CDP Private Cloud Data Services 1.5.3

ECS Day Two Operations Guide

Date published: 2023-12-16
Date modified: 2024-03-23

CLOUDERA

https://docs.cloudera.com/

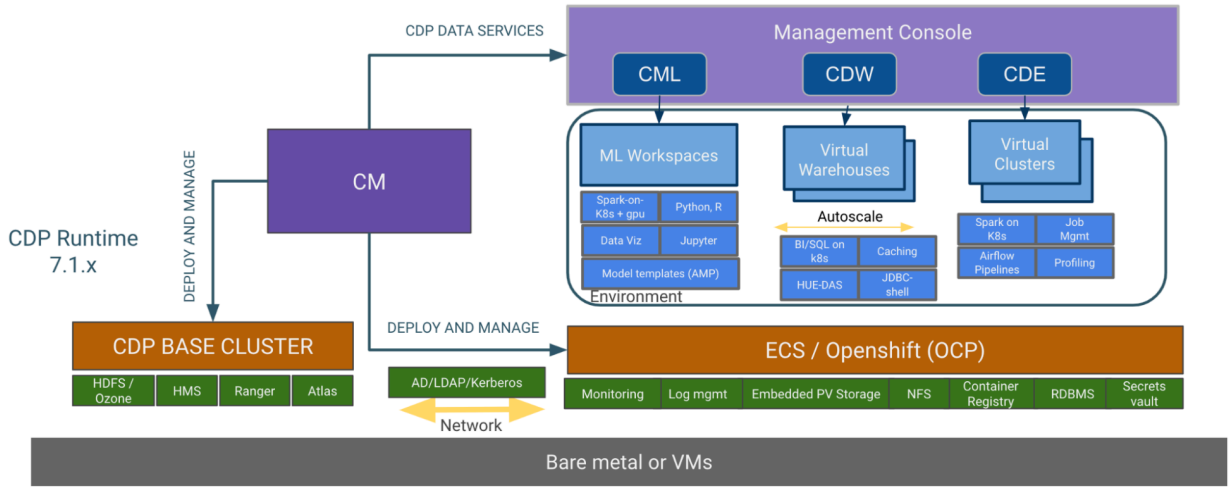
Legal Notice

Contents

Overview

This guide provides information for administrators about useful maintenance tasks after a new installation of CDP Private Cloud Data Services on the Embedded Container Services (ECS).

CDP Private Cloud Data Services Architecture

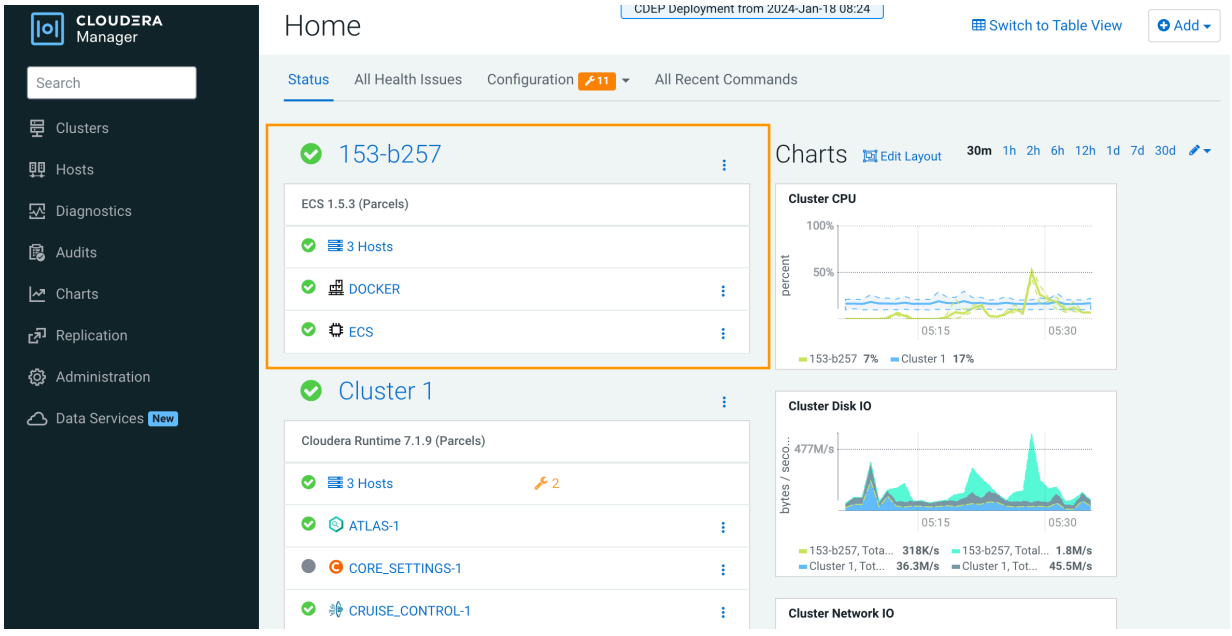


Prerequisites

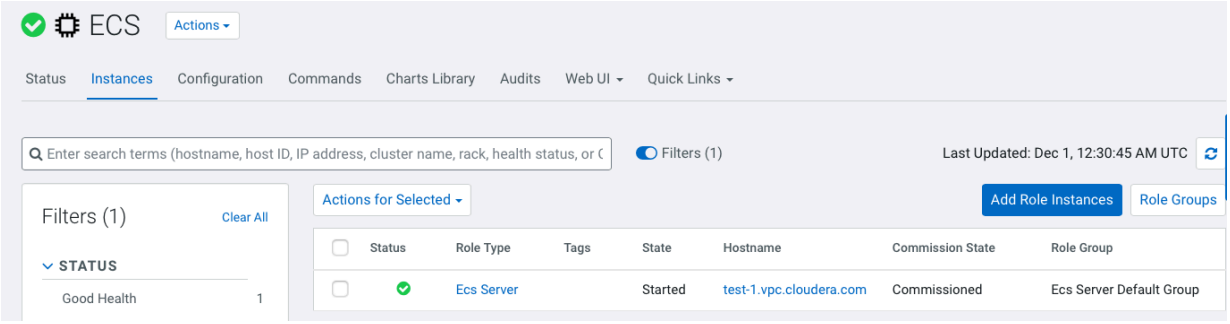
Perform the following steps to get started with ECS and the kubectl command line tool.

Getting started

- In Cloudera Manager, confirm that the ECS Cluster is healthy:



- Identify the ECS server host. Click ECS in the ECS cluster, then click Instances. The ECS server host is listed in the Hostname column for the Ecs Server role type.



- Open a connection to the ECS server host:

```
ssh test-1.vpc.cloudera.com
```

Ensure that you are using the root user:

```
root@test-1 ~]# whoami
root
```

Set up Kubernetes and kubectl

You can use the kubectl command line tool to interact with Kubernetes.

- Cluster management – kubectl allows you to manage various aspects of Kubernetes clusters, including deploying applications, inspecting and managing cluster resources, and viewing logs.
- Interacting with nodes and pods – kubectl provides commands to interact with and manage nodes, pods, and other resources within a Kubernetes cluster.
- Deployment and application management – You can use kubectl to create, update, and delete applications running on Kubernetes, as well as control scaling and rollout of updates.
- Debugging and diagnostics – kubectl includes various commands for checking the health of resources, diagnosing issues, and accessing logs.

To set up Kubernetes configuration and the kubectl alias, add the following lines to your bash profile. This eliminates the need to set this up for each session.

```
KUBECONFIG=/etc/rancher/rke2/rke2.yaml
alias kubectl=/var/lib/rancher/rke2/bin/kubectl
```

Basic operations

This topic describes a few basic kubectl command line tool operations.

View kubeconfig settings

```
[root@test-1 ~]# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://127.0.0.1:6443
  name: default
contexts:
```

```
- context:
    cluster: default
    user: default
  name: default
current-context: default
kind: Config
preferences: {}
users:
- name: default
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

Get all nodes in the Kubernetes environment

```
[root@test-1 ~]# kubectl get nodes
NAME                       STATUS   ROLES
 AGE    VERSION
test-1.vpc.cloudera.com   Ready    control-plane,etcd,master              2d
4h   v1.25.14+rke2r1
test-2.vpc.cloudera.com   Ready    <none>                                 2
d4h   v1.25.14+rke2r1
```

Get all namespaces

```
[root@test-1 ~]# kubectl get namespaces
NAME                                  STATUS   AGE
cdp                                   Active   2d4h
cdp-drs                               Active   2d4h
cdp-services                          Active   2d4h
default                               Active   2d4h
ecs-webhooks                          Active   2d4h
infra-prometheus                      Active   2d4h
k8tz                                  Active   2d4h
kube-node-lease                       Active   2d4h
kube-public                           Active   2d4h
kube-system                           Active   2d4h
kubernetes-dashboard                  Active   2d4h
liftie-wjtncjzm-ns                    Active   2d4h
local-path-storage                    Active   2d4h
longhorn-system                       Active   2d4h
observability                         Active   2d4h
pod-reaper                            Active   2d4h
test-1-5ea742bf-monitoring-platform   Active   2d4h
vault-system                          Active   2d4h
yunikorn                              Active   2d4h
```

Check all pods in a namespace

Use the following command format to check all pods in a namespace:

```
kubectl get pods -n <namespace_name>
```

For example, to get the pods and their status in the vault-system namespace:

```
[root@test-1 ~]# kubectl get pods -n vault-system
NAME                                    READY                STATUS
     RESTARTS             AGE
helm-install-vault-pd842                 0/1                 Completed
         0                  2d6h
```

```
vault-0                                          1/1                 Running
          0                 2d6h
vault-exporter-84bd8f848d-s9grm                  1/1                 Running
         0                2d6h
```

Get the containers in a pod

Use the following command format to get the containers in a pod:

```
root@test-1 ~]# kubectl get pods -n <namespace_name> <pod_name> -o=jsonpath=
'{.spec.containers[*].name}'
```

For example, to get the containers in the fluentd-aggregator-0 pod in the cdp namespace:

```
root@test-1 ~]# kubectl get pods -n cdp fluentd-aggregator-0  -o=jsonpath='{
.spec.containers[*].name}'
thunderhead-diagnostics-api
fluentd-aggregator[
```

Get logs from a specific pod

Use the following command format to get logs from a specific pod:

```
kubectl logs -n <namespace_name> <pod_name>
```

For example, to get the logs from the vault-0 pod in the vault-system namespace:

```
[root@test-1 ~]# kubectl logs -n vault-system vault-0
==> Vault server configuration:

             Api Address: https://10.42.0.15:8200
                     Cgo: disabled
         Cluster Address: https://vault-0.vault-internal:8201
   Environment Variables: GODEBUG, HOME, HOSTNAME, HOST_IP, KUBERNETES_POR
T,
KUBERNETES_PORT_443_TCP, KUBERNETES_PORT_443_TCP_ADDR,
KUBERNETES_PORT_443_TCP_PORT, KUBERNETES_PORT_443_TCP_PROTO,
KUBERNETES_SERVICE_HOST, KUBERNETES_SERVICE_PORT,
KUBERNETES_SERVICE_PORT_HTTPS, NAME, PATH, POD_IP, PWD, SHLVL, SKIP_CHOWN,
SKIP_SETCAP, VAULT_ADDR, VAULT_API_ADDR, VAULT_CACERT, VAULT_CLUSTER_ADDR,
VAULT_K8S_NAMESPACE, VAULT_K8S_POD_NAME, VAULT_PORT, VAULT_PORT_8200_TCP,
VAULT_PORT_8200_TCP_ADDR, VAULT_PORT_8200_TCP_PORT, VAULT_PORT_8200_TCP_PROT
O,
VAULT_PORT_8201_TCP, VAULT_PORT_8201_TCP_ADDR, VAULT_PORT_8201_TCP_PORT,
VAULT_PORT_8201_TCP_PROTO, VAULT_SERVICE_HOST, VAULT_SERVICE_PORT,
VAULT_SERVICE_PORT_HTTPS, VAULT_SERVICE_PORT_HTTPS_INTERNAL, VERSION
              Go Version: go1.20.1
              Listener 1: tcp (addr: "[::]:8200", cluster address: "[::]:8
201",
max_request_duration: "1m30s", max_request_size: "33554432", tls: "enabled")
               Log Level:
                   Mlock: supported: true, enabled: false
           Recovery Mode: false
                 Storage: file
                 Version: Vault v1.13.1, built 2023-03-23T12:51:35Z
             Version Sha: 4472e4a3fbcc984b7e3dc48f5a8283f3efe6f282

==> Vault server started! Log data will stream in below:
2023-11-28T20:34:33.998Z [INFO]  proxy environment: http_proxy="" https_pro
xy="" no_proxy=""
```

```
2023-11-28T20:34:33.998Z [INFO]  core: Initializing version history cache
for core
2023-11-28T20:34:36.013Z [INFO]  core: security barrier not initialized
2023-11-28T20:34:36.014Z [INFO]  core: seal configuration missing, not init
ialized
2023-11-28T20:34:36.014Z [INFO]  core: security barrier not initialized
2023-11-28T20:34:36.015Z [INFO]  core: security barrier initialized: stored=
1 shares=1 threshold=1
2023-11-28T20:34:36.016Z [INFO]  core: post-unseal setup starting
2023-11-28T20:34:36.029Z [INFO]  core: loaded wrapping token key
```

Get logs from a specific container

To get the logs from a specific container, use the following command format:

```
kubectl logs -n <namespace_name> <pod_name> -c container_name
```

For example, the following command lists all of the containers:

```
kubectl get pods cdp-release-thunderhead-environment-85bdfdb466-gprcb -n e2e
-djwl0v -o jsonpath='{.spec.containers[*].name}'
thunderhead-environment fluentbit
```

Then the kubectl logs command can be used with the -c option to return the logs from the thunderhead-environment container:

```
kubectl logs cdp-release-thunderhead-environment-85bdfdb466-gprcb -c thunder
head-environment -n e2e-djwl0v
```

Tunnel into a container

The following example demonstrates how to tunnel into one of the containers above, and then execute a list command:

```
[root@test-1 ~]# kubectl -n cdp exec -it pod/fluentd-aggregator-0 -c fluentd
-aggregator -- bash
[cloudera@fluentd-aggregator-0 /]$ ls -lrth
total 152K
drwxr-xr-x   2 root root    6 Dec 14  2017 srv
drwxr-xr-x   2 root root    6 Dec 14  2017 mnt
drwxr-xr-x   2 root root    6 Dec 14  2017 media
dr-xr-xr-x   2 root root    6 Dec 14  2017 boot
drwxr-xr-x   1 root root   19 Jan 17  2023 usr
lrwxrwxrwx   1 root root    8 Jan 17  2023 sbin -> usr/sbin
lrwxrwxrwx   1 root root    9 Jan 17  2023 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    7 Jan 17  2023 lib -> usr/lib
lrwxrwxrwx   1 root root    7 Jan 17  2023 bin -> usr/bin
drwxr-xr-x   1 root root   17 Jan 17  2023 var
drwxr-xr-x   1 root root   22 Feb 27  2023 opt
drwxr-xr-x   1 root root   21 Feb 27  2023 run
dr-xr-x---   1 root root   30 Feb 27  2023 root
drwxr-xr-x   1 root root   22 Feb 27  2023 home
drwxr-xr-x   1 root root   64 Feb 27  2023 etc
dr-xr-xr-x 656 root root    0 Nov 28 20:39 proc
dr-xr-xr-x  13 root root    0 Nov 28 20:39 sys
drwxr-xr-x   3 root root   17 Nov 28 20:40 fluentd
drwxr-xr-x   5 root root  360 Nov 28 20:40 dev
drwxrwxrwt   1 root root  104 Nov 28 20:40 tmp
```

Collecting diagnostic data

You can collect diagnostic data using the kubectl command line tool, or by downloading diagnostic data bundles.

Using kubectl to collect logs

You can use the following kubectl commands to collect log files:

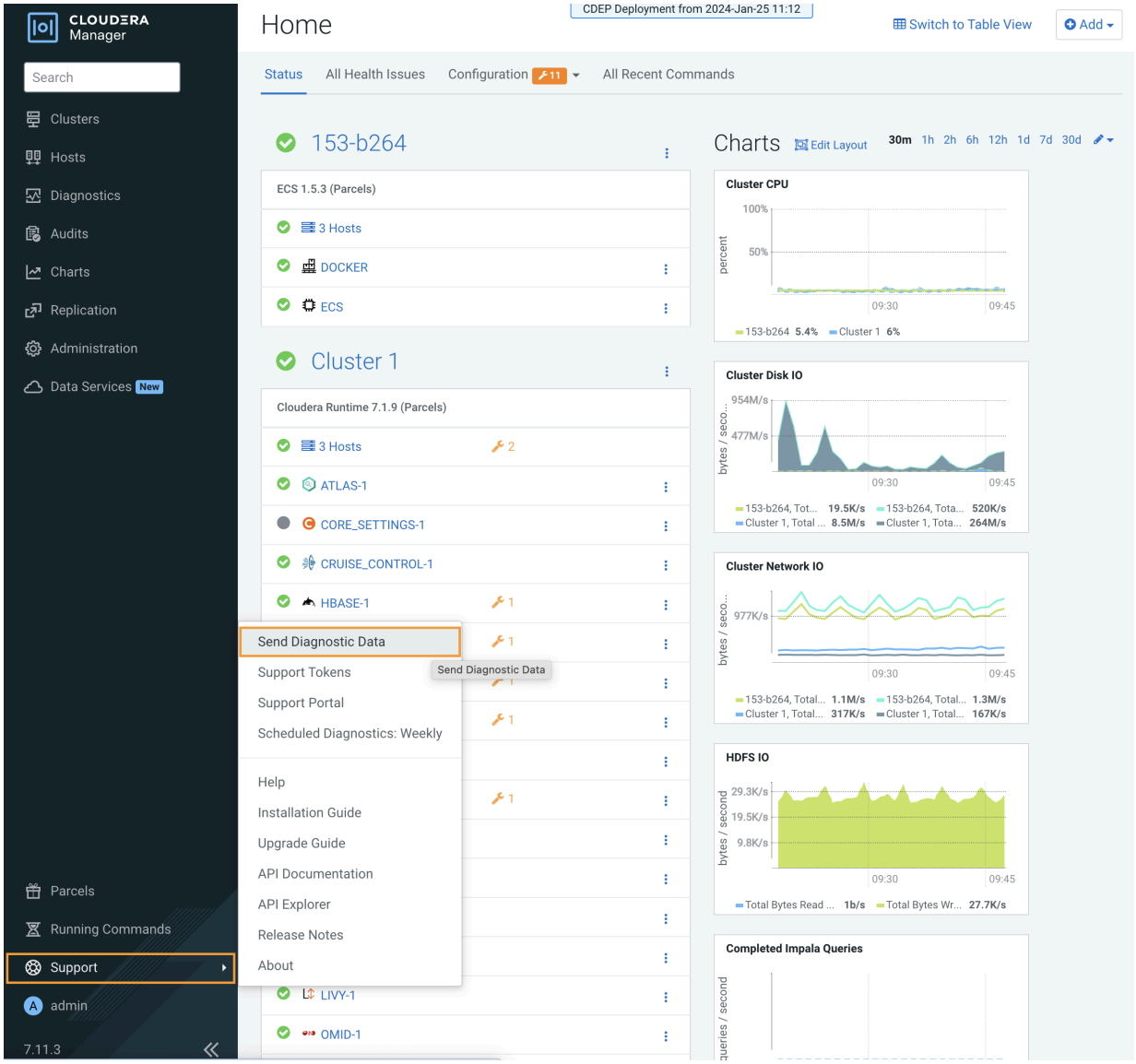
```
alias kubectl='/var/lib/rancher/rke2/bin/kubectl --kubeconfig /etc/rancher/r
ke2/rke2.yaml'

kubectl get pods -A -o wide --kubeconfig /etc/rancher/rke2/rke2.yaml > /tmp/
pods_status_all.txt
echo "+++PODS cdp+++" >> /tmp/pods_status.txt
kubectl -n cdp get pods --kubeconfig /etc/rancher/rke2/rke2.yaml >> /tmp/
pods_status.txt
echo "+++PODS kubernetes-dashboard+++" >> /tmp/pods_status.txt
kubectl -n kubernetes-dashboard get pods >> /tmp/pods.txt

echo "+++EVENTS +++" >> /tmp/pods_status_all.txt
kubectl get event -n kubernetes-dashboard >> /tmp/pods.txt

