# Tenable.io Container Security

Last Updated: March 18, 2020

# Table of Contents

# Welcome to Tenable.io Container Security

Last Updated: March 18, 2020

This user guide describes Tenable.io® Container Security. Tenable.io Container Security stores and scans container images as the images are built, before production. It provides vulnerability and malware detection, along with continuous monitoring of container images. By integrating with the continuous integration and continuous deployment (CI/CD) systems that build container images, Tenable.io Container Security ensures every container reaching production is secure and compliant with enterprise policy.

**Tip:** If you are new to Tenable.io Container Security, see the workflow to get started.

**Video:** Introducing Tenable.io Container Security

## Other Tenable.io Products

### Tenable.io Vulnerability Management

See the User Guide

Tenable.io Vulnerability Management allows security and audit teams to share multiple Nessus scanners, scan schedules, scan policies, and scan results with an unlimited set of users or groups.

By making multiple resources available for sharing among users and groups, Tenable.io Vulnerability Management provides endless possibilities for creating customized workflows for vulnerability management programs, while accommodating the numerous regulatory or compliance drivers that demand you keep your business secure.

Tenable.io Vulnerability Management can schedule scans, push policies, view scan findings, and control multiple Nessus scanners from the cloud. This enables the deployment of Nessus scanners throughout networks to both public clouds, private clouds, and physical locations.

### Tenable.io Web Application Scanning

See the User Guide

Tenable.io Web Application Scanning offers significant improvements over the existing **Web Application Tests** policy template provided by the Nessus scanner which is incompatible with modern web applications that rely on Javascript and are built on HTML5. This leaves you with an incomplete understanding of your web application security posture.

Tenable.io Web Application Scanning provides comprehensive vulnerability scanning for modern web applications. Tenable.io Web Application Scanning has accurate vulnerability coverage that minimizes false positives and false negatives, ensuring that security teams understand the true security risks in their web applications. The product offers safe external scanning that ensures production web applications are not disrupted or delayed, including those built using HTML5 and AJAX frameworks.

# Get Started with Tenable.io Container Security

Complete the following tasks in the order listed to get started with Tenable.io Container Security.

1. Activate your account and log in to the web portal.
2. Review the requirements described in Tenable.io Container Security Requirements.
3. Review the user permissions assigned to each user role.
4. Generate Access and Secret keys for the Tenable.io API.
5. Import and scan your container images.

   If your images do not appear on the dashboard within 24 hour, contact support.

   - If you want to upload a specific image to Tenable.io Container Security for scanning, download the image from your external registry and push the image to Tenable.io Container Security.
   - If you want to import all the images from a registry to Tenable.io Container Security for scanning, configure a connector to import images from a registry.
   - If you want to scan an image directly from your organization's local registry, or from your machine, download and run the Tenable.io Container Security Scanner.

After you complete these initial tasks, you can navigate the Tenable.io Container Security dashboard to view and manage your scan data.

> **Note**: Tenable.io Container Security imports and rescans your images at regular intervals, beginning when you first import and scan the images.

# Tenable.io Container Security Requirements

You can access Tenable.io Container Security from any machine that meets the System Requirements described in the *Tenable.io Vulnerability Management User Guide*.

## Supported Container Image Formats

Tenable.io Container Security supports the following image formats:

| Import and Scan Method | Supported Image Types |
|---|---|
| Push a Container Image to Tenable.io Container Security | Docker images |
| Configure Connectors to Import and Scan Images | Docker images |
| Configure and Run the Tenable.io Container Security Scanner | • Docker images<br>• Open Containers Initiative (OCI) images |

## Supported Registries

The container registries that Tenable.io Container Security supports depends on the method you use to import and scan images.

Tenable tests and verifies successful import and scanning for the following registries:

| Import and Scan Method | Supported Image Types |
|---|---|
| Push a Container Image to Tenable.io Container Security | Docker registry |
| Configure Connectors to Import and Scan Images | • Amazon Web Service (AWS) Elastic Container Registry (ECR)<br>• JFrog Artifactory registry<br>• Docker registry |
| Configure and Run the Tenable.io Container | • Amazon Web Service (AWS) Elastic Container Registry (ECR) |

| | |
|---|---|
| Security Scanner | • Azure Container registry<br>• Docker registry<br>• Google Cloud Platform (GCP) Google Container Registry (GCR)<br>• JFrog Artifactory registry<br>• Nexus Repository Manager registry |

**Note**: Tenable.io Container Security supports importing and scanning from tested and verified registries that are compatible with Docker Registry API version 2.0.
If you choose to import and scan images from registries that have not been tested and verified, Tenable Support cannot assist with your configurations.

# Log in to Tenable.io Container Security

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

**Note**: This topic describes the new Tenable.io interface. To log in to Tenable.io Container Security in the classic interface, see Log in to Tenable.io Container Security in the Classic Interface.

To access Tenable.io Container Security and view and manage your container assets, you must first log in to the Tenable.io Container Security dashboard.

## Before you begin:

- Obtain credentials for your Tenable.io user account.

  **Note**: If you are an administrator logging in to your Tenable.io instance for the first time, Tenable provides your first-time credentials during setup. After you log in for the first time, you can set your new password. If you are logging in to Tenable.io after initial setup, your username is the email address you used to register for your Tenable.io account.

- Review the Tenable.io System Requirements in the *General Requirements User Guide* and confirm that your computer and browser meet the requirements.

## To log in to Tenable.io Container Security in the new interface:

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

   The Tenable.io login page appears.

2. In the username box, type your Tenable.io Container Security username.

3. In the password box, type the Tenable.io Container Security password you created during registration.

4. (Optional) To remain logged in until you sign out or close the browser, select the **Remember Me** check box. Otherwise, Tenable.io Container Security logs you out after a period of inactivity.

5. Click **Sign In**.

   The Tenable.io landing page appears. The landing page displays the Vulnerability Management dashboard.

6. In the upper-left corner, click the ☰ button.

The left navigation plane appears.

7. Click **Container Security**.

   The **Container Security** dashboard appears.

## To log in to Tenable.io Container Security via a Docker command:

> **Note**: If you use a Docker command to log in to Tenable.io Container Security Scanner, you can push images via the Docker command line interface (CLI). However, you cannot navigate the dashboard or use other interface features unless you log in through a browser.

1. Generate your API access and secret keys.
2. In the Docker CLI, run the following command:

   ```
   docker login registry.cloud.tenable.com
   ```

   The CLI prompts you to provide a username.
3. Type your API access key.
4. Press **Enter**.

   The CLI prompts you to provide a password.
5. Type your API secret key.
6. Press **Enter**.

   The Docker CLI logs you in to the Tenable.io Container Security registry.

# Log in to Tenable.io Container Security in the Classic Interface

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

**Note**: This topic describes the classic Tenable.io interface. To log in to Tenable.io Container Security in the new interface, see Log in to Tenable.io Container Security.

To access Tenable.io Container Security and view and manage your container assets, you must first log in to the Tenable.io Container Security dashboard.

## Before you begin:

- Obtain credentials for your Tenable.io user account.

  **Note**: If you are an administrator logging in to your Tenable.io instance for the first time, Tenable provides your first-time credentials during setup. After you log in for the first time, you can set your new password. If you are logging in to Tenable.io after initial setup, your username is the email address you used to register for your Tenable.io account.

- Review the Tenable.io System Requirements in the *General Requirements User Guide* and confirm that your computer and browser meet the requirements.

## To Log in to Tenable.io Container Security in the classic interface:

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

   The Tenable.io login page appears.

2. In the username box, type your Tenable.io Container Security username.

   **Note**: Your username is the email address you used to register for your Tenable.io account.

3. In the password box, type the Tenable.io Container Security password you created during registration.

4. (Optional) To remain logged in until you sign out or close the browser, select the **Remember Me** check box. Otherwise, Tenable.io Container Security logs you out after a period of inactivity.

5. Click **Sign In**.

   The Tenable.io landing page appears. The landing page displays the Vulnerability Management dashboard.

6. In the top navigation bar, click **Vulnerability Management**.
7. In the drop-down box, click **Container Security**.

   The **Container Security** dashboard appears.

# User Roles in Tenable.io Container Security

Your ability to view and configure features of Tenable.io Container Security depends on the Tenable.io user role you are assigned.

Though the user roles are the same throughout the Tenable.io platform, the permissions available to each role are specific to each product. For example, the permissions assigned to the Scan Operator role in Tenable.io Container Security are different from permissions assigned to the Scan Operator role in Tenable.io Vulnerability Management.

The following table briefly describes the available user roles and related permissions in Tenable.io Container Security. For detailed permissions information, see User Role Permissions.

| Role | Description |
|---|---|
| Basic | Limited to viewing, searching, and filtering Tenable.io Container Security data. |
| Scan Operator and Standard | Can import, manage, and delete images and image repositories, but may only use policies set by a scan manager user or higher. |
| Scan Manager | In addition to scan operator privileges, can create, manage, and enforce policies. |
| Administrator | Has all permissions, is responsible for setting up the account, adding and managing users, and configuring connections to registries. |

# User Role Permissions

User roles allow you to manage permissions for user accounts in Tenable.io Container Security, controlling which Tenable.io Container Security resources users can access once logged in.

The following table describes the available roles and corresponding permissions in Tenable.io Container Security. Each user role encompasses the permissions of lower roles and adds new permissions.

