



# Red Hat OpenShift and Entrust CloudControl

Integration Guide

2024-02-12

© 2025 Entrust Corporation. All rights reserved.

# Table of Contents

1. Introduction
1.1. Product configurations
1.2. Requirements
2. Procedures
2.1. Download the CloudControl software
2.2. Deploy the CloudControl VM from the OVA
2.3. Power on the appliance
2.4. Configure the CloudControl virtual appliance
2.5. Set up the CloudControl GUI
2.6. OpenShift prerequisites
2.7. OpenShift setup
2.8. On-board the OpenShift Cluster
2.9. View OpenShift Kubernetes cluster inventory
2.10. Transfer root access control to CloudControl
2.11. Create a Trust Manifest Access Control Policy
2.12. Image registries
2.13. Image Deployment Control Policy
3. Additional resources and related products
3.1. CloudControl
3.2. Entrust products
3.3. nShield product documentation

# Chapter 1. Introduction

This guide describes how to integrate Red Hat OpenShift Kubernetes clusters with Entrust CloudControl. Entrust CloudControl organizes cluster inventory into categories to help the user find information about the OpenShift deployment. Entrust CloudControl uses role and asset-based access control to help define who can do what to which cluster objects. It also uses image deployment control policies that can be applied to cluster infrastructure ensuring ongoing compliance with the organization security policies.



Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

# 1.1. Product configurations

Entrust has successfully tested the integration of Entrust CloudControl with Red Hat OpenShift in the following configurations:

System	Version
OpenShift Server Version	4.11.20
OpenShift Kubernetes Version	v1.24.6+5658434
Entrust CloudControl	6.6.0

# 1.2. Requirements

Before starting the integration process, familiarize yourself with:

- The documentation and setup process for Red Hat OpenShift.
- The documentation and setup process for Entrust CloudControl. The online documentation contains everything needed to successfully install and deploy CloudControl.

# **Chapter 2. Procedures**

This guide uses a standalone CloudControl deployment and does not use Active Directory. All users are local to the system. CloudControl supports a cluster environment. For more information refer to Entrust CloudControl Installation Guide in the online documentation.

## 2.1. Download the CloudControl software

- 1. Go to https://my.hytrust.com/s/software-downloads.
- 2. Log in and select HyTrust CloudControl.
- 3. Open the folder HTCC\_6.6.0\_2023-02-24. This folder contains version 6.6.0 that was used in this guide.
- 4. Select the Entrust-CloudControl-6.6.0.660934.zip link to download the file.

fi -	Cases	s Knowledge Base 🗸	Product Documentation	Licenses	Software Downloads	Upgrade Center	Videos	
	sofv Fo	<sup>vare Downloads</sup> Iders and Files						
	• Hytrus	t CloudControl						
		Folder Name						
		HTCC_MoveRPVTool_2020-10-1	15					
		HTCC_Migration_Tool_2021-05-	19					
	-	HTCC_6.6.0_2023-02-24						
		Action Name						Size
		Entrust_CloudControl_F	Release_Notes_v6.6.pdf					0.19 MB
		Entrust-CloudControl-6.	6.0.660934.zip					6399.99 MB
	~	Entrust-CloudControl-6.	6.0.660934.zip.sha256sum.txt					104 Bytes
	~	Entrust-CloudControl-6.	6.0.660934.zip.sha384sum.txt					136 Bytes
		Entrust-CloudControl-6.	6.0.660934_upgrade.zip					3788.62 MB
		Entrust-CloudControl-6.	6.0.660934_upgrade.zip.sha256st	um.txt				112 Bytes
		Entrust-CloudControl-6.	6.0.660934_upgrade.zip.sha384si	<u>um.txt</u>				144 Bytes

5. After the file has been downloaded, open the ZIP file to access to the OVA file.

## 2.2. Deploy the CloudControl VM from the OVA

- 1. Log in to vCenter.
- 2. Select the cluster to create the CloudControl VM in.
- 3. Select the Actions Menu and select Deploy OVF template.

Deploy OVF Template	Select an OVF template	×
1 Select an OVF template	Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer such as a local hard drive a network share or a CD/DVD drive	
2 Select a name and folder	O url	
3 Select a compute resource	http://temoteserver-address/filetodeploy.ovf   .ova	
4 Review details	Local file	
5 Select storage	UPLOAD FILES No files selected.	
6 Ready to complete	Select a template to deploy. Use multiple selection to select all the files associated with an OVP template (ovf, vmok, etc.)	×

- 4. Select Local File and upload the CloudControl OVA file.
- 5. Select Next.

Follow the instructions during the deployment as needed.



For more information refer to Installing CloudControl from an OVA in the online documentation.

## 2.3. Power on the appliance

- 1. Log in to the vSphere Client.
- 2. Locate the Entrust CloudControl virtual machine in the inventory.
- 3. Right-click the CloudControl virtual machine and select **Power > Power On**.

## 2.4. Configure the CloudControl virtual appliance

This guide uses a Standalone Node setup. For more information refer to Creating a Standalone Node in the online documentation.

# 2.5. Set up the CloudControl GUI

After the standalone node has been configured, finish the setup using the GUI. For more information refer to Setting Up the CloudControl GUI in the online documentation.

# 2.6. OpenShift prerequisites

There are some OpenShift Prerequisites so it can be used in CloudControl.

For more information refer to OpenShift Prerequisites in the online documentation.

# 2.7. OpenShift setup

This guide does not describe the deployment of a Red Hat OpenShift cluster. Refer to the Red Hat documentation for details.

Assuming there is a Red Hat OpenShift cluster set up, the cluster kubeconfig file is required to be able to onboard the cluster into CloudControl. A client machine with the OpenShift binaries installed is also required to run the commands specified in this guide. This examples in this guide use a Red Hat 8 virtual machine running OpenShift Client Version: 4.11.20.

