. Click **Delete** to confirm the deletion of the policy.

Tenable Cloud Security deletes the policy from the policy list.



---

## Manage Policy Groups

---

Tenable Cloud Security combines related policies in a policy group. You can also create custom policy groups and add policies to it. You can view the list of all policy groups on the **Policies** page. You can edit and delete only custom policy groups.

See the following topics for more information about policy groups:

- [Create a Custom Policy Group](#)
- [View Policy Groups](#)
- [Edit a Policy Group](#)
- [Delete Policy Groups](#)




---

# Create a Custom Policy Group

---

You can create custom policy groups and add policies to it.

To add or create a custom policy group:

1. On the left navigation bar, click the  button.
2. Click **Custom policy**.
3. Click **Add policy group**.

The **Create Policy Group** page appears.

4. In the **Select policies to add to policy group** section, select the policies that you want to add to the policy group.
  - a. To search and filter the policies:
    - Use the **Search Policy** box to search for specific policies.
    - Filter the policies by:
      - **Severity** – Filters by the severity of the policy violation – High, Medium, or Low.
      - **Provider** – Filters by the cloud provider – AWS, Azure, GCP, or Image.
      - **Category** – Filters by policy categories – Compliance Validation, Configuration and Vulnerability Analysis, Data Protection, Identity and Access Management, Infrastructure Security, Logging and Monitoring, Resilience, or Security Best Practices.
      - **Resource Type** – Filters by the resource type of the policy.
      - **Benchmarks** – Filters by policy benchmarks.
      - **Policy Group** – Filters by the policy group.
  - b. Click **Continue**.
5. In the **Summarize policy group details** section, provide the following:



- a. In the **Provide a name** box, type a name for the custom policy group.
- b. Select your cloud provider:
  - AWS
  - Azure
  - Google Cloud Platform
  - Image

**Note:** Use **Image** for a creating policy group for policies for container image compliance.

- c. Select the policy mode:
  - Monitor
  - Enforce

**Note:** For **Image** policy group, only Monitor and Enforce policy modes apply.

For more information, see [Policy Modes](#).

6. Click **Done**.

A message confirms that Tenable Cloud Security created a new custom policy group.



# View Policy Groups

You can view the list of all policy groups on the **Policies** page.

To view policy groups:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears.

2. Click the **Policy Groups** tab.

The **Policy Groups** tab shows the following details:

Column	Description
<b>Name</b>	Displays the name of the policy group. Tenable Cloud Security has a policy group for each cloud provider.
<b>Provider</b>	Displays the cloud provider of the policy group.
<b>Managed by</b>	Specifies whether the policy group is created and managed by Tenable (Accurics Inc) or is a custom policy group (User).
<b>Project</b>	Displays the projects to which the policy group is assigned.
<b>Policy count</b>	Displays the number of policies in that policy group.

3. To filter the policy groups, click one of the following filters:

- **Cloud Provider** – Filters by the cloud provider: AWS, Azure, GCP, and Kubernetes.
- **IaC Types** – Filters by IaC types: Terraform, Terragrunt, Kustomize, Helm, CloudFormation, and Application.
- **Policy Type** – Filters by policy types: Custom, User Input, and Accurics Managed.

4. Click a policy group.

The **Policies** pane appears and lists all the policies associated with the policy group. You can also view the benchmarks supported by the policy group.





- Use the **Search policies** box to search for specific policies.
- Use the **Filter** drop-down to filter the policies. You can filter by the following:
  - **Severity** to filter the policies by severity – High, Medium, and Low.
  - **Benchmarks**



---

## Edit a Policy Group

---

You can edit a policy group by adding or deleting policies from the group. You can only edit custom policy groups, and not built-in policy groups.

To edit a policy group:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears.

2. Click the **Policy Groups** tab.

3. In the row of the policy group that you want to edit, click **> Edit Group**.

The **Edit Policy Group** window appears and lists the policies in the group.

4. Click **Edit Selection**.

The list of all policies appears.

5. Select the check box corresponding to the policies that you want to add to the policy group.

6. Click **Continue**.

7. In the **Summarize policy details** section, do the following:

- a. Edit the policy group name, if required.
- b. Select the cloud provider.
- c. Select one of the policy modes:
  - Monitor
  - Enforce

For more information, see [Policy Modes](#).

8. Click **Done**.

Tenable Cloud Security saves the policy group with the updated policies.



---

# Delete Policy Groups

---

You can only delete custom policy groups, and not built-in policy groups.

Before you begin:

- Remove all the projects assigned to the policy group from the **Policy Groups** tab.

For more information, see [Assign multiple projects to a policy group](#).

To delete a policy group:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears.

2. Click the **Policy Groups** tab.

3. In the row of the policy group that you want to delete, click **> Delete Group**.

The **Delete Policy Group** window appears.


4. Click **Delete** to confirm the deletion of the policy group.



## Associate Policies with a Project

By default, Tenable Cloud Security associates a policy group to a project when you create the project. You can assign policy groups and associated policies to a project.

To associate a policy group with a project:

1. Click the **Projects & Connections** tab.  
The **Projects & Connections** page appears.
2. Select the project with which you want to associate policy groups.  
The project details pane appears.
3. In the **Active policy groups** section, click the  button.  
The **Edit policy group** window appears.
4. Select one or more policy groups.

**Note:** Use the **Search** box to search for specific policy groups.

5. Click **Save**.  
Tenable Cloud Security displays the project details pane.
6. Click the **X** button to close the project details pane.  
Tenable Cloud Security associates the selected policy groups with the projects.



---

## Set up Drift Analysis

---

Any change to a cloud resource configuration is a potential security policy violation of the cloud security best practices. Tenable Cloud Security helps you analyze drifts and identify resource drifts and violations both in the IaC code and the resources deployed on the cloud. Then, Tenable Cloud Security facilitates to review and remediate the violations. Setting up drift analysis allows you to assess the posture of cloud deployment continuously and flag any drifts from the posture defined through the code.

To calculate drifts, Tenable Cloud Security maps your IaC resources to the corresponding cloud resources in your cloud account. A mapped resource is any resource in the cloud that has a matching configuration in IaC. An unmapped resource is any resource in the cloud that does not have a matching configuration in IaC.

Tenable Cloud Security helps you to analyze the following drifts along with information on how you can review and remediate the drifts.

- **IaC Drifts** – IaC drifts or code-to-cloud drifts occurs when a cloud resource is mapped with an IaC resource, but the cloud configuration parameter values of that resource are different from the configuration parameter value in the IaC.
- **Cloud Drifts** – Cloud to Cloud drift counts the resources that have configuration changes between two consecutive cloud scans. You can also set a baseline for a project to calculate the drift of the current scan from the baseline.

If both IaC and Cloud drifts exist for a resource, the IaC drift takes precedence.

See the following topics:

[Set a Baseline for a Project](#)

[View Cloud Drifts](#)

[View IaC Drifts](#)

[Review Drifts](#)

[Remediate Drifts](#)



---

## Set a Baseline for a Project

---

Tenable Cloud Security allows you to set a baseline for a project by recording the time stamp of the scan when the baseline is set. A baseline allows you to compare and identify cloud-to-cloud drifts between scans. Tenable recommends that you set a baseline for your project to review the drifts in every scan from the initial configuration when the baseline was set. You can also reset the baseline to a new time, if required. If you do not set a baseline for a project, Tenable Cloud Security calculates cloud-to-cloud drifts by comparing the current scan with the previous scan.

To set a baseline for a project:

1. On the Tenable Cloud Security home page, click **Projects & Connections**.

The **Projects** tab with the list of all projects appears by default.

2. In the row for the project for which you want to set a baseline, click **⋮ > Set baseline**.

A confirmation message appears.

3. Click **OK**.

The project baseline is set with the current time and date.

**Note:** To reset the baseline to the current date and time, click the baseline time on the project and click **OK** in the confirmation message.



---

## View Cloud Drifts

---

Cloud drift counts the resources that have configuration changes between two consecutive cloud scans. If you have set a [baseline](#), the Cloud drift is the difference in configuration between the current scan and the baseline. For example, you run a cloud scan that detects an EC2 instance with termination protection enabled. After the scan, you disable the termination protection of this EC2 instance. Now, in the next cloud scan, Tenable Cloud Security detects this change and shows it as a Cloud drift. Cloud drifts only happen on unmapped resources.

To view the cloud drift:

1. In the left navigation bar, click **Resources**.

The **Resources** page appears.

2. Click the **Resources with Drift** tab.

The list of all resource types with drifts appears.

3. Click the  **Filters** icon.

Tenable Cloud Security shows the available resource filters.

4. In the **Compliance state** section, select **Has Cloud Drifts**.

5. (Optional) Use the following filter options to further filter the resource types:

- **Projects** – Filters by project names.
- **Cloud Accounts** – Filters by cloud accounts.
- **Repository** – Filters by repositories.
- **K8s clusters** – Filters by Kubernetes clusters.
- **Source** – Filters by types: IaC, Cloud, State File, Mapped (IaC & Cloud).
- **Insights** – Filters by the types of violations found: Exposed blob stores, Exposed databases, Read/write IAM, and Exposed security groups
- **Compliance State** – Filters by compliance states: **Has Violations**, **Has IaC Drifts**, and **Has Cloud Drifts**.



- **Resource Type** – Filters by resource types.
- **VPC Filter** – Filters by VPC source.

6. Select the required filters and click **Apply**.

Tenable Cloud Security shows the results on the **Resources** page.

7. Click the resource type that you want to view.

All resources with drift for that resource type appear.

8. Click the resource ID that you want to view.

The **Resource Details** tab appears.

9. Click **Drifts**.

Tenable Cloud Security shows the comparison of the previous or baseline configuration with the current configuration.

	Previous/Baseline Cloud Configuration	Current Cloud Configuration
<input type="checkbox"/> Config	Cloud (aws_rds_cluster_instance.database-2-delta-instance-1)	Cloud (aws_rds_cluster_instance.database-2-delta-instance-1)
<input type="checkbox"/> instance_class	instance_class: "db.r5.large"	"db.r5.large" "db.t3.medium" <span style="float: right;">⋮</span>

10. Click the **Drift values** filter to select the type of drift:

- **Computed** – Configuration that is computed at run time. For example, IaC does not have a value for ARN, but the cloud equivalent configuration usually has an ARN value. In this case, the ARN might show as **Computed** on the IaC side or not show at all.
- **Missing in IaC** – Configuration that does not exist in IaC, but exists in the cloud. Therefore, it is a new parameter added or modified in the cloud.
- **Missing in Cloud** – Configuration that was configured in IaC, but Tenable Cloud Security could not find a matching configuration in the cloud. The configuration could be missing due to some of the following reasons:
  - The IaC configuration was not pushed and therefore, the configuration was not propagated to the cloud.





- The IaC configuration does not have an equivalent cloud value.
- Someone disabled or removed the configuration from the cloud.



---

## View IaC Drifts

---

For mapped resources, your IaC code configuration might differ from that on the cloud, which raises an IaC or a code-to-cloud drift. IaC drifts happen only on mapped resources.

If the `tfstate` (Terraform state) file is provided via the file or URL, Tenable Cloud Security can accurately map between the IaC and cloud resources. The `tfstate` data file includes unique IDs (ARNs/resource IDs) that can be used to link an IaC resource with a cloud resource.

If the `tfstate` file is not available, Tenable Cloud Security first creates a fingerprint for each IaC resource for matching against cloud resources. The fingerprint is sampled from multiple resource attributes, and the attributes used to form each fingerprint differ between resource types. By default, the following attributes are used for resources for each cloud provider:

- AWS: Resource Type + Name Tag
- Azure: Resource Type + Resource Group + Resource Name
- GCP: Resource Type + Project + Name
- Kubernetes: Resource Type + Namespace + Name

Mapping resources can be complex in larger environments since IaC to cloud resources can be many-to-many relationships. For example,

- IaC utilizing Terragrunt or Kustomize represents one-to-many IaC-to-Cloud relationship, since many different resources can be created using a single resource definition.
- When considering multiple repository (Git) branches, it is possible that many versions of an IaC resource correspond to a single cloud resource.

To view the code to cloud drift:

1. In the left navigation bar, click **Resources**.  
The **Resources** page appears.
2. Click the **Resources with Drift** tab.  
The list of all resource types with drifts appears.
3. Click the **Filters** icon.  
The list of available resource filters appears.



4. In the **Compliance state** section, select **Has IaC Drifts**.
5. (Optional) Use the following filter options to further filter the resource types:
  - **Projects** – Filters by project names.
  - **Cloud Accounts** – Filters by cloud accounts.
  - **Repository** – Filters by repositories.
  - **K8s clusters** – Filters by Kubernetes clusters.
  - **Source** – Filters by types: IaC, Cloud, State File, Mapped (IaC & Cloud).
  - **Insights** – Filters by the types of violations found: Exposed blob stores, Exposed databases, Read/write IAM, and Exposed security groups.
  - **Compliance State** – Filters by compliance states: **Has Violations**, **Has IaC Drifts**, and **Has Cloud Drifts**.
  - **Resource Type** – Filters by resource types.
  - **VPC Filter** – Filters by VPC source.
6. Select the required filters and click **Apply**.

Tenable Cloud Security shows the results on the **Resources** page.
7. Click the resource type that you want to view.

All resources with drift for that resource type appear.
8. Click the resource ID that you want to view.

The **Resource Details** tab appears.
9. Click **Drifts**.

Tenable Cloud Security shows the comparison of the IaC code and cloud code mapping.