The Administrator role has the most permissions. The Basic role has the fewest.

| Area | User Roles and Permissions | | | | |
|---|---|---|---|---|---|
| | Administrator | Scan Manager | Standard | Scan Operator | Basic |
| Dashboards | view | view | view | view | view |
| Usage Data | view [1] | view | view | view | view |
| Images | view, push to Tenable.io, delete [2] | view, push to Tenable.io, delete | view, push to Tenable.io, delete | view, push to Tenable.io, delete | - |
| Image Repository | view, search, delete | view, search, delete | view, search, delete | view, search, delete | view, search |
| Containers | view | view | view | view | view |
| Policies | create, view, edit, set permissions, delete | create, view, edit, set permissions, delete | - | - | - |
| Connectors | create, configure, view, delete | - | - | - | - |

[1] User with the Administrator role can view license information that is not available to other roles.

[2] Besides user with the Administrator role, users can delete only images that they imported. Administrator users can delete images for all users on an account.

| | | | | | |
|---|---|---|---|---|---|
| CS Scanner | download, view, configure, run | download, view, configure, run | download, view, configure, run | download, view, configure, run | download |
| Scan Results | view, search | view, search | view, search | view, search | view, search |
| Connectors | view, configure to import registries | view | view | view | view |

# Glossary of Terms

Tenable.io Container Security product documentation uses the following terms:

| Term | Description |
|---|---|
| CD System | A Continuous Deployment system. Typically used to monitor for successful builds that have passed tests, and to take those successful builds and push them to production environments, thus automating the deployment of the successful builds. |
| CI System | A Continuous Integration system. Typically used to monitor source control commits, such as merged pull requests in GitHub, to automatically trigger a build (to test) as the change in source control is detected. |
| CI/CD System | A Continuous Integration and Continuous Deployment system. Typically used to monitor source control commits, such as merged pull requests in GitHub, to automatically trigger a build (to test) as the change in source control is detected, and upon successful completion of the build and test phase, to take those successful builds and push them to production environments, thus automating the deployment of the successful build. |
| Container | A running instance of a container image. A container image that has been started or otherwise executed. |
| Container Image | An application hosted inside of a container image file (for example, ubuntu:14.04). |
| Container Image Tag | A specific release or version of an application hosted inside of a container (for example, 14.04). |
| Container Registry | A storage location for Container Images. Provides developers and continuous integration systems the ability to store containers that are pushed. |
| Continuous Deployment | A development practice where operations (or DevOps) automatically push successfully tested builds to production environments, making them immediately available. |
| Continuous Integration | A development practice where developers integrate code into a shared source control repository, regularly, as changes are made. |
| Image | An application hosted inside of a container image file (for example, ubuntu:14.04). |
| Image Tag | A specific release or version of an application hosted inside of a container (for |

| Term | Description |
| --- | --- |
| | example, 14.04). |
| Organization Admin | The role assigned to the first user registering for Tenable.io Container Security, at the time the Organization is created. If you have registered without an invitation, you were automatically assigned the role of Organization Admin and a new Organization was created for your account. |
| Registry | A storage location for Container Images. Provides developers and continuous integration systems the ability to store containers that are pushed. |
| Repository | A storage location or namespace, within the registry, for an image (for example, /org/tenable_io_container_security/approved/). |
| Tag | A specific release or version of an application hosted inside of a container (for example, 14.04). |
| User | The role assigned to invited users registering for Tenable.io Container Security, for pre-existing Organizations. If you have registered via an invitation, you were automatically assigned the role of User and you were added to the same Organization of the user who invited you. |

# Push a Container Image to Tenable.io Container Security

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

Use Docker commands to download the image from the external registry where it resides and import it to Tenable.io Container Security.

## Before you begin:

- Log in to Tenable.io Container Security via the Docker command.

## To push container image to Tenable.io Container Security:

1. Use the `docker pull` command to download the image from an external registry.

   ```
   docker pull alpine:latest
   ```

2. Use the `docker tag` command to add the `registry.cloud.tenable.com` tag.

   ```
   docker tag alpine:latest registry.cloud.tenable.com/alpine:latest
   ```

   **Note:** The `registry.cloud.tenable.com` tag prompts Docker to push the image to Tenable.io Container Security. If you do not add the `registry.cloud.tenable.com` tag, Docker automatically pushes the image to the Docker central repository.

3. Use the `docker push` command to push the tagged image to Tenable.io Container Security.

   ```
   docker push registry.cloud.tenable.com/alpine:latest
   ```

   Docker pushes the image to Tenable.io Container Security. Tenable.io Container Security scans the images for vulnerabilities.

## What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.

# Tenable.io Container Security Scanner

The Tenable.io Container Security Scanner (Tenable.io CS Scanner) allows you to securely scan container images without sending the images outside your organization's network. The Tenable.io CS Scanner takes an initial inventory, or snapshot, of the images you want to scan and sends the inventory to Tenable.io for analysis. You can then view scan data for the images alongside data for images imported normally to Tenable.io.

With the Tenable.io CS Scanner, you can scan:

- A specific image exported from a registry and stored locally on the machine where you install the scanner.
- All images hosted in a specific registry (e.g., a Docker registry).

You can configure and run the Tenable.io CS Scanner on any machine that meets the system requirements.

First, download the Tenable.io CS Scanner to your machine. Then, configure and run the Tenable.io CS Scanner.

After your scan completes, you can view the scan results in the Tenable.io Container Security dashboard.

# Tenable.io CS Scanner System Requirements

The machine where you want to run the Tenable.io Container Security Scanner must meet the following requirements.

## Software and Hardware Requirements

| Deployment Type | Software Requirements | RAM | Temporary Storage | CPU |
|---|---|---|---|---|
| Local | Able to run Linux containers | 2 GB | 15 GB | 64-bit multi-core, x86 compatible |

## Internet

The machine where you want to run the Tenable.io CS Scanner must have access to the Internet when you download and run the scanner.

## SSL Certificate Requirements

If the registry that hosts your images requires the HTTPS protocol, you must have an SSL certificate signed by a trusted Certificate Authority (CA) installed on the registry. Refer to your registry's documentation for installing an SSL certificate.

Mozilla's CA Certificate Store is the Tenable.io Container Security Scanner's trusted certificate authority.

**Note**: If you want the Tenable.io CS Scanner to scan the registry without verifying that a trusted CA signed the certificate, you must include the `ALLOW_INSECURE_SSL_REGISTRY` variable when you run the scanner. For more information, see Environment Variables.

## Supported Container Image Formats

The Tenable.io CS Scanner supports the following image formats:

- Docker images
- Open Containers Initiative (OCI) images

# Download the Tenable.io CS Scanner

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

Download the Tenable.io CS Scanner Docker image to the machine where you want to configure and run the Tenable.io CS Scanner.

## Before you begin:

- Confirm your machine meets the system requirements, as described in CS Scanner System Requirements.

## To download the CS Scanner:

1. In the **Connectors** section of the **Container Security** dashboard, click **Import**.

   The **Select a Connector** plane appears.

2. Under **CONTAINER SECURITY**, click **CS Scanner**.

   The **CS Scanner** plane appears with login credentials.

3. Copy or take a screenshot of the credentials to use later in the download process.

4. In the command line interface (CLI) on the machine where you want to download the Tenable.io CS Scanner, type:

   ```
   docker login tenableio-docker-consec-local.jfrog.io
   ```

5. Press **Enter**.

   The CLI prompts you to provide a username and password.

6. Update the fields using the credentials provided on the **CS Scanner** plane.

7. **Press Enter.**

   You are logged in to the Tenable.io CS Scanner.

8. Type the following to pull the latest version of the Tenable.io CS Scanner image:

```
docker pull tenableio-docker-consec-local.jfrog.io/cs-scanner:latest
```

9. Press **Enter**.

What to do next:

- Configure and run the Tenable.io CS Scanner, as described in Configure and Run the Tenable.io CS Scanner.

# Tenable.io CS Scanner Environment Variables

You must use the CLI on your computer to configure your environment variables and run the Tenable.io CS Scanner

You can configure and run the Tenable.io CS Scanner as many times as necessary, using any combination of registries and registry sources.

## Environment Variables

| Variable | Description | Type | Required | Supported Mode |
|---|---|---|---|---|
| TENABLE_ACCESS_KEY | Your Tenable.io API access key. | String | Yes | • Image Inspect<br>• Registry Import |
| TENABLE_SECRET_KEY | Your Tenable.io API secret key. | String | Yes | • Image Inspect<br>• Registry Import |
| IMPORT_REPO_NAME | The name of the Tenable.io CS Scanner registry where you want to import the image. This name cannot contain spaces. | String | Yes | • Image Inspect<br>• Registry Import |
| REGISTRY_URI | The URI of the registry from which you want to import the image. | String | No | Registry Import |
| REGISTRY_USERNAME | Your username for authenticating to the registry you want to scan.<br>Set this variable if you want to authenticate to the registry.<br>Your username variable depends on the registry you want to scan: | String | No | Registry Import |

| | | | | |
|---|---|---|---|---|
| | • Amazon Web Services (AWS) Elastic Container Registry (ECR) — Type your AWS access key ID as your username. For information about how to obtain your access key ID, see the *AWS Documentation*.<br>• Azure registry — Type your service principal ID for the registry. For more information about how to create a service principal, see *Azure Documentation*.<br>• Google Cloud Platform (GCP) Google Container Registry (GCR) — Type your GCR account client email as it appears in the `client_email` field in the service account private key JSON file. For information about how to create and download your service account private key, see the *Google Container Registry Documentation*.<br>• All other registries — Type the username you use to authenticate to the registry. | | | |
| REGISTRY_ PASSWORD | Your password for authenticating to the registry from which you want to import the image.<br>Set this variable if you want to authenticate to the registry.<br>Your password depends on the registry you want to scan. | String | No | Registry Import |