- 1. Log into the OpenShift client machine.
- 2. Set the KUBECONFIG environment variable to point to the kubeconfig file for the cluster:

% export KUBECONFIG=PATH\_TO\_KUBECONFIG\_FILE/config

3. Log into the cluster:

% oc login

The login process will depend on the user's specific setup. An API token may be needed to be able to login, or use the cluster login credentials. For example, using a token:

% oc login --token=0JTNnDj2743syq89ERQeNmTJ\_qRP-zB4b8fEbUoanWw --server=https://api.openshiftxxxxxx.net:6443

4. After logging in, test the connection by listing all the nodes:

% oc get nodes				
NAME ocp411-wqrcr-master-0 ocp411-wqrcr-master-1 ocp411-wqrcr-master-2 ocp411-wqrcr-worker-fd52p ocp411-wqrcr-worker-mtx46 ocp411-wqrcr-worker-v9xc8	STATUS Ready Ready Ready Ready Ready Ready	ROLES master master worker worker worker	AGE 66d 66d 66d 66d 66d 66d	VERSION v1.24.6+5658434 v1.24.6+5658434 v1.24.6+5658434 v1.24.6+5658434 v1.24.6+5658434 v1.24.6+5658434 v1.24.6+5658434

Now add the OpenShift cluster into CloudControl, see On-board the OpenShift Cluster.

## 2.8. On-board the OpenShift Cluster

On-boarding is the process of adding the OpenShift Kubernetes Cluster into CloudControl.

For more information refer to Adding an OpenShift Cluster in the online documentation.

1. From the Home tab, select Inventory > Kubernetes Clusters.



2. On the Clusters page, select Actions > Add Kubernetes Cluster.

If there are no clusters in the system, the user can also select **Add Kubernetes Cluster** on the **Kubernetes Clusters** page.

- 3. On the Add Kubernetes Cluster Import tab, choose one of the following:
  - ° Select Import File, then select Browse and choose the kubeconfig file to import.
  - ° Select Enter Text, then paste the contents of the kubeconfig file as plain text.

G

A kubeconfig file is a configuration file written in YAML that describes the cluster. It is the file used to set the KUBECONFIG environment variable earlier in this guide.



- 4. Select Continue.
- 5. On the **Clusters** tab, select the cluster to add and select **Continue**.

Only one cluster can be selected.

*	🛞 Kubernetes Clusters 🛛 🗙 🕇 Add	Kubernete	es Cluster 🗶				
Add Kubernetes Cluster							
1: Imp	oort 🗸 – 2: Clusters – 3: About	4: De	tails				
🕜 The	uploaded Kubeconfig file contains the	clusters	listed below. Select the cluster that	you	wish to add.		
	Name	≎ IP	)	٥	FQDN	٥	Port
	ocp411		1254, 148, 45		api.ocp411.		6443

6. On the **About** tab, enter the following:

- Friendly Name: Enter the user-facing name for the cluster.
- User: Enter the user name for the OpenShift Credentials.
- <sup>°</sup> Password: Enter the password for the OpenShift Credentials.

Add Kubernetes Cluster  ( e) ocp411	
1: Import 🖌 — 2: Clusters 🖌 — 3: About — 4: Details	
Vendor Type © OpenShift	SSH Access Credentials
Friendly Name	A highly privileged user is required to perform Configuration Hardening
оср411	operations on Kubernetes master nodes.
OpenShift Credentials	
OpenShift user credentials required to genenerate access tokens.	
User Name	
kubeadmin	
Password	
••••••	
Back	

7. Select Continue.



If the following error appears, the OpenShift Prerequisites for CloudControl have not been met. Set up the cluster for CloudControl and try again.

Forbidden, User "kubeadmin" doesn't have permission. nodes is forbidden: User "system:anonymous" cannot list resource "nodes" in API group "" at the cluster scope.

8. On the **Details** tab, monitor the process.



9. Select Continue.

- 10. In the **Enable Access Control** dialog box, select **Enable Access Control** to enable the ROOT Access Control Trust Manifest, or **No, not now** to wait until later.
  - If ROOT Access Control is enabled, CloudControl will take control of managing access to the cluster and a Trust Manifest Access Control Policy will have to be created to be able to create and manage objects in the cluster.
  - If ROOT Access Control is not enabled, that's OK as this feature can be enabled after the cluster has been onboarded.

lf Acc applie	sess Control is enabled, the ROOT Access Control Trust Manifest will be ed to this cluster.
View	ROOT Access Control Trust Manifest
A	Using the ROOT Access Control Trust Manifest may cause access disruption to cluster resources. To avoid this make sure that the relevant users and groups have been granted access in the Access Control Trust Manifest assigned to the ROOT.
This s Contr	setting can be changed in the "Actions" menu by choosing "Change Access of".

After this, the user will be able to view the dashboard for the newly added cluster. The user will also enable root access control.



11. After the cluster has been imported into CloudControl the cluster dashboard is shown.

## 2.9. View OpenShift Kubernetes cluster inventory

From the **Home** page, select **Inventory > Kubernetes Clusters** to view the **Kubernetes Clusters** page. From here, in depth information of all of the objects in that cluster can be viewed, as well as any tags or policies related to those objects. This information is included in a dashboard or a resource page.

For more information refer to Viewing Kubernetes Cluster in the online documentation.

Also refer to Navigating Kubernetes Inventory View Pages in the online documentation.

## 2.10. Transfer root access control to CloudControl

When root access control is enabled for the Kubernetes cluster, access control of the OpenShift cluster is transferred to CloudControl. This means that management of objects in the cluster will be controlled by CloudControl rules. These rules must be created and defined in CloudControl to be able to create, edit, and delete objects in the cluster.

#### 2.10.1. Before root access control is enabled

During the on-boarding process, root access was not enabled for the imported cluster. As a result, the user can still create objects at the OpenShift level.

For example, a pod can still be created. The pod.yaml file is below:

```
apiVersion: v1
kind: Pod
metadata:
    name: tutum-centos3
spec:
    containers:
        name: tutum-ssh-server3
        image: tutum/centos
```

#### To create the pod:

```
% export KUBECONFIG=~/git/cloudcontrolopenshift/files/OpenShift-KubeConfig.txt
% oc login
You must obtain an API token by visiting https://oauth-openshift.xxxxxxx.net/oauth/token/request. For instance:
% oc login --token=OJTNnDj2743syq89ERQeNmTJ_qRP-zB4b8fEbUoanWw --server=https://api.openshiftxxxxxxx.net:6443
Logged into "https://api.openshiftxxxxxxx.net:6443" as "kube:admin" using the token provided.
You have access to 57 projects, the list has been suppressed. You can list all projects with 'oc projects'
Using project "default".
% oc create -f pod.yaml
```

Security control policies have been implemented with the latests versions of Red Hat Openshift. The YAML files used in this guide do not take into account these control policies. Warning messages about them may appear and they can be ignored for the purpose of this guide. Messages like this for example:



Warning: would violate PodSecurity "restricted:latest": allowPrivilegeEscalation !=
false (container "tutum-ssh-server3" must set
securityContext.allowPrivilegeEscalation=false), unrestricted capabilities (container
"tutum-ssh-server3" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot !=
true (pod or container "tutum-ssh-server3" must set securityContext.runAsNonRoot=true),
seccompProfile (pod or container "tutum-ssh-server3" must set
securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")

The pod is created successfully. This process will fail if root access control is enabled using default HyTrust Global Access Control Policy, which denies all operations. See Enable root access control.