kubectl get pods -A | grep -vai 'running' | grep -vai 'completed'
```

If you see issues with a specific pod, collect the following:

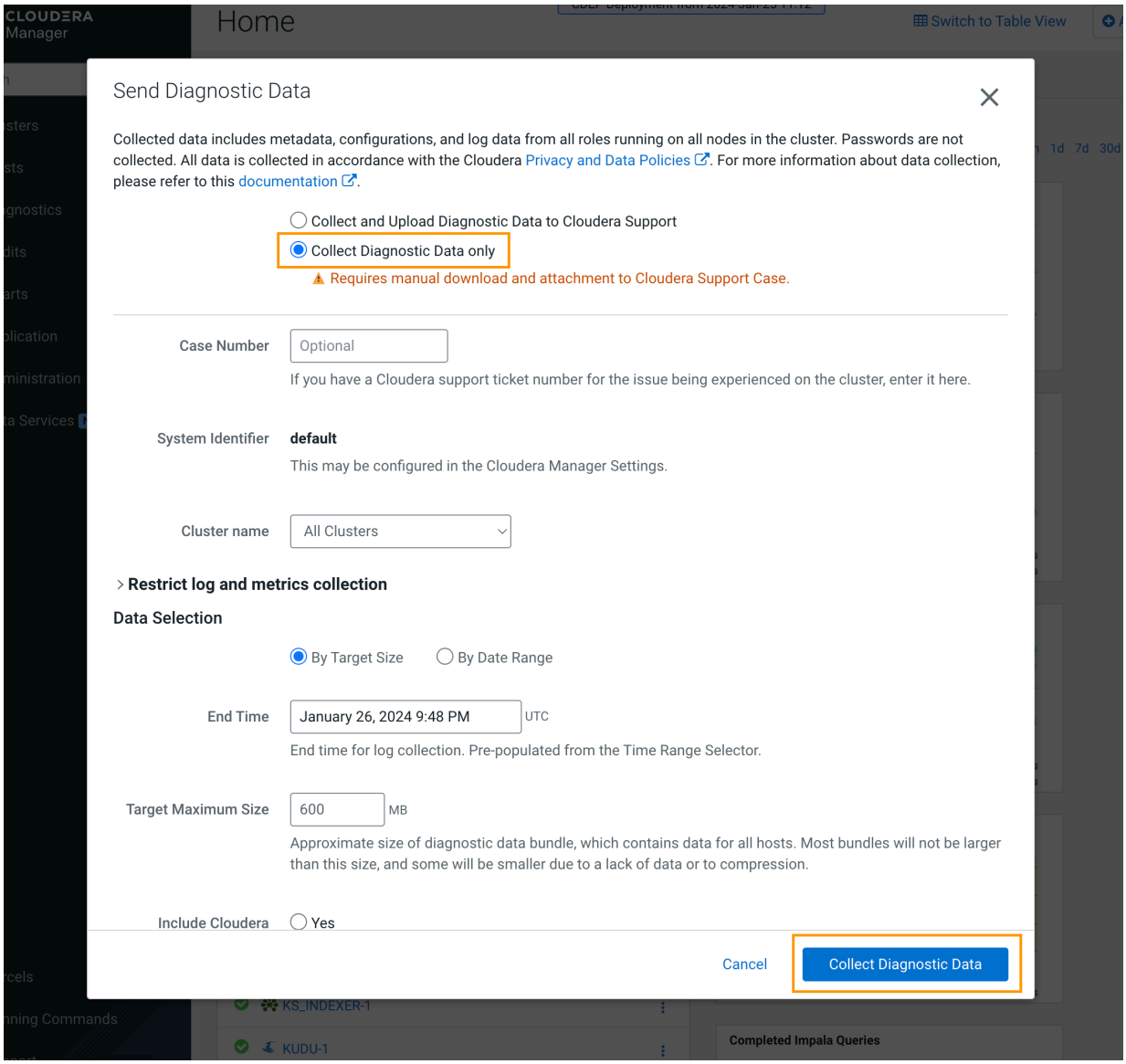
```
kubectl logs <POD_NAME> -n <Namespace> --all-containers=true
kubectl describe pod <POD_NAME> -n <Namespace>
kubectl get events -n <Namespace>
```

Downloading ECS cluster diagnostic data

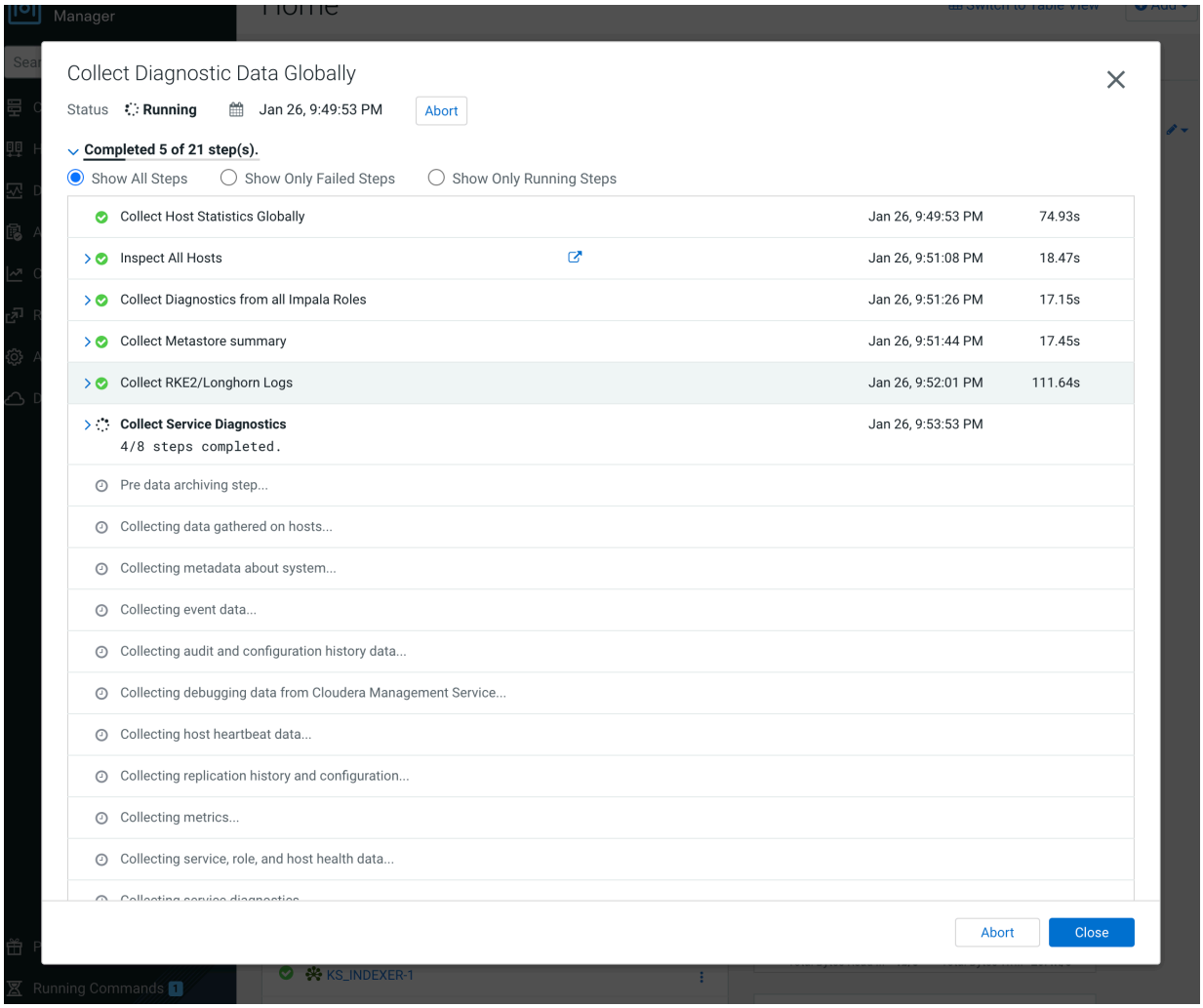
1. In Cloudera Manager, click Support, then click Send Diagnostic Data.



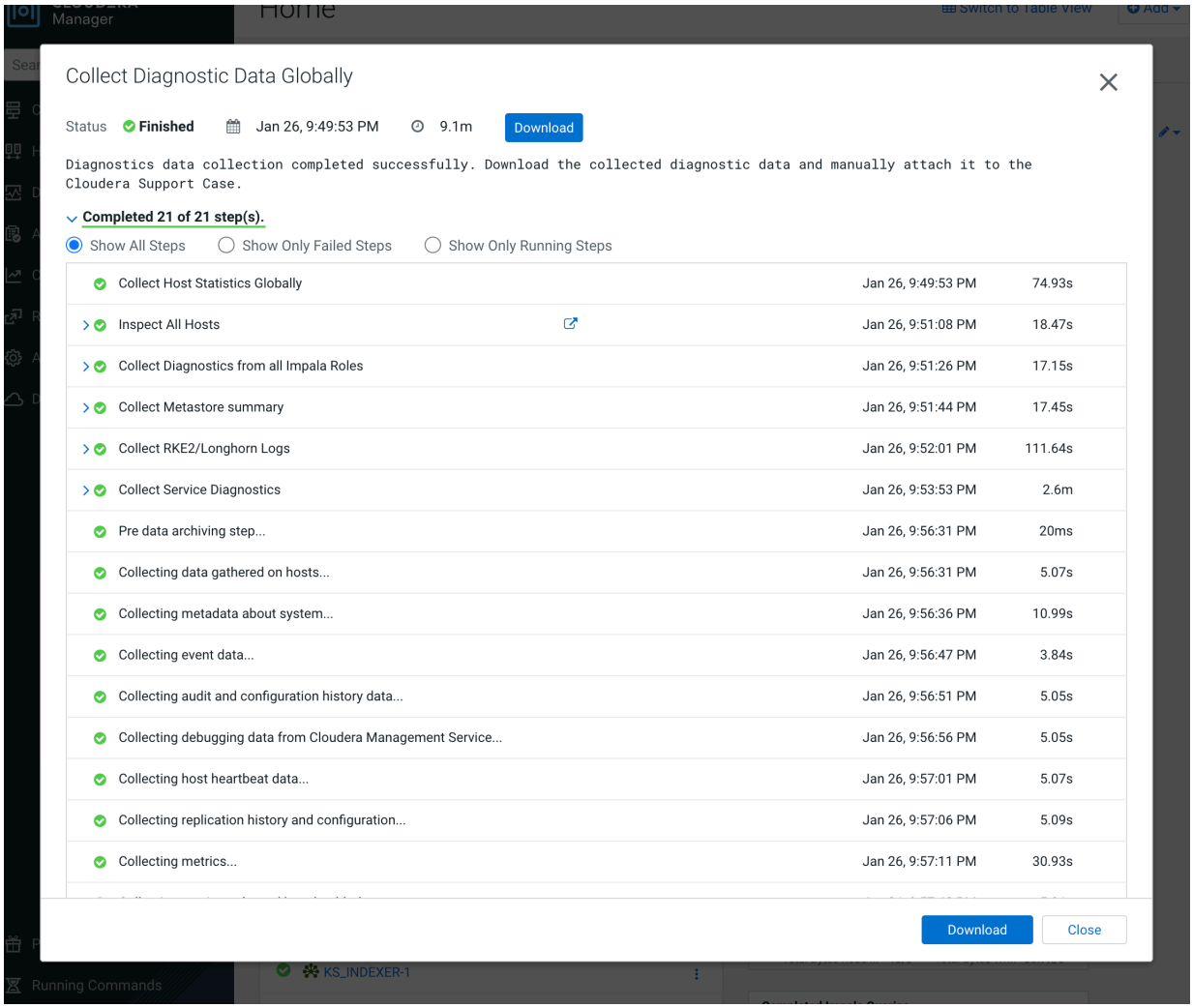
2. On the Send Diagnostic Data pop-up, select Collect Diagnostic Data only, then click Collect Diagnostic Data.



3. The Collect Diagnostic Data Globally pop-up displays the data collection progress.



4. When the data collection process is complete, click Download to download the ECS cluster diagnostic data.



See also: Log support in Cloudera Manager for ECS cluster

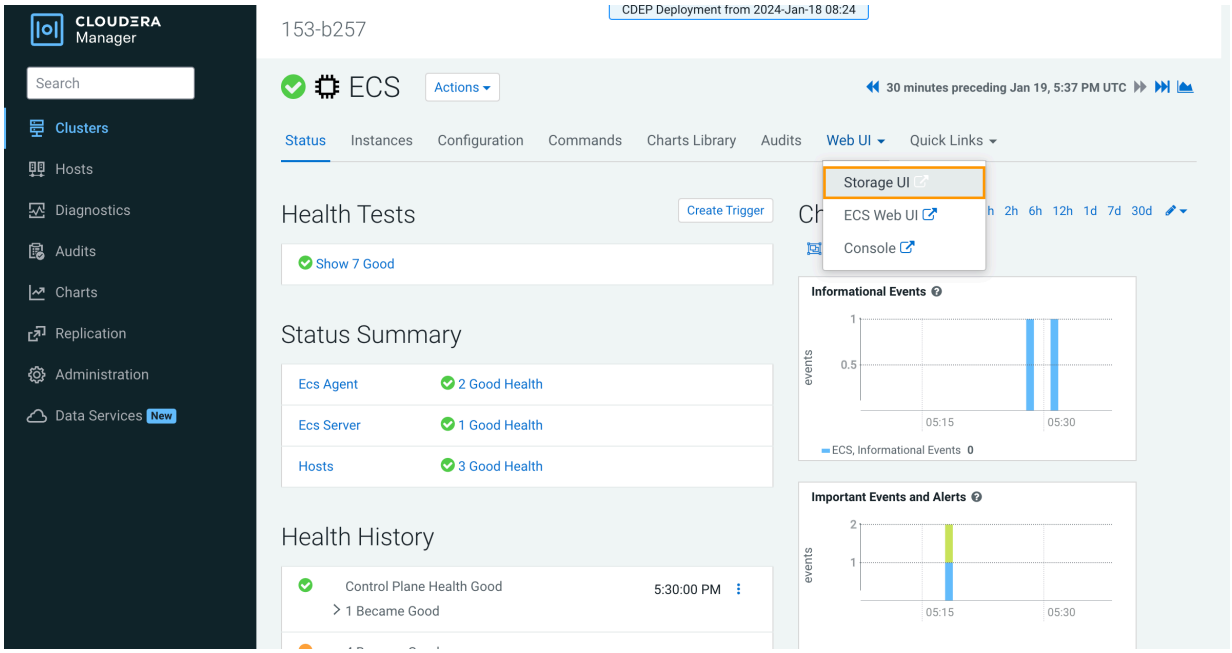
Downloading a Longhorn storage support bundle

For suspected storage issues, you can use the ECS Storage UI to download a Longhorn support bundle.

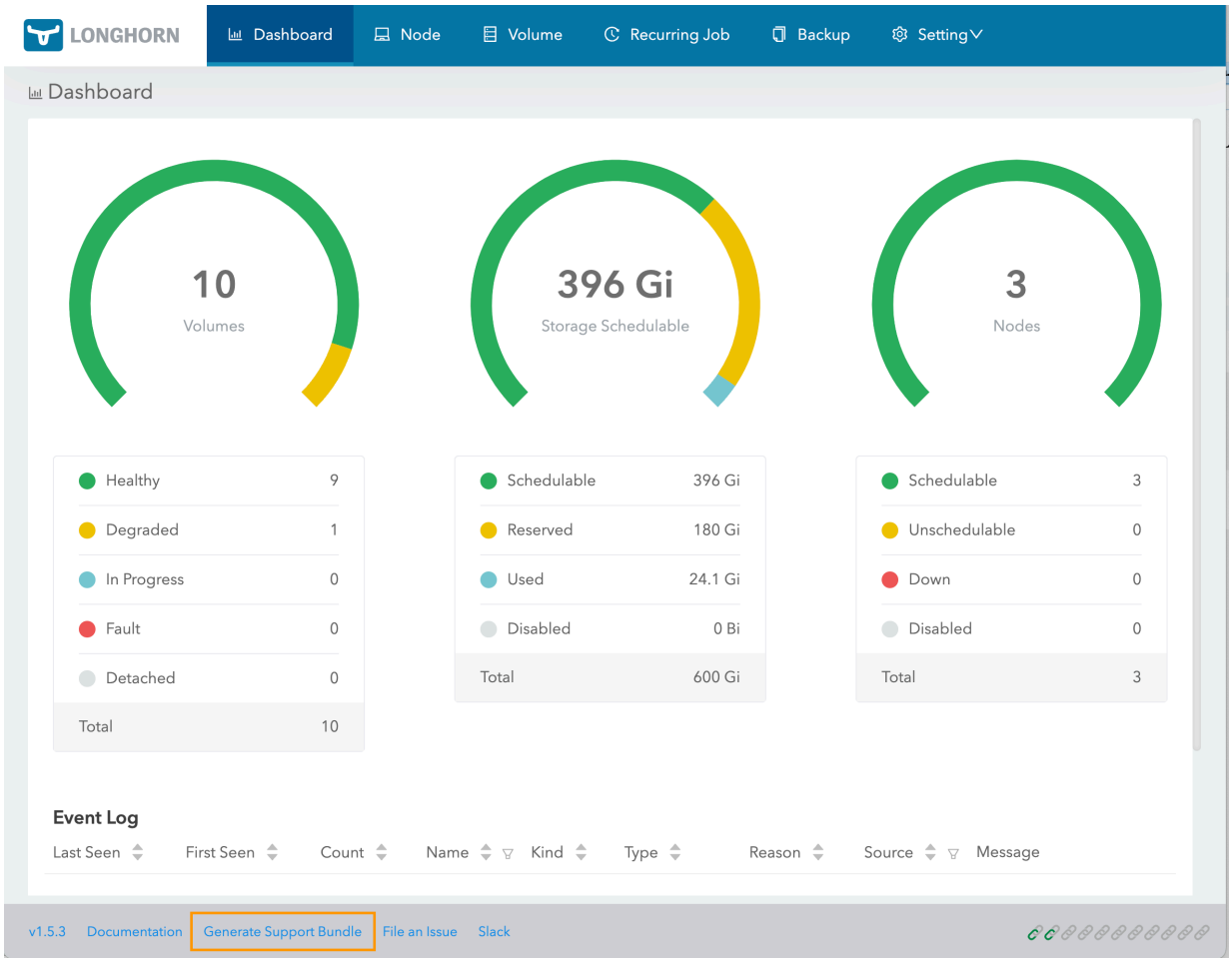
Note:

The Longhorn bundle is included in the Cloudera Manager bundle. If you have already collected the Cloudera Manager bundle, there is no need to download the Longhorn bundle. Also, if the Longhorn UI is down, you can collect the Cloudera Manager bundle as an alternative.

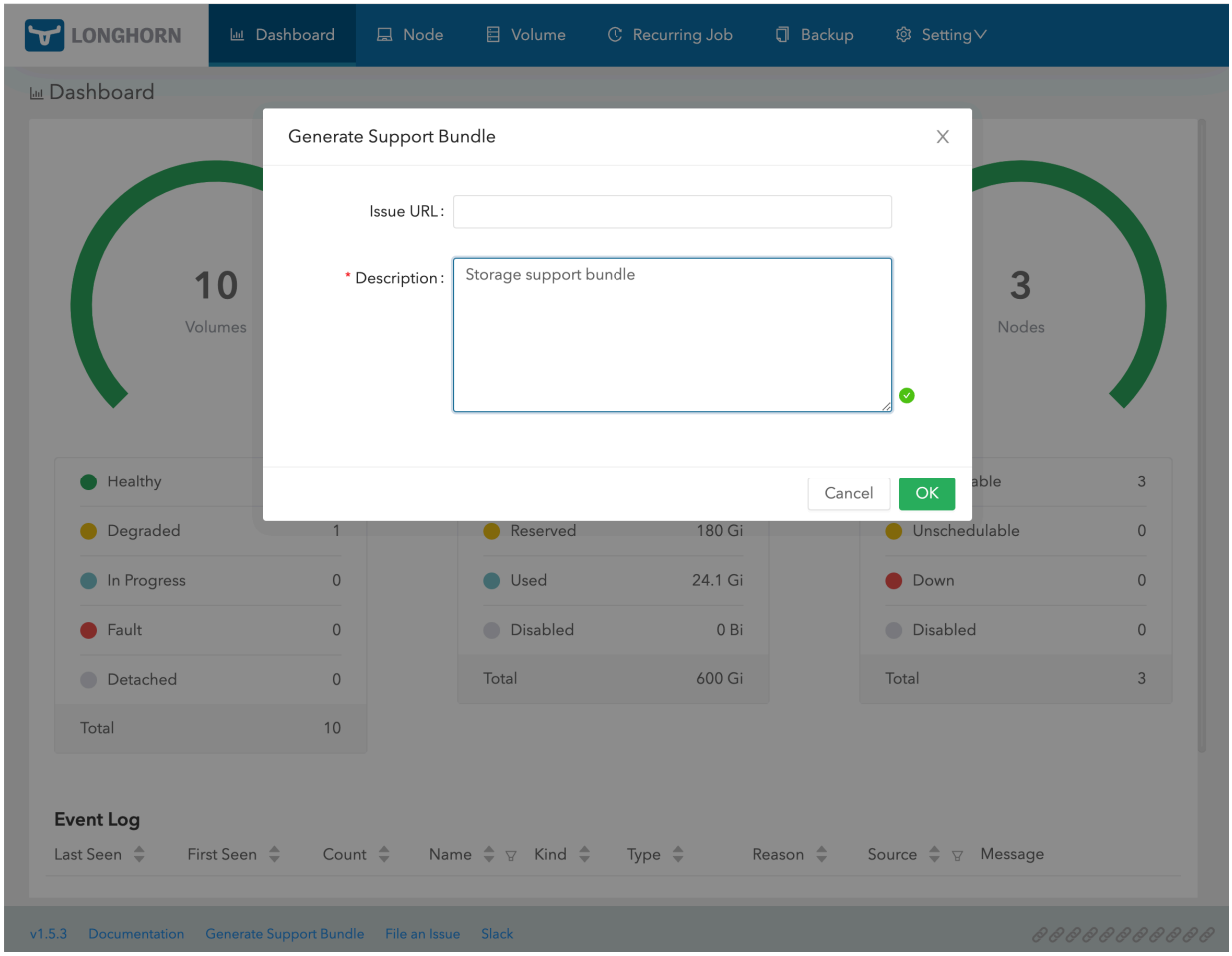
1. In Cloudera Manager, click ECS, then select Web UI > Storage UI.



2. Click Generate Support Bundle at the bottom of the Longhorn storage UI.

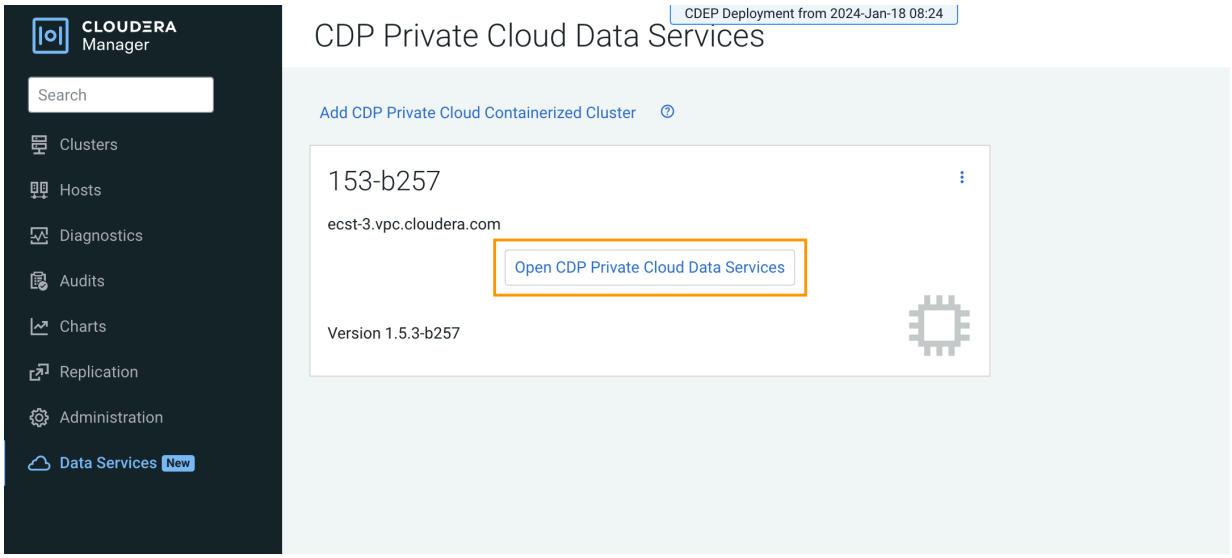


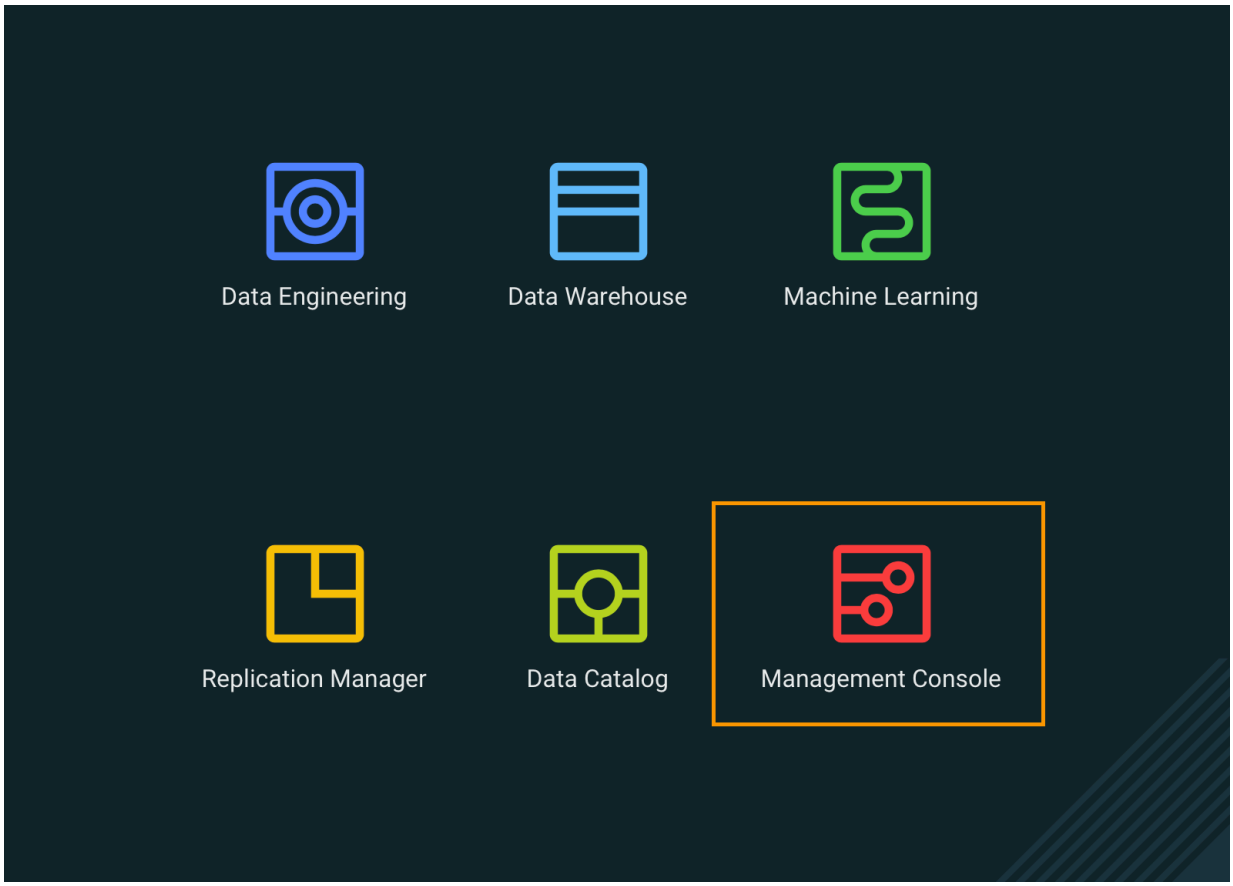
3. On the Generate Support Bundle pop-up, enter a description (Issue URL is optional), then click OK to download the Longhorn storage support bundle.



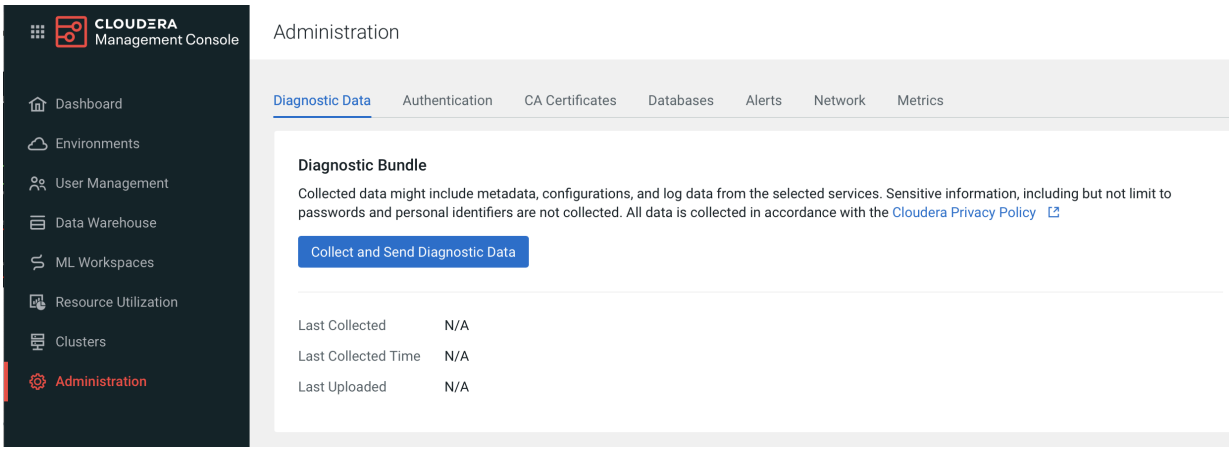
Downloading CDP Private Cloud diagnostic data

1. To access the Management Console, click Data Services in Cloudera Manager, then click Open CDP Private Cloud Data Services, and then select Management Console.

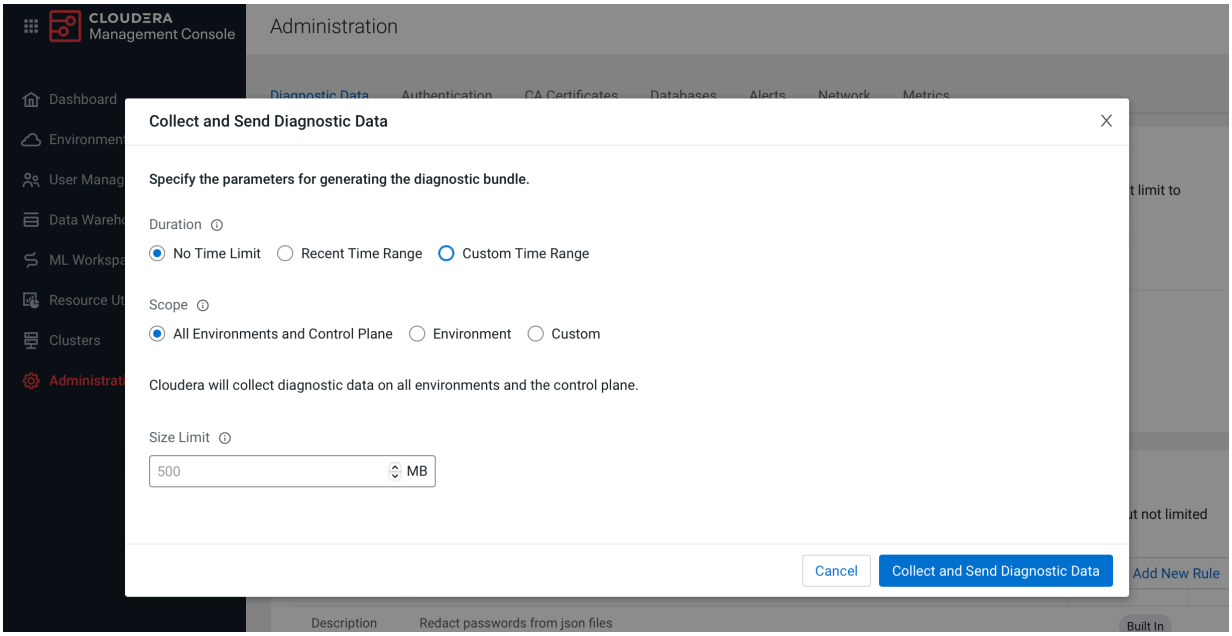




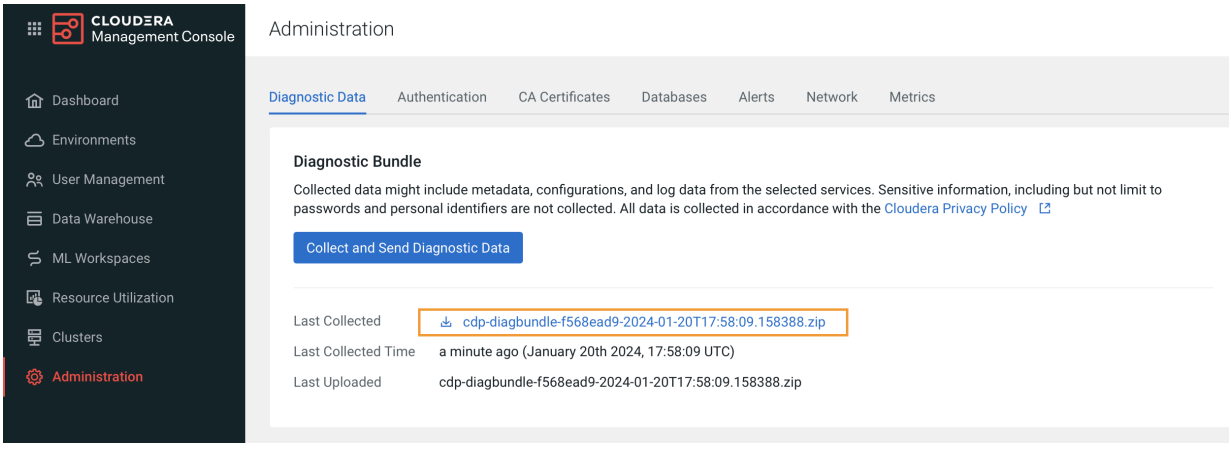
2. To download a diagnostic bundle, click Administration, then click Collect and Send Diagnostic Data.



3. On the Collect and Send Diagnostic Data pop-up, specify the duration, scope, and a size limit for the data, then click Collect and Send Diagnostic Data.



4. When the data collection process is complete, the .zip download file appears on the Diagnostic Data page.



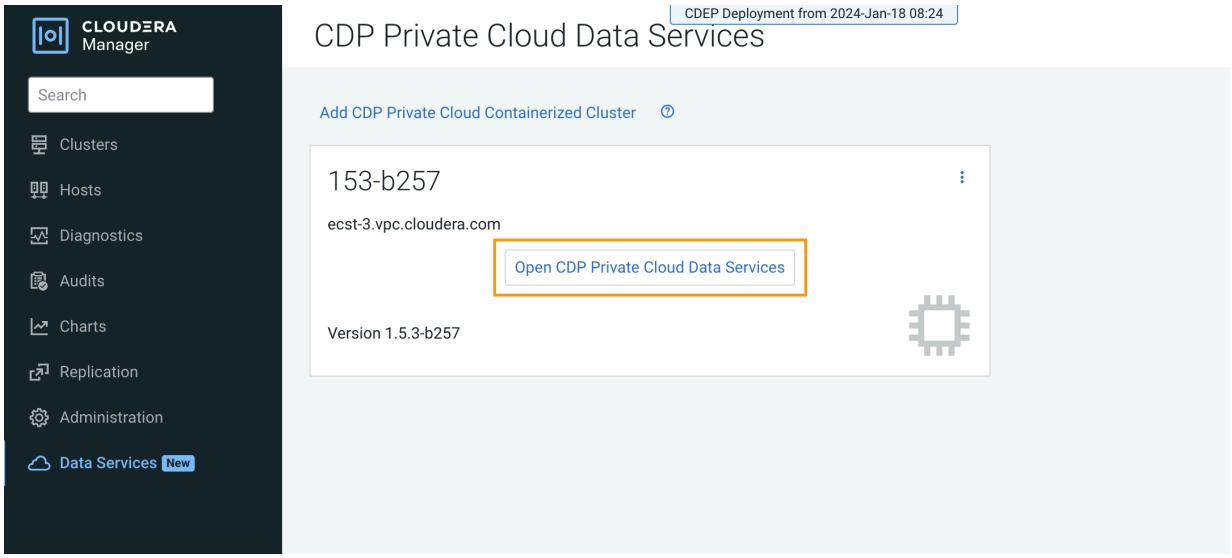
See also: Working with CDP Private Cloud diagnostic data

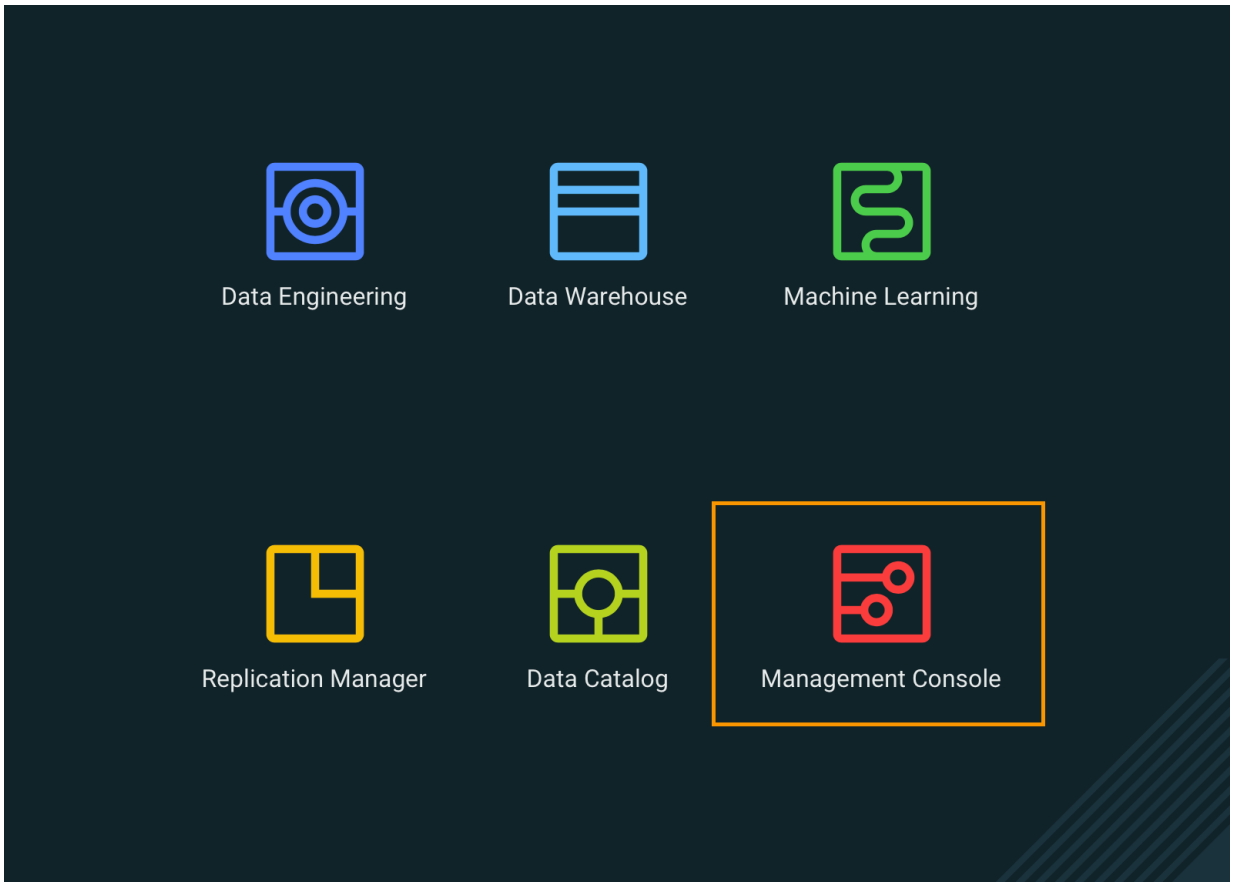
Proactive monitoring

You can define alert rules for your CDP Private Cloud Data Services deployment based on PromQL expressions. The alerts are automatically triggered when specific events occur in your deployment. You can view the triggered alerts on the Management Console dashboard. Any alert receivers that you have already configured start sending notifications to specified endpoints.

Configuring alert rules

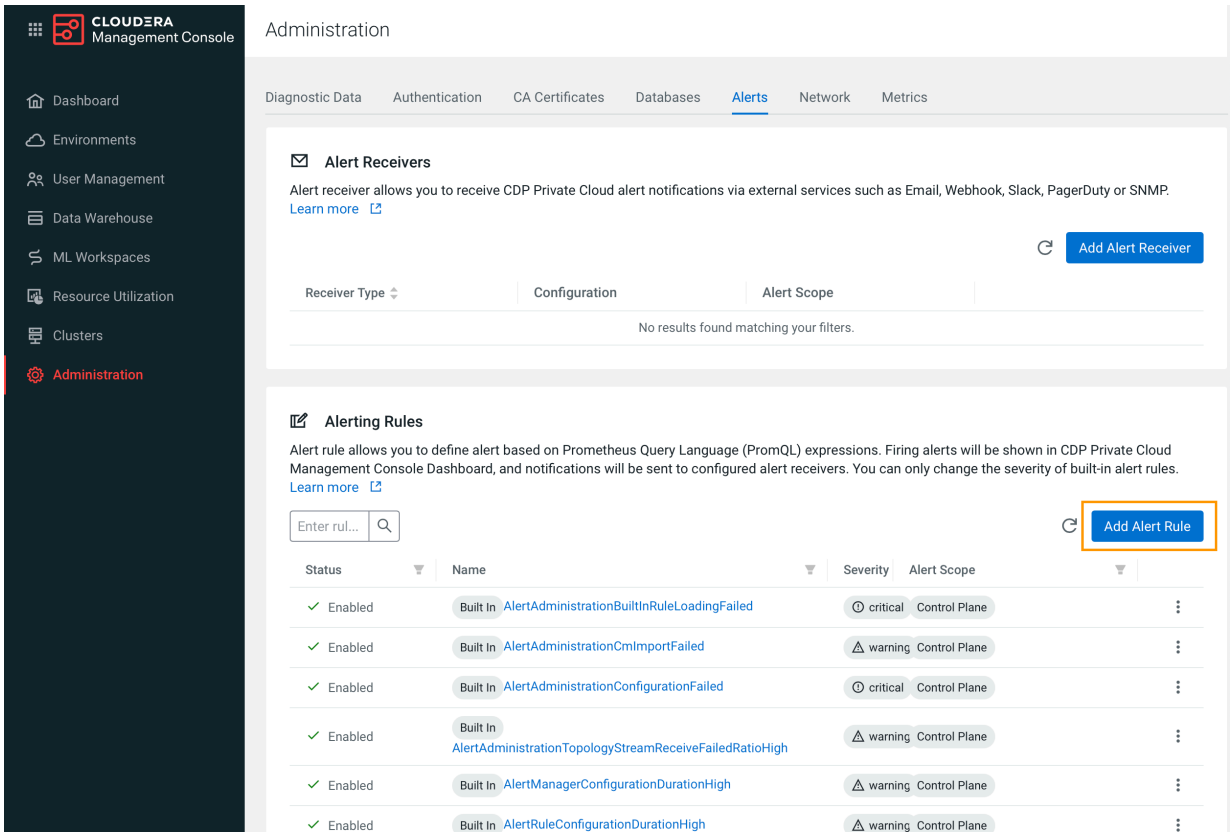