Config	laC ( )	Cloud ( )
<input type="checkbox"/> security_rule ◀ ▶	<pre>security_rule: [   {     "access": "Allow",     "description": "",     "destination_address_prefix": "*",     "destination_address_prefixes": [],      "destination_application_security_group_ids": [],     "destination_port_range": "",     "destination_port_ranges": [],     "direction": "Inbound",     "name": "ssh",     "priority": 100,     "protocol": "TCP",     "source_address_prefix": "",     "source_address_prefixes": [],     "source_application_security_group_ids":   ],   "source_port_range": "*",   "source_port_ranges": [   ],   {     "access": "Allow",     "description": "",     "destination_address_prefix": "*",</pre>	<pre>[   {     "access": "Allow",     "description": "",     "destination_address_prefix": "*",     "destination_address_prefixes": [],      "destination_application_security_group_ids": [],     "destination_port_range": "",     "destination_port_ranges": [],     "direction": "Inbound",     "Outbound",     "name": "ssh",     "priority": 100,     "protocol": "TCP",     "source_address_prefix": "",     "source_address_prefixes": [],     "source_application_security_group_ids":   ],   "source_port_range": "*",   "source_port_ranges": [   ],   {     "access": "Allow",     "description": "",</pre>

10. Click the **Drift values** filter to select the type of drift:

- **Computed** – Configuration that is computed at run time. For example, laC does not have a value for ARN, but the cloud equivalent configuration usually has an ARN value. In this case, the ARN might show as **Computed** on the laC side or not show at all.
- **Missing in laC** – Configuration that does not exist in laC, but exists in the cloud. Therefore, it is a new parameter added or modified in the cloud.
- **Missing in Cloud** – Configuration that was configured in laC, but Tenable Cloud Security could not find a match for it in the cloud. The configuration could be missing due to some of the following reasons:
  - The laC configuration was not pushed and therefore, the configuration was not propagated to the cloud.
  - The laC configuration does not have an equivalent cloud value.
  - Someone disabled or removed the configuration from the cloud.



---

## Review Drifts

---

In Tenable Cloud Security, you can review the IaC and the cloud drifts in your account. Reviewing drifts helps you view and understand violations, drifts that occurred, your change history, and configuration details.

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Resources**.  
The **Resources** page appears.
3. Click the **Resources with Drift** tab.  
The **Resources with Drift** page appears.

**Note:** Use the **Categories** pane on the left to change the display based on resource types or failing policies.

4. On the **Resources with Drift** page, do one of the following:
  - Select the required resource type to view the details.
  - Use the **Search** box to search and select specific resources. You can search using these options: **Research ID**, **Resource Name**, **Resource ARN**, **Source**, **Region**, and **Cloud VPC**.



- Use the following filters to list to select the required resource types.

Filter	Description
Projects	Filters resource types by projects.
Cloud accounts	Filters resource types by cloud accounts.
Repositories	Filters resource types by repositories.
Source Type	Filters resource types by IaC or Cloud.
More filters	Filters the results by <b>Resource Types</b> , <b>Compliance state</b> , <b>VPC source</b> , <b>Source location</b> , or <b>Mapped</b> .
Clear filters	Clears the filters.
Show Results	Displays the filtered results.

The **Resource Type** details page appears.

5. In the **Resources ID** column, select the required resource ID to view its details.  
Tenable Cloud Security displays the details of the selected resource type.
6. Click the **Drifts** tab to open the **Drifts** section.
7. In the upper-right corner, click **Filter** to select the type of drift:
  - **Computed**
  - **Missing in IaC**
  - **Missing in Cloud**

Tenable Cloud Security displays the selected drift types. For more information, see [View Cloud Drifts](#) and [View IaC Drifts](#) .



## Remediate Drifts

You can remediate drifts that occurred in your cloud or IaC accounts. Tenable Cloud Security provides an option to create a Jira ticket to resolve the drift and remediate the violation. You can also share the violation by sending alerts.

To remediate drifts:

1. In the left navigation bar, click **Resources**.  
The **Resources** page appears.
2. Click the **Resources with Drift** tab.  
The list of all resource types with drifts appears.
3. Click the **Filters** icon.

The list of filter options appears:

Filter	Description
<b>Projects</b>	Filters by project names.
<b>Cloud Accounts</b>	Filters by cloud accounts.
<b>Repository</b>	Filters by repositories.
<b>K8s clusters</b>	Filters by Kubernetes clusters.
<b>Source</b>	Filters by types: IaC, Cloud, State File, Mapped (IaC & Cloud).
<b>Insights</b>	Filters by the types of violations found: Exposed blob stores, Exposed databases, Read/write IAM, and Exposed security groups.
<b>Compliance State</b>	Filters by compliance states: <b>Has Violations</b> , <b>Has IaC Drifts</b> , and <b>Has Cloud Drifts</b> .
<b>Resource Type</b>	Filters by resource types.
<b>VPC Filter</b>	Filters by VPC source.

4. Select the required filters and click **Apply**.



Tenable Cloud Security shows the filtered results on the **Resources** page.

5. Click the resource type that you want to view.  
All resources with drift for that resource type appear.
6. Click the resource ID that you want to view.  
The **Resource Details** tab appears.
7. Click **Drifts**.
8. Select the check box next to the drift that you want to remediate.  
Tenable Cloud Security enables **Remediate**.
9. Do one of the following:
  - a. Click **> Create Ticket**.  
For more information about creating a ticket, see [Create a Ticket for an Issue](#).
  - b. Click **> Share**.
  - c. For more information about escalating an issue, see [Escalate or Share an Issue](#).





---

# Configure Alerts

---

Tenable Cloud Security provides options for you to set up alerts in every project. With alerts, Tenable Cloud Security can notify users with a summary of project key events.

You can set up the alerts for the following channels:

- [Configure Email Alerts](#)
- [Configure Slack Alerts](#)
- [Configure Splunk Alerts](#)
- [Configure Microsoft Teams Alerts](#)
- [Configure AWS SNS Alerts](#)



---

# Configure Email Alerts

---

You can create and configure email alerts from a project. Tenable Cloud Security generates an email alert only for security alerts.

## Before you begin

- Configure the email addresses to which you want to send alerts within Tenable Cloud Security.


To configure email alerts:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar of the Tenable Cloud Security page, click **Home**.
3. Click the **Projects & Connections** tab.
4. In the projects list, click the project for which you want to configure email alerts.

The project details panel appears.

5. In the **Alerts** section, click .

The **Project alerts** page appears.

6. In the **Choose alert channels** section, select the check box for the **Email** channel and click **Select to setup** .

The **Configure Channel** window appears.

7. From the list of email addresses, select the check box for the email that you want to configure for alerts.
8. Click **OK**.

The **Project alerts** page appears. Tenable Cloud Security saves the alert configuration and sends alerts for all project events.

9. Click **Save**.



---

# Configure Slack Alerts

---

You can create and configure Slack alerts from a project. Tenable Cloud Security sends Slack notifications that summarize project key events. Tenable Cloud Security supports integrating with Slack (using OAuth) to publish new Tenable Cloud Security Cloud alerts into a specific Slack channel.

## Step 1: Slack Configuration

To configure the integration in Slack:

1. [Create a new Slack application](#)
2. Add the following User Token Scopes to the application.
  - **chat:write**: Send messages on user's behalf.
  - **im:write**: Start direct messages with people on user's behalf.
3. Install the application into the workspace OR directly into the slack channel where you want Tenable Cloud Security to send notifications.

## Step 2: Tenable Cloud Security Configuration


To configure the integration in Tenable Cloud Security:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar of the Tenable Cloud Security page, click **Home**.
3. Click the **Projects & Connections** tab.
4. In the projects list, click the project for which you want to configure Slack alerts.

The project details panel appears.

5. In the **Alerts** section, click .

The **Project alerts** page appears.

6. In the **Choose alert channels** section, select the check box for the **Slack** channel and click **Select to setup** .



---

The **Configure Channel** window appears.

7. In the **Channel Name** box, type the Slack channel name into which you want Tenable Cloud Security to send notifications.
8. In the **Slack API Token** box, type your User OAuth Token generated in the Slack app.
9. Select the required check boxes for the type (severity) of the violations that you want to report.
10. Click **Save**.

The **Project alerts** page appears.

11. Click **Save**.



# Configure Microsoft Teams Alerts

You can integrate Tenable Cloud Security with Microsoft Teams to report violations.


To integrate Tenable Cloud Security with Microsoft Teams:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar of the Tenable Cloud Security page, click **Home**.
3. Click the **Projects & Connections** tab.
4. In the projects list, click the project for which you want to configure Microsoft Teams.

The project details panel appears.

5. In the **Alerts** section, click .

The **Project alerts** page appears.

6. In the **Choose alert channels** section, select the check box for the **Microsoft Teams** channel and click **Select to setup** .

The **Configure Channel** window appears.

7. In the **Webhook URL** box, type the incoming webhook URL for the integration.

**Note:** Incoming webhooks are special URLs in Microsoft Teams that provide a simple way to share content in team channels. For more information, see [Create an incoming webhook](#).

8. Click **Save**.

The **Project alerts** page appears.

9. Click **Save**.



# Configure Splunk Alerts

Tenable Cloud Security can integrate with Splunk Cloud Platform to manage your incident logs. You must configure the HTTP Event Collector (HEC) in Splunk for Tenable Cloud Security that lets you send notifications over the HTTP and Secure HTTP (HTTPS) protocols using a token-based authentication model.

For more information about the HEC, see [Set up and use HTTP Event Collector in Splunk Web](#).

For each incident, Tenable Cloud Security sends the following information to Splunk:

- category
- severity
- title
- resource
- firstDetection
- Date
- guideline
- violationId

For example,

```
{"message":{"violationId":"ACS_AWS_S3_15","resource":"arn:aws:s3::scanners-ac--809694787632","firstDetectionDate":"2020-05-19T11:53:40.573Z","title":"Ensure all data is transported from the S3 bucket securely","category":"S3","guideline":"<<Tenable Cloud Security guidelines>>"},"severity":"HIGH"}
```

## Step 1: Splunk Configuration

To configure the integration in Splunk:

1. Access the Splunk platform.
2. Click **Settings > Data Inputs**.

The **Data inputs** page appears.



3. In the **HTTP Event Collector** type, click **+Add new** in the **Actions** column.

Complete the steps in the **Add Data** wizard:

- a. In the **Select Source** page, type a name for the token in the **Name** box.
- b. Click **Next**.

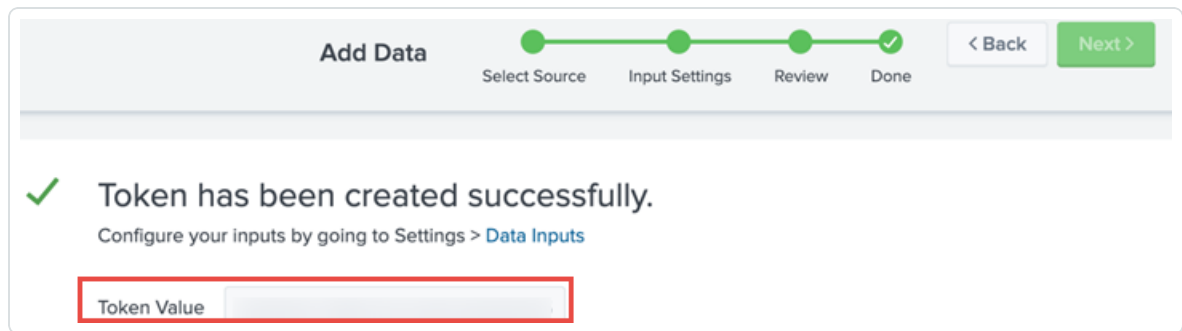
The **Input Settings** page appears.

- c. In the **Source type** section, click **Select** and then select **\_json** from the **Select Source Type** drop-down box.
- d. In the **Index** section, select **Default** from the **Default Index** drop-down box.
- e. Click **Review**.

Review the information provided for the Splunk configuration.

- f. Click **Submit**.

A confirmation message with the token value appears.



## Step 2: Tenable Cloud Security Configuration


1. [Access Tenable Cloud Security](#).
2. In the left navigation bar of the Tenable Cloud Security page, click **Home**.
3. In the left navigation bar of the Tenable Cloud Security page, click **Home**.
4. Click the **Projects & Connections** tab.
5. In the projects list, click the project for which you want to configure Microsoft Teams.

The project details panel appears.



6. In the **Alerts** section, click .

The **Project alerts** page appears.

7. In the **Choose alert channels** section, select the check box for the **Splunk** channel and click **Select to setup** .

The **Configure Splunk HTTP Event Collector (HEC)** page appears.

8. From the **Project** drop-down list, select the project for which you want to manage your incident logs.

9. Provide values for the following fields to configure Splunk:

- **Splunk HEC URL:** The standard form for the HEC URI in Splunk Cloud Platform is as follows:

```
<protocol>://http-inputs-<host>.splunkcloud.com:<port>/<endpoint>
```

Where:

- <protocol> is either http or https.
- <host> is the Splunk instance that runs HEC.
- <port> is the HEC port number, which is 8088 by default, but you can change it in the HEC Global Settings.
- <endpoint> is the HEC endpoint you want to use. Use the `/services/collector/event` endpoint for JSON-formatted events.
- **Splunk HEC Token:** Enter the Splunk token obtained during [Configuration in Splunk](#).
- **Event Source Type:** Type `_json`.

10. Select the **Verify SSL** check box to enable the SSL verification.

11. Select **Enable Alert** check box to enable the alerts for violations.

12. Select the required check boxes for the type (severity) of the violations that you want to report.

13. Click **Submit**.

The **Project alerts** page appears.