To delete the pod:

% oc get pods NAME READY STATUS RESTARTS AGE tutum-centos3 1/1 Running 0 3m57s% oc delete pod tutum-centos3 pod "tutum-centos3" deleted

The pod is deleted successfully. This process will fail if root access is enabled using default HyTrust Global Access Control Policy, which denies all operations. See Enable root access control.

#### 2.10.2. Enable root access control

Enabling root access control for the Kubernetes cluster gives access control of the OpenShift cluster to CloudControl.

To enable root access control:

1. From the Home tab, select Inventory > Kubernetes Clusters.

oudControl						<b>::</b> 4 ?
		(		-	ŧ	
Inv	ventory	Secu	rity	Syste	<b>P</b>	
AWS Accounts	Kubernetes Clusters	NSX-T	vSphere	(77) VCF	IIII / IIII/ Image Registries	

2. Select Management to view the clusters.

Kubernetes Clusters       Views       Actions	H Subernetes Cluste	ers 🗙						
Management    Nodes   1    1    Current      Current Configuration Hardening   Image: Configuration Hardening   Current Configuration Hardening   Image: Configuration Hardening   Current Configuration Hardening   Image: Configuration Hardening   TypeError: Reduce of empty array with no initial value   Image: Container Configuration for Co	Kubernetes Cluste	ers					Views	Actions   -
Current Configuration Hardening Current Configuration Hardening TypeError: Reduce of empty array with no initial value Deployment Control Allowed Denied ocp411 Container Runtime Violations Container (active and paused) compty with the policy. Container (active and paused) c	Management 1 Cluster	⊖ Nodes 6	Namespaces	l Dep	oloyments	<sup>⊕ Pods</sup> 297	<ul><li>✤ Containers</li><li>482</li></ul>	¢\$ Services <sup>⊮*</sup> 82
TypeError: Reduce of empty array with no initial value	Ci	urrent Configuratio	on Hardening			Configura	tion Hardening Trendir	ng
Deployment Control     and the point of the	TypeError: Re	educe of empty a	rray with no initial value	9	There	is currently no tr	rending information for Hardening	r Configuration
Allowed Denied		Deployment C	control	e <sup>p</sup>		Contair	ner Runtime Violations	<sup>بر</sup> ي
0 02/12/2023 03/05/2023 03/14/2023 0cp411	564 400 200		Allowed Der	nied	All contain     Coutrainets     Coutrainets	ners (active and pause	ed) comply with the policy.	• ocp411
Olivera Charles Contraction Contra	o	ocp4	111		02/12/2	023	03/05/2023 Last 30 days	03/14/2023

3. Select the **Cluster Name** link to enable root access.

This action will display the details about the cluster selected.

O Nodes	Namespaces	Deployments	Pods	Containers	¢ <sup>8</sup> Services	~
6	68	58	296	481	82	
	Details	× <sup>p</sup>		Container Deployment Control		2
Туре	ContainerOrchestrator			Allowed	Requests	
IP Address	10.104.148.49			3	•	
API Server Port	6443					
Master Nodes	3					
SSH Port	22					
API Version	4.11.20					
Build Date	Jan 5, 2023, 5:33:53 PM					
Access Control	Disabled					
Master URL	https://api.ocp411. 6443/					
Vendor Type	C Open Shift					
Trust Manifests	4					
Tags	Assign Now					
		Container Runtim	e Violations			2
<ul> <li>All containers (active and p</li> </ul>	aused) comply with the policy.					
All containers (active and p	aused) comply with the policy.					
<ul> <li>All containers (active and p</li> <li>1</li> </ul>	aused) comply with the policy.					
<ul> <li>All containers (active and p</li> <li>1</li> </ul>	aused) comply with the policy.					
All containers (active and p	aused) comply with the policy.					
✓ All containers (active and p 1 4 4 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4	aused) comply with the policy.					
All containers (active and p     Output     Define a second	aused) comply with the policy.					
All containers (active and p	aused) comply with the policy.					
All containers (active and p	aused) comply with the policy.					
Continues (active and p	aused) comply with the policy.			13092023	03/13/2	123

The Details section of the page shows that Access Control is Disabled.

4. Select the Disable link to Enable Root Access Control.



5. In the Access Control dialog, set Status to Enabled and select Apply.

After root access is enabled, the user cannot work with objects directly. For example, creating a pod:

```
% oc create -f pod.yaml
Error from server (Forbidden): error when creating "pod.yaml": admission webhook "accesscontrol.hytrust.com"
denied the request: Permission Denied by HyTrust Security Policy.
```

In this example, the user is unable to create the pod, as CloudControl now controls permissions on the cluster. Now give users permissions to perform actions, see Create a Trust Manifest Access Control Policy.

## 2.11. Create a Trust Manifest Access Control Policy

After root access control is enabled in the cluster, users are unable to create objects in the cluster. This is because cluster permissions are then managed by CloudControl.

This section describes the process to create a Trust Manifest Access Control policy, so a user can manage objects in the cluster.

#### 2.11.1. Log analysis

Use the logs in the system to check why access has been denied for a request. From this, create a Trust Manifest Access Control Policy to allow the user to perform the request

successfully. For more information refer to Log Analysis in the online documentation.

- 1. Go to the Home tab, select Security > Log Analysis.
- 2. Select the record that shows the **Deny** status to see the reason for the denial.

Look for an entry similar to the following:

Authorization denied due to no rules applying to the user via the configured access control policy for the resource(s) with name(s) '[openshift- multus]'. There needs to be at least one direct role association by way of user name or group(s)						
Privileges	Compute.ContainerNamesp	ace.Edit	Date	Mar 14, 2023, 2:36:35 PM		
Resources	openshift-multus (Kubernet	esNamespace)	Priority	A WARN		
Source	front out office along on	( )	Status	Deny		
Destination	front over state articles, or	( )	User	system:serviceaccount:openshift-network-operator:default		
Protocol	RESTAPI		Groups			
Policy	Enforced		Roles			
Msg ID	AUZ0001		Action	Compute.ContainerNamespace.Edit		
Category	AUZ		Vendor Action	Compute.ContainerNamespace.Edit		
			Trust Manifest	HyTrust Global Trust Manifest for Access Control		

In this example, a request to create a pod was denied.