| | | | | |
|---|---|---|---|---|
| | • Amazon Web Services (AWS) Elastic Container Registry (ECR) — Type your AWS access secret key as your password. For information about how to obtain your access secret key, see the *AWS Documentation*.<br>• Azure registry — Type your service principal password for the registry. For more information about how to create a service principal, see *Azure Documentation*.<br>• Google Cloud Platform (GCP) Google Container Registry (GCR) — Type your GCR service account private key as it appears in the `private_key` field in the service account private key JSON file. For information about how to create and download your service account private key, see the *Google Container Registry Documentation*.<br>• All other registries — Type the password you use to authenticate to the registry. | | | |
| IMAGE_NAME_WHITELIST | Image name or tag assigned to images that you want the Tenable.io CS Scanner to include in your registry scan.<br>Include this variable if you want to run the Tenable.io CS Scanner in | String | No | Registry Import |

| | | | | |
|---|---|---|---|---|
| | Registry Import mode and you want the scanner to include only images with a certain name or tag in the scan. If you do not set this variable, Tenable.io CS Scanner scans all the images in your registry.<br><br>**Note**: You cannot include an IMAGE_NAME_WHITELIST variable and an IMAGE_NAME_BLACKLIST variable in the same scan configuration.<br><br>Your whitelist variable depends on whether you want to include images based on name, tag, or both.<br><br>• Name — Type the name assigned to images that you want included in the scan.<br><br>For example, if you type `-e IMAGE_NAME_WHITELIST=alpine`, the Tenable.io CS Scanner scans only images named `alpine`.<br><br>• Tag — Type the tag assigned to images that you want included in `*:<tag>` format.<br><br>For example, if you type `-e IMAGE_NAME_WHITELIST=*:latest`, the Tenable.io CS Scanner scans only images with the `latest` tag.<br><br>• Both — Type the image name and tag set assigned to images that you want included in | | | |

| | | | | |
|---|---|---|---|---|
| | `<image>:<name>` format.<br><br>For example, if you type `-e IMAGE_NAME_WHITELIST=alpine:latest`, only images named `alpine` that also have the `latest` tag are included in the scan.<br><br>**Tip**: You can specify multiple whitelist variables by separating each with a comma (e.g., `-e IMAGE_NAME_WHITELIST=alpine1, alpine2, alpine3, *:latest`). | | | |
| IMAGE_NAME_BLACKLIST | Image name or tag assigned to images that you want the Tenable.io CS Scanner to exclude from your registry scan.<br><br>Include this variable if you want to run the Tenable.io CS Scanner in Registry Import mode and you want the scanner to exclude certain images from the scan. If you do not set this variable, Tenable.io CS Scanner scans all the images in your registry.<br><br>**Note**: You cannot include an IMAGE_NAME_BLACKLIST variable and an IMAGE_NAME_WHITELIST variable in the same scan configuration.<br><br>Your blacklist list variable depends on whether you want to exclude images based on name, tag, or both.<br><br>• Name — Type the name | String | No | Registry Import |

assigned to images that you want excluded from the scan.

For example, if you type `-e IMAGE_NAME_BLACKLIST=alpine`, the Tenable.io CS Scanner excludes only images named `alpine`.

- Tag — Type the tag assigned to images that you want excluded from the scan in `*:<tag>` format.

  For example, if you type `-e IMAGE_NAME_BLACKLIST=*:latest`, the Tenable.io CS Scanner excludes only images with the `latest` tag.

- Both — Type the image name and tag set assigned to images you want excluded in `<image>:<name>` format.

  For example, if you type `-e IMAGE_NAME_BLACKLIST=alpine:latest`, only images named `alpine` that also have the `latest` tag are excluded from the scan.

**Tip**: You can specify multiple blacklist variable sets by separating each set with a comma (e.g., `-e IMAGE_NAME_BLACKLIST=alpine1, alpine2, alpine3, *:latest`).

| | | | | |
|---|---|---|---|---|
| IMPORT_INTERVAL_MINUTES | The frequency, in minutes, you want the Tenable.io CS Scanner to import and scan images from the selected registry.<br><br>Set this variable if you want the scanner to run repeatedly at set intervals.<br><br>If you do not set this variable, the Tenable.io CS Scanner imports and scans images from the selected registry only the first time you scan your registry.<br><br>**Note**: You can schedule the scanner to run at set intervals only when you scan a registry. You cannot set a schedule when you configure and run the scanner in Image Inspect mode. | Integer | No | Registry Import |
| DEBUG_MODE | If true, the Tenable.io CS Scanner adds additional information to the scan's log to assist with debugging.<br><br>**Note**: Tenable recommends that you include this variable only if Tenable Support requests it. | Boolean | No | • Image Inspect<br>• Registry Import |
| ALLOW_INSECURE_SSL_REGISTRY | If true, the Tenable.io CS Scanner accepts the registry's SSL certificate without verifying that a trusted Certificate Authority (CA) issued the certificate.<br><br>**Caution**: If Tenable accepts an SSL certificate without verifying that a trusted CA issued the certificate, your certificate may not be valid and your connections may not be secure. Therefore, Tenable recom- | Boolean | No | Registry Import |

| | | | | |
|---|---|---|---|---|
| | mends that you include this variable only during testing or debugging procedures. | | | |
| HTTP_CONNECTION_TIMEOUT_SECONDS | The amount of time, in seconds, that the Tenable.io CS Scanner waits for a response after sending a connection request to the registry. If the registry does not accept the connection request within this time span, Tenable.io CS Scanner cancels (times out) the request.<br><br>By default, the Tenable.io CS Scanner times out unanswered connection requests after 10 seconds. | Integer | No | • Image Inspect<br>• Registry Import |
| HTTP_IDLE_TIMEOUT_SECONDS | The amount of time, in seconds, that the Tenable.io CS Scanner waits for a response after sending a request for image data to the registry. If the registry does not respond within this time limit, the Tenable.io CS Scanner cancels (times out) the request.<br><br>By default, the Tenable.io CS Scanner times out unanswered requests after 60 seconds. | Integer | No | • Image Inspect<br>• Registry Import |
| HTTP_REQUEST_TIMEOUT_SECONDS | The amount of time, in seconds, that the Tenable.io CS Scanner allows a request to remain active (i.e., the amount of time the Tenable.io CS Scanner waits for the registry to accept a connection request and respond to a request for image data). If a request is still active after this time limit has passed, the Tenable.io CS Scanner cancels (times out) the request. | Integer | No | • Image Inspect<br>• Registry Import |

| | | | | |
|---|---|---|---|---|
| | By default, the Tenable.io CS Scanner times out active requests after 60 seconds. | | | |

# Configure and Run the Tenable.io CS Scanner

When you run the Tenable.io Container Security Scanner, you can configure it to scan a single image or all images hosted in a repository.

To scan a single image, configure and run the Tenable.io CS Scanner in Image Inspect mode.

To scan all images in a repository, configure and run the Tenable.io CS Scanner in Registry Import mode.

# Scan an Image via the Tenable.io CS Scanner

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

Run the Tenable.io CS Scanner in Image Inspect mode to scan a single image.

## Before you begin:

- Download the image you want to scan to your local machine.
- Confirm your local machine meets the system requirements, as described in CS Scanner System Requirements.
- Download the Tenable.io CS Scanner, as described in Download the CS Scanner.
- Prepare your environment variable value, as described in the Environment Variables.

## To run the Tenable.io CS Scanner in Image Inspect mode:

1. In the CLI of the machine where you want to run the scanner, type the customized configuration and command for your deployment type using the parameters defined below.

   **Note**: Some of the following variables are not required to run the scanner. For information about these variables and their definitions, see Environment Variables.

   ```
   docker save <your image name as it appears in the repository> | docker run \
   -e TENABLE_ACCESS_KEY=<variable> \
   -e TENABLE_SECRET_KEY=<variable> \
   -e IMPORT_REPO_NAME=<variable> \
   -i tenableio-docker-consec-local.jfrog.io/cs-scanner:latest inspect-image <Image name as you want it to appear in Tenable.io> \
   ```

2. Press **Enter**.

   The Tenable.io CS Scanner scans the image.

## What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.

# Scan a Registry via the Tenable.io CS Scanner

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

Run the Tenable.io CS Scanner in Registry Import mode to scan all images in a registry.

## Before you begin:

- Confirm your machine meetings the system requirements described in Tenable.io CS Scanner System Requirements.
- Download the Tenable.io CS Scanner, as described in Download the CS Scanner.
- Prepare your environment variable values, as described in the Environment Variables.
- (Optional) To scan images hosted in an Amazon Web Services (AWS) Elastic Container Registry (ECR), an Azure registry, or a Google Container Registry (GCR), prepare your registry as described in Prepare your Registry.

## To run the Tenable.io CS Scanner in Registry Import mode:

1. In the CLI of the machine where you want to run the scanner, type the customized configuration and command for your deployment type using the parameters defined below.

   **Note**: Some of the following variables not required to run the scanner. For information about these variables and their definitions, see Environment Variables.

```
docker run \
-e TENABLE_ACCESS_KEY=<variable> \
-e TENABLE_SECRET_KEY=<variable> \
-e IMPORT_REPO_NAME=<variable> \
-e REGISTRY_URI=<variable> \
-e REGISTRY_USERNAME=<variable> \
-e REGISTRY_PASSWORD=<variable> \
-e IMPORT_INTERVAL_MINUTES=<variable> \
```

```
-i tenableio-docker-consec-local.jfrog.io/cs-scanner:latest import-registry
```

2. Press **Enter**.

   The Tenable.io CS Scanner scans all images in the registry.

## What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.

# Prepare your Registry

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

You must prepare the following registries before you scan the registries via the Tenable.io CS Scanner.