14. Click **Save**.



# Configure AWS SNS Alerts

In Tenable Cloud Security, you can configure alerts for the Amazon Web Services (AWS) Simple Notification Service (SNS).

**Note:** AWS generates a corresponding ARN number whenever you create a new topic. For more details, see [Creating an Amazon SNS topic](#).

Before you begin:

- Create an IAM role with permissions to publish to the SNS topic. For more information, see [Set up a Role for AWS SNS Alerts](#).


To configure AWS SNS alerts:

1. [Access Tenable Cloud Security](#).
2. Click the **Projects & Connections** tab.
3. In the projects list, click the project for which you want to configure AWS SNS.

The project details panel appears.

4. In the **Alerts** section, click .

The **Project alerts** page appears.

5. In the **Choose alert channels** section, select the check box for the **SNS** channel and click **Select to setup** .

The **Configure SNS Alerts** window appears.

**Note:** You can also configure AWS SNS alerts by clicking **Integrations > Configure AWS SNS**. In this case, select the project for which you want to configure the AWS SNS alert and follow the remaining steps.

6. In the **Topic ARN** box, enter the ARN of the SNS topic.

**Note:** AWS generates the **Topic ARN** when you create the topic.



7. Select the required check boxes for the type (severity) of the violations that you want to report – High, Medium, and Low.

8. Click **Submit**.

The **Project alerts** page appears.

9. Click **Save**.



# Set up a Role for AWS SNS Alerts

To integrate with AWS SNS, Tenable Cloud Security requires a role with publish permission to the SNS topic.

Before you begin:

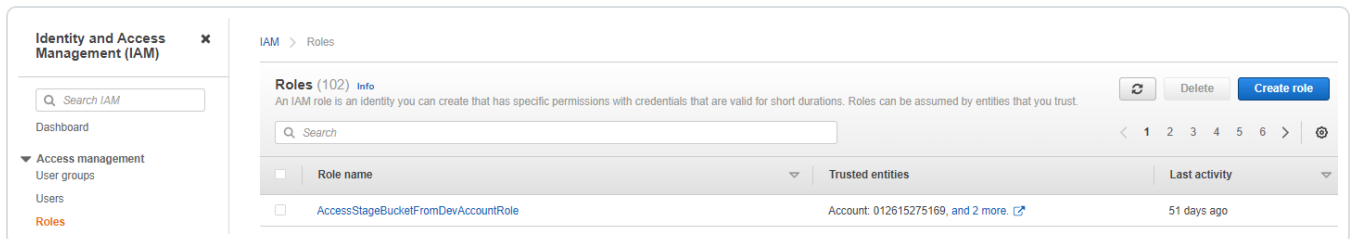
- Log in to the AWS web console with a user account with permission to create IAM roles.

For more information about IAM roles, see Amazon's [AWS Identity and Access Management User Guide](#).

To create a role for AWS SNS:

1. In the AWS web console, go to **Identity and Access Management (IAM)**.
2. On the left navigation pane, click **Roles**.

The **Roles** page appears.



3. Click **Create Role**.

The Create Role wizard appears.

4. In the **Select trusted entity** page, do the following:
  - a. In the **Trusted entity type** section, select **AWS Account**.
  - b. In the **An AWS Account** section, select **Another AWS Account**.
  - c. In the **Account ID** box, type **012615275169**.

**Note:** 012615275169 is the account ID of the Tenable AWS account that you are establishing a trust relationship with to support AWS role delegation.



- d. Under **Options**, click the **Require External ID** check box and type your Tenable Vulnerability Management Container UUID in the External ID box.

**Note:** In Tenable Vulnerability Management, navigate to **Settings > License** to get your container UUID. For more information, see [View Information about Your Tenable Vulnerability Management Instance](#).

- e. Click **Next**.

Step 2  
Add permissions

Step 3  
Name, review, and create

### Trusted entity type

**AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

**AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

**Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

**SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

**Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

### An AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

This account (576993307204)

**Another AWS account**

Account ID  
Identifier of the account that can use this role

012615275169

Account ID is a 12-digit number.

Options

**Require external ID (Best practice when a third party will assume this role)**  
You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. [Learn more](#)

External ID

<insertTjioContainerUUID>

**Important:** The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account iam:AssumeRole calls. [Learn more](#)

**Require MFA**  
Requires that the assuming entity use multi-factor authentication.

5. On the **Add permission policies** page, create a policy with the following JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sns:Publish",
      "Resource": "*"
    }
  ]
}
```



6. In the **Name, review, and create** page, do the following:
  - a. In the **Role Details** section, type a **Role Name** for the role.
  - b. (Optional) Add a role description in the **Description** box.
  - c. (Optional) Click **Add Tags** to add key-value pairs to AWS resources.
  - d. Click **Create Role**.
7. To get the **Role ARN** and **External ID** of this new role for Tenable Cloud Security, do the following:
  - a. On the left navigation pane, click **Roles**.
  - b. Search for the role that you created.
  - c. In the **Summary** section, note the **Role ARN** value.
  - d. Click the **Trust Relationships** tab and note the value of the **ExternalId** field.
8. Note down the following values:
  - **Role ARN**
  - **External ID**

You need these values when configuring AWS SNS in Tenable Cloud Security.



## Integrate with Atlassian Jira

You can integrate Tenable Cloud Security with Atlassian Jira to manage cloud alerts and create Jira tickets for issues. Tenable Cloud Security requires access to your Jira domain, email account, and API token for integration.

Before you begin:

- As a Jira administrator, add a new issue type named **Security Issue** in your Jira project to integrate with Tenable Cloud Security and to escalate violations. Configure this **Security Issue** issue type with the following **Required** fields:
  - **Project**
  - **Summary**
  - **Description**
  - **Reporter**
  - **Assignee**
  - **Priority**

**Caution:** Do not set any other fields as mandatory, as this can cause the Jira ticket creation to fail.

For more information, see [Add a new issue type](#) and [Specifying field behavior](#) in the Atlassian documentation.

- Generate a Jira API token for your Atlassian account.

For more information, see [Create an API token](#) in Atlassian documentation.

- Ensure generic users have the appropriate permissions to create issues within the JIRA project selected when integrating with Tenable Cloud Security.

To integrate Jira Cloud with Tenable Cloud Security:

1. [Access Tenable Cloud Security](#).
2. In the left pane, click **Integrations**.

The **All Integrations** page appears.



3. Click **Configure Jira Integration**.

The **Jira Integration** page appears.

4. Select the **Jira Cloud** option to integrate Jira with Tenable Cloud Security.

5. Click **Continue**.

6. Select the **API Token** option to use Jira API token for authentication.

7. Click **Continue**.

8. In the **Enter your credentials** section, provide the following details:

- In the **Jira Domain** box, type the name of Jira domain that you want to integrate with Tenable Cloud Security.
- In the **Email** box, type the email ID associated with the Jira API token.
- In the **Token** box, copy and paste the value of the [Jira API token](#) for your Atlassian account.

9. Click **Continue**.

10. In the **Set up Jira Configuration** section, provide the following details:

- In the **Project** drop-down box, select the Jira project.  
All Jira projects in your domain appear in the **Project** drop-down box.
- In the **Issue type** drop-down box, select the issue type as **Security Issue**.
- In the **Reporter** drop-down box, select the username of the reporter for the Jira tickets.
- In the **Assignee** drop-down box, select the username of the assignee for the Jira tickets.

**Note:** Only the users that are part of the selected Jira **Project** are displayed in the **Reporter** and **Assignee** fields.

You can view a read-only generic summary of issues. When you create a Jira ticket, Tenable Cloud Security automatically populates the issue summary and description based on the failing policy.

11. Click **Continue**.





12. Click **Setup Jira**.

Tenable Cloud Security integrates with Jira. You can now create issues for any failing policies from Tenable Cloud Security. For more information about creating tickets, see [Create a Ticket for an Issue](#).



---

## View Alerts

---

You can view the log details of all events occurring on your cloud accounts on the **Alerts** page.

Tenable Cloud Security classifies alerts by the following severity type:

- **Error**
- **Security**
- **Info**
- **Success**

All events except **Security** are considered informational events. You can view the number of critical alerts on the Tenable Cloud Security Misconfigurations Dashboard.

To view alerts:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Alerts**.

The **Alerts** page lists the log details of all scans and categorizes them based on projects, log types, scan source, and the current scan status.



## Configure Alert Rules

In Tenable Cloud Security, you can enable alert rules to receive notifications about any change to your account.

To receive notifications:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Alerts**.
3. Click the **Alert Rules** tab.  
A page opens with a list of rules.
4. To enable violation alerts, select the checkboxes next to the appropriate alert rules.  
Tenable Cloud Security enables **Manage Alerts**.
5. Click **Manage Notifications**.  
The **Manage Notifications** pane appears.
6. Hover over your project and click the toggle next to the project name.

**Note:** You can select more than one project to assign the alert rule.

The toggle changes to blue. A message confirms that Tenable Cloud Security enabled the alert rule for the project.

7. Click **X** to close the **Manage Notifications** pane to return to the **Alert Rules** tab.

**Note:** A green bell icon indicates that Tenable Cloud Security enabled the alert rule for one or more projects. A blue bell icon indicates that the alert rule is disabled.



## Alerts Page Information

The **Alerts** page displays the logs generated for all scans. You can filter the alerts based on projects, log type, scan source, and actions. For more information, see [Filter Options](#).

Column	Description
<b>Summary</b>	A summary of the alert. You can click the summary text to view: <ul style="list-style-type: none"><li>• <b>Summary</b></li><li>• <b>Created</b></li><li>• <b>Type</b></li><li>• <b>Source</b></li><li>• <b>Action</b></li><li>• <b>Project</b></li></ul>
<b>Project</b>	The project name.
<b>Source</b>	The violation source: <ul style="list-style-type: none"><li>• <b>IaC</b></li><li>• <b>Cloud</b></li></ul>
<b>Resource Type</b>	The resource type associated with the alert.
<b>Cloud Account</b>	The cloud account associated with the scan.
<b>Created</b>	The IP address of the machine and the time when Tenable Cloud Security reported the issue.

### Filter Options

Option	Description
<b>Project</b>	Filter the alerts based on the project.



Option	Description
<b>Resource Type</b>	Filter the alerts based on the type of resource type.
<b>Source</b>	Filter the alerts based on the source type of scans: <ul data-bbox="467 394 708 569" style="list-style-type: none"><li>• Cloud violation</li><li>• IaC violation</li><li>• CloudTrail</li></ul>



---

## Set an Alert Rule for a Policy

---

You can enable alerting for a project for policy violations. Whenever one or more policies fails for the resources in that project, Tenable Cloud Security sends an alert to all the alert channels configured for the project.

Before you begin:

- Configure at least one alert channel for the project. For more information, see [Configure Alerts](#).

To set an alert rule for a policy:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears and displays the **Policies** tab.

2. In the row of the policy for which you want to set an alert, click **> Notify**.