1. To access the Management Console, click Data Services in Cloudera Manager, then click Open CDP Private Cloud Data Services, and then select Management Console.





2. On the Management Console home page, select Administration > Alerts.

3. On the Alerts page, click Add Alert Rule.

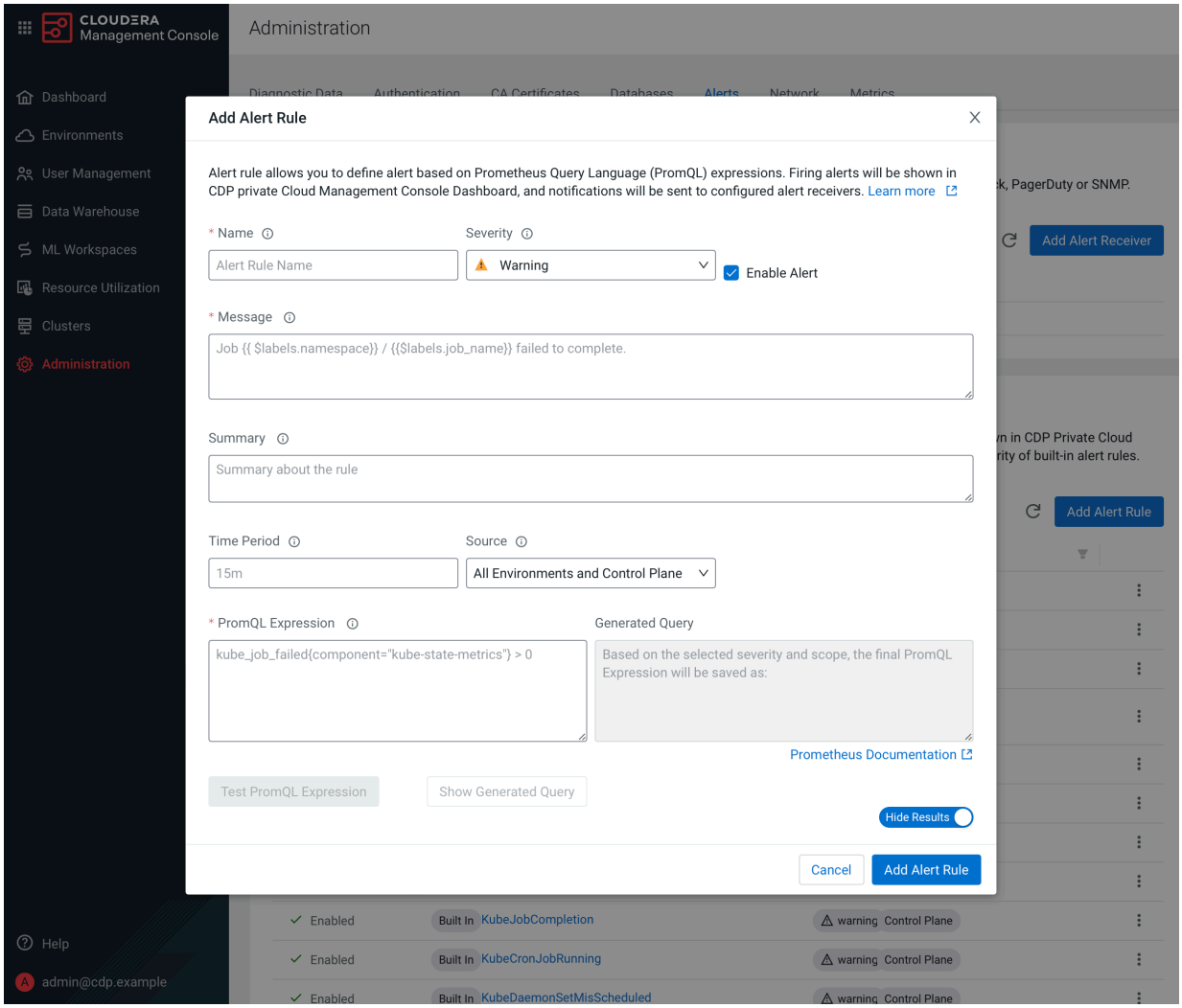


4. On the Add Alert Rule pop-up, enter the following alert rule options (required fields are indicated in bold) then click Add Alert Rule.

Field	Description
Name	The name of the alert rule. You cannot use spaces or special characters in the name.
Severity	Specify the severity: Critical or Warning.
Enable Alert	Select this checkbox to enable the alert rule.
Message	The alert rule text. You can use PromQL labels to denote entities such as jobs in the text. For more information about using PromQL labels, see Alerting Rules.
Summary	A brief summary of the alert rule.
Time Period	The duration for which the PromQL expression must be true. If the expression continues to be true after the specified duration, the configured alert is automatically triggered.
Source	The CDP Private Cloud components for which the alert rule applies. You can select one of the following options as the source: • All Environments and Control Plane • Environments • Control Plane • A specific environment from the list of configured environments

Field	Description
PromQL Expression	The query expression in PromQL. The alert is issued when this expression is true for the time period specified in the For Clause.  **Important:** Metrics reported by the environments always contain the following labels: appId and appName. Therefore, the result of the alert rule's query expression also must contain these labels. To ensure that the result contains the labels, include the by (appId, appName) clause when using aggregation operators in the query expression. For example, instead of the count(my_metric) > 0 expression, use the count(my_metric) by (appId, appName) > 0 expression.
Generated Query	The query that is generated for a selected workload type depending on the specified PromQL expression. You can view the query by clicking Show Generated Query.

Field	Description
Test PromQL Expression	You can click this option to test the query expression generated for the combination of a selected source and workload type. If you select one of All Environments and Control Plane, Environments, or Control Plane as the source for the PromQL query, it runs *only* on the control plane. To run the query on an environment, you must select a specific environment as the source. **Note:** If you click this option *before* saving the alert rule and if the PromQL expression is invalid, an unexpected error appears. In addition, you may lose all of the information entered for configuring the alert rule. Therefore, you should save the alert rule and then test the PromQL expression.



The new alert rule is listed on the Alerts page under Alerting Rules.

Environment health checks

You can use environment health checks to verify the health of various ECS components. If you are experiencing issues, these tests can help you diagnose and solve the problem.

Host health checks

Check the status of all nodes in the Kubernetes environment