- Amazon Web Service (AWS) Elastic Container Registry (ECR)
- Azure Registry
- Google Cloud Platform (GCP) Google Container Registry (GCR)

You do not need to prepare other registry types before scanning.

## Amazon Web Service (AWS) Elastic Container Registry (ECR)

For information about how to make specific configurations to your AWS ECR, see the *AWS Documentation*.

### To prepare your AWS ECR:

1. Configure your AWS ECR.
2. Obtain your AWS access keys.

   **Note**: Your AWS access keys consist of two parts: an access key ID and an access secret key. The access key ID is your registry username variable, and the secret access key is your registry password variable. For more information, see Tenable.io CS Scanner Environment Variables.

### What to do next:

- Scan your repository, as described in Scan a Registry via the Tenable.io CS Scanner.

## Azure Registry

For information about how to make specific configurations to your Azure registry, see the *Azure Documentation*.

### To prepare your Azure registry:

1. Configure your Azure registry.
2. Create a service principal for your Azure registry and assign the `AcrPull` role to the service principal.

### What to do next:

- Scan your repository, as described in Scan a Registry via the Tenable.io CS Scanner.

## Google Cloud Platform (GCP) Google Container Registry (GCR)

For information about how to make specific configurations to your GCP GCR, see the *Google Container Registry Documentation*.

### To prepare your GCP GCR:

1. Create a service account in GCR with the Project Viewer role.
2. Authenticate to your registry by creating and downloading a service account key as a JSON file.

### What to do next:

- Scan your repository, as described in Scan a Registry via the Tenable.io CS Scanner.

# The Tenable.io Container Security Dashboard

The **Container Security** dashboard acts as landing page for Tenable.io Container Security. This dashboard contains widgets that display high-level information about your containers, images and image repositories, and policies. Click a widget on the dashboard to view details about the item type or to import data items (e.g., images) into Tenable.io Container Security.

**Note**: For information about how Tenable.io Container Security evaluates risks for your assets, see Risk Metrics.

From the **Container Security** dashboard you can:

- View Container Details
- Configure Connectors to Import and Scan Images
- View Scan Results for Container Images
- Manage Image Repositories
- Delete an Image
- Manage Policies
- View Data Usage

**Note:** Tenable.io Container Security uses the new Tenable.io interface. For more information about navigating the new interface, see:

- Filter a Table in the New Tenable.io Interface
- Search a Table in the New Tenable.io Interface
- Log Out of the New Tenable.io Interface

# View Container Details

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

A container is a running instance of an image. You create containers from an image each time you run the image on your application. You can create multiple containers from a single image, and you can make changes to those containers without affecting the image from which you created them.

When you perform a scan on your system using Nessus or Nessus Agent, Tenable.io Container Security identifies the images and containers in the system and analyzes the containers for risk.

Tenable.io Container Security then displays the containers by scan status and risk level in the **Identified Containers** widget on the Container Security dashboard based on the results of the most recent scan.

**Note**: Tenable.io Container Security imports and rescans your images at regular intervals, beginning when you first import and scan the images.

## Before you begin:

- If Tenable.io Container Security has not yet scanned the source image used to create the container you want to analyze, use one of the following methods to import the image for scanning:
  - Pull an individual image from your repository and then push the image to Tenable.io Container Security.
  - Configure Connectors to Import and Scan Images stored in your organization's local registry.
  - Use the Tenable.io Container Security Scanner to scan your images directly from your organization's local registry or your machine.
- Run a Nessus scan on the network where your containers run, selecting the **Basic Network Scan** template and providing your network authentication credentials. For more information about scan templates, see Scan and Policy Templates in the *Nessus User Guide*.

  **Note**: Tenable.io Container Security imports data from Nessus to determine if there have ever been any changes made to files on the container. If Nessus detects file changes, Tenable recommends that you check your images and repositories and confirm that no one has accessed them without authorization.

> **Tip**: Alternatively, you can run a Nessus Agent scan on the network where the container runs. For more information, see the Nessus Agent User Guide.

## To view container details:

1. In the **Container Security** dashboard, find the **Identified Containers** widget. This widget categorizes your containers by risk and scan status.

   > **Note**: For information about how Tenable.io Container Security calculates container risk, see Container Risk.

2. Click the **Identified Containers** widget.
   The **Identified Containers** page appears. The identified containers table lists all the containers created from images scanned by Tenable.io Container Security.

3. In the identified containers table, you can:

   - Filter the identified containers table.
   - Search the identified containers table.
   - View the summary for your identified containers in the identified containers table.

| Column | Description |
| --- | --- |
| **Container ID** | The ID that the software your container runs on assigned to the container. |
| **Repository/Image:Tag** | The repository name, image name, and image tag (e.g., latest). |
| **Risk Score** | The risk score on a scale of 1-10. |
| **Scan Status** | Indicates whether Tenable.io Container Security has scanned the container's source image.<br>• ✓ — Tenable.io Container Security has scanned the source image.<br>• ⓘ — Tenable.io Container Security has never scanned the source image. |

| | |
|---|---|
| | **Note**: When you initiate an image import, Tenable.io Container Security immediately queues the image to be scanned. However, Tenable.io Container Security does not always complete the scan immediately. To prevent undetected vulnerabilities, Tenable recommends that you confirm any images marked as not scanned are imported for scanning. For information about how to import and scan images, see Get Started with Tenable.io Container Security. |
| **File Changed** | Indicates whether the Nessus scan detected any changes to container files.<br>**Note**: If file changes are detected, Tenable recommends that you check your images and repositories and confirm that no one has accessed them without authorization.<br>• ✓ — Nessus did not detect file changes during its scan.<br>• ⓘ — Nessus detected file changes during its scan. |
| **Vulnerabilities** | The number of vulnerabilities detected in the container. |
| **Malware** | The number of malware items detected in the container. |
| **Host IP** | The IP address for the server where the container runs. |

- View details for a specific container.
  a. In the identified containers table, click the row for the container you want to view. The identified containers details page appears.
  b. On the identified containers details page, you can:

| Tab | Action |
|---|---|
| **Vulnerabilities** | • View details for each vulnerability identified in the image your identified container links to:<br>  • In the **Severity** column, view the severity rating |

| | |
|---|---|
| | Tenable.io Container Security assigned the image.<br><br>**Note**: For information about how Tenable.io Container Security determines image risk, see Image Risk.<br><br>• In the **Exposure ID** column, view the vulnerability's ID.<br><br>**Note**: The authority that identifies a given vulnerability determines the vulnerability's ID format.<br><br>• In the **Risk Score** column, view the CVSSv2 score.<br><br>• In the **Release Date** column, view the date when the software on which the container runs released the vulnerability.<br><br>• Click a row in the vulnerabilities table.<br><br>The vulnerability details plane appears, containing details and remediation recommendations for the vulnerability. |
| **Malware** | • View details about malware detected in the identified container:<br><br>• In the **Infected File** column, view the name of each infected file as it appears on the container.<br><br>• In the **Risk Score** column, view the CVSSv2 score for each infected file. |
| **Images** | • View details about the image your container links to.<br><br>• In the **Image ID** column, view the image ID.<br><br>**Note**: The image ID automatically generates when the software that hosts your image (e.g., Docker) creates the image. |

| | |
|---|---|
| | • In the **Repository** column, view the local repository where the image resides.<br>• In the **Image Name** column, view the image name as it appears in the repository.<br>• In the **Tag** column, view the tag associated with the image (e.g., latest).<br>• Click a row in the image table.<br>The details page appears for the image your identified container links to. For information about the image details, see View Scan Results for Container Images. |
| **Package Inventory** | View details about the package in the image your identified container links to, including the package name, version, license, and type. |

# Configure Connectors to Import and Scan Images

**Required User Role:** Administrator

Connectors act as links to local or third-party registries. You can use connectors to access these registries and then import image data from them to Tenable.io Container Security.

To import and analyze container images, you must configure a connector to a registry or, in certain cases, to the registry's own connector.

After you configure your connectors, you can view and manage your connectors from the **Settings** page in Tenable.io. For more information about connectors, see Connectors in the *Tenable.io Vulnerability Management User Guide*.

**Note**: Tenable.io Container Security does not support connector configurations for Azure Container Registries (ACR). To import images from an ACR registry, use the Tenable.io Container Security Scanner.

## Tenable.io Container Security Connectors

Tenable.io Container Security supports image imports via the following connectors.

**Note:** Tenable.io Container Security does not support registry imports from Docker Hub.

| Connector | Description |
|---|---|
| Tenable.io Container Security Scanner | A command line operated, on-premises scanning tool that allows you to scan images without importing them into Tenable.io Container Security. To configure the Tenable.io Container Security Scanner, see Tenable.io Container Security Scanner. |
| Amazon Web Service (AWS) Elastic Container Registry (ECR) | Connector for assets hosted in an AWS Elastic Container Registry. To configure an AWS ECR connector and import assets, see Configure an AWS ECR Connector to Import Images.<br><br>**Note**: To import assets from an AWS ECR, Tenable.io Container Security requires read-only access to your AWS account. |

| | |
|---|---|
| Docker | Connector for assets hosted in a Docker-compatible registry.<br><br>**Note**: If your registry is not listed but is Docker-compatible, select this connector. For information about Docker-compatible connectors, see Docker documentation at https://docs.docker.com. |
| Docker EE | Connector for assets hosted in a Docker Enterprise Edition (EE) registry. |
| JFrog Artifactory | Connector for assets hosted in a JFrog Artifactory registry. |

**Note**: To configure a connector for a Docker, Docker EE, or JFrog Artifactory registry and import assets from the registry, see Configure a Local Connector to Import Images.

# Configure an AWS ECR Connector to Import Images

**Required User Role:** Administrator

To import and analyze images hosted in an Amazon Web Service (AWS) Elastic Container Registry (ECR), you must configure your AWS ECR connector.