The **Manage Notifications** pane appears.

3. Click the toggle next to the project for which you want to set an alert for the selected policy.

Tenable Cloud Security updates the alert rules and sends an alert when a violation against the policy is reported.



---

## View Findings

---

The **Findings** page lists the misconfigurations and vulnerabilities that are detected during the Tenable Cloud Security cloud scans.

For more information, see the following:

- [View Misconfigurations](#)
- [View Vulnerabilities](#)
- [View Ignored Misconfigurations](#)



## View Misconfigurations


Tenable Cloud Security shows misconfigurations when resources fail to comply with the configured policies. You can view and download a CSV report of misconfigurations from the **Misconfigurations** page. You can also view the resources impacted by these misconfigurations and remediate the impacted resources. You can perform the following tasks from the **Misconfigurations** page:

- [Download the Misconfigurations report](#)
- [View impacted resources](#)
- [Remediate an impacted resource](#)

To view misconfigurations and download the Misconfigurations report:

1. In the left navigation bar, click **Findings**.

The **Misconfigurations** page appears.

2. On the **Misconfigurations** page, do one of the following:
  - Use the **Search** box to search for specific failing policies.
  - Use the following filters:
    - a. Click the  **Filters** icon to open the **Filter Misconfigurations** box.
    - b. Select the following filters as needed.

Filter	Description
<b>Projects</b>	Filters the failing policies by projects.
<b>Cloud account</b>	Filters the failing policies by cloud accounts.
<b>Repository</b>	Filters the failing policies by repositories.
<b>Category</b>	Filters by resource category.
<b>Severity</b>	Filters by the severity of the failing policy: <b>High, Medium, Low, Info</b> .





<b>Source</b>	Filters by the source for the policy violation: <b>Cloud</b> or <b>IaC</b> .
<b>K8s cluster</b>	Filters by Kubernetes clusters.
<b>Policy group</b>	Filters by policy groups. Applicable only for custom policies.
<b>Benchmarks</b>	Filters by policy benchmarks.

3. Click **Export > CSV** to download the misconfigurations report in the CSV format.

The report provides a project-wise listing of all failing policies and includes the following details:

- Project
- Source (IaC or Cloud)
- Cloud Provider
- Cloud Account
- Region
- VPC
- Policy Group
- Severity
- Policy ID
- Failing Policy
- Resource Type
- Cloud ARN
- Cloud Resource ID
- Cloud Resource Name
- Remediation
- IaC Resource ID



- IaC Resource Name
- IaC Repository
- IP Address
- DNS
- Cloud Tag
- Date and time the violation was last seen
- Date and time the violation was first seen

To view impacted resources:

1. On the **Misconfigurations** page, click the policy that you want to view.  
The policy pane with the list of impacted resources appears.
2. In the **Impacted Resources** section, do one of the following:
  - Select the impacted resource that you want to remediate and click the impacted resource name.
  - Use the following filters to filter and select a specific impacted resource:

Filter	Description
<b>Projects</b>	Filters the impacted resources by project.
<b>Resource Types</b>	Filters the impacted resources by resource types.
<b>Source Types</b>	Filters the impacted resources by source – IaC or Cloud.
<b>Cloud Accounts</b>	Filters the impacted resources by cloud account name.

3. Click an impacted resource to view the resource details.

The following details are displayed:



- Resource details

Filter	Description
Violation Source	Source of the policy violation – IaC or Cloud.
Cloud ID	ID of the cloud resource.
IaC ID	ID of the IaC resource.
Resource Type	The resource type to which the resource belongs.
Cloud Provider	The cloud provider – AWS, Azure, or GCP.
Cloud Account	The cloud account name.
Repository	Link to the repository of the IaC resource.
Cloud Tags	The label associated with the cloud resource by the cloud provider.
IaC Tags	The label associated with the IaC resource.

- Resource Configuration JSON: Shows the IaC or cloud resource configuration and the remediation resource configuration.
- Remediation: Provides the remediation steps for the policy violation.

To remediate an impacted resource:

1. Click the check box next to an impacted resource.  
Tenable Cloud Security enables **Create a ticket**, **Create a PR**, and **Exclude Policy**.
2. Click one of the following remediation options:
  - **Create a ticket** – Creates a Jira ticket for the selected issue. For more information, see [Create a Ticket for an Issue](#).
  - **Create a PR** – Creates a pull request. This option is enabled only for IaC resources. For more information, see [Create a Pull Request for an Issue](#).



- **Exclude a Policy** – Ignores the violation. For more information, see [Ignore Mis-configurations](#).

**Note:** Tenable Cloud Security shows the remediation steps to fix a failing policy on the policy pane.



## View Vulnerabilities

The **Vulnerabilities** tab of the **Findings** page displays the vulnerabilities detected during the Agentless Assessment of EC2 instances and Azure virtual machines.

1. [Access Tenable Cloud Security](#).

The **Dashboard** page appears.

2. In the left navigation pane, click **Findings**.

The **Misconfigurations** tab appears.

3. Click the **Vulnerabilities** tab.

The **Vulnerabilities** tab appears with the list of vulnerabilities. The **Vulnerabilities** table displays the following details:


Column	Description
<b>Severity</b>	This is the severity level of the vulnerability whether <b>Critical</b> , <b>High</b> , <b>Medium</b> , <b>Low</b> , and <b>Info</b> . For more information about how Tenable calculates severity, see <a href="#">CVSS vs. VPR</a> .
<b>Name</b>	The name of the vulnerability.
<b>CVSS3 Score</b>	The NVD-provided CVSSv3 impact score for the vulnerability. If the NVD did not provide a score, Tenable Cloud Security shows a Tenable-predicted score.
<b>Plugin family</b>	The plugin family for the vulnerability.
<b>Impacted resources</b>	The number of impacted resources.
<b>VPR Score</b>	The Vulnerability Priority Rating (VPR) assigned to the vulnerability.
<b>Last detected</b>	This is the time when the vulnerability was last detected.

4. To view the details of a vulnerability, click the vulnerability name.

The **Vulnerability details** plane appears with the following information:



Section	Description
<b>Vulnerability information</b>	Includes the details about the vulnerability such as the severity, plugin family, plugin ID, the ease of exploitation, and the patch publication date.
<b>VPR Key Drivers</b>	Gives the key drivers that Tenable uses to calculate the VPR of a vulnerability.
<b>Description</b>	Provides a description of the vulnerability.
<b>Solution</b>	Provides the solution to fix the vulnerability.
<b>Impacted Resources</b>	Lists the impacted resources and the detection date of the vulnerability on the resource.

5. To view specific vulnerabilities on the **Vulnerabilities** tab, do one of the following:
- Use the **Search** box to search by CVE or Plugin ID.
  - Use the following filters:
    - a. Click the  **Filters** icon to open the **Filter Vulnerabilities** box.
    - b. Select the following filters as needed.

Filter	Description
<b>Severity</b>	Filters the list by severity: critical, high, medium, or low.
<b>Plugin family</b>	Filters the list of vulnerabilities by plugin family name. Use the search box to search for a specific plugin family.
<b>VPR</b>	Filters by the vulnerability priority rating (VPR) score.
<b>Projects</b>	Filters the list by projects.
<b>Cloud provider</b>	Filters the list by cloud providers.




<b>Cloud accounts</b>	Filters the list by cloud accounts.
<b>Source</b>	Filters by the source of the vulnerability – Cloud or Image.

- c. Click **Apply Filters**.

Tenable Cloud Security applies the filters and displays the filtered vulnerabilities.


6. To export the list of vulnerabilities as a CSV, click  **Export > CSV**.

7. To add or remove columns from the **Vulnerabilities** table:

- a. Click  to display the column names.
- b. Select or deselect the check boxes next to the column name as needed.

Tenable Cloud Security displays the selected columns.

**Note:** You cannot remove the **Severity** and **Name** columns from the table and these are disabled.

8. Click  to refresh the vulnerabilities list.



# View Ignored Misconfigurations

You can ignore a policy for a resource if the policy is not applicable to a resource or you do not want to report the violation for that policy.

To view ignored misconfigurations:

1. In the left navigation bar, click **Findings**.  
The **Misconfigurations** page appears.
2. Click the **Ignored Misconfigurations** tab.  
The **Ignored Misconfigurations** page appears.
3. Click the ignored policy that you want to view.  
The **Ignored policy** pane appears.
4. In the **Ignored Resources** section, do one of the following:
  - Select the required ignored policy to view its details.
  - Filter and select the required ignored policy using one of the following filters:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by source.
Inference	Filters the impacted resources by inference.

5. Click **Export > CSV** to download the ignored misconfigurations report in the CSV format.

The report provides a project-wise listing of all ignored failing policies and includes the following details:

- Project
- Source (IaC or Cloud)
- Cloud Account
- VPC





- Policy Group
- Severity
- Failing Policy
- Resource Type
- Cloud ARN
- Cloud ID
- IaC ID
- IaC Repository
- Date and time the violation was last seen

6. To unignore or create a Jira ticket for the ignored policy:

- a. Select the checkbox next to the ignored resource that you want to unignore or create a Jira ticket.

Tenable Cloud Security enables **Un-ignore** and **Create a ticket**.

- b. Select one of the following options:

- **Un-ignore** – Tenable Cloud Security removes the issue from the ignored list. For more information, see [Unignore an Issue](#).
- **Create a ticket** – Creates a Jira ticket for the ignored issue. For more information, see [Create a Ticket for an Issue](#).

7. Click an ignored resource name to view the resource details. For more information, see [View Resource Details](#).



# View Resource Configuration

The configuration view in Tenable Cloud Security provides details about the resource that has policy violations. You can verify the configuration with the policy and update it to resolve the issue.

To view the configuration of a resource:

1. In the left navigation bar, click **Findings**.

The **Misconfigurations** page appears.

2. On the **Misconfigurations** page, click the failing policy that you want to view.

The policy pane with the list of impacted resources appears.

3. (Optional) In the **Impacted resources** section, use the following filters to filter the impacted resources:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by source.
Inference	Filters the impacted resources by inference.

4. Hover over the impacted resource that you want to view and click **Show Config**.

The **Config** window appears.

The screenshot shows a 'Config' window with a close button (X) in the top right corner. Below the title bar, there are fields for 'Resource Name' and 'Resource Type'. A 'Show Accurics Recommendation' checkbox is visible in the top right of the main content area. The main content is a code editor displaying a JSON configuration. The configuration is as follows:

```
1 {
2   "creation_token": " ",
3   "encrypted": true,
4   "kms_key_id": null,
5   "provisioned_throughput_in_mibps": " ",
6   "tags": {
7     "CostCenter": " ",
8     "Name": " ",
9     "Stack": "${var.environment}",
10    "environment": "${var.environment}",
11    "terraform_managed": "true"
12  },
13   "throughput_mode": "bursting"
14 }
```



# Compare Resource Configurations

When reviewing violations in the Tenable Cloud Security console, you can compare versions of the resource from different sources, such as IaC vs. Cloud or Cloud vs. Cloud. You can even view the configuration of the impacted resources. You can either view the configuration of individual resources or view the configuration of resources in comparison.

To view the resources impacted due to policy violations:

1. In the left navigation bar, click **Findings**.

The **Misconfigurations** page appears.

2. On the **Misconfigurations** page, do one of the following:
  - Use the **Search** box to search for specific failing policies.
  - Filter the failing policies using one of the following filters:

Filter	Description
Projects	Filters the failing policies by projects.
Cloud accounts	Filters the failing policies by cloud accounts.
Repositories	Filters the failing policies by repositories.
More filters	Filters the results by <b>Severity</b> , <b>Policy Groups</b> , or <b>Source Types</b> .
Show Results	Displays the filtered results.

3. Click the required failing policy name to view its details and perform the following steps:
  - a. In the **Impacted resources** section, click the required impacted resource.  
The **Resource Details** tab appears.
  - b. Click **Drifts** to view the comparisons between cloud and IaC.
  - c. Click **Resource Configurations** to view the Cloud and IaC configurations.



## View Resources

The **Resources** page shows all the IaC and cloud resources connected to Tenable Cloud Security.

To view resources:

1. In the left navigation bar, click **Resources**.



The **Resources** page appears. The **Resources** page includes the following two tabs:

- **Resources** – Displays the list of resource types and the number of resources, findings, or configuration drifts.
- **Resources with Drift** – Displays only the list of resources types that have drifts.

Both tabs display the following details:

Section	Description
<b>Search Resource</b>	<p>Use the <b>Search Resource</b> box to search for specific resources. A drop-down next to the <b>Search Resource</b> box lets you filter the resources by:</p> <ul style="list-style-type: none"><li>• <b>Resource ID</b> – Filters by resource ID.</li><li>• <b>Resource Name</b> – Filters by resource name.</li><li>• <b>Resource ARN</b> – Filters by resource ARN.</li><li>• <b>Source</b> – Filters by source type: IaC or cloud.</li><li>• <b>Region</b> – Filters by regions.</li><li>• <b>Cloud VPC</b> – Filters by Virtual Private Cloud (VPC).</li></ul>
<b>Categories</b>	<p>Displays the number of resource types in each resource category. The number of <b>Findings</b> show the total number of vulnerabilities and mis-configurations for that resource type.</p> <p>You can view the resource types and findings for all or individual categories such as IaaS, Networking, Object storage, RDBMS, Serverless, and Others.</p>



	<p>You can click the  button to show or hide <b>Categories</b>. When you hide <b>Categories</b>, the category icons are still visible and you can click the category icons to filter the resource types.</p>
<b>Filter drop-down box</b>	<p>Displays the filter options by which you can filter the resource types. Click the  <b>Filters</b> icon to open the <b>Filter Resources</b> box. The following filter options are available:</p> <ul style="list-style-type: none"><li>• <b>Projects</b> – Filters by project names.</li><li>• <b>Cloud Accounts</b> – Filters by cloud accounts.</li><li>• <b>Repository</b> – Filters by repositories.</li><li>• <b>K8s clusters</b> – Filters by Kubernetes clusters.</li><li>• <b>Source</b> – Filters by types: IaC, Cloud, State File, Mapped (IaC &amp; Cloud).</li><li>• <b>Insights</b> – Filters by the types of violations found: Exposed blob stores, Exposed databases, Read/write IAM, and Exposed security groups</li><li>• <b>Compliance State</b> – Filters by compliance states: <b>Has Violations</b>, <b>Has IaC Drifts</b>, and <b>Has Cloud Drifts</b>.</li><li>• <b>Resource Type</b> – Filters by resource types.</li><li>• <b>VPC Filter</b> – Filters by VPC source.</li></ul> <p>Select the required filters and click <b>Apply Filters</b>.</p>
<b>Resources</b>	<p>Displays the number of resources.</p>
<b>Findings</b>	<p>Displays the number of vulnerabilities and misconfigurations. Misconfigurations are results from a Misconfiguration Scan. Vulnerabilities are results from <a href="#">Agentless Assessment</a>. For more information, see <a href="#">Cloud Scans</a>.</p>
<b>Config drifts</b>	<p>The number of configuration changes for each resource type. It</p>



	includes the total IaC and cloud drifts for all resources in that resource type.
--	--

2. Click a resource type link to view the details of resources that belong to that resource type. The **Resource Types** table displays the following details:

Column	Description
<b>Resource ID</b>	Displays the resource ID with its name below the ID.
<b>Source</b>	Displays source type: IaC or Cloud.
<b>Cloud Account</b>	Displays the cloud account ID.
<b>Region</b>	Displays the region where the resource is located.
<b>Findings</b>	Displays the total findings that is the sum of the number of mis-configurations, vulnerabilities, and drifts.
<b>Cloud tag</b>	The label associated with the resource by the cloud provider.

You can view a summary of total resources, total findings, and configuration drifts for the selected resource type at the top of the **Resource Types** table.

3. Click the resource ID to view the [Resource Details](#) pane.



## View Resource Details

On the [Resources](#) page, you can click a resource ID to view the **Resource Details** page. The **Resource Details** pane displays the following details:

- [Resource Details](#)
- [Vulnerabilities](#)
- [Misconfigurations](#)
- [Drifts](#)

### Resource Details

This section displays the following resource details:

Section	Description
<b>Assets Information</b>	Provides details about the assets such as the cloud provider, cloud ID or IaC ID, resource ID, resource name, resource type, and so on.
<b>Additional Information</b>	Provides information such as drift, if mapped to cloud, compliance state, and repository.

### Vulnerabilities

This section displays the following details of the vulnerabilities:

Column	Description
<b>Severity</b>	The severity of the vulnerability: <b>High</b> , <b>Medium</b> , or <b>Low</b> .
<b>Failing policy</b>	The failing policy name.




<b>Source</b>	The source type where the vulnerability was detected: Cloud.
<b>Last detected on</b>	The date and time of the last detection.

## Misconfigurations

This section displays the following details of the misconfigurations:

Column	Description
<b>Severity</b>	The severity of the misconfiguration: <b>High</b> , <b>Medium</b> , or <b>Low</b> .
<b>Failing policy</b>	The failing policy name. Click View remediation details to view the remediation steps.
<b>Source</b>	The source type where the misconfiguration was detected: Cloud or IaC.
<b>Last detected on</b>	The date and time of the last detection.

Click the check box next to a failing policy name to enable the **More actions** button or click . You can perform the following tasks for the selected failing policy:

- **Escalate:** [Escalate or Share an Issue](#).
- **Create Ticket:** [Create a Ticket for an Issue](#).
- **Ignore:** [Ignore Misconfigurations](#).
- **Create PR:** [Create a Pull Request for an Issue](#).

**Note:** You can create pull requests only for IaC scans in policies that support remediation (version 2).

## Drifts

This section displays the configuration drifts between the previous or baseline cloud configuration with the current cloud configuration. You can also compare the resource configuration in IaC and cloud. Filter the results using the following drift values:

- Computed
- Missing in IaC – Filters by the missing code in IaC resource.
- Missing in Cloud – Filters by the missing code in cloud resource.





All the three drift values are selected by default.

Click the check box next to a resource to enable the **More actions** button. You can perform the following tasks for the selected resource:

- **Create Ticket:** [Create a Ticket for an Issue](#).
- **Share:** [Escalate or Share an Issue](#).

For more information, see [Set up Drift Analysis](#).



---

## Remediate Issues

---

Remediation is the process of correcting issues to bring resources into compliance. In Tenable Cloud Security, you can accomplish the remediation for the policy violations in different ways. You can either enable auto-remediation or manually take the necessary actions that update the configuration of the existing or new resources from the connected cloud providers.

Tenable Cloud Security provides the following options to take appropriate actions to remediate the policy violations.

- [Set up Auto-Remediation](#)
- [Set up Inline Reviews](#)
- [Escalate the issue](#)
- [Create a pull request \(PR\) for the issue](#)
- [Create a ticket for the issue](#)
- [Ignore an issue](#)
- [Unignore an Issue](#)
- [View and Remediate the Line of Change in IaC](#)



# Set up Auto-Remediation


You can use the **Auto-Remediate** setting as the remediation type for your repositories to automatically create pull requests when Tenable Cloud Security detects any violation in the IaC scan. The working of auto-remediation depends on whether you have enabled the webhook for monitoring the repositories.

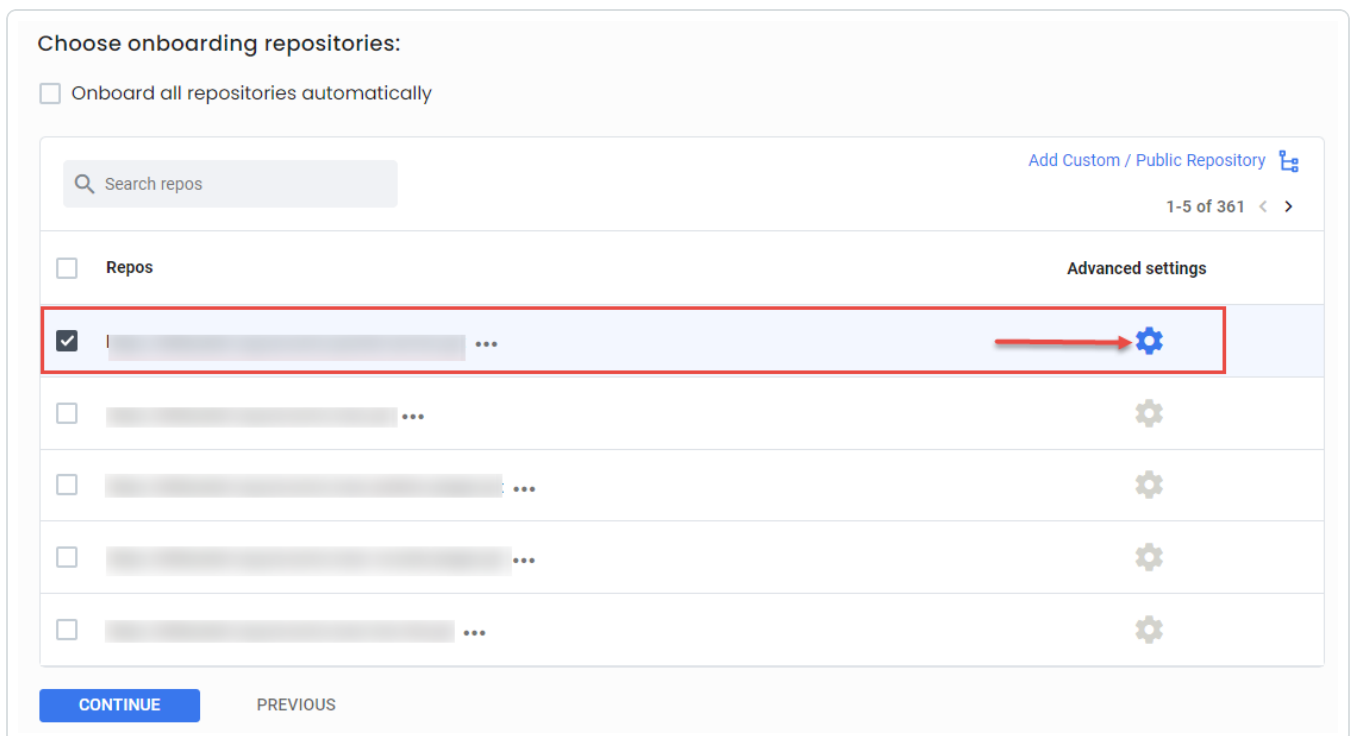
Before you begin:

The following permissions are required:

- Admin-level privileges to the repository to enable webhook.
- Write access to the repository to enable auto-remediation.


To set up auto-remediation for your repositories:

1. Navigate to the [Connect to repository](#) page and select the version control workflow.
2. On the **Choose onboarding repositories** section, select the repository and click the  icon.








Choose onboarding repositories:

Onboard all repositories automatically

Search repos Add Custom / Public Repository 

1-5 of 361 < >

<input type="checkbox"/> Repos	Advanced settings
<input checked="" type="checkbox"/> [Redacted] ...	
<input type="checkbox"/> [Redacted] ...	
<input type="checkbox"/> [Redacted] ...	
<input type="checkbox"/> [Redacted] ...	
<input type="checkbox"/> [Redacted] ...	

**CONTINUE** PREVIOUS

3. In the **Advanced settings** window, perform the following:



- a. In the **laC Engine Type** box, select **Terraform** or **Terragrunt**.
- b. (Optional) Click the **Enable Webhook** toggle to allow Tenable Cloud Security to continuously monitor your repository for any changes.

If this option is enabled, Tenable Cloud Security continuously monitors the repositories and triggers an automatic laC scan for any code change in the monitored branch of the repository.

- c. From the **Remediation type** drop-down list, select the **Auto-remediate** option.

The behavior of the **Auto-Remediate** setting depends on the webhook setting in the previous step.

- **Webhook Enabled** – If webhook is enabled, Tenable Cloud Security continuously monitors the repositories in the project. Whenever there is a code change in the monitored branch (through a pull request, merge, or commit), Tenable Cloud Security triggers an automatic laC scan. If any violations are detected in the laC scan, Tenable Cloud Security automatically creates a pull request with fixes in that repository.
- **Webhook Disabled** – If webhook is disabled, you must manually run an laC scan. If any violations are detected in the laC scan, Tenable Cloud Security automatically creates a pull request with fixes in that repository.

- d. Click **Save** to save the changes.



laC engine type      Select terraform version

Terraform      0.12.x

Enable Webhook  
Optionally, you can enable the webhook so that the repository changes can be continuously monitored.

Remediation type

Auto-Remediate

While the webhook is disabled, the PR's and inline reviews will be generated only when the IaC scan is run manually.

Tenable Cloud Security scans the IaC code in the specified repository and then automatically adds the remediation code and creates a pull request to merge the changes to the branch, if any violations are found.



# Set up Inline Reviews


You can use the **Inline Reviews** setting as the remediation type when you want Tenable Cloud Security to add issues to the configured repository for any violations. The working of inline review depends on whether you have enabled the webhook for monitoring the repositories.

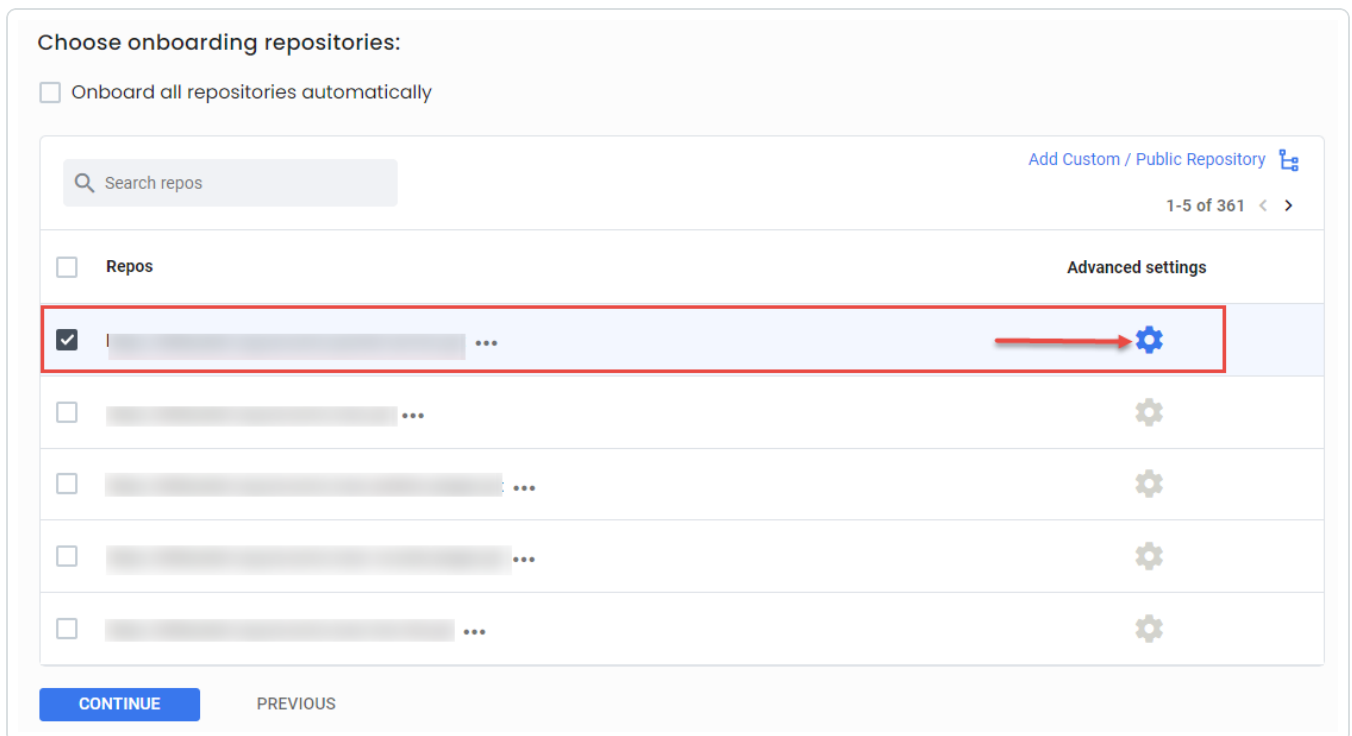
Before you begin:

The following permissions are required:

- Admin-level privileges to the repository to enable webhook.
- Write access to the repository to enable inline reviews.

To set up inline reviews for your repositories:

1. Navigate to the [Connect to repository](#) page and select the version control workflow.
2. On the **Choose onboarding repositories** section, select the repository and click the  icon.



3. In the **Advanced settings** window, perform the following:



- a. In the **laC Engine Type** box, select **Terraform** or **Terragrunt**.
- b. (Optional) Click the **Enable Webhook** toggle to allow Tenable Cloud Security to continuously monitor your repository for any changes.

If this option is enabled, Tenable Cloud Security continuously monitors the repositories and triggers an automatic laC scan for any code change in the monitored branch of the repository.