Now create a Trust Manifest Access Control Policy to allow the user to create the pod.

## 2.11.2. Create the Access Control Policy

This section describes how to use an Access Control Policy to allow users to create pods in the OpenShift cluster.

In the example below, the Access Control Policy will allow the kubeadmin user to create pods in the cluster. Any other user in the system will be denied access.

For details of how to create an Access Control Policy, refer to Creating an Access Control Trust Manifest from the CloudControl GUI in the online documentation.

- 1. From the Home tab, select Security > Trust Manifests.
- 2. On the **Manage Trust Manifests** page, select **Create Trust Manifest**. That is, the plus sign in the GUI.
- On the Details tab of the Create Trust Manifest page, enter the Name and optional Description for the Trust Manifest.
- 4. For Policy Type, select Access Control.
- In the Access Control Policy section, enter the name of the rule. That is, ACP -Creation Rule.
  - For Rule Type, select allow.
- 6. For Role, select ASC\_ContainerInfraAdmin.

This is the role that controls all operation related to creating objects in cluster.

7. Under Subjects:

- a. For Type, select Kubernetes.
- b. For Group or User, add:
  - k8s::user:system:admin
  - k8s::user:kube:admin
  - k8s::user:system:serviceaccount:openshift-machine-api:machine-apioperator
- 8. Select Validate to validate the policy.
- 9. Select **Save** to save the policy.
- 10. Select **Publish** to publish the policy.

When the policy is published it will ask the user to assign resources to the policy. Select the OpenShift cluster and select **Assign**.

11. Select Close.

### 2.11.3. Test the Access Control Policy

To test if the Access Control policy is working, create the pod:

```
% oc create -f pod.yaml
```

```
pod/tutum-centos3 created
```

The request is successful.

To test the deletion of a pod:

```
% oc delete pod tutum-centos3
```

Error from server (Forbidden): admission webhook "accesscontrol.hytrust.com" denied the request: Permission Denied by HyTrust Security Policy, Resource not found.

In this example, the resource is not found in CloudControl. This is because the CloudControl has not yet updated its cluster inventory. When a cluster is onboarded, it creates an internal job that runs every 5 minutes to update the cluster inventory. This means that the only way to delete the pod at this time is to:

- Wait for the cluster inventory job to complete.
- Manually sync the inventory.

To manually sync the inventory:

1. In the **Cluster Detail** tab, select **Actions > Sync Inventory**.

#### 2. Select Initiate Sync.

After the sync completes, delete the pod. For example:

% oc delete pod tutum-centos3

pod "tutum-centos3" deleted

The pod deletes successfully.

## 2.12. Image registries

An image registry is a service that stores repositories and images. Each repository contains one or more version of the same image. All images in a repository must have the same name and be differentiated by tags. The tag name corresponds to the version of the image. The most recent image is also tagged as 'latest'.

Image registries are not protected by CloudControl, but adding a registry allows CloudControl to discover valuable information about the registry, such as the number of images and their specific vulnerabilities.

This allows more detailed rules when creating an Image Deployment Control Policy which will be discussed later in this guide. Entrust strongly suggests adding private image registries to CloudControl for better control during image deployment in Kubernetes.

Add the private registry to CloudControl: xxxx.yyyy.zzz

1. In the **Home** tab, Select **Inventory > Image Registries** to view the registries that have been added to CloudControl.



2. On the Image Registries page, select Actions > Add Registry.

If there are no registries in the system, select the **Add Registry** link on the **Image Registries** page.

3. On the Add Registry page, in the About tab enter the following information:

- a. **Name** Name of the registry
- b. IP/FQDN Enter the IP Address or FQDN for the registry.
- c. Port Enter the registry port used in configuration of the local registry.
- d. **Authorization Schema** Choose one of the following to use for authorization: **BASIC** or **OAUTH**.
- e. User Enter the user name for the registry.
- f. **Password** Enter the password for this user.
- g. Description Enter an optional description.
- 4. Select Continue.

	ontrol	
🖌 📓 Image Registries 🗙	+ Add Registry	×
Add Registry		
1: About – 2: Details		
Name*		
registry.		
Description		
Docker registry used by	luritaries Lab	
IP/FQDN*		
registry.		
Port*		
443		
Authorization Scheme		
BASIC		<b>.</b>
User Name *		
constraint and a second s		
Password *		

If a certificate authority has not been added, the system will prompt the user to add one.

- 1. In the Missing Certificate Authority window, select Install Certificate Authority Now.
- 2. On the Install Certificates page, do one of the following:
  - a. Select **Import** and then select **Browse** to locate the certificate file.
  - b. Select **Enter Text** and then paste the contents of the certificate as plain text into **Certificate Data**.

- 3. Select Continue.
- 4. On the **Details** page, monitor the process.
- 5. Select **Continue** to view the dashboard for the newly added registry.

() EN	ITRUST	CloudCo	ontrol								== A	?
# 8	Image Registries	×	🗱 registry. 🗙 🗙									
registry.									ons   •			
Details           Name         registry.           ID         15dff365-4054-303-addd           Resource Type         ContainerRegistry           Platform         None detected								Ti No tags are cu Assig	ags wrrently assigned yn Now			
						lma	iges					
Filter											Actions	;   -
	Name	-	Image ID	٥	Image Tag 🛛 🗘	Vulnera	bilities	Signature 0	Containers	Create Time		
	aw-centos7nshie	ldibm	sha256:4b4b5a4726d3b4		latest	Not assse	essed	true	0	Jun 2, 2021, 9:25:10 AM	<b>%</b> 0	<b>^</b>
	aw-nshieldproxy		sha256:b7d6c622e3972f6		latest	Not assse	essed	true	0	Jan 12, 2021, 5:35:17 PM	<b>%</b> 0	
	aw-rh8nshieldibm	1	sha256:d5e8cd3b1d86ccf		withTouch	Not assse	essed	true	0	Jun 8, 2021, 1:14:37 PM	<b>%</b> 0	
	aw-rh8nshieldibm	1	sha256:fb0627e7c24ba07		testing	Not assse	essed	true	0	Jun 17, 2021, 5:31:04 PM	<b>%</b> 0	
	aw-rh8nshieldibm	1	sha256:eb9035c83f8fbbdf		latest	Not assse	issed	true	0	Jun 15, 2021, 9:42:19 AM	<b>%</b> 0	
	busybox		sha256:2131f09e4044327		latest	Not assse	ssed	true	0	Jun 29, 2020, 4:21:41 PM	<b>%</b> 0	
	cv-12.71-nshield-	ap	sha256:6dedb82d073382:	۰.	latest	Not assse	essed	true	0	Sep 16, 2021, 9:52:37 AM	<b>%</b> 0	
	cv-12.71-nshield-	hw	sha256:df9ee225b06a90e		latest	Not assse	essed	true	0	Sep 16, 2021, 9:51:44 AM	<b>%</b> 0	-
Showing	g 1 to 8 of 47 records	s (0 Seleo	cted)									

Now create an Image Deployment Control Policy, see Image Deployment Control Policy.