```
[root@test-1 ~]# kubectl get nodes
NAME                       STATUS   ROLES                       AGE    VER
SION
test-1.vpc.cloudera.com    Ready    control-plane,etcd,master   2d4h   v1.25
.14+rke2r1
test-2.vpc.cloudera.com    Ready    <none>                      2d4h   v1.
25.14+rke2r1
```

Ensure that the namespaces are all active

```
[root@test-1 ~]# kubectl get namespaces
NAME                                  STATUS   AGE
cdp                                   Active   2d4h
cdp-drs                               Active   2d4h
cdp-services                          Active   2d4h
default                               Active   2d4h
ecs-webhooks                          Active   2d4h
infra-prometheus                      Active   2d4h
k8tz                                  Active   2d4h
kube-node-lease                       Active   2d4h
kube-public                           Active   2d4h
kube-system                           Active   2d4h
kubernetes-dashboard                  Active   2d4h
liftie-wjtncjzm-ns                    Active   2d4h
local-path-storage                    Active   2d4h
longhorn-system                       Active   2d4h
observability                         Active   2d4h
pod-reaper                            Active   2d4h
test-1-5ea742bf-monitoring-platform   Active   2d4h
vault-system                          Active   2d4h
yunikorn                              Active   2d4h
```

Vault health checks

Vault states

There are three possible states the Vault can be in:

- Initialization:

  This involves preparing the Vault's storage back end to accept data. This cannot be executed on a Vault cluster that has already been initialized. The Vault operates with a self-signed certificate, and the ecs_util.sh script encompasses all of the necessary steps for this process.
- Unsealing:

  If the Vault is resealed, restarted, or stopped, a minimum of three keys are required to unseal it to resume request handling. The Vault does not retain the generated root key, as the root key must be reconstructed using at least three keys, or the vault remains permanently sealed. Cloudera stores the root key in Cloudera Manager, and this is the key that is used when the unseal option is selected from the Cloudera Manager user interface.

- Startup:

  After completing initialization and unsealing, the Vault is ready to be started. Once operational, it can begin processing requests.

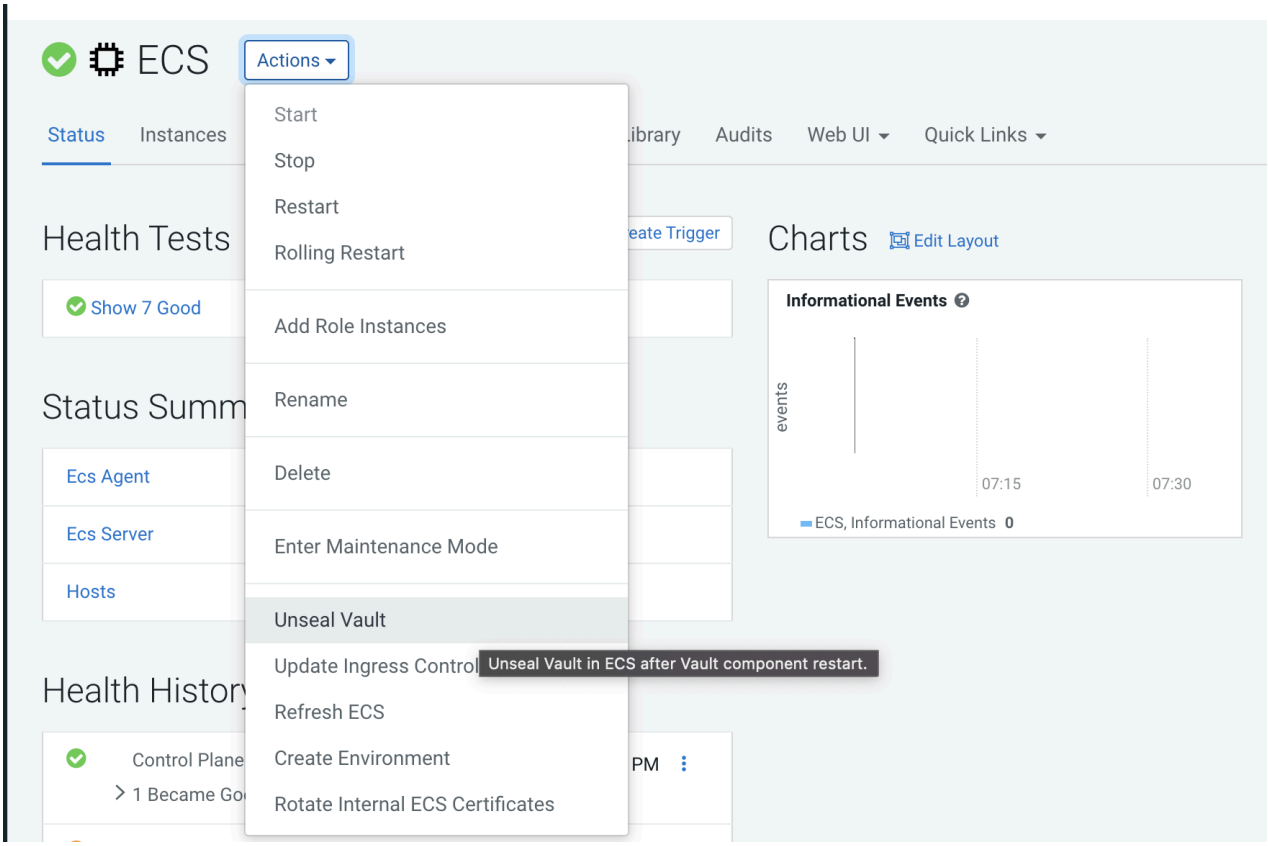
Check the Vault status

```
[root@test-1 ~]# kubectl get pods -n vault-system
NAME                                READY       STATUS          RESTARTS
 AGE
helm-install-vault-pd842            0/1         Completed         0
  2d6h
vault-0                             1/1         Running           0
 2d6h
vault-exporter-84bd8f848d-s9grm     1/1         Running           0
  2d6h
```

Unseal the Vault using Cloudera Manager

You should only unseal the Vault if there are issues reported in the logs about the Vault being sealed, or if pods in the Vault namespace are crash-looping.

To unseal the Vault, select the ECS cluster in Cloudera Manager, click ECS, then select Actions > Unseal Vault.



Storage health checks

Check storage mounts

```
[cloudera@fluentd-aggregator-0 /]$ mount | grep longhorn
```

```
/dev/longhorn/pvc-2f71ea50-744c-4eb9-875c-3f793d141961 on /var/log type ext4
 (rw,relatime,data=ordered)
```

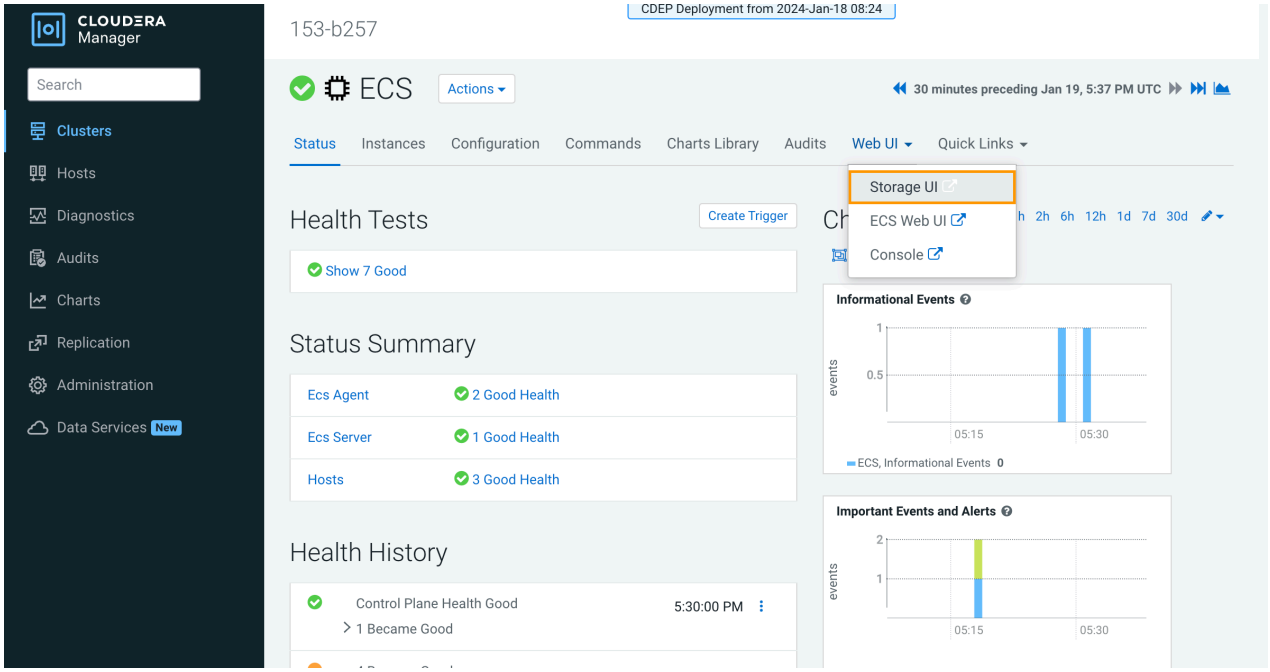
Check Longhorn status

```
[root@test-1 ~]# kubectl get pods -n longhorn-system
NAME                                                READY   STATUS      R
ESTARTS   AGE
csi-attacher-7b556d5f87-2rttk                       1/1     Running     0
          2d10h
csi-attacher-7b556d5f87-ldst4                       1/1     Running     0
          2d10h
csi-attacher-7b556d5f87-nsrnn                       1/1     Running     0
          2d10h
csi-provisioner-76f6697668-567c5                    1/1     Running     0
           2d10h
csi-provisioner-76f6697668-6smx5                    1/1     Running     0
          2d10h
csi-provisioner-76f6697668-w82z5                    1/1     Running     0
          2d10h
csi-resizer-5d8b75df89-gn7jk                        1/1     Running     0
          2d10h
csi-resizer-5d8b75df89-m2r87                        1/1     Running     0
           2d10h
csi-resizer-5d8b75df89-zthrl                        1/1     Running     0
          2d10h
csi-snapshotter-c54d8cbd8-2vmxs                     1/1     Running     0
          2d10h
csi-snapshotter-c54d8cbd8-52sjc                     1/1     Running     0
          2d10h
csi-snapshotter-c54d8cbd8-f49gj                     1/1     Running     0
           2d10h
engine-image-ei-791d1d81-7bv7b                      1/1     Running     0
          2d10h
engine-image-ei-791d1d81-zb2kv                      1/1     Running     0
          2d10h
helm-install-longhorn-zchvx                         0/1     Completed   0
          2d10h
instance-manager-e-050ae22aa5b0f98c28dc7da17d4e6ba2 1/1     Running     0
           2d10h
instance-manager-e-5830ecda079889e4a49271591835ceb2 1/1     Running     0
          2d10h
instance-manager-r-050ae22aa5b0f98c28dc7da17d4e6ba2 1/1     Running     0
          2d10h
instance-manager-r-5830ecda079889e4a49271591835ceb2 1/1     Running     0
          2d10h
longhorn-admission-webhook-6cb4bb94f-2252d          1/1     Running     0
           2d10h
longhorn-admission-webhook-6cb4bb94f-hfqvz          1/1     Running     0
          2d10h
longhorn-conversion-webhook-76fd55b9-rklhz          1/1     Running     0
          2d10h
longhorn-conversion-webhook-76fd55b9-sxkkb          1/1     Running     0
          2d10h
longhorn-csi-plugin-czjtm                           3/3     Running     0
           2d10h
longhorn-csi-plugin-f26j7                           3/3     Running     0
          2d10h
longhorn-driver-deployer-7b64685666-7nx6v           1/1     Running     0
          2d10h
longhorn-manager-tjz5h                              1/1     Running     0
          2d10h
```

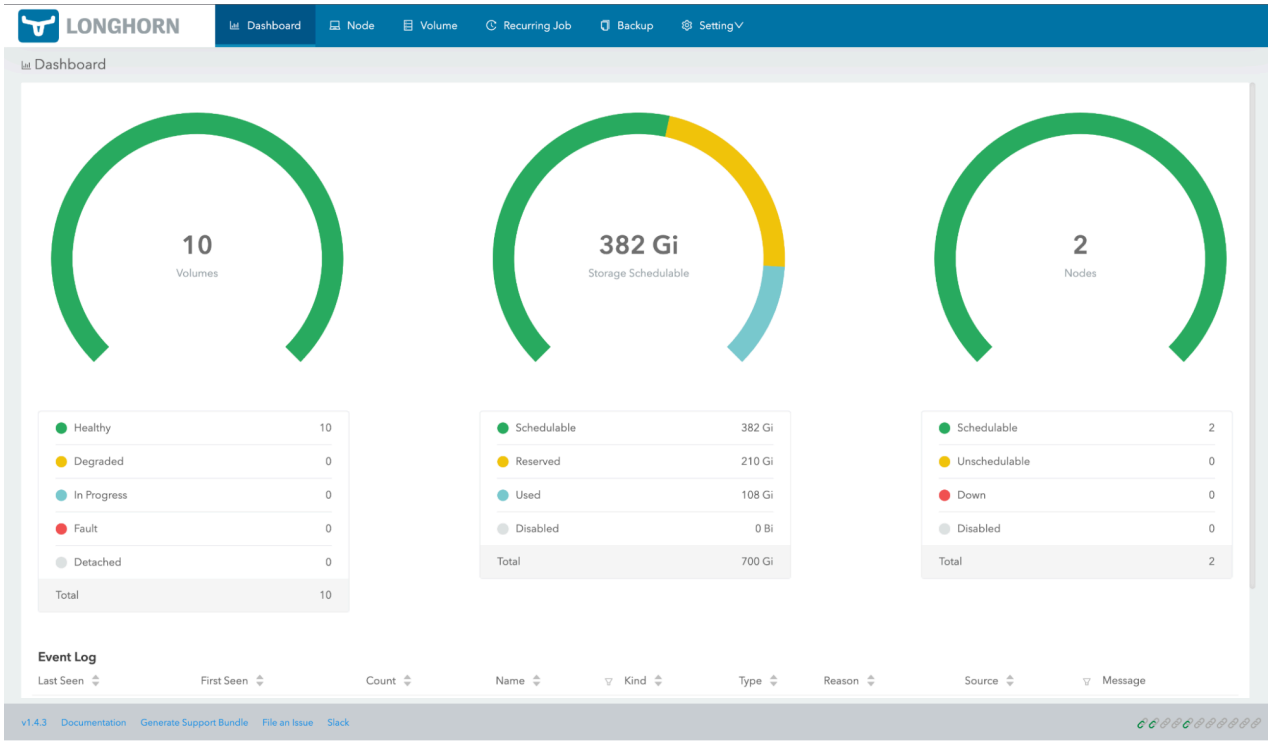
```
longhorn-manager-x9r26                          1/1     Running     0
          2d10h
longhorn-recovery-backend-fc6dccdcb-vnqb6       1/1     Running     0
          2d10h
longhorn-recovery-backend-fc6dccdcb-w4fdr       1/1     Running     0
          2d10h
longhorn-ui-79c96b46cb-4jqrq                    1/1     Running     0
          2d10h
longhorn-ui-79c96b46cb-fvn5g                    1/1     Running     0
          2d10h
```

Check Longhorn status using the UI

In Cloudera Manager, click ECS, then select Web UI > Storage UI.



A healthy system should show healthy volumes, schedulable storage, and schedulable nodes:



Persistent volume claims

Use the following command format to list the persistent volume claims in a namespace:

```
kubectl get pvc -n <namespace>
```

For example, to list the persistent volume claims in the cdp namespace:

```
[root@test-1 ~]# kubectl get pvc -n cdp
NAME                                                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
cdp-embedded-db-backend                             Bound    pvc-0c2691ba-1ec7-422d-be73-fce0a815da36   20Gi       RWO            longhorn       2d8h
cdp-release-prometheus-server                       Bound    pvc-8b68e520-4403-4b44-a3d2-efbbbcf492d5   20Gi       RWO            longhorn       2d8h
logs                                                Bound    pvc-2f71ea50-744c-4eb9-875c-3f793d141961   20Gi       RWO            longhorn       2d8h
storage-volume-cdp-release-prometheus-alertmanager-0   Bound    pvc-d74e4e7e-ef57-4229-93b6-220dcd9b55cc   2Gi        RWO            longhorn       2d8h
storage-volume-cdp-release-prometheus-alertmanager-1   Bound    pvc-9ff7debf-6135-43d5-86c5-c40aa0e3775d   2Gi        RWO            longhorn       2d8h
```

Common storage issues and workarounds

longhorn-manager not present

Check to see if the longhorn-manager daemonset exists in the longhorn-system namespace. If not, it may have been accidentally deleted. To restore it via the Helm chart:

```
export KUBECONFIG=/etc/rancher/rke2/rke2.yaml
cd /opt/cloudera/parcels/ECS/installer/install/bin/linux
./helm history longhorn -n longhorn-system (note down the latest revision)
./helm rollback longhorn <revision> -n longhorn-system
```

Volume fails to attach to node

When this issue occurs, Longhorn manager reports the following error:

```
time="2023-03-03T01:42:30Z" level=warning
```

```
msg="pvc-e930fca4-0c90-44b0-bedb-9d9d39ec197c-e-c87678d7: 2023/03/02 09:2
7:40
cannot create an available backend for the engine from the addresses
[tcp://10.42.0.21:10120]"
```

Checking the instance-manager pod logs, it shows a discrepancy between the actual and the expected volume size. The volume size has drifted from the requested pvc:

```
[pvc-e930fca4-0c90-44b0-bedb-9d9d39ec197c-r-57d7d0e6]
time="2023-03-03T01:48:08Z" level=info msg="Opening volume
/host/ecs/longhorn-storage/replicas/pvc-e930fca4-0c90-44b0-bedb-9d9d39ec1
97c-fb
bf1fa2, size 10737418240/512"