## Before you begin:

- Activate your account and log in to Tenable.io Container Security, as described in Log in to Tenable.io Container Security.
- Confirm the images you want to import are stored in your organization's container registry.

## To configure a connector to an AWS Elastic Container Registry:

1. In the **Connectors** section of the **Container Security** dashboard, click **Import**.

   The **Select a Connector** plane appears.

2. In the **Container Security** section, click **AWS Elastic Container Registry**.

3. In the **URL** box, type the fully-qualified domain name of your ECR deployment (e.g., **`579133718396.dkr.ecr.us-east-2.amazonaws.com`**).

4. In the **User Name** box, type *AWS*.

5. In the **Password** box, type the base 64-encoded password used in the **`docker login`** command, which is generated by AWS CLI.

   **Note:** AWS ECR passwords expire every 12 hours. You must refresh your AWS token if more than 12 hours passes between imports of the same registry.

   **Tip:** If your ECR is in the us-east-2 region, you can run the **`aws ecr get-login --region us-east-2`** command to get the **`docker login`** command.

6. Do one of the following:

- To save the connector, click **Save**.

  > **Note**: If you click **Save**, Tenable.io saves your configured connector but does not import your assets. To launch a manual import for the connector, see Launch a Connector Import Manually in the *Tenable.io Vulnerability Management User Guide*.

- To save the connector and import your assets from the registry, click **Save & Import**.

  > **Note**: There may be a short delay before your assets appear in Tenable.io.

7. (Optional) Click **Back** to configure another connector.

## What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.

# Configure a Local Connector to Import Images

**Required User Role:** Administrator

To import and analyze images hosted in a local registry, you must configure your registry's connector.

## Before you begin:

- Activate your account and log in to the web portal, as described in Log in to Tenable.io Container Security.
- Confirm the images you want to import are stored in your organization's container registry.

## To configure a connector to a local container registry:

1. In the **Connectors** section of the **Container Security** dashboard, click **Import**.

   The **Select a Connector** plane appears.

2. In the **Container Security** section, click the name of the type of container registry you want to use. Alternatively, type the name of the registry in the search box.

   **Note:** If you want to connect to a registry that is not listed, contact Tenable Support to let Tenable know that you want your container registry to be officially supported. If your registry is not listed but is Docker-compatible, select Docker. For information about Docker-compatible connectors, see Docker documentation at https://docs.docker.com.

3. In the **URL** box, type your registry's URL.
4. In the **Port** box, type your registry's port ID.
5. In the **Username** box, type your username.
6. In the **Password** box, type your password.
7. Use the **Schedule Import** toggle to enable or disable scheduled imports.

   **Note**: By default, Tenable.io requests new and updated asset records every 12 hours.

   If enabled:

- In the Import text box, type the frequency with which Tenable.io sends data requests to the registry.
- In the drop-down box, select *Minutes*, *Hours*, or Days.

8. Do one of the following:
   - To save the connector, click **Save**.

     > **Note**: If you click **Save**, Tenable.io saves your configured connector but does not import your assets. To launch a manual import for the connector, see Launch a Connector Import Manually in the *Tenable.io Vulnerability Management User Guide*.

   - To save the connector and import your assets from the registry, click **Save & Import**.

     > **Note**: There may be a short delay before your assets appear in Tenable.io.

9. (Optional) Click **Back** to configure another connector.

## What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.

# View Scan Results for Container Images

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

After Tenable.io Container Security scans your container images, you can view the detailed scan results on the Tenable.io Container Security dashboard.

## Before you begin:

- Scan the container image you want to analyze using any of the following processes:
  - Pull an individual image from your repository and then push the image to Tenable.io Container Security.
  - Configure your connectors to import and scan images stored in your organization's local registry.
  - Use the Tenable.io Container Security Scanner to scan your images directly from your organization's local registry or your machine.

## To view scan results for container images:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Images** widget.

   The **Images** page appears.

2. In the images table, you can:
   - Filter the images table.
   - Search the images table.
   - View details for the image:
     a. In the images table, click an image row.

        The **Image Details** page appears.

     b. On the **Image Details** page, you can:

| Tab | Action |
| --- | --- |
| **Vulnerabilities** | • View vulnerability details for each vulnerability iden- |

| | |
|---|---|
| | tified in the image:<br>• In the **Severity** column, view the severity rating Tenable.io Container Security assigned the image.<br>**Note**: For information about how Tenable.io Container Security determines image risk, see Image Risk.<br>• In the **Vulnerability** column, view the vulnerability ID.<br>**Note**: The authority that identifies a given vulnerability determines the vulnerability's ID format.<br>• In the **Risk Score** column, view the CVSSv2 score.<br>• In the **Release Date** column, view the date when the software on which the image is hosted released the vulnerability.<br>• Click a row in the vulnerabilities table.<br>A vulnerability details plane appears, containing details and remediation recommendations for the vulnerability. |
| **Malware** | View details about malware identified in the image, including a list of infected files, the file types, and the MD5 and SHA256 digests of the file. |
| **Package Inventory** | View details about the package in the image your identified container links to, including the package name, version, license, and type. |
| **Layer Digest** | View the digest IDs for each layer in the image. |
| **Identified Containers** | • In the **Container ID** column, view the ID that the software your container runs on assigned to each con- |

| | |
|---|---|
| | tainer.<br>• In the **Hostname** column, view the name of the network on which each container runs.<br>**Note**: Not all networks have a hostname; some only have an IP address.<br>• In the **Host IP** column, view the IP address for the network on which each container runs.<br>• In the **Start Date** column, view the date when the container most recently started running. |

# Manage Image Repositories

You automatically create an image repository when you push an image to the registry.

To manage image repositories in Tenable.io Container Security:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Repositories** widget.

   The **Repositories** page appears, displaying an overview description of the repository.

2. In the repositories table, you can:

   - Search the table.

     **Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

     a. In the text box, type your search term or terms.

     b. Click the 🔍 button.

        Tenable.io filters the table by your search criteria.

        **Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

   - View details for an image in the repository.

     **Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

     a. In the repositories table, click the row of the repository that contains the image you want to view.

        The **Repository Details** page appears with an overview description of the repository. On the **Repository Details** page, the **Container Images** table appears, listing each image stored in the repository.

     b. In the **Container Images** table, click an image row to view additional details.

        The **Tags** page appears.

     c. In the **Container Tag** table, click a row to expand the **Activity Log** details plane for

that tag.

> **Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

- Delete an image repository.

  > **Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

  a. In the repositories table, click the row of the repository you want to delete.

     The **Repository Details** page appears.

  b. In the details section, next to **ACTIONS**, click the × button.

     A confirmation window appears.

  c. Click **Delete** to confirm.

     > **Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

# Delete an Image

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

To delete an image:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Images** widget.

   The **Images** page appears. This page contains a table that lists the images Tenable.io Container Security has imported and scanned.

2. In the images table, click the × button next to the image you want to delete.

   A **Confirm Deletion** window appears.

3. Click **Delete** to confirm the deletion.

   Tenable.io Container Security removes the image and all the vulnerabilities associated with that image.

# Manage Policies

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

To manage policies in Tenable.io Container Security:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Policies** widget.

   The **Policies** page appears. This page contains a table that lists the policies Tenable.io Container Security uses to evaluate container images.

   The table lists the policies in order of priority, as determined by Tenable.io Container Security.

2. In the policies table, you can:

   - Search the policies table.
   - Add a policy.
   - Edit an existing policy.
   - Delete a policy.

# Add a Policy

**Required User Role:** Scan Manager or Administrator

To add a policy in Tenable.io Container Security:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Policies** widget.

   The **Policies** page appears. This page contains a table that lists the policies Tenable.io Container Security uses to evaluate container images.

   The table lists the policies in order of priority, as determined by Tenable.io Container Security.

2. Next to the **Policies** heading, click the ⊕ button.

   The add policy plane appears.

3. In the text box, type a meaningful name for the policy.

4. In the **Repositories** section, select the repositories where Tenable.io Container Security applies the policy:

   - To apply the policy to all repositories, select **All Repositories**.
   - To apply the policy to one repository:

     a. Select **Specific Repository**.

     b. In the drop-down box, type the name of the repository where you want to apply the policy.

     c. Select the repository.

5. In the **Conditions** section, set the condition that triggers the policy.

6. In the **Enforcement Action** section, select a policy enforcement setting.

7. Click **Create Policy**.

   The new policy appears at the top of policy list on the **Policies** page.

   **Note:** By default, the system assigns the policy the highest priority (1). If you want to modify the

priority setting, edit the policy.

**Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

# Edit a Policy

**Required User Role:** Scan Manager or Administrator

To edit a policy in Tenable.io Container Security:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Policies** widget.

   The **Policies** page appears. This page contains a table that lists the policies Tenable.io Container Security uses to evaluate container images.

   The table lists the policies in order of priority, as determined by Tenable.io Container Security.

2. In the policies table, click a policy row.

   The edit policy plane appears.

3. Edit the policy name.

4. In the **Priority** box, type a number representing the priority for the policy.

   Tenable.io Container Security evaluates container images against policies in the priority order you specify.

   If you type a priority number that is already associated to another policy, the system accepts the new priority number and lowers the priority numbers for all policies below it.

5. In the **Repositories** section, select the repositories where Tenable.io Container Security applies the policy:

   - To apply the policy to all repositories, select **All Repositories**.
   - To apply the policy to one repository:

     a. Select **Specific Repository**.

     b. In the drop-down box, type the name of the repository where you want to apply the policy.

     c. Select the repository.