## 2.13. Image Deployment Control Policy

The Image Deployment Control Policy controls how images are deployed in the OpenShift cluster managed by CloudControl. As part of a Trust Manifest, it allows the user to determine what images from a registry are safe to be added to the protected clusters.

CloudControl enforces image security using Image Deployment Control Policies, which are comprised of one or more deployment rules:

#### 2.13.1. Private registry rules

• Private Registries

Allows the user to enter a list of private registries, either onboarded or not, to be evaluated with the trust manifest. Only the images from registries listed in the **Allowed Registries** section are evaluated to see if they can be deployed. Images from all other registries will be denied.

Signature Rule

Allows denial of images that do not have an associated digital signature.

Attributes Rule

Allows the system to deny or deploy images based on their image ID or image name.

Vulnerabilities Rule

Allows the system to deny or deploy images based on CVSS scores or specific CVEs.

#### 2.13.2. Public registry rules

A public registry rule enables images from public registries to be deployed in the environment without any evaluation.



Entrust strongly recommends leaving the public registry rule set to ENABLED and do not allow public registries to be deployed. If a public registry is used, then leave the rule set to ENABLED and enter that specific registry into the Allowed Registries property. This will allow only that specific registry image to be deployed, and will prevent all other public registry images from deployment.

#### 2.13.3. Other considerations

Rules can either have a True or False value and can also include a 'stop processing' clause. Deployment rules in a policy are evaluated in the following order:

- If False, the image will not be deployed, and no further rules are evaluated.
- If True, AND there is a 'stop processing' clause, the image is allowed to be deployed and no further rules are evaluated.
- If True, the next rule in the policy is evaluated. If this is the only rule, then the image is allowed to be deployed.
- If all rules are True, then the image is allowed to be deployed.



Entrust recommends using the image SHA as it is a unique identifier for images. Pods can be created by using either an image name with tag, or an image name with SHA, and in many cases images with the same SHA could have been tagged with different tags. For example, a single image named TestImage could have different tags like TestImage:3, TestImage:4, and TestImage:5, but all these images will have the same SHA as the underlying image is the same for all three of them.

When the user creates any Image Deployment Control Policy rules, use the image name with SHA to ensure that the intended image is evaluated no matter what tags are there.

When an image name with tag is used, such as TestImage:3, then only the image that matches that specific tag will be selected. The other images, TestImage:4 and TestImage:5 will not be evaluated.

### 2.13.4. Create an Image Deployment Control Policy

This section will show how to use an Image Deployment Control Policy to control which images are allowed/denied in the deployment.

For more information refer to Creating a Deployment Control Trust Manifest from the CloudControl GUI in the online documentation.

- 1. From the Home tab, select Security > Trust Manifests.
- 2. On the **Manage Trust Manifests** page, select **Create Trust Manifest**. That is the plus sign in the GUI.
- On the Details tab of the Create Trust Manifest page, enter the Name and optional Description for the trust manifest.
- 4. For Policy Type, select Deployment Control.
- 5. In the Deployment Control Rules: Private registries section:
  - a. For Allowed Registries, enter the registries to be allowed. That is:
    - Enter the registry that was onboarded: **xxxxx.yyyy.zzz**.
    - Registries can be existing onboarded registries or registries that are planned to be onboarded.
    - Registries that have not been onboarded are depicted with a yellow warning icon.
- 6. In the Deployment Control Rules: Rules section:
  - a. For **Signature Rule**, select either **Enabled** or **Disabled**. This determines whether to deny an image when no signature is present.
- 7. In the Deployment Control Rules: Rules section:
  - a. For **Attribute Rule**, select **Enabled** or **Disabled** to determine whether to evaluate using attributes, and then complete the following:
    - i. **Name** Enter the name of the rule. The name cannot contain any special characters.
    - ii. Exemption List Deploy on Match

Select **ENABLED** or **DISABLED** to determine whether to use this when evaluating.

If **ENABLED**, use the + and - symbols to add the following criteria:

- Image ID Enter the image ID in SHA format to match.
- Image Name Enter the Name and Tag Regex to match.

If there is a match, the image will immediately be deployed, and no other deployment policy rules will be evaluated.

If there is no match, continue to the next enabled step.

If there are no other steps, continue to the next rule in the deployment policy.

iii. Whitelist - Deny on No Match

Select **ENABLED** or **DISABLED** to determine whether to use this when evaluating.

If **ENABLED**, use the + and - symbols to add the following criteria:

- Image ID Enter the image ID in SHA format to match.
- Image Name Enter the Name and Tag Regex to match.

If there is no match, the image will immediately be denied, and no other deployment policy rules will be evaluated.

If there is a match, continue to the next enabled step.

If there are no other steps, continue to the next rule in the deployment policy.

iv. Blacklist - Deny on Match

Select **ENABLED** or **DISABLED** to determine whether to use this when evaluating.

If **ENABLED**, use the + and - symbols to add the following criteria:

- Image ID Enter the image ID in SHA format to match.
- Image Name Enter the Name and Tag Regex to match.

If there is a match, the image will immediately be denied, and no other deployment policy rules will be evaluated.

If there is no match, continue to the next rule in the deployment policy.

b. Optional. In the **Public Registries** section, add public registries to be deployed without any evaluation.

Entrust recommends leaving this section enabled, but do not enter any values in **Allowed Registries**.

- 8. Select one of the following:
  - Validate Validate the draft or existing trust manifest.
  - ° **Save** Save the trust manifest as a draft.
  - ° **Publish** Publish the trust manifest.