2023-01-30T14:59:53.514816555-08:00 stderr F
[pvc-84f1c799-284c-4676-9c3a-34a7fdcfe8cc-e-3b7dabc9]
time="2023-01-30T22:59:53Z" level=warning msg="backend tcp://10.42.1.47:1
0000
size does not match 2147483648 != 64424509440 in the engine initiation pha
se"
```

This can be resolved by updating the volume size to the original expected size:

1. SSH into the node that has the replica.
2. cd into the replica folder, for example:

   ```
   cd /longhorn/replicas/pvc-126d40e2-7bff-4679-a310-1a5dc941
   ```

3. Change the size field from its current value to the expected value in the volume.meta file.

Host-level tasks

Starting, stopping, restarting, and refreshing Embedded Container Service Clusters

Procedures to start, stop, restart, and refresh Private Cloud Experience clusters

Starting a Embedded Container Service Cluster

Procedure

1. On the HomeStatus tab, click the Actions Menu to the right of the Embedded Container Service cluster name and select Start.
2. Click the Start button that appears in the next screen to confirm. The Command Details window shows the progress of starting services.

Results

When the All services successfully started message appears, the task is complete and you can close the Command Details window.

Stopping a CDP Private Cloud Data Services Cluster

Procedure

1. On the HomeStatus tab, click the Actions Menu to the right of the Embedded Container Service cluster name and select Stop.
2. Click the Stop button in the confirmation screen. The Command Details window shows the progress of stopping services.

Results

When the All services successfully stopped message appears, the task is complete and you can close the Command Details window.

Note: The cluster-level Stop action does not stop the Cloudera Management Service. You must stop the Cloudera Management Service separately.

Restarting a Embedded Container Service Cluster

Procedure

1. On the HomeStatus tab, click the Actions Menu to the right of the cluster name and select Restart.
2. Click the Restart button that appears in the next screen to confirm.
   The Command Details window shows the progress of stopping services. When the All services successfully started message appears, the task is complete and you can close the Command Details window.
3. Click ActionsUnseal Vault

Refreshing a Embedded Container Service Cluster

Procedure

To refresh a cluster, in the HomeStatus tab, click the Actions Menu to the right of the cluster name and select Refresh Cluster.

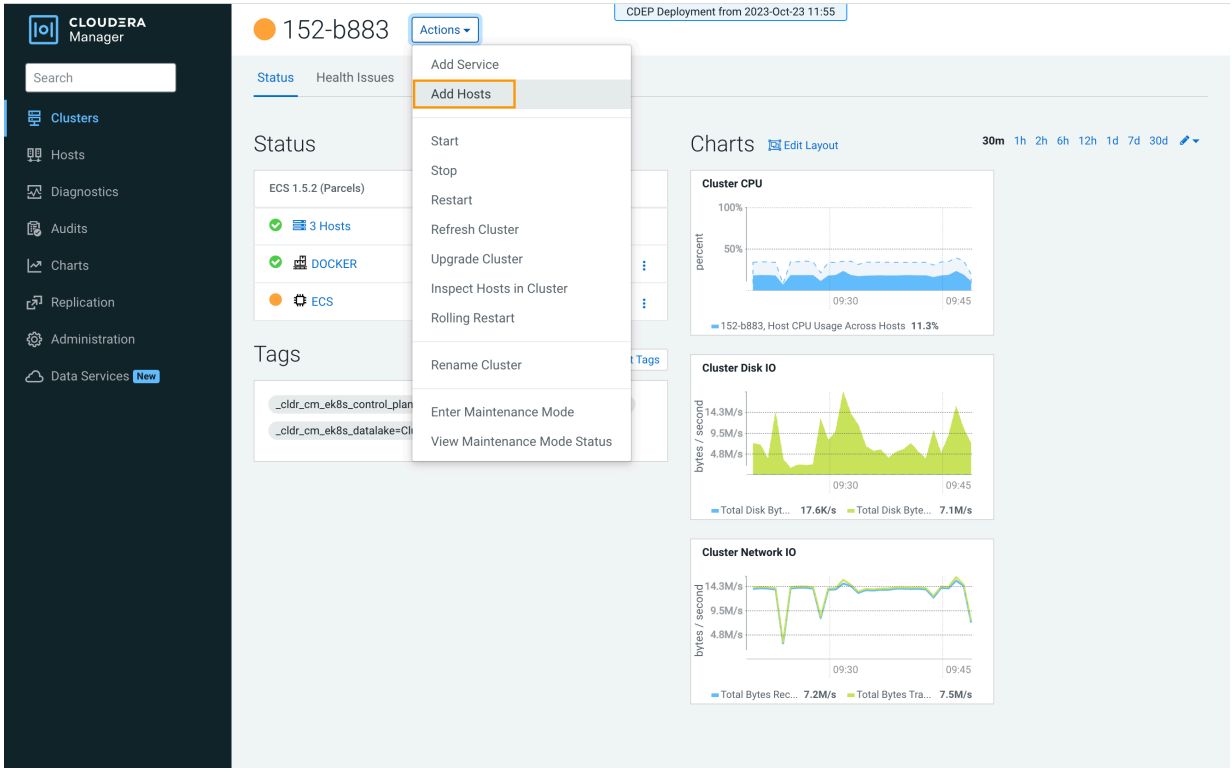
Adding hosts to a Embedded Container Service Cluster

You can add hosts to a Embedded Container Service (ECS) cluster to increase capacity and performance.

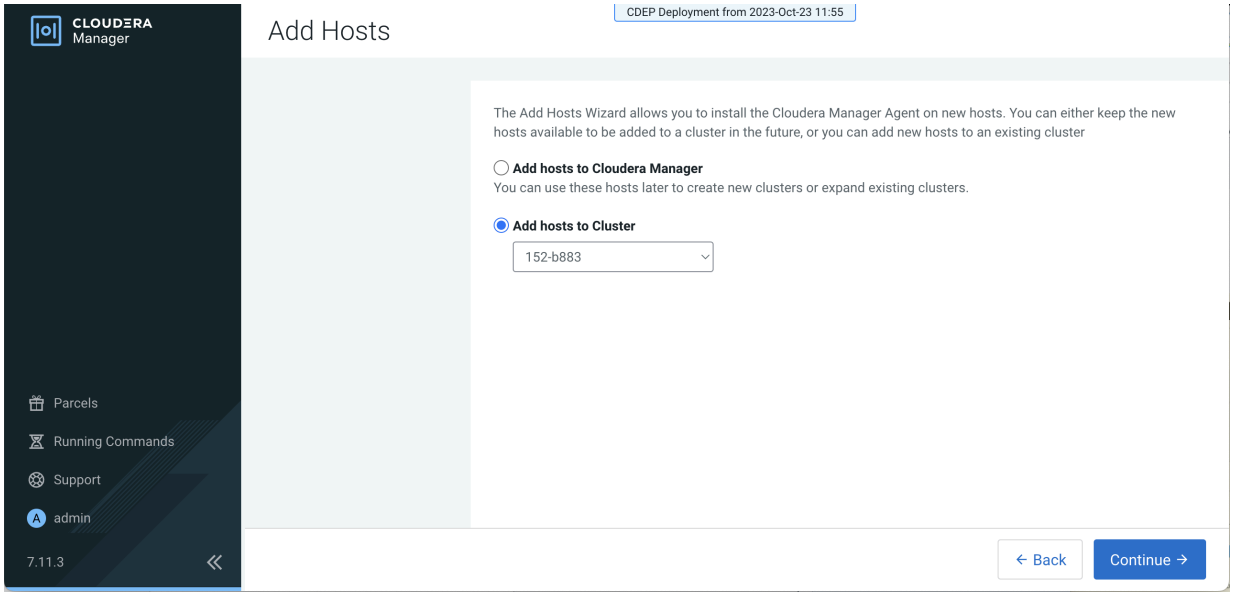
About this task

Procedure

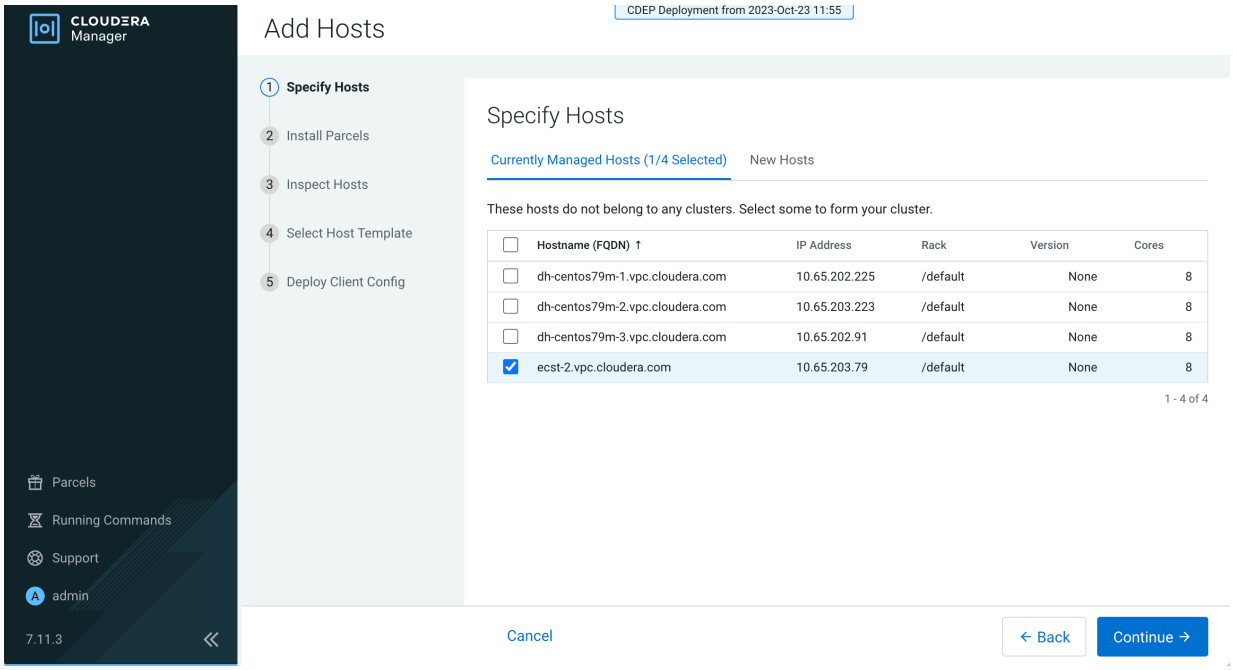
1. On the Cloudera Manager home page, click the ECS Cluster, then select Actions > Add Hosts.



2. On the Add Hosts page, click Add Hosts to Cluster and select the ECS Cluster, then click Continue.