6. In the **Conditions** section, set the condition that triggers the policy.

7. In the **Enforcement Action** section, select a policy enforcement setting.

8. Click **Save**.

Tenable.io Container Security saves your changes and displays the updated information on the **Policies** page.

> **Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

# Delete a Policy

**Required User Role:** Scan Manager or Administrator

## To delete a policy in the policies table:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Policies** widget.

   The **Policies** page appears. This page contains a table that lists the policies Tenable.io Container Security uses to evaluate container images.

   The table lists the policies in order of priority, as determined by Tenable.io Container Security.

2. In the policies table, click the × button next to the policy you want to delete.

   **Tip:** Roll over the policy row to reveal the × button for that policy.

3. Click **Delete** to confirm the deletion.

## To delete a policy while viewing the policy configuration:

1. In the **Statistics** section of the **Container Security** dashboard, click the **Policies** widget.

   The **Policies** page appears. This page contains a table that lists the policies Tenable.io Container Security uses to evaluate container images.

   The table lists the policies in order of priority, as determined by Tenable.io Container Security.

2. In the policies table, click the row of the policy you want to delete.

   The **Edit Policy** plane appears.

3. In the **Actions** section, click the × button.

4. Click **Delete** to confirm the deletion.

   **Tip:** In the top navigation bar, click a link in the breadcrumb trail to return to a previous page.

# Policy Condition Settings

You can set one of the following conditions to trigger a policy in Tenable.io Container Security:

| Option | Description |
| --- | --- |
| CVSS | To set the maximum CVSS value that triggers the policy:<br>1. Click **Max CVSS Value**.<br>2. Select an operator from the drop-down box.<br>3. Type the CVSS trigger value. |
| CVE | To set a CVE or CVEs that trigger the policy:<br>1. Click **CVE**.<br>2. In the text box, type one or more CVE values in decimal format (0.0) in a comma-separated list. |
| Malware | To set the policy to trigger on malware:<br>1. Click **Malware**.<br>2. Select **True** in the drop-down box. |

# Policy Enforcement Settings

You can select one of the following enforcement actions for a policy in Tenable.io Container Security:

| Option | Description |
| --- | --- |
| Set Compliance Status to False | Use this action if you want to query Tenable.io Container Security for the policy compliance status of scanned container images.<br><br>If a scan of a container image identifies the condition specified in the policy, any API queries for the policy compliance status of the container image receive a *false* response (security test failed). For more information, see the description of the **/policycompliance** endpoint in the Tenable.io Container Security API guide.<br><br>This action is useful if you integrate Tenable.io Container Security with your CI/CD pipeline. For example, you can configure Jenkins to mark a build unstable if a container receives a failed compliance status from Tenable.io Container Security. |
| Prevent/Block "docker pull" | Prevents Docker from pulling any image from the Tenable.io Container Security registry that Tenable.io Container Security scanned and identified as having a condition specified in the policy. |

# Risk Metrics

Tenable.io Container Security uses the metrics described below to categorize your images and containers on the Tenable.io Container Security dashboard.

## Image Risk

Tenable.io Container Security assigns all vulnerabilities in an image a static severity category based on the vulnerability's CVSSv2 score.

Tenable.io Container Security designates severity for each vulnerability using the categories described below.

| Severity | Description |
|---|---|
| **Critical** | The vulnerability's CVSSv2 score is between 9.0 and 10.0. |
| **High** | The vulnerability's CVSSv2 score is between 7.0 and 8.9. |
| **Medium** | The vulnerability's CVSSv2 score is between 4.0 and 6.9. |
| **Low** | The vulnerability's CVSSv2 score is between 0.1 and 3.9. |
| **Unscored** | Tenable.io Container Security has not yet determined the vulnerability's risk score. |

## Container Risk

Tenable.io Container Security calculates a container's overall risk score by determining which vulnerability on the container has the highest CVSSv2 score, then rounding that score to the nearest whole number.

For example, if the highest risk score for a vulnerability on a container is 9.2, Tenable.io Container Security assigns the entire container a risk score of 9.

Tenable.io Container Security designates risk for each container using the categories described below.

| Category | Description |
|---|---|
| **Unscanned** | The container was created from an image that Tenable.io Container Security has never scanned for vulnerabilities. |
| **Low/Medium** | Tenable.io Container Security scanned the image and container and assigned a |

| | |
|---|---|
| **Risk** | risk score of 0–7. |
| **High Risk** | Tenable.io Container Security scanned the image and container and assigned a risk score of 8–10. |

# View Data Usage

**Required User Role:** Basic, Scan Operator, Standard, Scan Manager, or Administrator

Tenable.io Container Security displays your data capacity by used and available data in the **Usage** widget on the Container Security dashboard.

The **Usage** widget categorizes your data by licensed container images or gigabytes (GB), depending on which metric your license specifies. For more information about your license metrics, contact your Tenable representative.

## To view your data usage:

1. In the Container Security dashboard, locate the **Usage** widget.
2. View the following details about your data usage:

| Widget Section | Description |
|---|---|
| **Licensed Space** or **Licensed Images**, depending on your licensing scheme | The amount of data licensed to your account. |
| **Licensed Space Limit** or **Licensed Images Limit**, depending on your licensing scheme | The amount of licensed data still available. |
| **Space used** or **Licensed Images used**, depending on your licensing scheme | The amount of licensed data already in use, displayed as a percentage of your licensed data limit.<br><br>To calculate the data in use, Tenable.io Container Security:<br><br>• Identifies each image by the combination of container name, image registry, and version tag.<br><br>• Includes only the three most recent tags of the image against your licensed usage.<br><br>As a result, the Image widget may display an image count that does not match the amount of |

| | |
|---|---|
| | used licensed data the **Usage** widget displays.<br><br>For example, if your licensed image limit is 20, and you have 10 images already in use, your **Licensed Images used** percentage is 50%. |

# Integrations

You can push an image to Tenable.io Container Security from the following integrated platforms:

- Bamboo
- CircleCI
- Codeship
- Distelli
- Drone.io
- Jenkins
- Shippable
- Solano Labs
- Travis CI
- Wercker
- Kubernetes (supported for theTenable.io CS Scanner only)

# Bamboo

## Before You Begin

These instructions describe how to push a Docker image from Bamboo to Tenable.io Container Security.

These steps assume you are already comfortable using Bamboo and are already pushing Docker images to a public or private registry. If you are already using Bamboo, but have not built Docker container images, familiarize yourself with the Bamboo documentation Configuring the Docker task in Bamboo.

## Steps

1. Create a new Docker task for the relevant job.
2. In the **Task** box, type a description for the task.
3. Depending on whether you want the task to run, select or clear the **Disable this task** check box.
4. Select **Push a Docker image to a Docker registry command** and complete the settings.

   Bamboo builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# CircleCI

## Before You Begin

These instructions describe how to push a Docker image from CircleCI to Tenable.io Container Security.

These steps assume you are already comfortable using CircleCI and are already pushing Docker images to a public or private registry. If you are already using CircleCI, but have not built Docker container images, familiarize yourself with the CircleCI documentation Continuous Integration and Delivery with Docker.

## Click here for information about the `circle.yml` file.

If you are using CircleCI to build Docker container images, you should have a `circle.yml` file in your project source control repository that looks similar to the following example:

```
machine:
services:
- docker

dependencies:
override:
- docker info
- docker build -t circleci/elasticsearch .

test:
override:
- docker run -d -p 9200:9200 circleci/elasticsearch; sleep 10
- curl --retry 10 --retry-delay 5 -v http://localhost:9200

deployment:
hub:
branch: master
commands:
- docker push circleci/elasticsearch
```

The following lines in `circle.yml` instruct CircleCI to leverage Docker for the build process:

```
machine:
```

```
services:
- docker
```

The following lines in `circle.yml` instruct CircleCI to build the elasticsearch image in the `circleci/` repository:

```
dependencies:
override:
- docker info
- docker build -t circleci/elasticsearch .
```

The following are the most important lines for adding Tenable.io Container Security integration to CircleCI environments. These lines instruct CircleCI to use Docker to log in to the registry (in this case to Docker Hub, since no private registry is specified) and push `cicleci/elasticsearch` to the registry:

```
deployment:
hub:
branch: master
commands:
- docker login -u $DOCKER_USER -p $DOCKER_PASS
- docker push circleci/elasticsearch
```

## Steps

1. To add environment variables for the project in the CircleCI console, open the project, click **Project Settings**, then click **Environment Variables**.
2. Define the following variables:

| Variable | Description |
|---|---|
| TENABLE_IO_ CONTAINER_ SECURITY_EMAIL | The email that you use to log in to Tenable.io Container Security. |
| TENABLE_IO_ CONTAINER_ | The user name that you use to log in to Tenable.io Container Security. You can find this on the **Settings** page in Tenable.io Container |

| Variable | Description |
| --- | --- |
| SECURITY_USER | Security. |
| TENABLE_IO_CONTAINER_SECURITY_ENDPOINT | For hosted cloud users of Tenable.io Container Security, this value is `registry.cloud.tenable.com`. |

3. To add support for Tenable.io Container Security, update the `circle.yml` file as follows:

```
machine:
environment:
VERSION: 2.1.1
TAG: ${VERSION}
services:
- docker

dependencies:
override:
- docker info
- docker version
- docker build -t $TENABLE_IO_CONTAINER_SECURITY_
ENDPOINT/circleci/elasticsearch .

test:
override:
- docker run -d -p 9200:9200 $TENABLE_IO_CONTAINER_SECURITY_
ENDPOINT/circleci/elasticsearch; sleep 10
- curl --retry 10 --retry-delay 5 -v registry.cloud.tenable.com

deployment:
hub:
branch: master
commands:
- docker login -u $TENABLE_IO_ACCESS_KEY -p $TENABLE_IO_SECRET_KEY
- docker tag $TENABLE_IO_CONTAINER_SECURITY_ENDPOINT/circleci/elasticsearch
$TENABLE_IO_CONTAINER_SECURITY_ENDPOINT/circleci/elasticsearch:${TAG}
- docker push $TENABLE_IO_CONTAINER_SECURITY_
ENDPOINT/circleci/elasticsearch:${TAG}
- docker logout
```

CircleCI builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Codeship

## Before You Begin

These instructions describe how to push a Docker image from Codeship to Tenable.io Container Security.