#### 2.13.5. Image Deployment Control Policy examples

In this example, an Image Deployment Control Policy is created. It will only allow images that are in the private registry that was onboarded earlier. Any other image that is not in the private registry will be blocked and will not run.

To be able to publish the policy:

- The Restrict Public Registry rule will be enabled.
- A fake registry name **abc** will be added to the exception list. This will force the policy to only allow images in the private registry.

lame *	
MyOpenshiftDeploymentControlPolicy	
Description	469 Characte
Image Deployment Control Policy	
Jeployment Control Rules	
✓ Private Registries	
Allowed Registries *	ENABLED
≝ registry. 443 ×	×
Rules Evaluation will happen in the following order:	Expand All   Collapse All
> Signature Rule	DISABLED
> Attributes Rule	DISABLED
> Vulnerabilities Rule	DISABLED
V Public Registries	
✓ Restrict Public Registry Rule	ENABLED
Adding public registries will allow all images in these registries to be deployed without any further evaluation.	
This is against recommended best practice.	
Allowed Registries	
0 abc ×	×
Cancel	Validate Apply

After this policy is published, users can attempt to deploy an image that is not in the registry.

For example, using the **pod.yaml** file again:

```
apiVersion: v1
kind: Pod
metadata:
    name: tutum-centos3
spec:
    containers:
        name: tutum-ssh-server3
        image: tutum/centos
```



The YAML file is not using an image from the private registry.

#### % oc create -f pod.yaml

Error from server (Forbidden): error when creating "pod.yaml": admission webhook "deploymentcontrol.hytrust.com" denied the request: Permission Denied by HyTrust Security Policy.

In this example, the pod was not deployed because it uses an image from the **tutum** registry.

To use an image from a private registry using a different YAML file. For example, podinternal-registry.yaml with the following content:

```
apiVersion: v1
kind: Pod
metadata:
    name: internal-registry-centos
spec:
    imagePullSecrets:
        name: regcred
        containers:
            name: internal-registry-centos
            image: xxxxx.yyyy.zzz/cv-centos:latest
```

#### Create the pod:

```
% oc create -f pod-internal-registry.yaml
pod/internal-registry-centos created
```

The deployment is successful, because the Image Deployment Control Policy allows images from the private registry.

Add the **tutum** external registry to the external registry exception list, so an external image can also be deployed. For example:

Name MyOpenshiftDeploymentControlPolicy	
Description mage Deployment Control Policy	
Deployment Control Rules	
Private Registries     Allowed Registries	ENABLED
🚆 registry. 443	
Rules Evaluation will happen in the following order:	Expand All   Collapse All
✓ Signature Rule	DISABLED
Images without a valid signature will be denied.	
> Attributes Rule	UMABLED
> Vulnerabilities Rule	UISABLED
✓ Public Registries	
V Restrict Public Registry Rule	ENABLED
Adding public registries will allow all images in these registries to be deployed without any further evaluation. This is against recommended best practice.	
Allowed Registries	

Create the pod using **pod.yaml** file again:

% oc create -f pod.yaml
pod/tutum-centos3 created

The deployment policy is working as designed.

Expand the policy by further restricting what images can be deployed from the private registry.

To determine what images are in the private registry:

1. In the **Home** tab, Select **Inventory > Image Registries** and select the private registry that was onboarded.

JEN		CloudC	ontrol									-	
	Image Registries	×	📰 registry. 🗙										
reç	gistry.	net.									A	ction	s
			Details						Та	gs			
me		registry	5 fdE4 4502 oddd										
sourr	се Туре 🛛	Contain	erRegistry										
tform	n I	None de	etected						No tags are cu Assig	rrently assigned n Now			
						Images							
ilter											Acti	ons	I
ilter	Name	•	Image ID	0	lmage Tag ≎	Vulnerabilit	ies	Signature 0	Containers	Create Time	Acti	ons	1
lter	Name aw-centos7nshield	<b>a</b> dibm	Image ID sha256:4b4b5a4726d3b4	0	Image Tag ≎ latest	Vulnerabilit	ies /	Signature 0	Containers 0	Create Time Jun 2, 2021, 9:25:10 AM	Actie	ons	
lter	Name aw-centos7nshield aw-nshieldproxy	<b>a</b> dibm	Image ID sha256.4b4b5a4726d3b44 sha256.b7d6c622e3972f6	0	Image Tag ≎ latest latest	Vulnerabilit Not assessed Not assessed	ies 1	Signature true	Containers 0 0	Create Time Jun 2, 2021, 9:25:10 AM Jan 12, 2021, 5:35:17 PM	Actio	ons > 0 > 0	
lter	Name aw-centos7nshield aw-nshieldproxy aw-rh8nshieldibm	đibm	Image ID sha256.4b4b5a4726d3b44 sha256.b7d6c622e3972f6 sha256.d5e8cd3b1d86ccf	0	Image Tag o latest latest withTouch	Vulnerabiliti Not assessed Not assessed Not assessed	ies i i	Signature ¢ true true	Containers 0 0 0	Create Time Jun 2, 2021, 9:25:10 AM Jan 12, 2021, 5:35:17 PM Jun 8, 2021, 1:14:37 PM	Actio	ons > 0 > 0 > 0	
	Name aw-centos7nshield aw-nshieldproxy aw-rh8nshieldibm aw-rh8nshieldibm	dibm	Image ID sha256-4b4b5a4726d3b4 sha256-b766c622e3972f6 sha256-b568cd3b1686ccf sha256-b568cd3b1686ccf sha256-b5682r97c24ba07	0	Image Tag ≎ latest latest withTouch testing	Vulnerabiliti Not assessed Not assessed Not assessed Not assessed	ies / /	Signature ¢ true true true true	Containers 0 0 0 0 0	Create Time Jun 2, 2021, 9:25:10 AM Jan 12, 2021, 5:35:17 PM Jun 8, 2021, 1:14:37 PM Jun 17, 2021, 5:31:04 PM	Activ	ons > 0 > 0 > 0 > 0	
	Name aw-centos7nshield aw-nshieldproxy aw-rh8nshieldibm aw-rh8nshieldibm aw-rh8nshieldibm	dibm	Image ID sha256-4b4b5a4726d3b4 sha256-b7d6c622e397216 sha256-b5d8cd3b1d86ccf sha256-b5d8cd3b1d86ccf sha256-b6627e7c24ba07 sha256-b69035c8318bbdf	0	Image Tag o latest i latest with Touch testing i latest i	Vulnerabiliti Not assessed Not assessed Not assessed Not assessed Not assessed	les 1 1 1	Signature ¢ true true true true	Containers 0 0 0 0 0 0 0 0	Create Time           Jun 2, 2021, 9:25:10 AM           Jan 12, 2021, 5:35:17 PM           Jun 8, 2021, 1:14:37 PM           Jun 17, 2021, 5:31:04 PM           Jun 15, 2021, 9:42:19 AM	Activ	ons > 0 > 0 > 0 > 0 > 0 > 0 > 0	
	Name aw-centos7nshield aw-nshieldproxy aw-rh8nshieldibm aw-rh8nshieldibm aw-rh8nshieldibm busybox	đibm	Image ID sha256.4b4b5a4726d3b4 sha256.b7d6c52ca3972f6 sha256.b7d6c52ca3972f6 sha256.d5e6cd3b1d86ccf sha256.tb6627e7c24ba07 sha256.eb0035c8318bbdf sha256.2t31f09e4044327	0	Image Tag C C Iatest Ia	Vulnerabiliti Not assessed Not assessed Not assessed Not assessed Not assessed	ies 1 1 1 1	Signature ¢ true true true true true true	Containers 0 0 0 0 0 0 0 0 0 0	Create Time           Jun 2, 2021, 9:25:10 AM           Jan 12, 2021, 5:35:17 PM           Jun 8, 2021, 1:14:37 PM           Jun 17, 2021, 5:31:04 PM           Jun 15, 2021, 9:42:19 AM           Jun 29, 2020, 4:21:41 PM	Activ	ons > 0 > 0 > 0 > 0 > 0 > 0 > 0 > 0	
	Name aw-centos7nshield aw-nshieldproxy aw-rh8nshieldibm aw-rh8nshieldibm aw-rh8nshieldibm busybox cv-12.71-nshield-d	dibm	Image ID sha256-4b4b5a4726d3b4 sha256-b7d6c622a3972f6 sha256-b7d6c622a3972f6 sha256-b66276724ba07 sha256-b60276724ba07 sha256-sh035c338bbbd sha256-2131f094044327 sha256-6bd6b82d073382	0	Image Tag o latest latest latest testing latest latest latest latest latest latest l	Vulnerabiliti Not assessed Not assessed Not assessed Not assessed Not assessed Not assessed	les / / / / / / /	Signature ¢ true true true true true true true true true true	Containers 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Create Time           Jun 2, 2021, 9:25:10 AM           Jan 12, 2021, 5:35:17 PM           Jun 8, 2021, 1:14:37 PM           Jun 17, 2021, 5:31:04 PM           Jun 15, 2021, 9:42:19 AM           Jun 29, 2020, 4:21:41 PM           Sep 16, 2021, 9:52:37 AM	Activ	ons       >0       >0       >0       >0       >0       >0       >0	