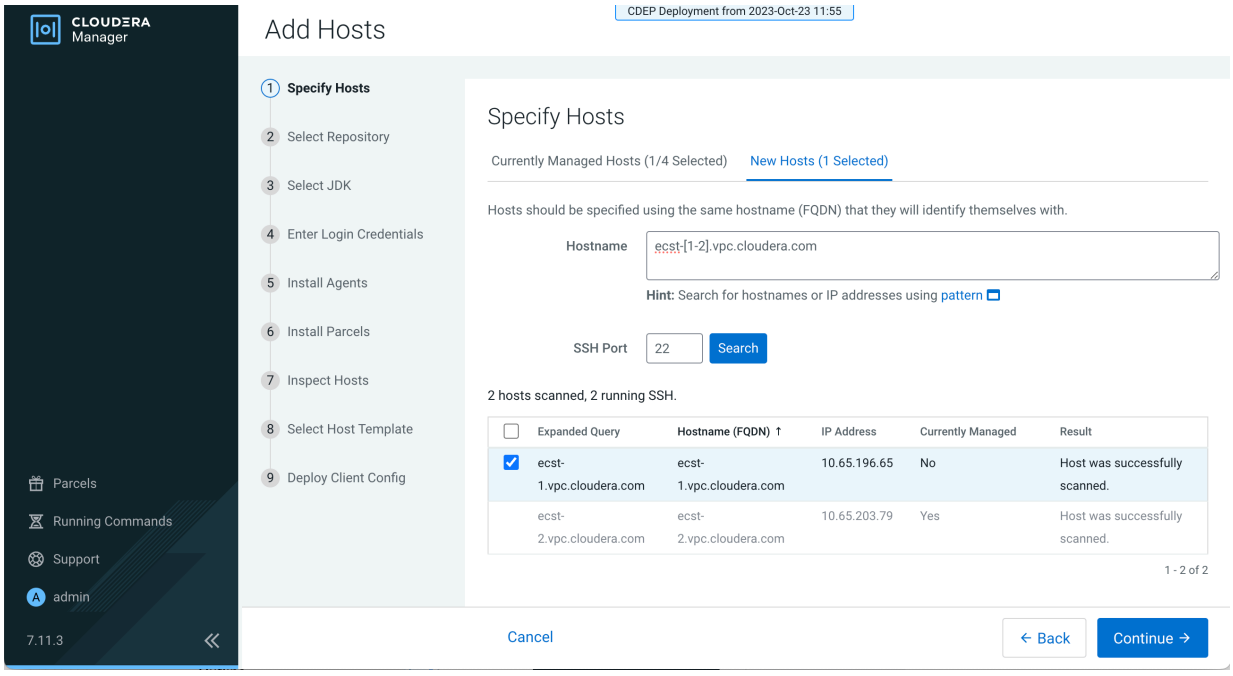


3. On the Specify Hosts page, hosts that have already been added to Cloudera Manager are listed on the Currently Managed Hosts tab. You can select one or more of these hosts to add to the ECS cluster.



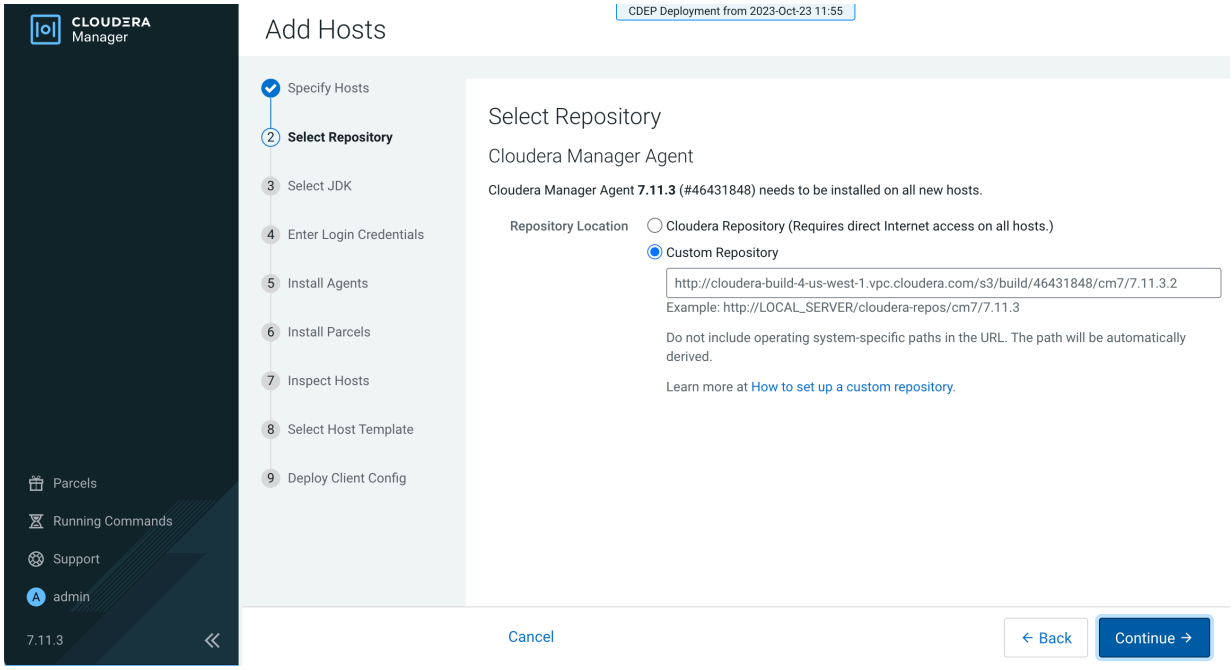
You can also click the New Hosts tab to specify one or more hosts that have not been added to Cloudera Manager. Enter a Fully Qualified Domain Name in the Hostname box, then click Search.

Note: Click the pattern link under the Hostname box to display more information about allowed FQDN patterns.

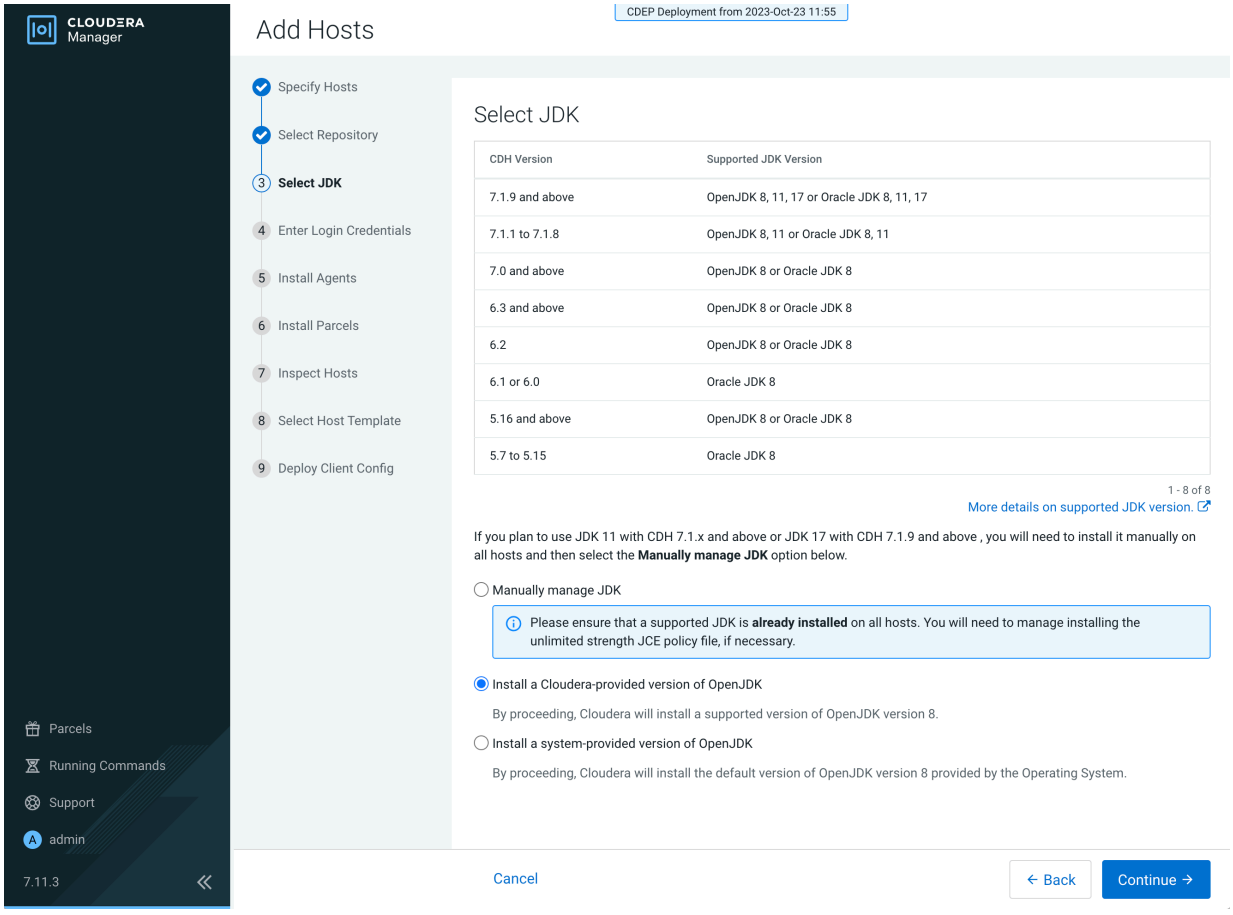


After you have finished specifying the ECS hosts, click Continue.

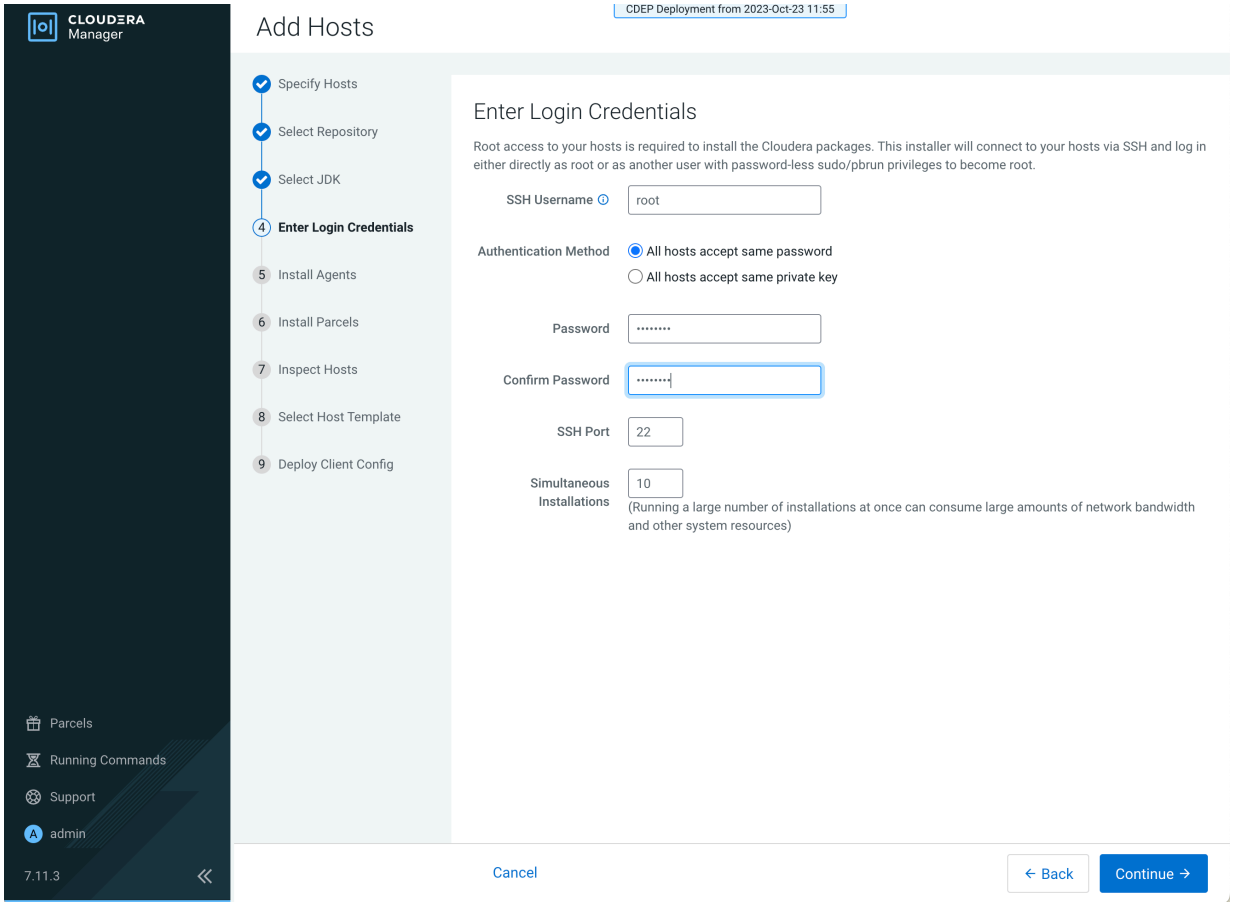
4. On the Select Repository page, the applicable Cloudera Manager Agent repository location is selected by default. Click Continue.



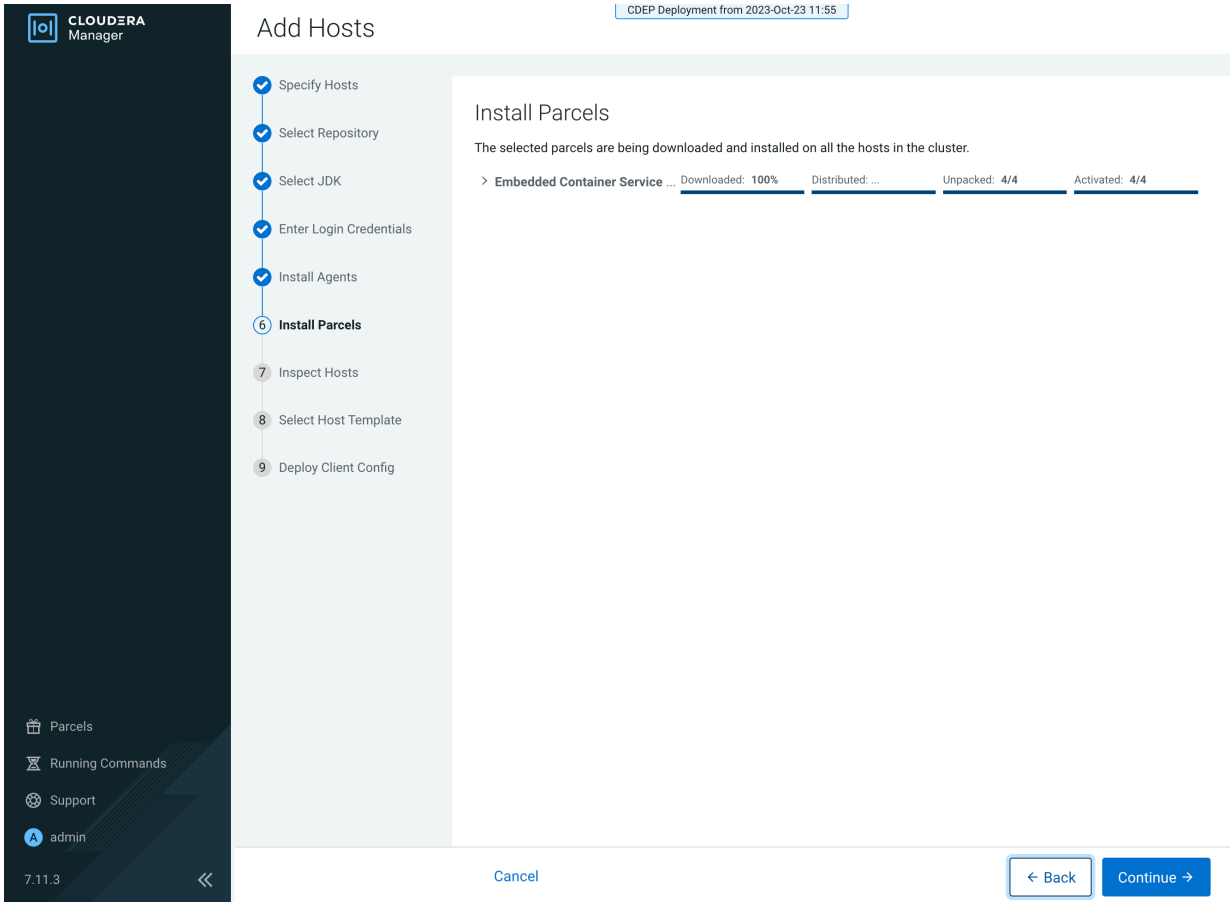
5. Select a JDK option on the Select JDK page, then click Continue.



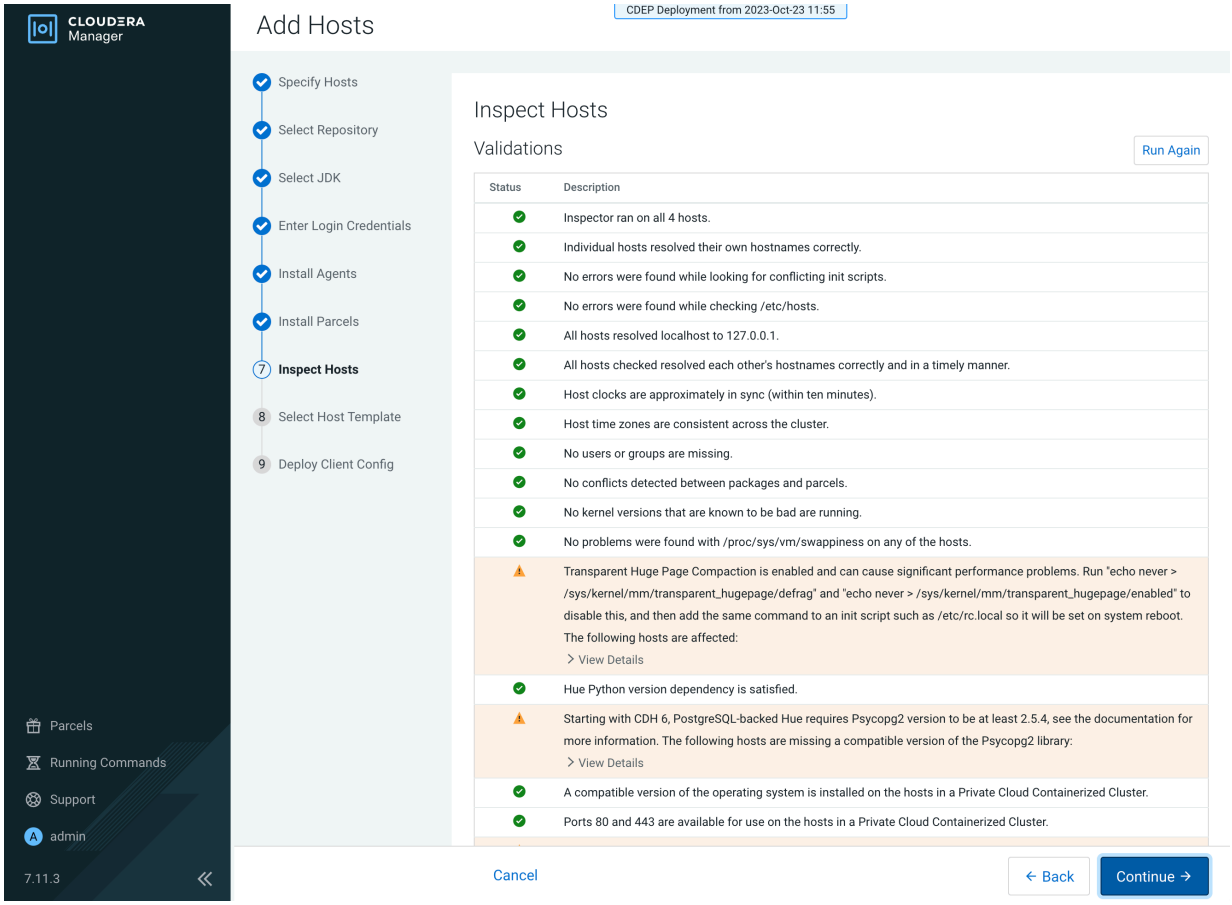
6. On the Enter Login Credentials page, All hosts accept the same password is selected by default. Enter the user name in the SSH Username box, and type in and confirm the password. You can also select the All hosts accept the same private key option and provide the Private Key and passphrase.



7. The Cloudera Manager agents are installed, and then the Install Parcels page appears. The selected parcel is downloaded to the Cloudera Manager server host, distributed, unpacked, and activated on the ECS cluster hosts. Click Continue.



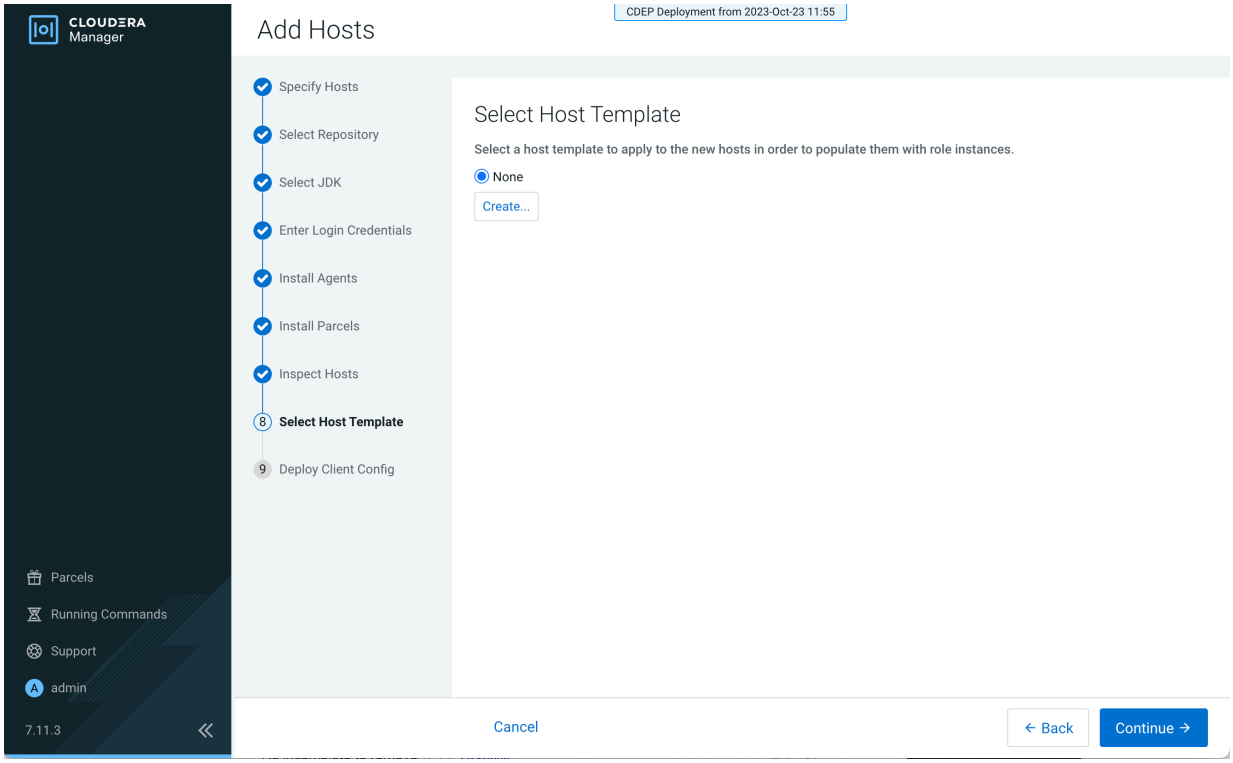
8. Review the Validations list on the Inspect Hosts page. If issues are detected, you can fix the issues, then click Run Again to repeat the host inspection. Click Continue.



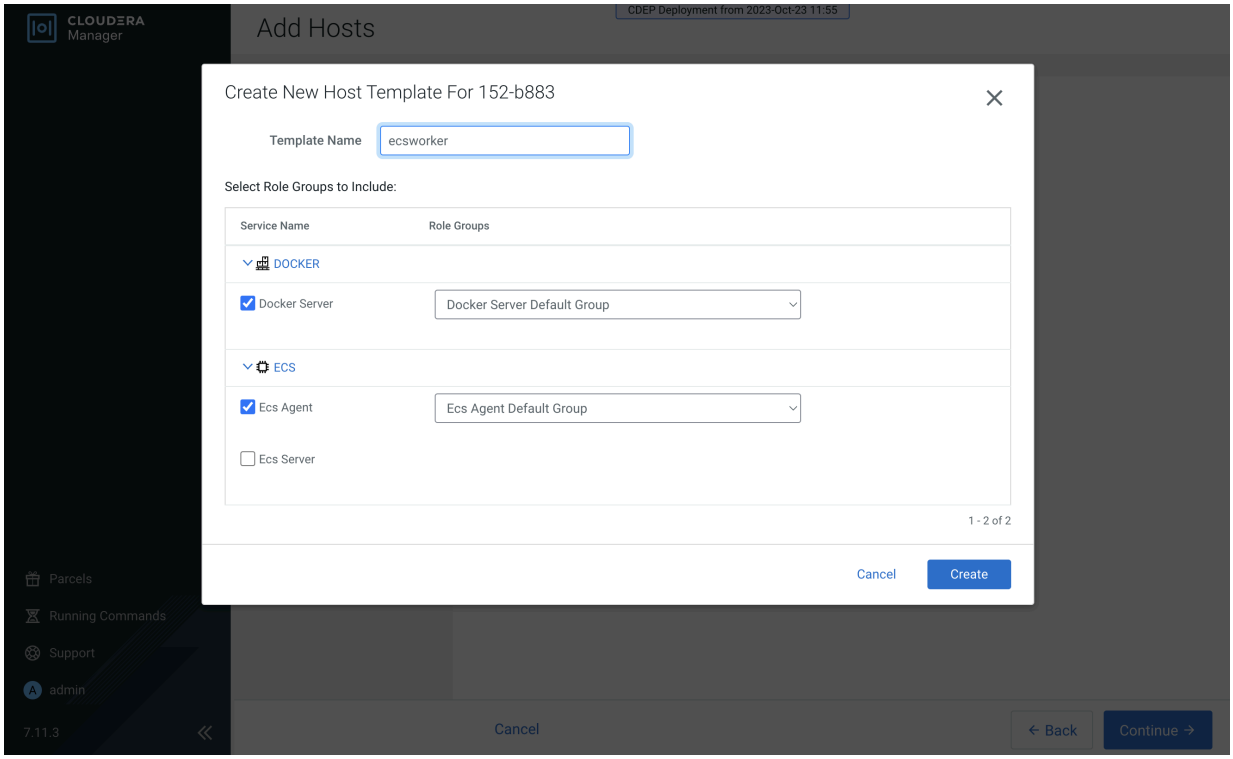
9. The Select Host Template page lists available host templates. Click Create.

Note:

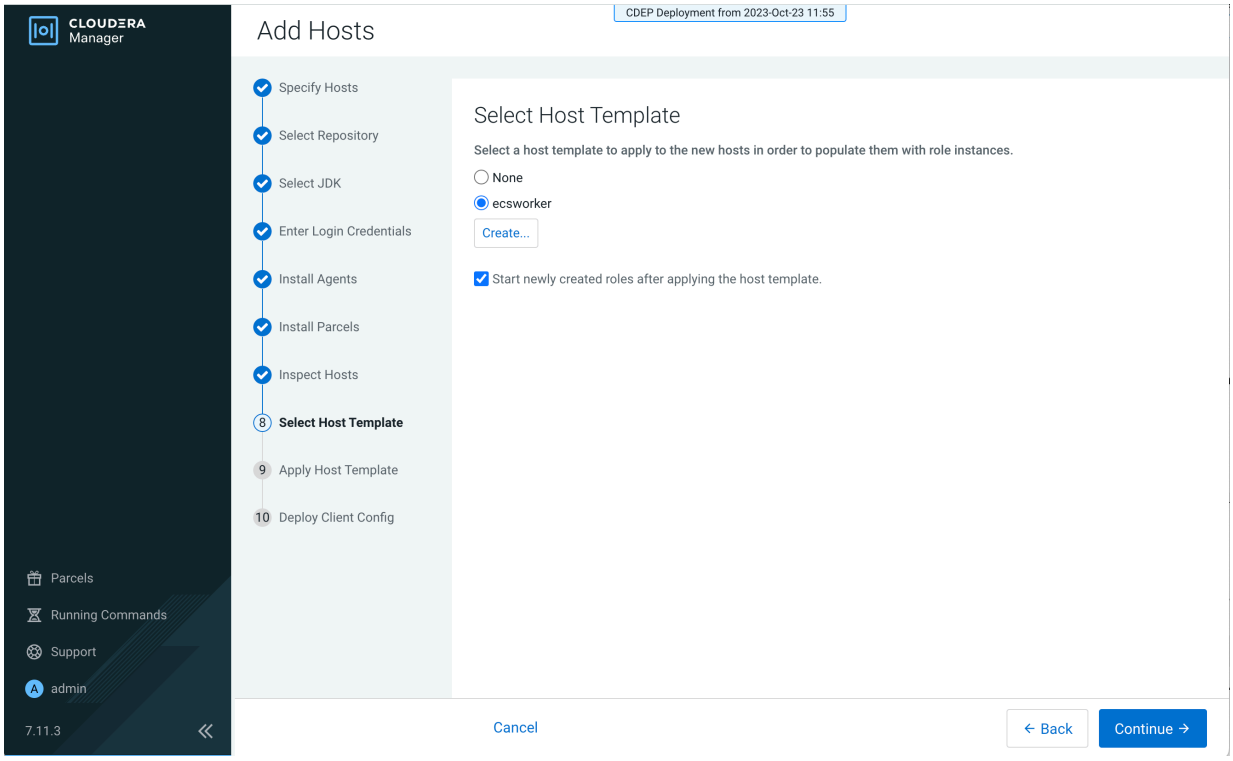
The following three steps describe how to create a host template to assign the Docker Server and Ecs Agent role groups to the new host. You can also select None and add these role instances after adding the new host to the cluster, as described at the end of this topic.



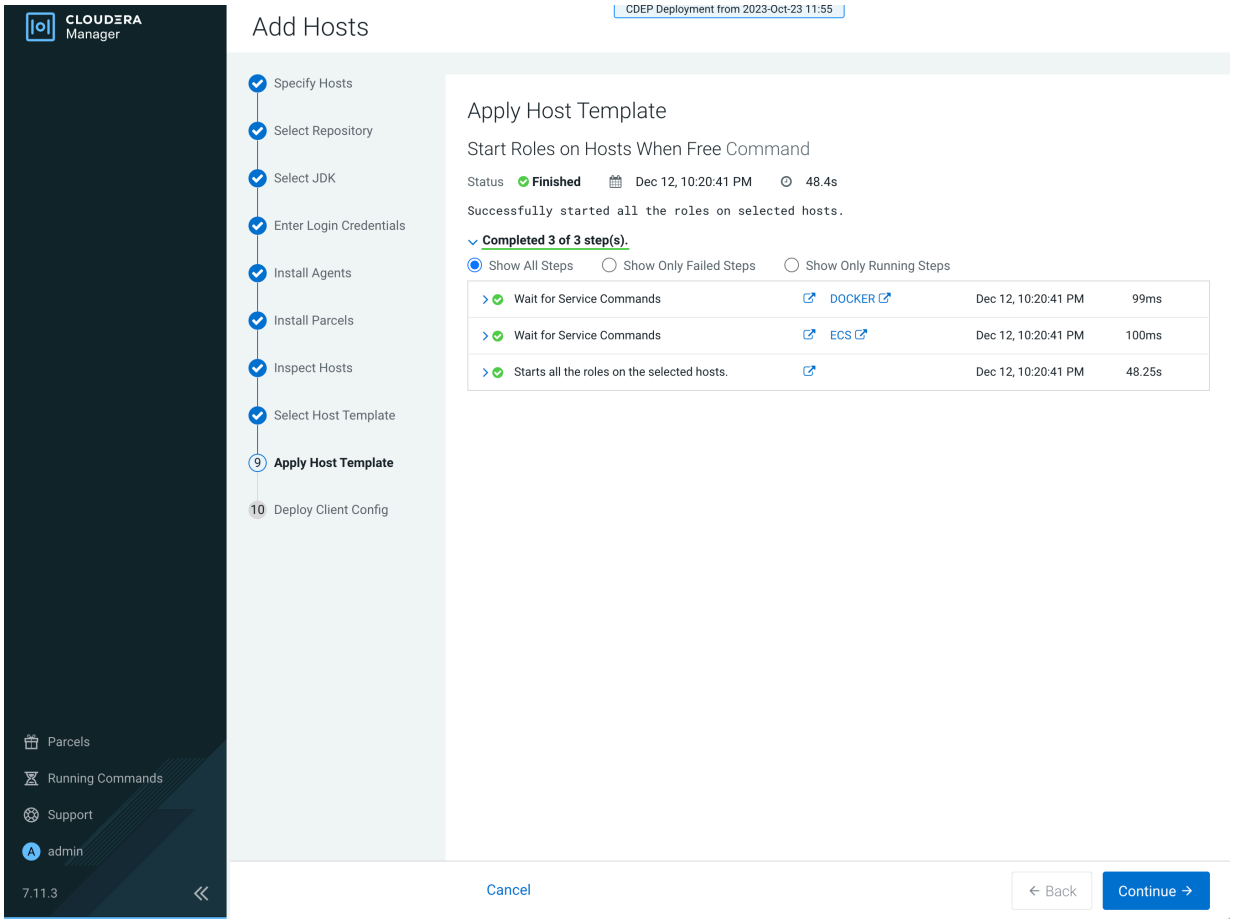
10. On the Create New Host Template pop-up, enter a template name and select the Docker Server and Ecs Agent role groups, then click Create.



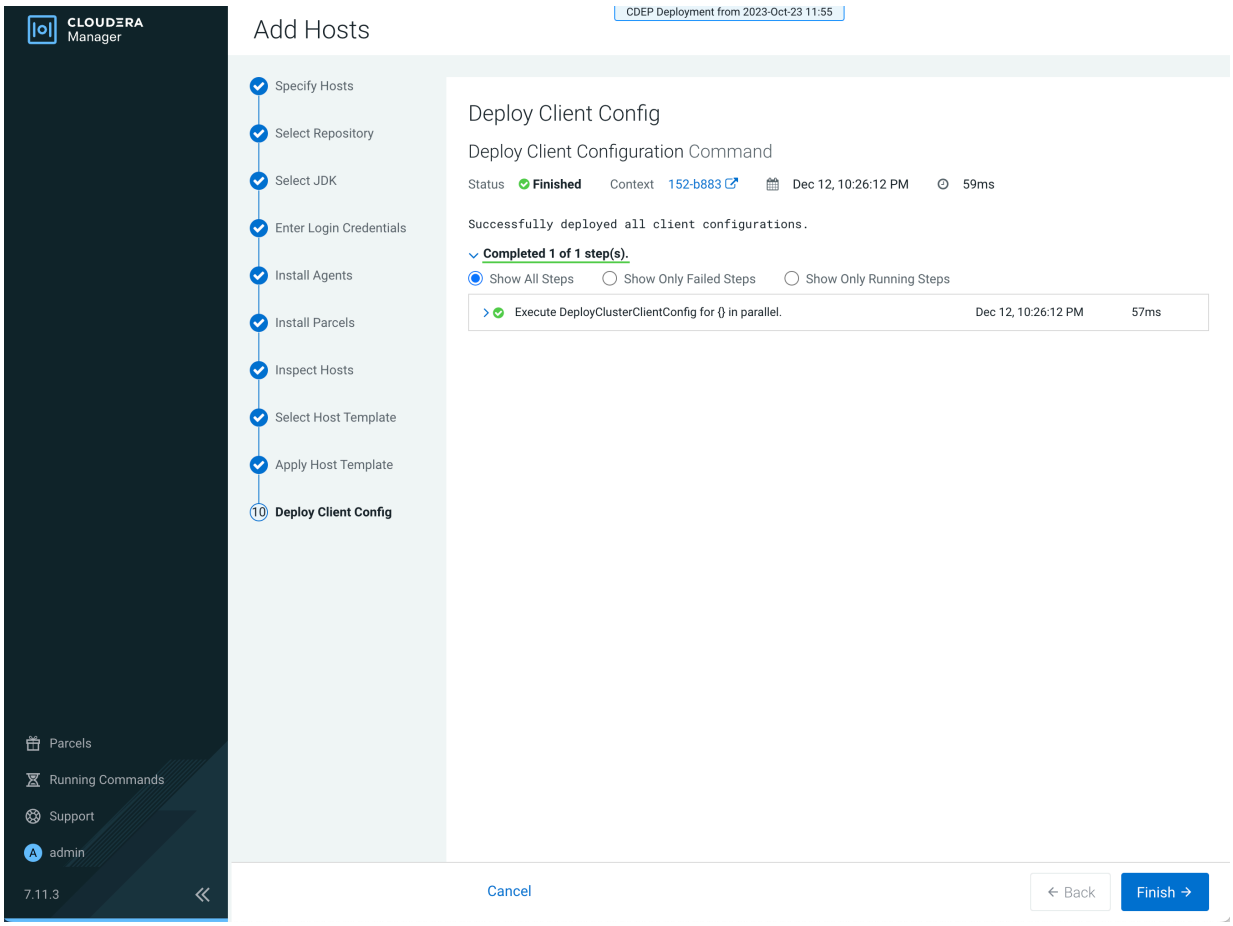
11. On the Select Host Template page, select the new template, then click Continue.



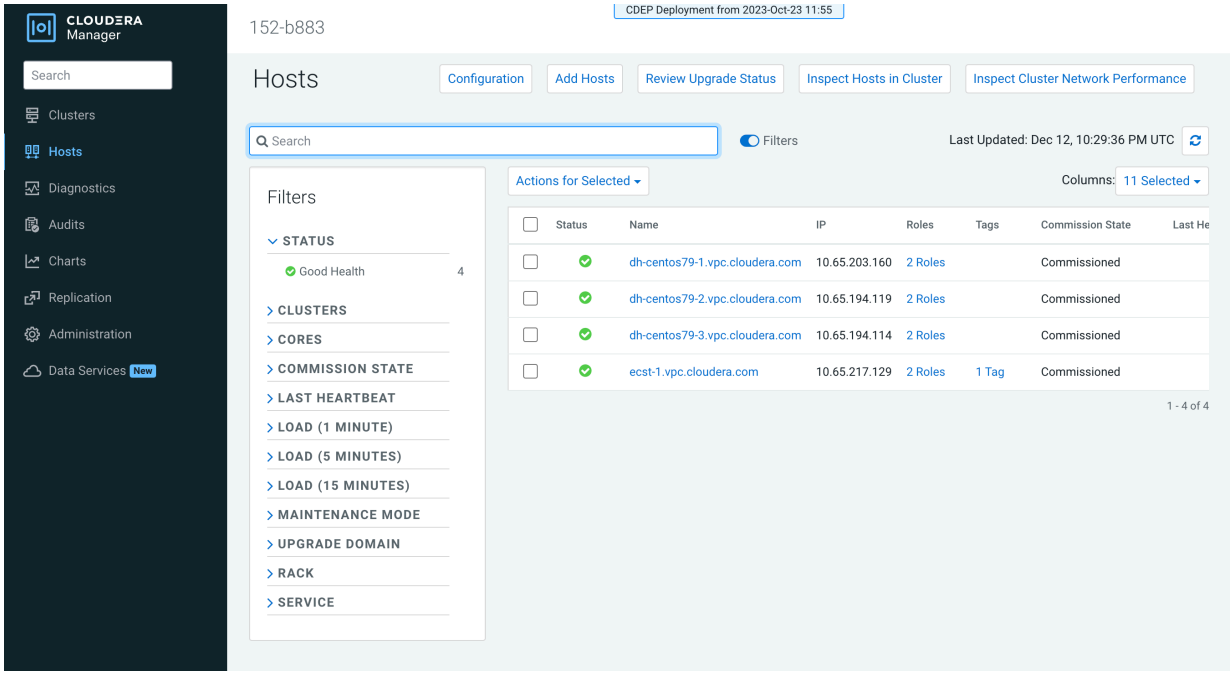
12. The Apply Host Template page appears. After the roles have successfully started, click Continue.



13. The Deploy Client Config page appears. After all client configurations have been successfully deployed, click Finish.



14. The new host is listed on the ECS cluster Hosts page.