These steps assume you are already comfortable using Codeship and are already pushing Docker images to a public or private registry. If you are already using Codeship, but have not built Docker container images, familiarize yourself with the Codeship documentation Pushing to a remote registry.

## Steps

1. Edit the **`codeship-services.yml`** file to use the repository name and image name specified in Tenable.io Container Security.

   ```
   app:
   build:
   image: repository_name/image_name
   dockerfile_path: Dockerfile
   ```

   **Note:** If this is the first time you are pushing an image into the repository, there is not a pre-configured image name. The image name is added automatically after the push from Codeship.

2. Edit the service section of the the **`codeship-steps.yml`** file to look similar to the following example:

   ```
   service:
   app type: push
   image_name: repository_name/image_name
   registry: registry.cloud.tenable.com
   encrypted_dockercfg_path: dockercfg.encrypted
   ```

   Codeship builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Distelli

## Before You Begin

These instructions describe how to push a Docker image from Distelli to Tenable.io Container Security using the Distelli WebUI Manifest.

These steps assume you are already comfortable using Distelli and are already pushing Docker images to a public or private registry. If you are already using Distelli, but have not built Docker container images, familiarize yourself with the Distelli documentation on the Distelli Manifest. You can use the Distelli manifest file by either using the Distelli WebUI Manifest, or by editing the **`distelli-manifest.yml`** file directly.

## Steps

1. Log in to Distelli and navigate to an application.
2. Click the **Manifest** tab.

   The **Build** section displays content similar to the following example:

   ```
   docker build --quiet=false -t $DOCKER_REPO:$DISTELLI_BUILDNUM .
   docker login -u $DOCKER_USERNAME -p $DOCKER_PW
   docker push $DOCKER_REPO:$DISTELLI_BUILDNUM
   ```

3. To add support for Tenable.io Container Security, modify the **Build** section to look like the following example:

   ```
   bash docker build --quiet=false -t $TENABLE_IO_CONTAINER_SECURITY_
   REPO:$DISTELLI_BUILDNUM . docker login -u $TENABLE_IO_ACCESS_KEY -p $TENABLE_
   IO_SECRET_KEY registry.cloud.tenable.com docker push $TENABLE_IO_CONTAINER_
   SECURITY_REPO:$DISTELLI_BUILDNUM
   ```

   This modification adds the Tenable.io Container Security URI to docker login.

   Distelli builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Drone.io

## Before You Begin

These instructions describe how to push a Docker image from Drone.io to Tenable.io Container Security.

These steps assume you are already comfortable using Drone.io and are already pushing Docker images to a public or private registry. If you are already using Drone.io, but have not built Docker container images, familiarize yourself with the Drone.io documentation How to build and publish Docker images.

If you use Drone.io to build Docker container images, you should already have a build script (usually a **`build.sh`** file) that looks like the following:

```
$ docker build -t docker-registry/image-name .
$ docker push docker-registry/image-name
```

## Steps

1. Open the **`build.sh`** file.
2. Append a docker login directive before the docker push directive in the script, as in the following example:

   ```
   $ docker build -t docker-registry/image-name .
   $ docker login -u $TENABLE_IO_ACCESS_KEY -p $TENABLE_IO_SECRET_KEY
   registry.cloud.tenable.com
   $ docker push docker-registry/image-name
   ```

   Drone.io builds for this project are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Jenkins

## Before You Begin

These instructions describe how to push a Docker image from Jenkins to Tenable.io Container Security.

These steps assume you are already comfortable using Jenkins and are already pushing Docker images to a public or private registry. If you are already using Jenkins, but have not built Docker container images, familiarize yourself with the documentation for the Jenkins CloudBees Docker Build and Publish plugin.

## Click here for instructions on how to install the CloudBees Docker Build and Publish plugin.

1. Log in to Jenkins.
2. Click **Manage Jenkins**, then click **Manage Plugins**.
3. Click **Installed**.

   A list of installed plugins appears.
4. Click **Available**.
5. In the **Filter** box, type **CloudBees Docker Build and Publish plugin**.
6. Select the check box that corresponds to the plugin.
7. Install the plugin.

   The CloudBees Docker Build and Publish plugin is installed and ready for use by Jenkins jobs.

## Steps

1. On the Jenkins dashboard, select the job you want to modify.
2. Click **Configure**.
3. In the **Build** section, click **Add build step**.
4. In the drop down box, select **Docker Build and Publish**.

5. Type the details for the following configuration parameters:

    - **Repository Name:** The repository name and image name. For example, if you build a rabbitmq container image, you can name the repository rabbitmq and the image rabbitmq. In this example, in the **Repository Name** box, type *rabbitmq/rabbitmq*.
    - **Tag:** The tag name. The simplest tag name to use is *latest*.
    - **Docker Host URI:** The Jenkins path to the Docker Host. If the Docker Host is running on localhost, then in the **Docker Host URI** box, type *tcp://127.0.0.1:4243*.
    - **Docker registry URL:** The Tenable.io Container Security API endpoint, which in this case is *registry.cloud.tenable.com*.
    - **Registry credentials:** The registry credentials that you select from the box.

      ### Adding registry credentials

      1. Click **Add**.
      2. Click **Username with password**.
      3. In the **Username** box, type your Tenable.io Container Security user name.
      4. In the **Password** box, type your Tenable.io Container Security password.
      5. Click **Add**.

         The credentials are added.

6. Click **Save**.

   Jenkins builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Shippable

## Before You Begin

These instructions describe how to push a Docker image from Shippable to Tenable.io Container Security.

These steps assume you are already comfortable using Shippable and are already pushing Docker images to a public or private registry. If you are already using Shippable, but have not built Docker container images, familiarize yourself with the Shippable documentation Building a Docker image.

## Steps

1. Log in to Shippable.
2. In the upper right corner of the screen, click the **Account Settings** button.
3. Click **Integrations**, and then click **Add Integration**.
4. In the **Master Integration** section, click **Private Docker Registry**.
5. In the **Name** box, type **Tenable.io Container Security**.
6. In the **URL** box, type **registry.cloud.tenable.com**.
7. In the **Username** box, type your Tenable.io Container Security user name.
8. In the **Password** box, type your Tenable.io Container Security password.
9. In the **Email** box, type the email address associated with your Tenable.io Container Security account.
10. Click **Save**.

    Your Tenable.io Container Security account is now available for hosting container images built by Shippable.
11. Access your project page, and click **Settings**.
12. Click **Hub**, and select the Tenable.io Container Security integration that you just created.
13. In the **Push Build** field, click **Yes**.

14. In the **Push image to** box, type the name of your repository and image in Tenable.io Container Security (e.g., `testrepo/nodejs`).
15. In the **Push Image Tag** box, select from the following options: **default**, **commitsha**, or **latest**.
16. Click **Save**.

    Shippable builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Solano Labs

## Before You Begin

These instructions describe how to push a Docker image from Solano Labs to Tenable.io Container Security.

These steps assume you are already comfortable using Solano Labs and are already pushing Docker images to a public or private registry. If you are already using Solano Labs, but have not built Docker container images, familiarize yourself with the Solano Labs documentation.

> **Note:** Solano Labs support for building Docker container images is in private beta. For customers interested in participating, Solano Labs recommends contacting Solano Labs support.

## Steps

1. Open the **`solano.yml`** file, which should look similar to the following example:

```
# Use docker-enabled workers (currently private beta - contact
support@solanolabs.com)
system:
docker: true
python:
python_version: 2.7
hooks:
pre_setup: |
set -ex
sudo apt-get update -qq
sudo docker pull jenkins
sudo docker build -t myrepo/jenkins-dsl-ready:my .
tests:
- python -m doctest build/resolve_jenkins_plugins_dependencies.py
```

2. Add a `post_build` phase with your Tenable.io Container Security user name.

```
# Use docker-enabled workers (currently private beta - contact
support@solanolabs.com)
system:
docker: true
```

```
python:
python_version: 2.7
hooks:
pre_setup: |
set -ex
sudo apt-get update -qq
sudo docker pull jenkins
sudo docker build -t myrepo/jenkins-dsl-ready .
post_build: |
docker login -u $TENABLE_IO_ACCESS_KEY -p $TENABLE_IO_SECRET_KEY
registry.cloud.tenable.com
docker push myrepo/jenkins-dsl-ready
tests:
- python -m doctest build/resolve_jenkins_plugins_dependencies.py
```

Solano Labs builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Travis CI

## Before You Begin

These instructions describe how to push a Docker image from Travis CI to Tenable.io Container Security.

These steps assume you are already comfortable using Travis CI and are already pushing Docker images to a public or private registry. If you are already using Travis CI, but have not built Docker container images, familiarize yourself with the Travis CI documentation Using Docker in Builds.