2. Look at the images that are available in the registry.

Change the deployment policy to only allow images with name **cv-centos** and with the **latest** tag. Any other image in the private registry that does not match this requirement in the name will be blocked.

To do this, add an **Attribute Rule** to the deployment policy. For example:

Name *			
CV Images Only			
Evaluation will happen in the following ord	er:		
Step 1. Exemption List			
Deploy on Match			
If any of the following criteria matches, g to the next enabled step.	eploy the image and stop processin	g the Deployment Po	olicy. If no criteria matches, con
Criteria *			
Image Name × - Name Regex	cv-centos Tag R	latest	0.0
Step 2. Whitelist	deny, the image and stop processing	the Denloyment Po	licy. If any criteria matches, co
Step 2. Whitelist Deny on No Match If none of the following criteria matches, to the next enabled step. Criteria *	<u>deny</u> the image and stop processing	g the Deployment Po	licy. If any criteria matches, <u>co</u>
Step 2. Whitelist	deny the image and stop processing	g the Deployment Po Regex * latest	licy. If any criteria matches, <u>co</u>
Step 2. Whitelist	deny, the image and stop processing cv.* Tag R eny, the image and stop processing	g the Deployment Polex * latest	icy. If any criteria matches, <u>co</u>

Now try to deploy the **busybox** image in the private registry. To do this, create a file called **pod-internal-busybox.yaml** with the following content:

```
apiVersion: v1
kind: Pod
metadata:
    name: internalregistry-busybox
spec:
    imagePullSecrets:
        - name: regcred
        containers:
            - name: internalregistry-busybox-test
            image: xxxxx.yyyy.zzz/busybox
```

When the system tries to deploy this, it should fail and deny the deployment. For example:

```
% oc create -f pod-internal-busybox.yaml
```

Error from server (Forbidden): error when creating "pod-internal-busybox.yaml": admission webhook "deploymentcontrol.hytrust.com" denied the request: Permission Denied by HyTrust Security Policy.

Now deploy the **cv-centos** image in the private registry. To do this, use the **pod-internal-registry.yaml** file again.

```
% oc create -f pod-internal-registry.yaml
```

```
pod/internal-registry-centos created
```

The deployment is successful.

This demonstrates how to harden the Image Deployment Control Policy.

#### 2.13.6. Image Deployment Control Policy based on vulnerabilities

When a private registry is onboarded into CloudControl, the system checks the number of vulnerabilities in each image in the registry. Below is an example of a private registry that was onboarded.

Name	•	Image ID	0	Image 🗘	Vulnerabilities	Signature 0	Contain	c		
ojw-ab5		sha256:7d2badb2abefa86b79515671f3581cb9f44895ac		latest	0	true	0	N	<b>&gt; 0</b>	•
ojw-ckst		sha256:365215cb7643101318e37b590519fd0e5173972		ocs1	0	true	0	N	<b>&gt; 0</b>	
ojw-ckst-kmdl		sha256:f03e3350c05abd4335a38e71baea030942fd4aa		setr	0	true	0	N	<b>&gt; 0</b>	
ojw-httpd		sha256:6e26c808ef1f055a2d82d8951c0feb94c5c6585c		latest	<mark>69</mark> 101	true	0	N	≫ 0	
ojw-httpd-ast		sha256:71c5d377ecfc3fe8276a4b235cef8e22f9d08107a		latest	69 101	true	0	N	<b>&gt; 0</b>	II.
ojw-httpd-err		sha256:3e319d6460418325a5ed1545bf514dc328f003b		latest	69 101	true	0	N	<b>&gt; 0</b>	
oiw-httpd-nsc		sha256:d2857a6b2afc24622d96e1562f57acfe557fc7cc		latest	69 101	true	0	N	<b>&gt; 0</b>	*