- c. From the **Remediation type** drop-down list, select the **Inline Reviews** option.

The behavior of the **Auto-Remediate** setting depends on the webhook setting in the previous step.

- **Webhook Enabled** – If webhook is enabled, Tenable Cloud Security continuously monitors the repositories in the project. Whenever there is a code change in the monitored branch (through a pull request, merge, or commit), Tenable Cloud Security triggers an automatic laC scan. If any violations are detected in the laC scan, Tenable Cloud Security adds issues to the monitored branch. Also, Tenable Cloud Security checks any upcoming pull requests for your monitored branch. If any violations are detected in the upcoming pull requests, Tenable Cloud Security adds comments to the pull requests.
- **Webhook Disabled** – If webhook is disabled, you must manually run an laC scan. If any violations are detected in the laC scan, Tenable Cloud Security adds issues to the monitored branch.

- d. Click **Save** to save the changes.



laC engine type      Select terraform version

Terraform      0.12.x

Enable Webhook

Optionally, you can enable the webhook so that the repository changes can be continuously monitored.

Remediation type

Inline Reviews

While the webhook is disabled, the PR's and inline reviews will be generated only when the laC scan is run manually.

Tenable Cloud Security scans the IaC code in the specified repository and then automatically creates issues for any violations found. The issue includes the line numbers that have the violation.





## Escalate or Share an Issue

If you want to notify a user about the misconfigurations for any resource, you can escalate the issue. Tenable Cloud Security sends an email alert for the misconfiguration.

Before you begin:

- Make sure that you configured in email alerts in Tenable Cloud Security. For more information, see [Email Alerts](#).

To escalate an issue:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.
3. Click the **Misconfigurations** tab.

The **Misconfigurations** page shows the failing policies and the number of impacted resources along with other details.

4. Do one of the following:
  - Select the required failing policy to view its details.
  - Use the **Search** box to search and select a specific failing policy.
  - Use the following filters to filter and select a specific failing policy:

Filter	Description
<b>Projects</b>	Filters failing policies by projects.
<b>Cloud</b>	Filters failing policies by cloud accounts.
<b>Repositories</b>	Filters failing policies by repositories.
<b>Severity</b>	Filters failing policies by the severity of the failing policy.
<b>Violations</b>	Filters failing policies by policy groups
<b>Source Type</b>	Filters failing policies by IaC or Cloud.

The policy details pane appears.



5. In the **Impacted resources** section, select the check box corresponding to the resource for which you want to raise an alert.

Use the following filters to select the impacted resources:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by IaC or Cloud.
Inference	Filters the impacted resources by inference.

6. Click **Share**.

Tenable Cloud Security sends an email alert for the selected issue and a message confirms the escalation.



## Create a Pull Request for an Issue

When code changes cause issues, Tenable Cloud Security makes the required fixes in the code and raises a pull request for the changes. When the pull request merges with the main repository, Tenable Cloud Security no longer reports the issue.

**Note:** Tenable Cloud Security can create pull requests only for the native IaC resources.

**Note:** You cannot create a single pull request for multiple violations. Create a separate pull request for each violation.

To create a pull request for an issue:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.
3. Click the **Misconfigurations** tab.

The **Misconfigurations** page shows the failing policies and the number of impacted resources along with other details.

4. Do one of the following:
  - Select the required failing policy to view its details.
  - Use the **Search** box to search and select a specific failing policy.



- Use the following filters to filter and select a specific failing policy:

Filter	Description
Projects	Filters failing policies by projects.
Cloud	Filters failing policies by cloud accounts.
Repositories	Filters failing policies by repositories.
Severity	Filters failing policies by the severity of the failing policy.
Violations	Filters failing policies by policy groups
Source Type	Filters failing policies by IaC or Cloud.

The policy details pane appears.

5. In the **Impacted resources** section, hover over the impacted resource that you want to remediate and click **> Create a pull request**.

Use the following filters to select the impacted resources:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by IaC or Cloud. <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"><b>Note:</b> You can create pull requests only for IaC resources.</div>
Inference	Filters the impacted resources by inference.

The **Remediation** window appears.

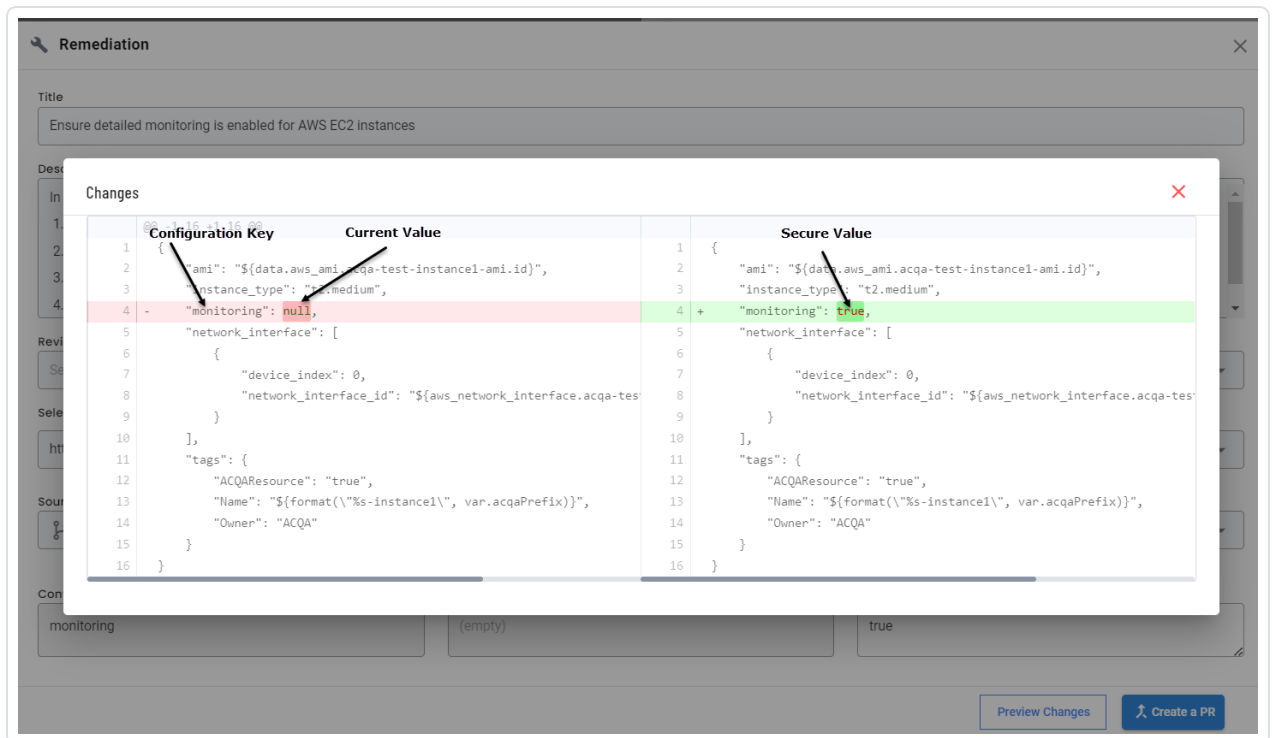
6. In the **Remediation** window, do the following:
  - a. (Optional) In the **Title** box, modify the title for the pull request.  
By default, the policy name is provided the title of the pull request.
  - b. (Optional) In the **Description** box, modify the default description for the pull request.



- c. In the **Reviewers** drop-down box, select a reviewer to review the changes before merging the change request with the main repository.
- d. In the **Source Branch** box, type the source branch.
- e. In the **Destination Branch** drop-down box, select the destination branch.
- f. In the **Secure Value** box, verify that the value displayed is correct.

The **Configuration Key** box displays the actual mismatched key and the **Current Value** box displays the value that you must replace.

- g. Click **Preview Changes** to view the changes.  
The **Changes** window appears.



- h. Click **X** to close the **Changes** window.
- i. Click **Create a PR**.

## 7. On the left navigation bar, click **Remediate > Fix PRs**.

Tenable Cloud Security displays all pull requests raised for the issues. The **Status** column displays the current status of the PR.



Service Tickets **0**

Fix PRs **1**

---

Projects Status **1** Clear filters Status Showing 1 results

Title	Repository	Source / Dest	Status ^	Last updated v	Reviewers
<a href="#">Ensure detailed monitoring is enabled for AWS EC2 instances</a>	AC10QA-ORG1/acqa-repo1-aws-tf12-part1	bugfix/accurics_remediation_9198926795039974 → master	<b>OPEN</b>	11.22.22 @ 05:10 PM	



## Create a Ticket for an Issue

You can create a Jira ticket for an issue. The Jira ticket allows you to assign and track the issue until its resolution.

Before you begin:

- Make sure that you have integrated Jira with Tenable Cloud Security. For more information, see [Integrate with Atlassian Jira](#).

To create a ticket for an issue:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.
3. Click the **Misconfigurations** tab.

The **Misconfigurations** page shows the failing policies and the number of impacted resources along with other details.

4. Do one of the following:
  - Select the required failing policy to view its details.
  - Use the **Search** box to search and select a specific failing policy.
  - Use the following filters to filter and select a specific failing policy:

Filter	Description
<b>Projects</b>	Filters failing policies by projects.
<b>Cloud</b>	Filters failing policies by cloud accounts.
<b>Repositories</b>	Filters failing policies by repositories.
<b>Severity</b>	Filters failing policies by the severity of the failing policy.
<b>Violations</b>	Filters failing policies by policy groups
<b>Source Type</b>	Filters failing policies by IaC or Cloud.

The policy details pane appears.



5. In the **Impacted resources** section, select the check box corresponding to the resource for which you want to create a ticket.

Use the following filters to select the impacted resources:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by IaC or Cloud.
Inference	Filters the impacted resources by inference.

6. Click **Create a ticket**.

The **Create a Jira ticket** window appears.

7. (Optional) Modify the **Assignee** and **Priority** of the issue.

8. Click **Submit**.

A message confirms that Tenable Cloud Security created the Jira ticket for the issue. You can click the link to the issue in the message to view the Jira ticket.

9. On the left navigation bar, click **Remediate**.

The **Service Tickets** tab appears and shows all the JIRA tickets.





---

# Ignore Misconfigurations

---

If a reported misconfiguration is not valid, you can ignore it. When you ignore the misconfiguration, Tenable Cloud Security does not consider it as a violation. You can ignore a misconfiguration in the following ways:

- [Ignore a misconfiguration from the Tenable Cloud Security console](#)
  - [Findings page](#)
  - [Policies page](#)
- [Ignore a misconfiguration by modifying the resource file](#)

## Ignore a misconfiguration from the Tenable Cloud Security console

To ignore a misconfiguration from the **Findings** page:

1. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.

2. Click the **Misconfigurations** tab.

The **Misconfigurations** page shows the misconfigurations and the number of impacted resources along with other details.

3. Do one of the following:
  - Select the required misconfiguration to view its details.
  - Use the **Search** box to search and select a specific misconfiguration.



- Use the following filters to filter and select a specific misconfiguration:

Filter	Description
Projects	Filters misconfigurations by projects.
Cloud accounts	Filters misconfigurations by cloud accounts.
Severity	Filters misconfigurations by the severity of the misconfiguration.
Source	Filters the misconfigurations by the source – Cloud, IaC, or both.
K8s cluster	Filters by the name of Kubernetes cluster.
Policy group	Filters misconfigurations by policy groups
Benchmark	Filters by policy benchmarks.

The policy details panel appears.

4. In the **Impacted resources** section, select the check box corresponding to the resource for which you want to ignore the violation.

Use the following filters to select the impacted resources:

Filter	Description
Projects	Filters the impacted resources by projects.
Resource type	Filters the impacted resources by resource types.
Source types	Filters the impacted resources by IaC or Cloud.
Cloud accounts	Filters the impacted resources by cloud account ID.

5. Do one of the following:
  - Click **Ignore Selected** to ignore one or more selected resources for the selected policy.
  - Click **Ignore All** to ignore all the resources for the selected policy.

The **Ignore policy for selected resources** window appears and displays the count of resources to ignore and the policy for which the resources are ignored.

6. In the **Select reason for ignoring** drop-down box, select the reason.



7. In the **Ignore for** drop-down box, select the duration for Tenable Cloud Security to ignore the misconfiguration: Forever, 6 months, 2 months, 1 month, 2 weeks, 1 week, or 1 day.
8. In the **Comment** box, type your reason for ignoring the violation.
9. Click **Submit**.  
A message confirms that Tenable Cloud Security ignored the violation. You can view the ignored misconfiguration and the count of ignored resources in the **Findings > Ignored Misconfigurations** page.

To ignore a misconfiguration from the **Policies** page:

1. In the left navigation bar, click **Policies**.  
The **Policies** page appears.
2. Click the Filter icon and set the **Status** filter to Non-Compliant.  
Tenable Cloud Security shows all non-compliant policies or policies that have misconfigurations.
3. Click a non-compliant policy.  
The **Policy** details plan appears.
4. In the **Impacted resources** section, select the check box corresponding to the resource for which you want to ignore the violation.

Use the following filters to select the impacted resources:

Filter	Description
<b>Projects</b>	Filters the impacted resources by projects.
<b>Resource type</b>	Filters the impacted resources by resource types.
<b>Source types</b>	Filters the impacted resources by IaC or Cloud.
<b>Cloud accounts</b>	Filters the impacted resources by cloud account ID.

5. Do one of the following:



- Click **Ignore Selected** to ignore one or more selected resources for the selected policy.
- Click **Ignore All** to ignore all the resources for the selected policy.

The **Ignore policy for selected resources** window appears and displays the count of resources to ignore and the policy for which the resources are ignored.

6. In the **Select reason for ignoring** drop-down box, select the reason.
7. In the **Ignore for** drop-down box, select the duration for Tenable Cloud Security to ignore the misconfiguration: Forever, 6 months, 2 months, 1 month, 2 weeks, 1 week, or 1 day.
8. In the **Comment** box, type your reason for ignoring the violation.
9. Click **Submit**.

A message confirms that Tenable Cloud Security ignored the violation. You can view the ignored misconfiguration and the count of ignored resources in the **Findings > Ignored Misconfigurations** page.

## Ignore a misconfiguration by modifying the resource configuration file

**Note:** This task is applicable only for Terraform resource configuration files.

To ignore a misconfiguration by modifying the resource configuration file:

1. In your repository, open the resource configuration file and add the following comment to the file:

```
#ts:skip=<Policy_ID> <Skip_reason>
```

where:

- **Policy\_ID** is the ID of the policy you want to exclude.  
To find the policy ID, in the **Policies** tab, click the policy to view its details along with the policy ID.
- **Skip\_reason** is the descriptive reason for ignoring the policy during scan.

**Note:** To ignore multiple policies for a resource, add a comment line for each policy.

### Example



```
resource "aws_ami" "awsAmiEncrypted" {  
  #ts:skip=AC_AWS_0005 need to skip this rule  
  
  name          = "some-name"  
  
  ebs_block_device {  
    device_name = "dev-name"  
    encrypted   = "false"  
  }  
}
```

Tenable Cloud Security ignores the AC\_AWS\_0005 policy for the `aws_ami` resource during scan and does not report it as a violation.



# Unignore an Issue

You can unignore a violation that you previously configured for ignoring.

To unignore an issue:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.
3. Click the **Ignored Misconfigurations** tab.  
The **Ignored Misconfigurations** window appears.
4. On the **Ignored Policies** page, do one of the following:
  - Click the required ignored policy to view its details.
  - Use the **Search Policy** box to search and select a specific ignored policy.  
The **Ignored Policy** pane appears.
5. In the **Ignored resources** section, do one of the following:
  - Select the checkbox next to the ignored resource that you want to unignore.
  - Use the following filters to filter and select the ignored resources:

Filter	Description
Resource type	Filters the impacted resources by resource types.
Source	Filters the impacted resources by source.
Inference	Filters the impacted resources by inference.

Tenable Cloud Security enables **Un-ignore**.

6. Click **Un-ignore**.  
A message confirms that Tenable Cloud Security unignored the issue.

**Note:** Tenable Cloud Security moves the ignored issues to the **Misconfigurations** page.



## View and Remediate the Line of Change in IaC

For an IaC scan violation, you can view the exact line of code that needs correction. Tenable Cloud Security also provides a recommended configuration to remediate the issue.

To view and remediate:

1. [Access Tenable Cloud Security](#).
2. In the left navigation bar, click **Findings**.  
The **Vulnerabilities** page appears.

3. Click the **Misconfigurations** tab.

The **Misconfigurations** page shows the failing policies and the number of impacted resources along with other details.

4. Do one of the following:
  - Select the required failing policy to view its details.
  - Use the **Search** box to search and select a specific failing policy.
  - Use the following filters to filter and select a specific failing policy:

Filter	Description
<b>Projects</b>	Filters failing policies by projects.
<b>Cloud</b>	Filters failing policies by cloud accounts.
<b>Repositories</b>	Filters failing policies by repositories.
<b>Severity</b>	Filters failing policies by the severity of the failing policy.
<b>Violations</b>	Filters failing policies by policy groups
<b>Source Type</b>	Filters failing policies by IaC or Cloud.


The policy details pane appears.

5. In the **Impacted resources** section, hover over the impacted resource that you want to remediate and click **> Show config**.  
The **Config** window with the impacted resource configuration appears.



6. Click the **Show Tenable Recommendation** checkbox.

Tenable Cloud Security shows the recommended configuration alongside the impacted resource configuration.

7. Click  to copy the configuration.

A message confirms that Tenable Cloud Security copied the configuration to clipboard. You can use the recommended configuration to correct the impacted resource configuration.





---

## Fix a Configuration Violation for a Project

---

Tenable Cloud Security allows you to remediate the configurations in the repositories of one or more projects. You can provide the fix value for a configuration key and Tenable Cloud Security automatically applies the configuration key to the specified projects.

To remediate a configuration violation:

1. In the left navigation bar, click **Policies**.

The **Policies** page appears and displays the **Policies** tab.

2. In the row of the policy that you want to remediate, click **⋮ > Fix Configuration**.

The **Fix configuration** window appears.

3. Provide a remediation for the configuration:

- a. From the **Select configuration** drop-down box, select the **Configuration Key**.
- b. In the **Fix Value** box, type a value.
- c. In the **Project** drop-down box, select a project.
- d. (Optional) Click + to add more remediation details.
- e. Click **Save**.



---

## View Tenable Cloud Security Dashboards and Reports

---

Tenable Cloud Security includes **Dashboards** that display analytics and statistics for all projects and timelines. The Tenable Cloud Security **Reports** page shows you the compliance coverage and identifies the resources that are not compliant. For more information, see the following:

- [View the Misconfigurations Dashboard](#)
- [Vulnerabilities Dashboard](#)
- [View and Download Compliance Report](#)



## View the Misconfigurations Dashboard

The Tenable Cloud Security **Misconfigurations Dashboard** page displays analytics and statistics for all projects and timelines.

To view analytics and statistics:

1. From the **Home** page, do one of the following:
  - To view more details about a specific item, in any widget, click a number or link.
  - To view the summary for one or all project, in the upper-right corner of the page, click **Projects**.

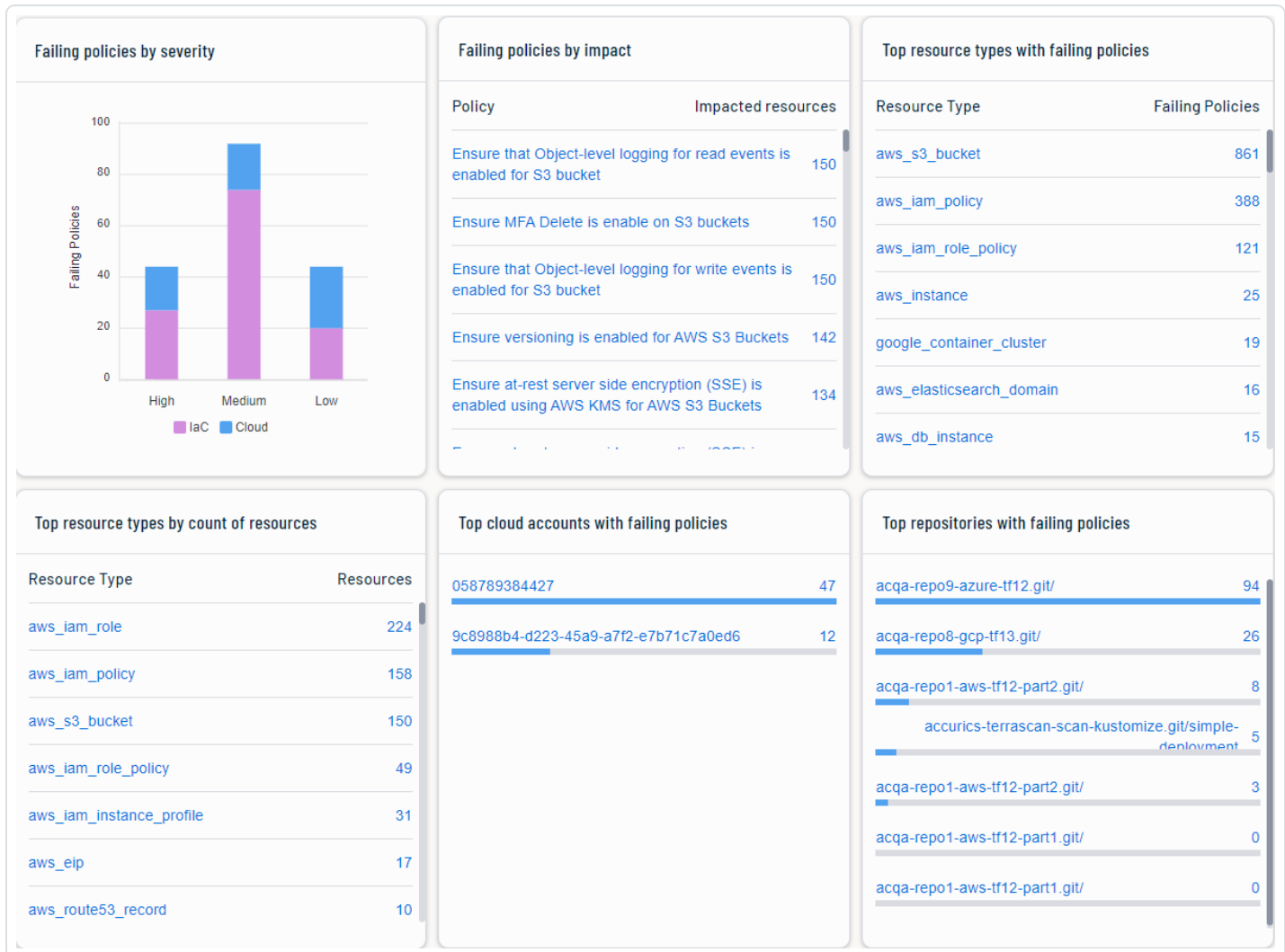
The following table describes the analytical widgets available on the **Dashboard** page:

Widget	Action
<b>Issues summary</b>	<p>This widget displays the total number of critical alerts and the total number of issues including the number of medium and high severity issues. It also displays the number of cloud and IaC drifts.</p> <p>Click a count to view the list of failing policies and issues. For more information, see <a href="#">View Misconfigurations</a>.</p>
<b>Critical security insights</b>	<p>This widget displays security insights for the policy violations and drifts detected on your resources, organized by:</p> <ul style="list-style-type: none"><li>• <b>Exposed BLOB Stores</b> – The number of unstructured Binary Large Object (BLOB) data stores in AWS, Azure, and Google Cloud Platform.</li><li>• <b>Exposed Databases</b> – The number of exposed databases on your account.</li><li>• <b>Read/Write IAM Roles</b> – The number of Amazon Web Service (AWS) Identity and Access Management (IAM) roles on your account with read and write permissions.</li><li>• <b>Exposed Security Groups</b> – The number of exposed security groups on your AWS account.</li></ul>



	<p>Click a count to view the details on the <b>Resources</b> page in Tenable Cloud Security. For more information, see <a href="#">View Resources</a>.</p>
<b>Remediation insights</b>	<p>This widget displays remediation insights for the issues detected on your resources, organized by:</p> <ul style="list-style-type: none"><li>• <b>Open Service Tickets</b> – The number of open vulnerability service tickets on your account.</li><li>• <b>Open "Fix" PR</b> – The number of service tickets on your account that remain open until you address the issues described in the corresponding pull requests.</li></ul> <p>Click a count to view the number of open tickets and pull requests on the <b>Remediate</b> page. For more information, see <a href="#">Remediate Issues</a>.</p>

On the **Misconfigurations Dashboard** page, you can also view the following statistical widgets:



Widget	Action
<b>Failing policies by severity</b>	This widget displays the failing policies of Infrastructure as Code (IaC) and cloud organized by policy type and severity (high, medium, and low).  Click a severity count to view the list of failing policies on the <b>Failing Policies</b> page.
<b>Failing policies by impact</b>	This widget displays the list of all the failing policies, organized in descending order by the number of impacted resources on each policy.  Click a failing policy name or the impacted resource count to view the details on the <b>Failing Policies</b> page.
<b>Top resource</b>	This widget displays the top resource types organized in descending order



<b>types with failing policies</b>	<p>by the number of failing policies.</p> <p>Click a resource type or a failing policy count to view the details on the <b>Resources</b> page.</p>
<b>Top resource types by count of resources</b>	<p>This widget displays the list of resource types organized in descending order by the number of resources.</p> <p>Click a resource type or a resource count to view the details on the <b>Resources</b> page.</p>
<b>Top cloud accounts with failing policies</b>	<p>This widget displays the top cloud accounts with the number of failing policies on the account, organized in descending order by the number of impacted resources on each policy.</p> <p>Click a cloud account or a failing policy count to view the details on the <b>Failing Policies</b> page.</p>
<b>Top repositories with failing policies</b>	<p>This widget displays the top repositories with the number of failing policies for the corresponding repository, organized in descending order by the number of impacted resources on each policy.</p> <p>Click a repository name or a failing policy count to view the details on the <b>Failing Policies</b> page.</p>



## View the Vulnerabilities Dashboard

The Tenable Cloud Security **Vulnerabilities** dashboard displays the vulnerabilities detected during a Vulnerability Scan using Agentless Assessment.

To view the **Vulnerabilities** dashboard:

1. [Access Tenable Cloud Security.](#)

The **Dashboards** page appears. The **Misconfigurations** tab is selected by default.

2. Click the **Vulnerabilities** tab.

The **Vulnerabilities** dashboard appears with several widgets showing key insights about the vulnerabilities detected by Tenable Cloud Security.

The following table describes the widgets on the **Vulnerabilities** dashboard:

Widget	Description
<b>Key Insights</b>	Provides a quick overview of actionable metrics, such as: <ul style="list-style-type: none"><li>• Total instances.</li><li>• Number of publicly exposed instances with vulnerabilities.</li><li>• Number of critical instances.</li><li>• Number of instances with critical vulnerabilities.</li><li>• Number of operating systems with critical vulnerabilities.</li></ul>
<b>Vulnerability distribution by severity</b>	Summarizes the number of vulnerabilities by <b>Critical</b> , <b>High</b> , <b>Medium</b> , and <b>Low</b> severity.
<b>Vulnerabilities per project</b>	Summarizes the number of vulnerabilities in each project, organized by Vulnerability Priority Rating (VPR). VPR is a dynamic metric that represents a vulnerability's likelihood for exploitation and its severity. Tenable recommends that you remediate these vulnerabilities with a higher VPR first.



<b>Top 5 vulnerabilities by impacted resources</b>	Lists the top five vulnerabilities with a high VPR affecting a high number of resources. Tenable recommends that you remediate these vulnerabilities first.
<b>Top OS / Hosts with critical &amp; high vulnerabilities</b>	Lists the top five operating systems or hosts affected with the maximum number of critical and high severity vulnerabilities. Tenable recommends that you remediate these vulnerabilities first.

3. Click a widget to view more details on the [Vulnerabilities](#) page.
4. Filter the vulnerabilities by clicking the **Source** filter – Cloud or Image.

By default, Tenable Cloud Security shows the total vulnerabilities in cloud and container images.





# View and Download Compliance Report

The Tenable Cloud Security **Reports** page shows the compliance reports for all resources based on the last scan. Use this report to view your compliance coverage and identify the resources that are not compliant. You can also download the reports in the CSV format.

To view compliance reports:

1. In the left navigation bar, click **Reports**.

The **Reports** page appears. The **Reports** page is grouped by **Benchmarks** by default. Click **Resource Type** to view the compliance report grouped by resource types.

The **Reports** page includes the following widgets:

Widget	Description
<b>Benchmark</b>	Select a benchmark from this drop-down list to filter the compliance report based on the selected benchmark. Click <b>Clear Filters</b> to clear the filters.  <b>Note:</b> Currently, Tenable Cloud Security does not map some policies with benchmarks. Compliance coverage percentage is calculated based on all applicable policies and might include policies that are not mapped to benchmarks.
<b>Compliance coverage</b>	The compliance coverage in percentage, calculated by dividing the number of passed policies from the total policies.
<b>Failed checks</b>	The number of failed policies.
<b>Last assessed</b>	The date and time of the last scan.

You can also view the compliance coverage in percentage for each policy category.

2. In the **Reports** page, do the following:



- Select one of the following filters to refine the compliance report:

Filter	Description
<b>Cloud provider</b>	Filters the compliance reports by cloud provider: AWS, Azure, or GCP. When you select a cloud provider using this filter, you can select only the relevant <b>Projects</b> , <b>Cloud accounts</b> , and <b>Repositories</b> for further filtering.
<b>Projects</b>	Filters the compliance reports by projects.
<b>Cloud accounts</b>	Filters the compliance reports by cloud accounts.

- In the **Policies** section, do one of the following:
  - Click any policy category to view the policies in that category. You can view the policy severity, cloud provider, resource type, compliance status (Compliant or Non-Compliant), and the date and time on which Tenable Cloud Security last assessed this policy.