## Click here for information about the **travis.yml** file.

If you are using Travis CI to build Docker container images, you should have a **travis.yml** file in your project source control repository that looks similar to:

```
sudo: required
language: ruby
services:
- docker
before_install:
- docker build -t carlad/sinatra .
- docker run -d -p 127.0.0.1:80:4567 carlad/sinatra /bin/sh -c "cd /root/sinatra;
bundle exec foreman start;"
- docker ps -a
- docker run carlad/sinatra /bin/sh -c "cd /root/sinatra; bundle exec rake test"
script:
- bundle exec rake test
```

The following lines in **travis.yml** instruct Travis CI to leverage Docker for the build process:

```
sudo: required
services:
- docker
```

The following lines in **travis.yml** instruct Travis CI to build the sinatra image in the carlad/ repository:

```
before_install:
```

```
- docker build -t carlad/sinatra .
```

## Steps

1. Open the **`travis.yml`** file.
2. Add your Tenable.io Container Security credentials.

   ```
   $ travis encrypt TENABLE_IO_CONTAINER_SECURITY_EMAIL=email@organization.com
   $ travis encrypt TENABLE_IO_CONTAINER_SECURITY_USER=username
   $ travis encrypt TENABLE_IO_CONTAINER_SECURITY_PASSWORD=password
   ```

3. Add your environment variables.

   ```
   env:
   global:
   - secure: "UkF2CHX0lUZ...VI/LE=" # TENABLE_IO_CONTAINER_SECURITY_EMAIL
   - secure: "Z3fdBNPt5hR...VI/LE=" # TENABLE_IO_CONTAINER_SECURITY_USER
   - secure: "F4XbD6WybHC...VI/LE=" # TENABLE_IO_CONTAINER_SECURITY_PASSWORD
   - COMMIT=${TRAVIS_COMMIT::8}
   ```

4. Add your connection information.

   ```
   after_success:
   - docker login -u $TENABLE_IO_ACCESS_KEY -p $TENABLE_IO_SECRET_KEY
   registry.cloud.tenable.com
   - export REPO=sebestblog/travis-demo
   - export TAG=`if [ "$TRAVIS_BRANCH" == "master" ]; then echo "latest"; else
   echo $TRAVIS_BRANCH ; fi`
   - docker build -f Dockerfile -t $REPO:$COMMIT .
   - docker tag $REPO:$COMMIT $REPO:$TAG
   - docker tag $REPO:$COMMIT $REPO:travis-$TRAVIS_BUILD_NUMBER
   - docker push $REPO
   ```

   Travis CI builds are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Wercker

## Before You Begin

These instructions describe how to push a Docker image from Wercker to Tenable.io Container Security.

These steps assume you are already comfortable using Wercker and are already pushing Docker images to a public or private registry. If you are already using Wercker, but have not built Docker container images, familiarize yourself with the Wercker documentation Containers.

## Steps

1. In your project source control repository, open the **`wercker.yml`** file.
2. Add support for Tenable.io Container Security by changing the deploy directive as follows:

```
deploy:
steps:
- internal/docker-push:
username: $USERNAME
password: $PASSWORD
tag: my-amazing-tag
repository: turing/bar
registry: registry.cloud.tenable.com
```

   Wercker builds for this project are sent to Tenable.io Container Security for storage, distribution, vulnerability scanning, and malicious code scanning.

# Tenable.io Container Security Scanner with Kubernetes

You can run the Tenable.io Container Security Scanner with Kubernetes to securely scan container images without sending the images outside your organization's network. For more information, see Tenable.io Container Security Scanner.

- Tenable.io CS Scanner System Requirements for Kubernetes
- Prepare Kubernetes Objects to Configure and Run the Tenable.io CS Scanner
- Configure and Run the Tenable.io CS Scanner in Kubernetes

# Prepare Kubernetes Objects to Configure and Run the Tenable.io CS Scanner

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

You must prepare your Kubernetes namespace and secret objects before you can configure and run the Tenable.io CS Scanner in Kubernetes. The Tenable.io CS Scanner refers to these objects when it scans an image in Kubernetes.

Secrets contain sensitive information associated with the TENABLE_ACCESS_KEY, TENABLE_SECRET_KEY, REGISTRY_USERNAME, and REGISTRY_PASSWORD environment variables described in Environment Variables. To run the Tenable.io CS Scanner in Kubernetes, you must configure these secrets and deploy them to the registry where the image you want to scan is stored.

For more information about how to create objects in Kubernetes, see the Kubernetes documentation at kubernetes.io.

## Before you begin:

- Download the Tenable.io CS Scanner, as described in Download the CS Scanner.

## To prepare Kubernetes to configure and run the Tenable.io CS Scanner:

1. Log in to the CLI on the machine where you want to configure and run the Tenable.io CS Scanner.
2. In a text editor, create a namespace file (`tiocsscanner-namespace.yaml`) for your CS Scanner. For example:

   ```
   apiVersion: v1
   kind: Namespace
   metadata:
     name: tiocsscanner
     labels:
       name: tiocsscanner
   ```

3. Save and close the file.
4. Deploy the `tiocsscanner-namespace.yaml` file to Kubernetes. For example:

```
kubectl apply -f tiocsscanner-namespace.yaml
```

Your namepsace is configured and deployed.

> **Note**: The above command works only if the file is saved to the current working directory. If the file is saved somewhere other than the working directory, include the full path directory in the command. For example:
>
> **kubectl apply -f /home/jsmith/images/tiocsscanner-namespace.yaml**

5. Configure secrets for your Tenable.io access and secret keys. For example:

```
$ kubectl create secret generic tio
--from-literal=username=<Your Tenable.io access key>
--from-literal=password=<Your Tenable.io secret key>
--namespace=tiocsscanner
```

Your Tenable.io access key and secret key secrets are configured.

6. Configure secrets for your private registry username and password. For example:

```
$ kubectl create secret generic private_registry
--from-literal=username=<Your private registry username>
--from-literal=password=<Your private registry password>
--namespace=tiocsscanner
```

Your private registry username and password secrets are configured.

7. Deploy your secrets to the registry where the image you want to scan is stored. For example:

```
kubectl create secret docker-registry jfrog-tio
--docker-server=https://tenableio-docker-consec-local.jfrog.io
--docker-username=<Your username from the Tenable.io Container Security
console>
--docker-password=<Your password from the Tenable.io Container Security
console>
--docker-email=<Your email address>
--namespace=tiocsscanner
```

Your secrets are deployed to the registry.

## What to do next:

- Configure and run the Tenable.io CS Scanner in Kubernetes, as described in Configure and Run the CS Scanner in Kubernetes.

# Configure and Run the Tenable.io CS Scanner in Kubernetes

**Required User Role:** Scan Operator, Standard, Scan Manager, or Administrator

To scan images with the Tenable.io CS Scanner in Kubernetes, create a Kubernetes deployment file and deploy the file via the CLI on the machine where you want to run the scan.

## Before you begin:

- Confirm your machine meets the system requirements, as described in Tenable.io CS Scanner System Requirements.
- Download the Tenable.io CS Scanner, as described in Download the Tenable.io CS Scanner.
- Prepare Kubernetes to configure and run the Tenable.io CS Scanner, as described in the Prepare Kubernetes Objects to Configure and Run the Tenable.io CS Scanner.

## To configure and run the Tenable.io CS Scanner in Kubernetes:

1. In a text editor, open a new file.
2. Save the file as `tiocsscanner-deployment.yaml`.
3. Copy and paste the following text into the file, typing your specific variables where applicable:

   For information about these variables and their definitions, see Environment Variables.

```
apiVersion: v1
kind: Service
metadata:
  name: tiocsscanner
  namespace: tiocsscanner
  labels:
    app: tiocsscanner
spec:
  selector:
    app: tiocsscanner
  type: ClusterIP
```

```
  ports:
  - name: http
    protocol: TCP
    port: 5000
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    app: tiocsscanner
  name: tiocsscanner
  namespace: tiocsscanner
spec:
  minReadySeconds: 10
  replicas: 1
  selector:
    matchLabels:
      app: tiocsscanner
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: tiocsscanner
    spec:
      containers:
        - image: "tenableio-docker-consec-local.jfrog.io/cs-
scanner:latest"
```

```
name: tiocsscanner
resources:
  limits:
    cpu: "3"
  requests:
    cpu: "1.5"
    memory: "2Gi"
args:
 - import-registry
env:
  - name: TENABLE_ACCESS_KEY
    valueFrom:
      secretKeyRef:
        name: tio
        key: username
  - name: TENABLE_SECRET_KEY
    valueFrom:
      secretKeyRef:
        name: tio
        key: password
  - name: REGISTRY_USERNAME
    valueFrom:
      secretKeyRef:
        name: private_registry
        key: username
  - name: REGISTRY_PASSWORD
    valueFrom:
      secretKeyRef:
        name: private_registry
        key: password
  - name: IMPORT_REPO_NAME
```

```
            value: "<variable>"
          - name: REGISTRY_URI
            value: "<variable>"
          - name: IMPORT_INTERVAL_MINUTES
            value: "<variable>"
```

**Note**: If you are not pulling the images directly from the repository where they are hosted, append the following command to the end of the file, starting on a new line after the last variable:

```
imagePullSecrets
 -name: jfrog-tio
```

4. Save and close the file.
5. In the CLI on the machine where you want to run the scan, type the following to deploy the file:

```
kubectl apply -f tiocsscanner-deployment.yaml
```

**Note**: The above command works only if the file is saved to the current working directory. If the file is saved somewhere other than the working directory, include the full path directory in the command. For example:

```
/home/jsmith/images/tiocsscanner-namespace.yaml
```

6. Press **Enter**.

   The Tenable.io CS Scanner runs on Kubernetes.

7. Run the following command to confirm the scan ran successfully:

```
kubectl get pods --namespace=tiocsscanner
```

   The scan status log appears.

**Note**: If you receive error messages in the scan data, follow the error prompts to correct the issue.

What to do next:

- View the results of your scan, as described in View Scan Results for Container Images.