For each image, CloudControl collects the number of vulnerabilities and their types. Select a link in the **Vulnerabilities** column to view a dialog with tabs that include the details.

iage 🖾 ojw-httpd (Last	Details V	ulnerabilities Containers
Severity = LOW >	Severity = NEGLIGIBLE >	3 ×
/ulnerability	≎ <u>Severity</u>	≎ Description
VE-2020-29363	NEGLIGIBLE	An issue was discovered in p11-kit 0.23.6 through 0.2
VE-2018-6829	NEGLIGIBLE	cipher/elgamal.c in Libgcrypt through 1.8.2, when use
VE-2021-20193	NEGLIGIBLE	A flaw was found in the src/list.c of tar 1.33 and earlier
VE-2019-9923	NEGLIGIBLE	pax_decode_header in sparse.c in GNU Tar before 1
VE-2005-2541	NEGLIGIBLE	Tar 1.15.1 does not properly warn the user when extra
VE-2018-1000654	NEGLIGIBLE	GNU Libtasn1-4.13 libtasn1-4.13 version libtasn1-4.13
VE-2019-17543	LOW	LZ4 before 1.9.2 has a heap-based buffer overflow in
VE-2011-3374	NEGLIGIBLE	It was found that apt-key in apt, all versions, do not co
VE-2019-18276	NEGLIGIBLE	An issue was discovered in disable_priv_mode in shell

The **Vulnerabilities** tab lists each vulnerability with their CVE, severity, and description. Select the CVE link to view details about the CVE. For example:

Image 🖬 ojw-ht	Vulnerability	Details		×		×
Filter Severity Vulnerability CVE-2020-29363 CVE-2018-6829 CVE-2021-20193 CVE-2019-9923	Name Severity Fixed By CVE Link Description	CVE-2019-17543 LOW https://security-tracked LZ4 before 1.9.2 has a LZ4, write32 (related 1 applications that call (This issue can also Io vendor states "only a API are at risk."	er.debian.org/tracker/CVE-2019-17543 heap-based buffer overflow in o LZ4_compress_destSize), affecting LZ4_compress_fast with a large input add to data corruption.) NOTE: the few specific / uncommon usages of t	t. te	rough 0.2 when use ind earlier before 1	
CVE-2005-2541	NEG	LIGIBLE	Tar 1.15.1 does not properly warn the	e user v	when extra	

With this information on hand, now create/modify the Image Deployment Control Policy to allow/deny deployments based on the number of vulnerabilities an image contain.

For example, modify the current Image Deployment Control Policy to allow any image from the private registry, but only if it has no vulnerabilities. This requires the use of two images from the private registry:

- cv-centos which has no vulnerability.
- ojw-httpd which has some vulnerabilities.

#### 2.13.6.1. Modify the Image Deployment Control Policy

Modify the policy by disabling the **Attributes Rule** and enabling the **Vulnerabilities Rule**. The default thresholds will be appropriate for testing.

- 1. From the **Home** tab, select **Security > Trust Manifests**.
- On the Manage Trust Manifests page, select the MyOpenShiftDeploymentControlPolicy policy.

- 3. Select Edit.
- 4. Disable the Attribute Rule section under Rules, under Private Registries.
- 5. Enable the Vulnerability Rule section.
  - a. Give a name for the rule, My Vulnerability Rule.
  - b. Take the defaults for the **Deny deployment thresholds** values.
    - i. The **ojw-httpd** image has more than 30 low severity vulnerabilities.
- 6. Select Validate and the select Apply.

	ate Reg	isules	ENA
Allov	wed Reg	istries *	
E	registry	. :443 ×	
Rule Evalu	s Jation wil	I happen in the following order:	Expand All   Collaps
	Signat	ure Rule	
		without a confidence of the standard	DIRABLEI
	Images	without a valid signature will be denied.	
>	Attribu	tes Rule	DIABLE
~	Vulner	abilities Rule	ENABLED
	Name	*	
	My V	ulnerability Rule	
	Deny	deployment if thresholds exceed:	
	0	or more defcon1 and critical severity vulnerabilities	
	1	or more high severity vulnerabilities	
	-	or more medium severity vulnerabilities	
	5		
	5 30	or more low and negligible severity vulnerabilities	
	30 White	or more low and negligible severity vulnerabilities	
	5 30 White	or more low and negligible severity vulnerabilities ist	
	5 30 White	or more low and negligible severity vulnerabilities Ist	

Now test the policy, see Test the policy.

#### 2.13.6.2. Test the policy

To test the policy, attempt to deploy the **cv-centos** image. Use the same **pod-internal-registry.yaml** file again.

```
% oc create -f pod-internal-registry.yaml
pod/internal-registry-centos created
```

The deployment succeeds.

Attempt to deploy the **ojw-httpd** image. To do this, create a file called **pod-internalcve.yaml** with the following content:

```
apiVersion: v1
```

```
kind: Pod
metadata:
    name: internalregistry-ojw-httpd
spec:
    imagePullSecrets:
    - name: regcred
    containers:
    - name: internalregistry-ojw-httpd
    image: xxxxx.yyyy.zzz/ojw-httpd:latest
```

It should fail and be denied.

% oc create -f pod-internal-cve.yaml

```
Error from server (Forbidden): error when creating "pod-internal-cve.yaml": admission webhook
"deploymentcontrol.hytrust.com" denied the request: Permission Denied by HyTrust Security Policy.
```

Examine the logs to see the denial.

- 1. Go to the Home tab, select Security > Log Analysis.
- 2. Select the record that shows the **Deny** status to see the reason for the denial.
- 3. Look for something like this:

Container Imag Vulnerabilities	eb94c5c6585c9c5cef90ad tag:(latest)` ha		
Privileges		Date	Mar 15, 2023, 1:01:01 PM
Resources	ocp411 (ContainerOrchestrator)	Priority	A WARN
Source	.com ( )	Status	Deny
Destination	.com ( )	User	system:admin
Protocol	RESTAPI	Groups	
Policy	unknown	Roles	
Msg ID	K8S0004	Action	Deploying Image
Category	POL	Vendor Action	Deploying Image

This feature from CloudControl allows users to put in place image deployment control policies that can harden the organization deployment requirements and tailor this capability according to the organization needs.

# Chapter 3. Additional resources and related products

- 3.1. CloudControl
- 3.2. Entrust products
- 3.3. nShield product documentation