  - Use the **Search** box to search for specific policies.
  - Click the **Expand All** check box to view an expanded view of all policies with their categories.
  - Filter the policies using one of the following filters:

Filter	Description
<b>Policy Status</b>	Filters the failing policies by one of the following statuses: <ul style="list-style-type: none"><li>• <b>Compliant</b>: Displays the policies that passed without any violations for all resources.</li><li>• <b>Non-Compliant</b>: Displays the policies that failed with violations for at least one resource.</li><li>• <b>Ignored</b>: Displays the policies that you have</li></ul>



	<p>ignored. For more information, see <a href="#">Ignore Mis-configurations</a>.</p> <div style="border: 1px solid blue; padding: 5px;"><p><b>Note:</b> The policy status becomes <b>Ignored</b> only when all the resources associated with the policy are ignored.</p></div> <ul style="list-style-type: none"><li>• <b>Not Assessed:</b> Displays the policies that are not applicable and are skipped from assessment.</li></ul>
<b>Severity</b>	Filters the failing policies by severity: <b>All</b> , <b>High</b> , <b>Medium</b> , or <b>Low</b> .

- Click a policy to view the policy details with the impacted resources for that policy.

You can view the IaC remediation code for the resource and the remediation steps for the policy violation.

3. Click **Export > CSV** to download the report in the CSV format.

a. Select the report that you want to download:

- **Summary Report:** Includes the summary of compliance coverage of all resources based on the last scan.
- **Detail Report:** Includes compliance summary and additional details, such as policy severity and status.

b. Click **Export**.



# Tenable Cloud Security Settings

The **Settings** page allows you to view and manage all of your settings and configurations. The Tenable Cloud Security **Settings** menu takes you to the Tenable Vulnerability Management **Settings** page.

To access the **Settings** page:

1. In the upper-left corner, click the ☰ button.

The left navigation plane appears.

2. Click **Settings**.

The **Settings** page appears.

Click on a tile to navigate to specific settings. For more information, see the following topics in the *Tenable Vulnerability Management User Guide*:

Topic	Description
<a href="#">General</a>	View and manage your general settings.
<a href="#">My Account</a>	View and manage your account settings.
<a href="#">SAML</a>	Manage SAML credentials and self service.
<a href="#">License</a>	View licensing details and statistics.
<a href="#">Access Control</a>	View and manage which hosts users can scan and can view in scan results and aggregated data.
<a href="#">Access Groups</a>	Manage access groups. <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"><p>Tenable is retiring access groups. Moving forward, Tenable recommends that you use <a href="#">permissions</a> to manage user and group access to resources on your Tenable Vulnerability Management instance and that you <a href="#">convert</a> your existing access groups into permission configurations. For more information, see <a href="#">Transition to Permission Configurations</a>.</p></div>
<a href="#">Activity Logs</a>	View activity logs for your organization's account.



<a href="#">Exports</a>	View export activity and manage scheduled exports.
<a href="#">Exclusions</a>	View and manage scanning restrictions.
<a href="#">Recast</a>	View and manage recast and accept rules.
<a href="#">Tagging</a>	View and manage tags and tagging rules.
<a href="#">Sensors</a>	Manage sensors and sensor groups.
<a href="#">Credentials</a>	View and manage scanning credentials.
<a href="#">Connectors</a>	Enable Frictionless Assessment and Cloud Connectors.



---

# Troubleshooting Issues with Tenable Cloud Security

---

This section lists common issues, their causes, and possible remediation actions.

When you contact Tenable Support for any scan issues, share the API token with Tenable Support to help troubleshoot your scan issue. For more information, see [Generate API Tokens](#).

See the following troubleshooting scenarios:

[Not Able to Find your Repository?](#)

[Seeing Duplicate Repositories?](#)

[Cloud Accounts cannot be Associated with this Project](#)

[Auto-Remediation not Working with On-Premises Scanner](#)



---

## Not Able to Find your Repository?

---

When onboarding repositories, one or more repositories do not appear in the **Connect to repository** page.

Repositories might be missing due to any of the following reasons:

- [Insufficient privileges to access the repositories.](#)
- [Repository in an unrecognized format or not an IaC repository.](#)
- [No authorization to access the GitHub organization.](#)
- [Connection to GitLab is reset.](#)
- [Repository inherited the third-party access setting from the parent repository.](#)
- [No admin access to the repository.](#)
- [Azure organization's security policies restrict access to the repositories.](#)

### Insufficient privileges to access the repositories

Tenable Cloud Security does not have sufficient privileges to access your private repositories.

#### **Solution:**

If the missing repositories are private repositories, grant access to Tenable Cloud Security to the private repositories. Depending on your version control system, use the following links to provide access to your repository.

- GitHub: [Approving OAuth Apps for your organization](#)
- GitLab: [Configure GitLab as an OAuth 2.0 authentication identity provider](#)
- Bitbucket: [Integrate another application through OAuth](#)

### Repository in an unrecognized or unsupported format

Tenable Cloud Security fails to discover the repositories because it was in an unrecognized format.

#### **Solution:**

Make sure the repository is in a format that Tenable Cloud Security supports. Tenable Cloud Security supports only the following IaC engine types:



- Terraform
- Terragrunt
- CloudFormation template
- Kubernetes YAML
- Helm Chart
- Kustomize YAML
- Azure Resource Manager

## No authorization to access the GitHub organization

Tenable Cloud Security does not have the authorization to access the GitHub organization of the repository.

### Solution:

Reset the connection of Tenable Cloud Security with GitHub by following these steps:

1. On the **Connect to repository** page, click **Previous** in the **Choose onboarding repositories** step.

The **Connect to a version control provider** step appears.

2. Click **Reset** to disconnect from GitHub.

A confirmation box appears.


3. Click **Yes** to confirm.

4. Click **GitHub** to connect to GitHub again.

Tenable Cloud Security Console redirects you to the sign-in page of the GitHub source code provider.

5. In the **Sign-in to GitHub** window, type your credentials.

6. Click **Sign in**.

Tenable Cloud Security connects to the source code provider. Once the connection succeeds, the **Reset** button and a  icon appear next to the source code provider.





## Connection to GitLab is reset

GitLab repositories are onboarded successfully, but these repositories disappear after some time. You might see this issue with GitLab repositories or on-premises scanner accessing GitLab repositories. The possible cause for this issue is that the connection to GitLab is automatically reset because the authentication token has expired. For more information about this issue, see [GitLab Token Unable To Refresh Due To Race Condition](#).

### Solution:

Reconnect and authenticate to GitLab.

- To connect to a GitLab repository, see [Integrate with GitLab](#).
- To connect to a GitLab repository using an on-premises scanner, see [Use an On-Premises Code Scanner to Scan GitLab Server IaCs](#).

## Repository inherits the third-party access setting from the parent repository

If the repository is forked from an existing repository, it inherits the **Third-party access** setting from the parent repository. If third-party access is restricted to the repository with this setting, Tenable Cloud Security cannot access your repository.

### Solution:

Allow the **Third-party access** setting from your repository.

## No admin access to the repository

Tenable Cloud Security does not have admin access to your repositories to set up a webhook. This webhook allows Tenable Cloud Security to test the pull requests and provide an accurate state of the vulnerabilities in your repositories.

### Solution:

Ask an administrator to grant you admin access to the repository via the repository's settings.

## Azure organization's security policies restrict access to the repositories



---

For Azure DevOps, Azure allows tenants to define which applications can gain access to Microsoft resources through their Conditional Access Policy (CAP) feature. It is possible that Tenable Cloud Security is unable to read the resources because of these policies.

**Solution:**

In the **Organization Settings** of Azure DevOps, ensure that the **Third-party application via OAuth** option under **Application policies** is enabled so that Tenable Cloud Security can read the repositories.

For more information about managing application connection policies, see [Change application connection & security policies for your organization](#) in Azure DevOps documentation.



---

## Seeing Duplicate Repositories?

---

After onboarding repositories, there are multiple entries for the same repository on the **Repositories** tab in the **Projects and Connections** page.

The **Repositories** tab shows multiple entries for the same repository in the following scenarios:

- The same repository is onboarded via the Tenable Cloud Security Console multiple times with different configuration parameters. For example, consider this scenario:
  - Onboard the repository `test-repo` with **Remediation Type** set to **Inline Review** and associate the repository with the project XYZ.
  - Onboard the repository `test-repo` with **Remediation Type** set to **Auto-Remediation** and associate the repository with the project XYZ.

In this case, Tenable Cloud Security two entries for the `test-repo` repository on the **Repositories** tab.

- The same repository is onboarded via the CLI multiple times by selecting different branches.
- The same repository is onboarded via the CLI multiple times by selecting different folders.

### Solution:

By design, Tenable Cloud Security shows multiple entries for a single repository if the repository is onboarded or scanned with multiple configurations.

To verify the repository configurations:

1. On the home page, click **Projects and Connections**.
2. Click the **Repositories** tab.
3. Click the repository name for which you want to view the configuration parameters.

The **Repository** pane appears.

4. Click **Settings**.

The configuration parameters set for the repository at the time of onboarding appear.



---

## Cloud Accounts cannot be Associated with this Project

---

You cannot associate cloud accounts to default projects created by Tenable Cloud Security. Tenable Cloud Security creates default projects in the following scenarios:

- Onboarding all repositories automatically

Tenable Cloud Security creates default projects for each SCM type. For example, **Default Gitlab Repositories**.

- Integrating Terraform cloud repositories

Tenable Cloud Security creates a default project, **Default\_TF\_Cloud\_Project**, when you start a new run for a Terraform repository.

### Solution:

Onboard the cloud account to a project and then move the repositories in the default project to the project you created for the cloud account.

Perform the following tasks:

1. [Create a project](#).
2. [Onboard a cloud account to the project](#).
3. In Tenable Cloud Security, go to **Projects and Connections**.
4. Click the **Repositories** tab.
5. Select the check box next to the repositories in the default project.
6. Click **Assign Project**.

The **Select a Project** page appears.

7. Select the project you created in [Step 1](#).
8. Click **Assign**.

The repository and cloud account now belong to the same project.

9. From the **Projects** tab, select the default project and click **Delete** to delete the default project.



# Auto-Remediation not Working with On-Premises Scanner

If you connect a GitLab repository to an on-premises code scanner and enable auto-remediation, automatic pull requests might not be created after the scan. Automatic pull request creation might fail if you use the IP address of the on-premises code scanner in the authorization callback URL instead of the fully qualified domain name.

## Solution:

Add the IP address of the on-premises scanner to the allow list of the GitLab server. Perform the following steps in GitLab:

1. On the top bar, select **Main menu > Admin**.
2. On the left sidebar, select **Settings > Network**.
3. Expand **Outbound requests**.
  - a. Select the **Allow requests to the local network from system hooks** check box.
  - b. In the **Local IP address and domain names that hooks and services may access** box, specify the IP address of the on-premise scanner host and port.

### Outbound requests Collapse

Allow requests to the local network from hooks and services. [Learn more.](#)

Allow requests to the local network from web hooks and services  
 Allow requests to the local network from system hooks

Local IP addresses and domain names that hooks and services may access

10.2 4:8 0

Requests to these domains and IP addresses are accessible to both system hooks and web hooks even when local requests are not allowed. IP ranges such as 1:0:0:0:0:0:0:0/124 and 127.0.0.0/28 are supported. Domain wildcards are not supported. To separate entries use commas, semicolons, or newlines. The allowlist can hold a maximum of 1000 entries. Domains must be IDNA encoded. [Learn more.](#)

Enforce DNS rebinding attack protection  
OutboundRequests|Resolve IP addresses once and uses them to submit requests.

For more information, see [Webhooks and insecure internal web services](#).