15. If your ECS hosts are running the CentOS 8.4, OEL 8.4, RHEL 7.9, or RHEL 8 operating systems, you must install iptables on all the ECS hosts.

    For CentOS 8.4, OEL 8.4, or RHEL 8, run the following command on each ECS host:

    ```
    yum --setopt=tsflags=noscripts install -y iptables
    ```

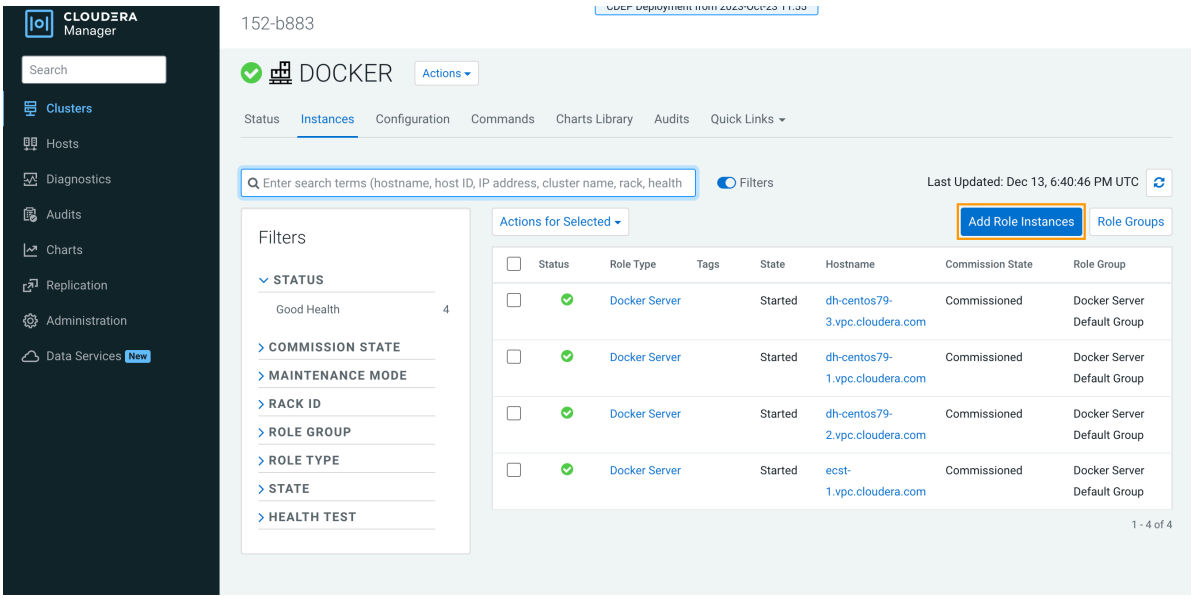
    For RHEL 7.9, run the following command on each ECS host:

    ```
    yum install -y iptables
    ```

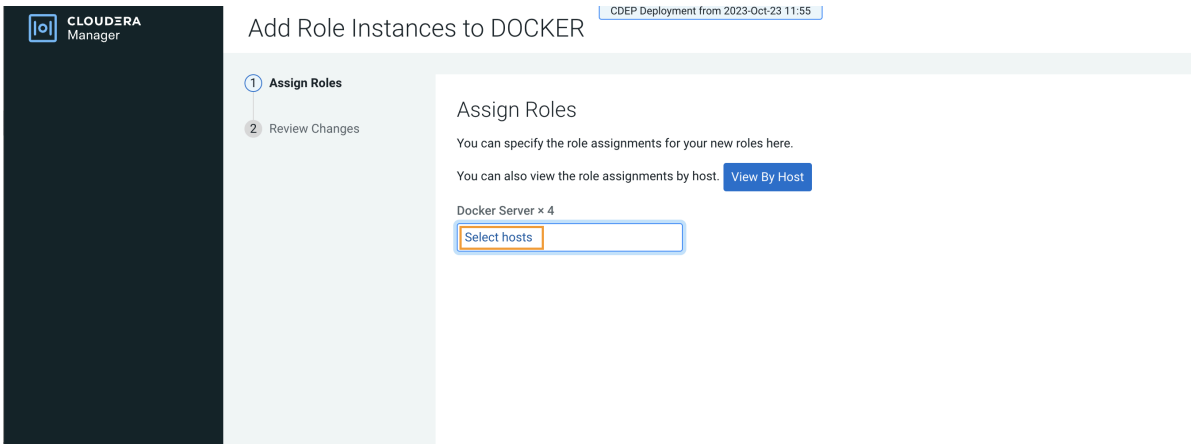
16. If you did not apply a host template to assign roles, perform the following steps to assign the Docker Server and Ecs Agent role groups to the new host.

To assign the Docker Server role group:

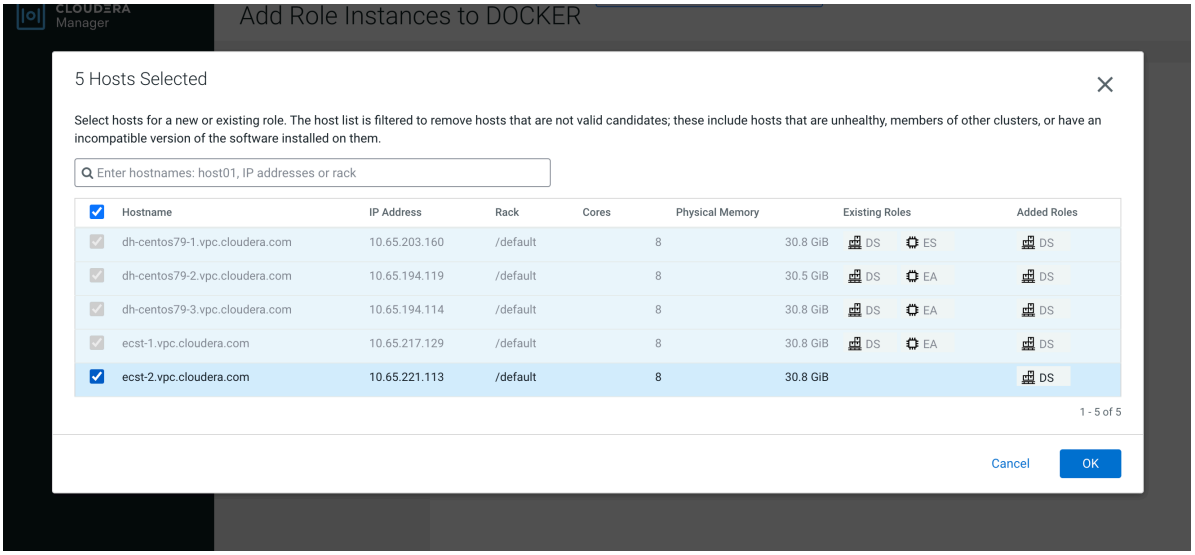
a. Click DOCKER on the ECS cluster home page, select Instances, then click Add Role Instances.



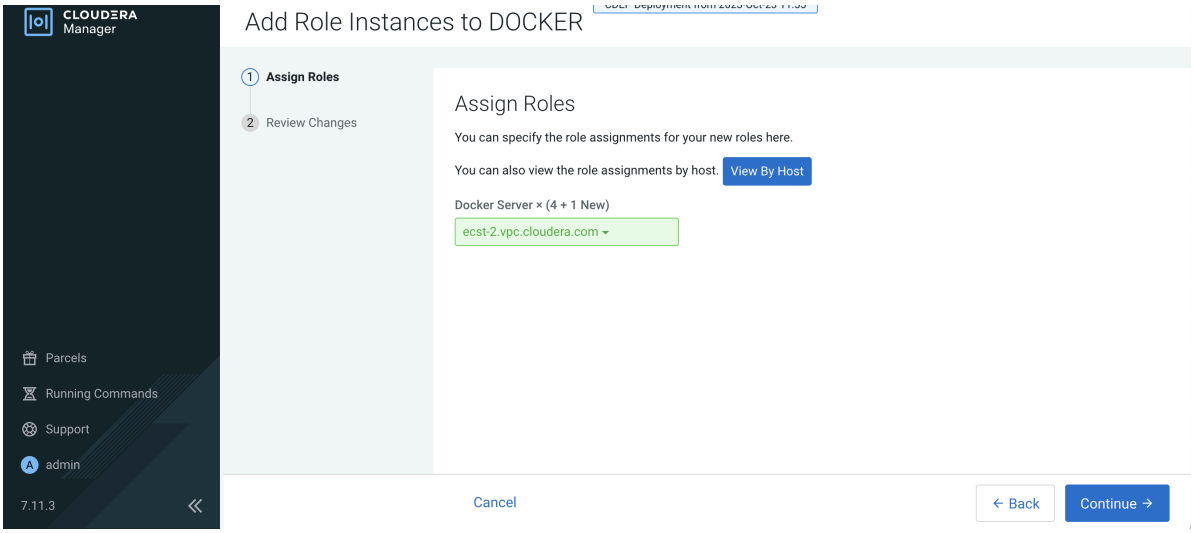
b. On the Add Role Instances to DOCKER page, click Select hosts.



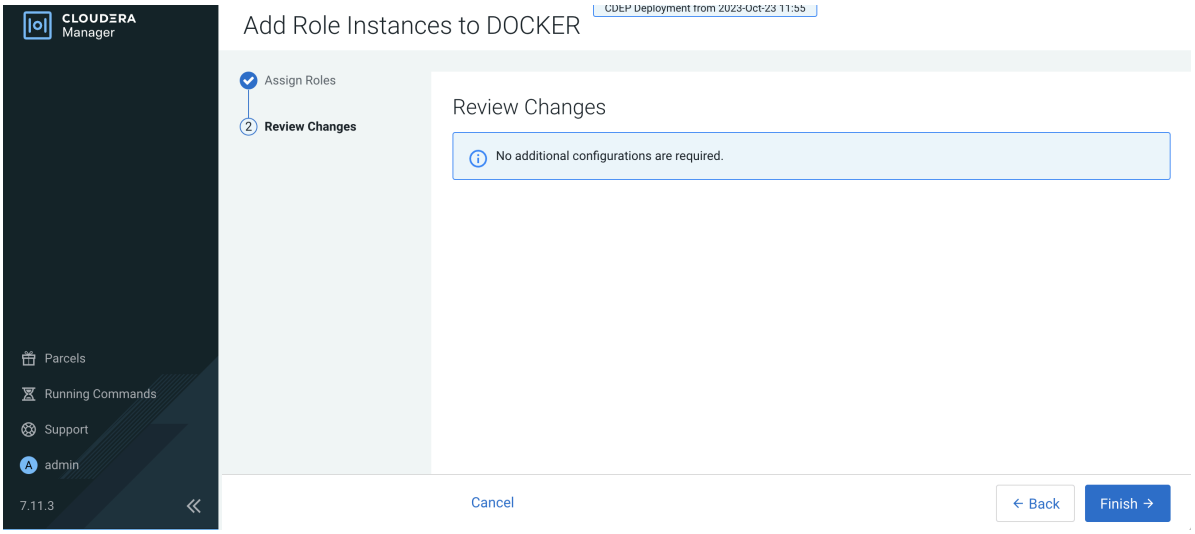
c. On the Hosts Selected pop-up, select the new host, then click OK.



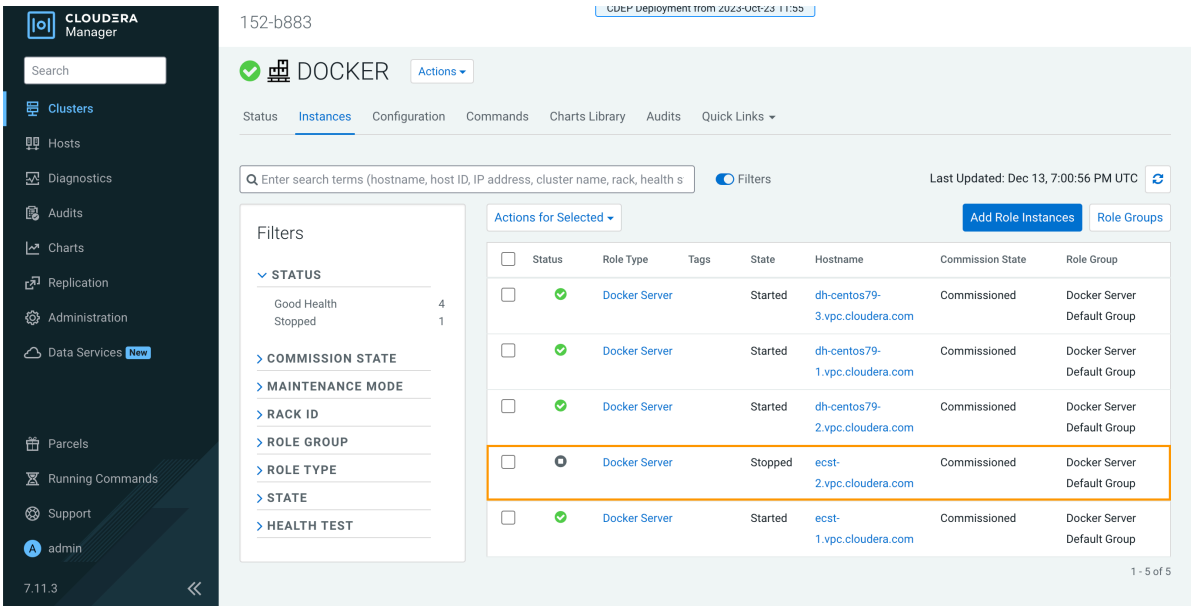
d. On the Assign Roles page, click Continue.



e. On the Review Changes page, click Finish.

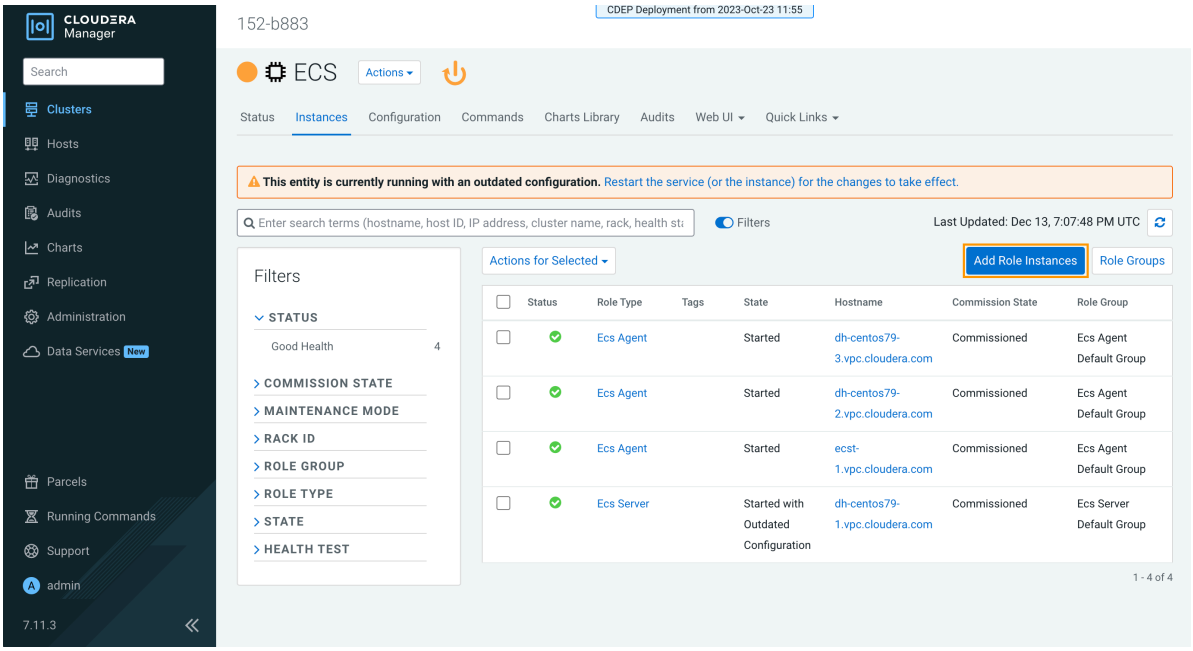


f. The new host is listed on the Docker Instances page.



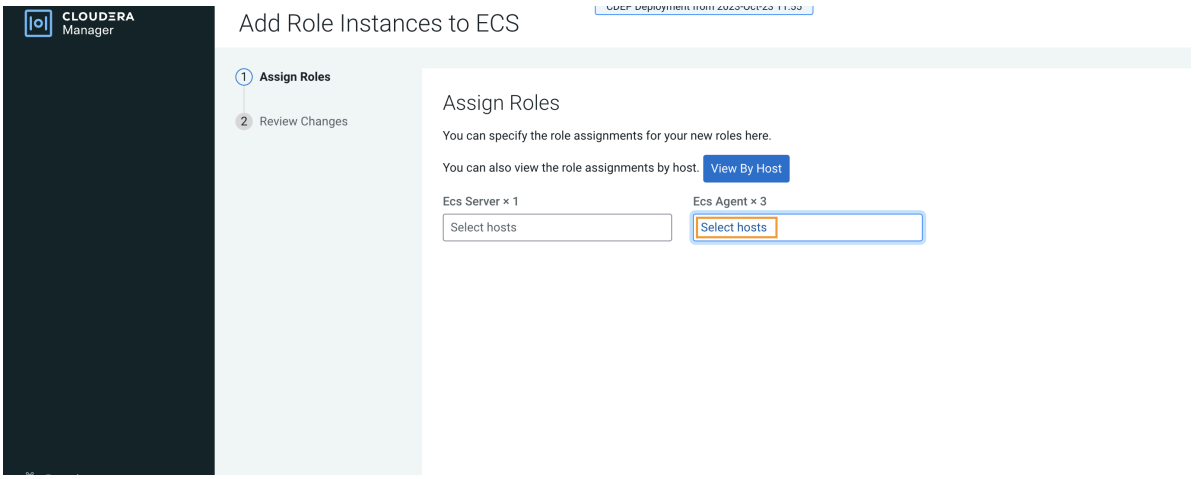
To assign the ECS Agent role group:

a. Click ECS on the ECS cluster home page, select Instances, then click Add Role Instances.

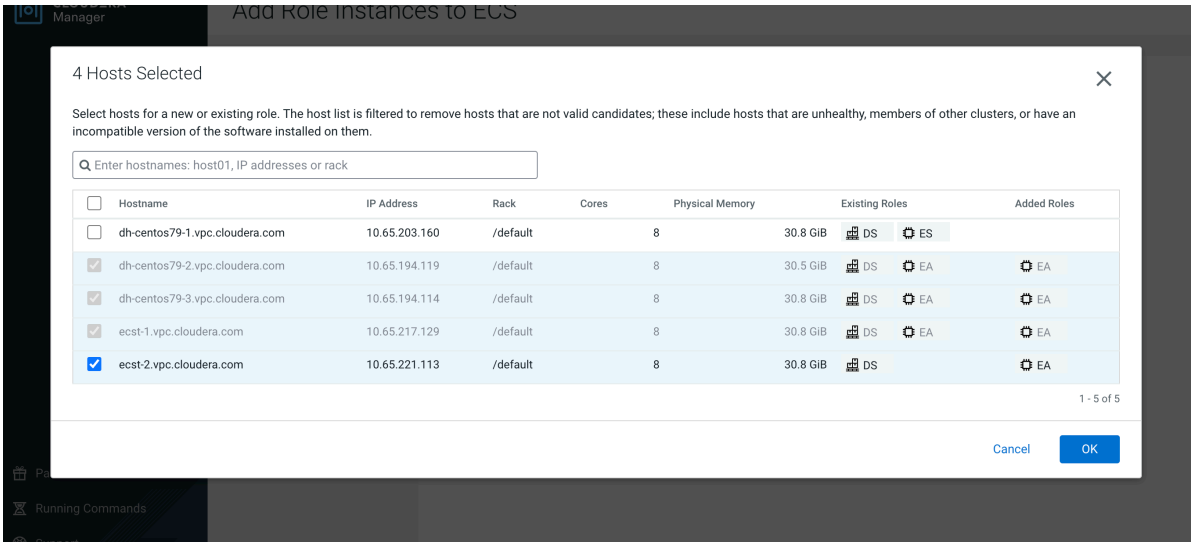


b. On the Add Role Instances to ECS page, in the Ecs Agent box, click Select hosts.

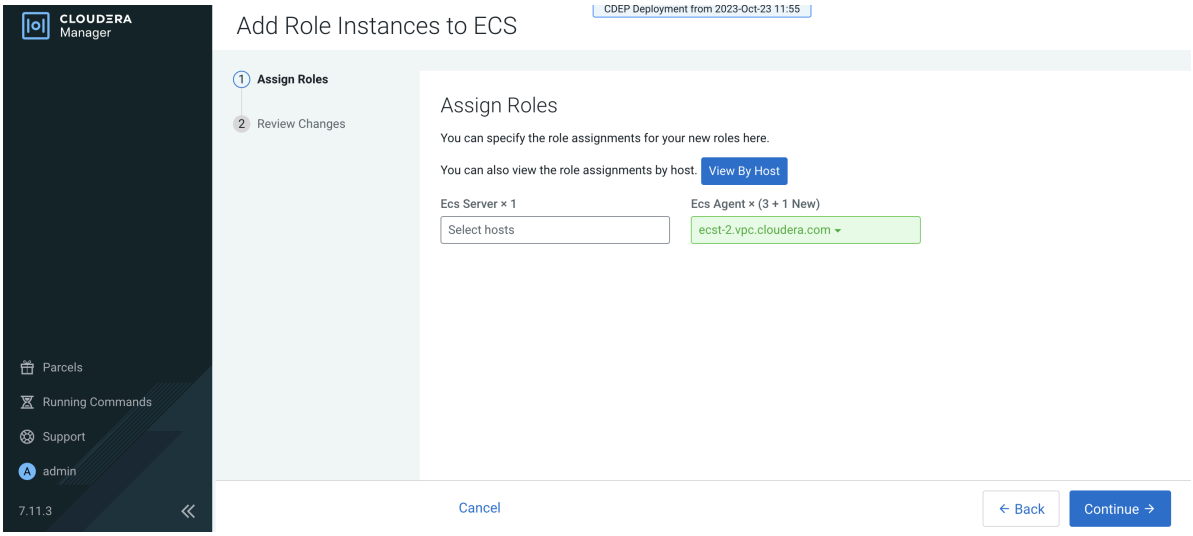
Important: Be sure to click Select hosts in the Ecs Agent box – do not click the link in the Ecs Server box.



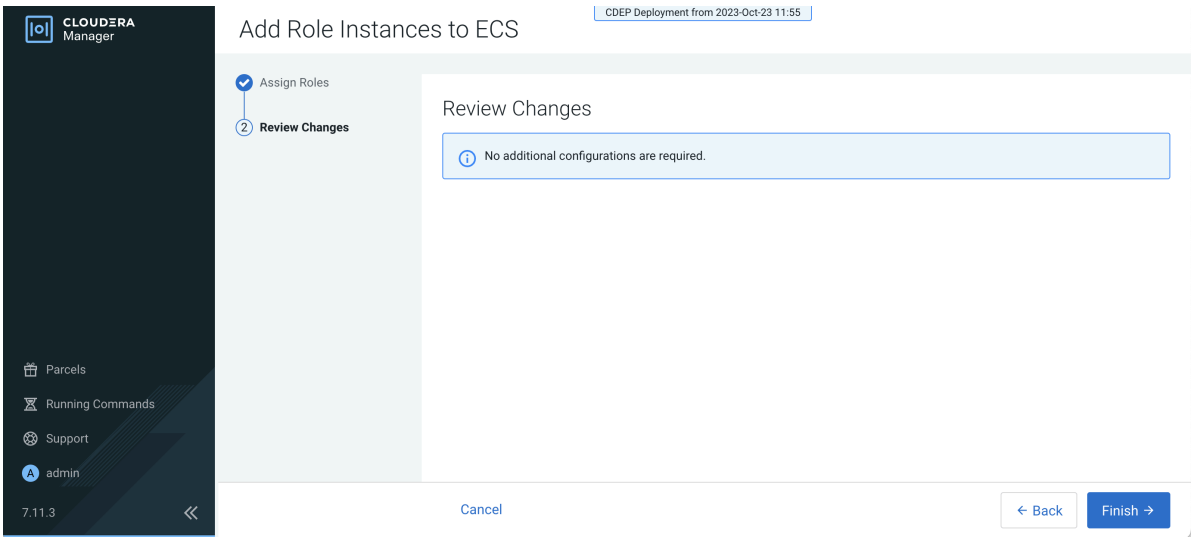
c. On the Hosts Selected pop-up, select the new host, then click OK.



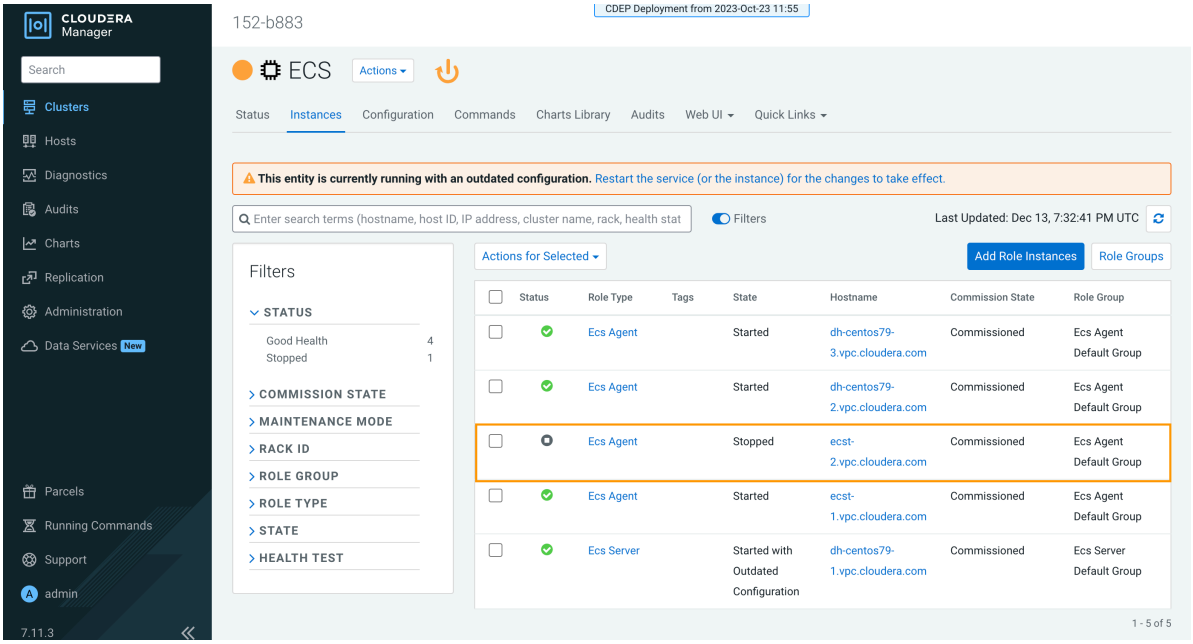
d. On the Assign Roles page, click Continue.



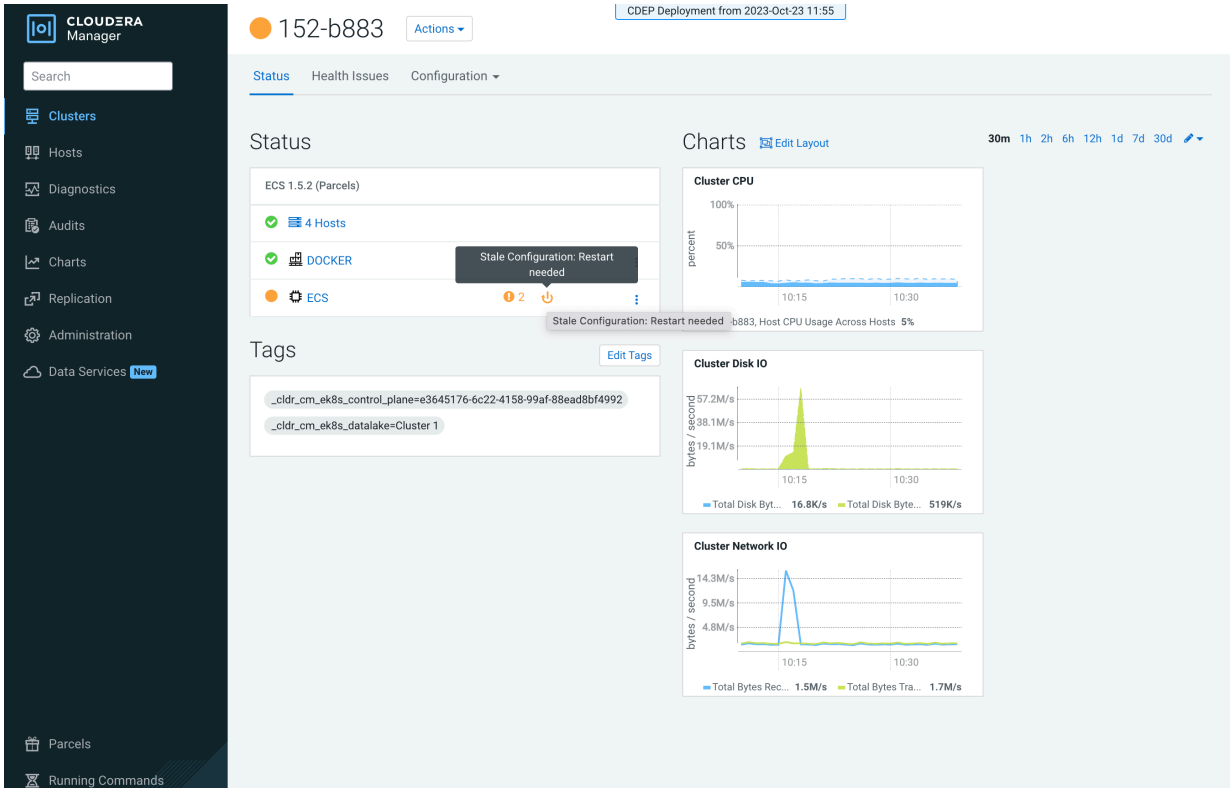
e. On the Review Changes page, click Finish.



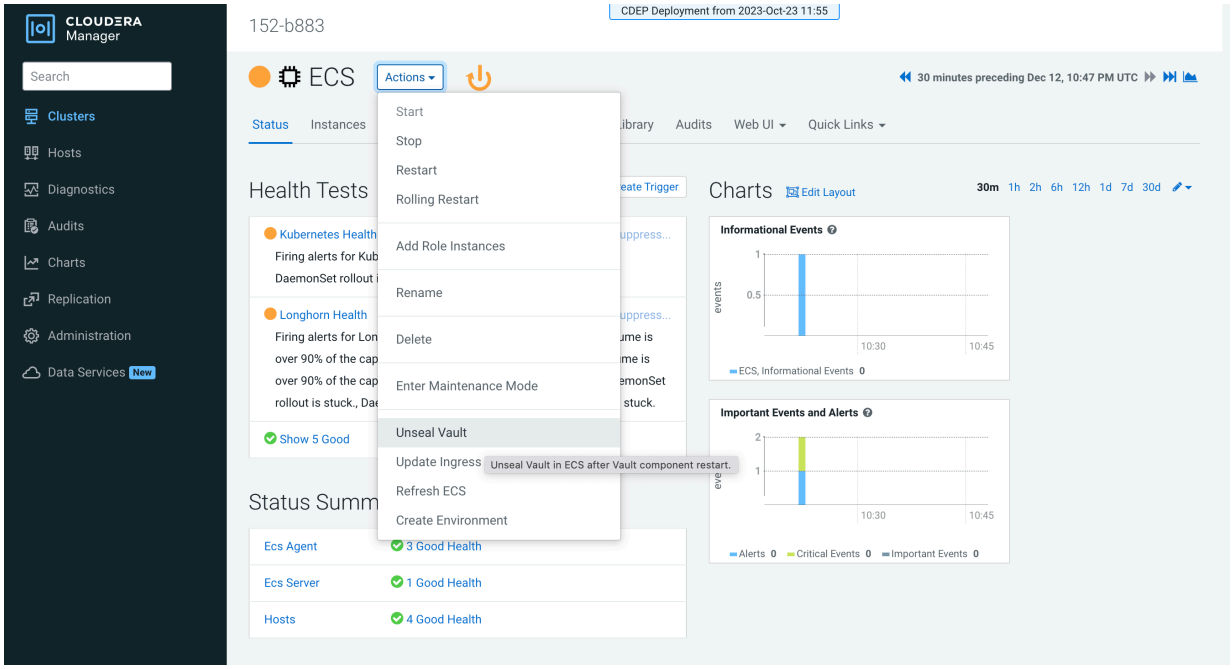
f. The new host is listed on the ECS Instances page.



17. Restart the ECS cluster by clicking the ECS Restart icon, or by selecting Actions > Restart on the ECS cluster home page.



18. Click ECS on the ECS cluster home page, then select Actions > Unseal Vault.

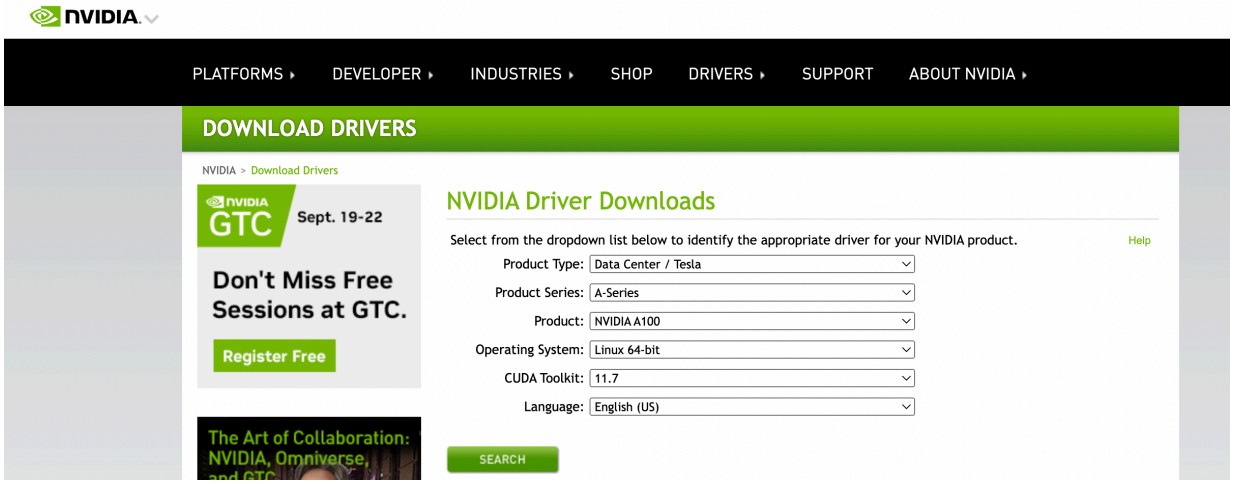


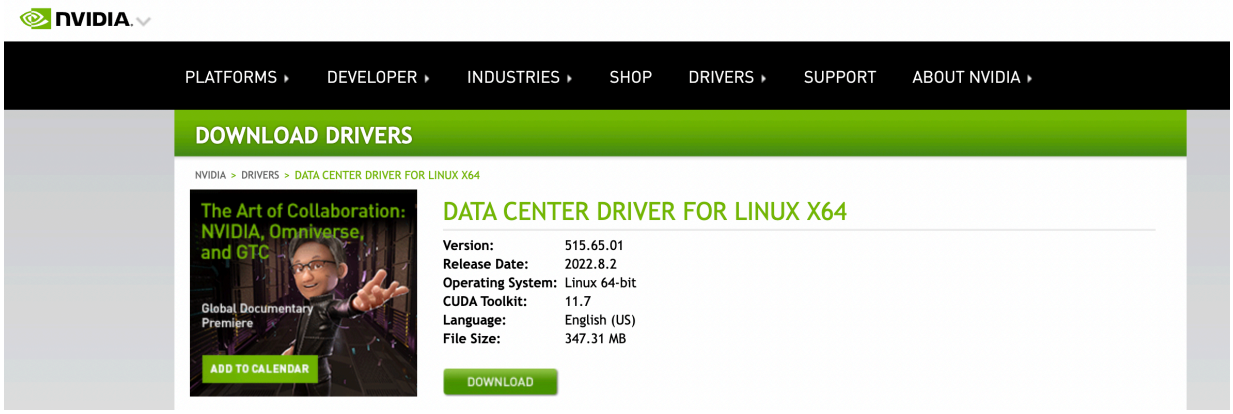
Installing NVIDIA GPU software in ECS

After you add a host containing a NVIDIA GPU card in an Embedded Container Service (ECS) cluster, you must install the NVIDIA GPU software driver and its associated software. You can then test the GPU card in the Cloudera Machine Learning (CML) workspace.

Installing the NVIDIA driver and container runtime

1. Use the NVIDIA Driver Downloads page to determine the software driver version required for your NVIDIA GPU card. This example uses a NVIDIA A100 GPU card, which requires driver version 515.65.01.





2. Run the following command to cordon the GPU worker node:

```
# kubectl cordon ecsgpu.cdpkvm.cldr node/ecsgpu.cdpkvm.cldr cordoned
```

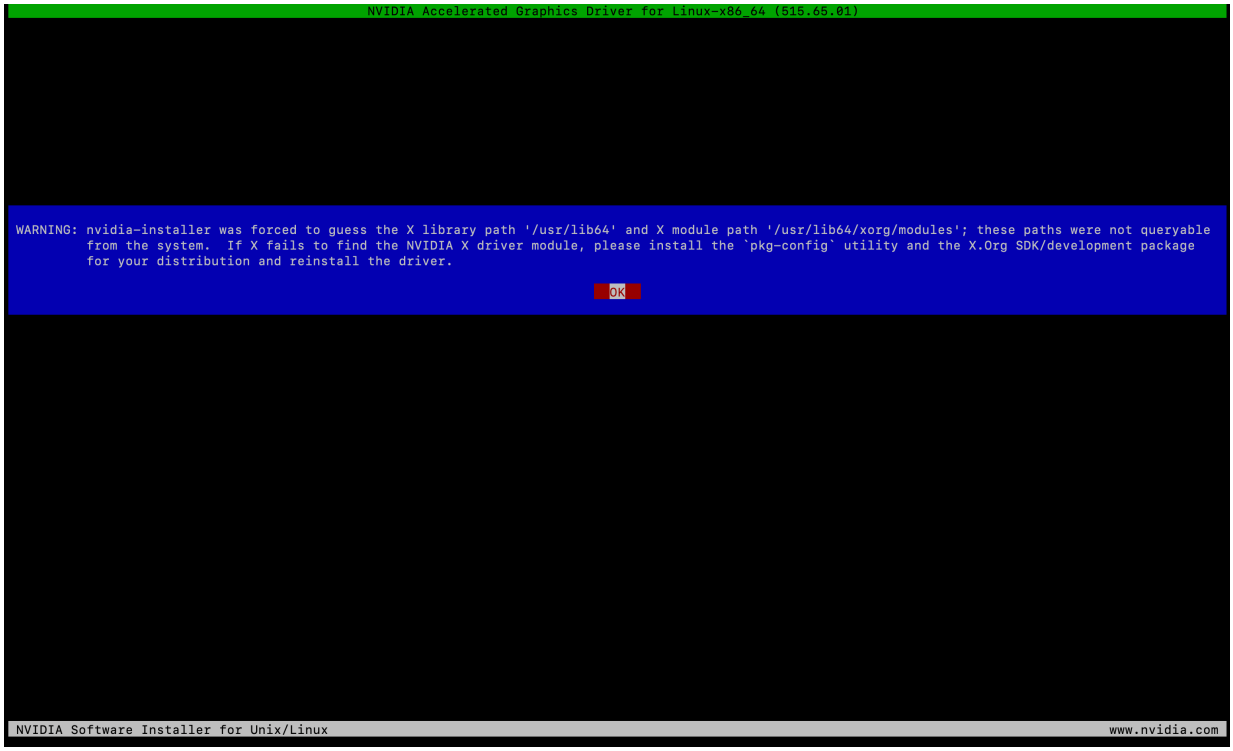
3. On the ECS host with the NVIDIA GPU card, install the required Operating System (OS) software packages as shown below, and then reboot the node. In this example, the host OS is Centos 7.9, and the host name of the node with the GPU card is ecsgpu.cdpkvm.cldr.

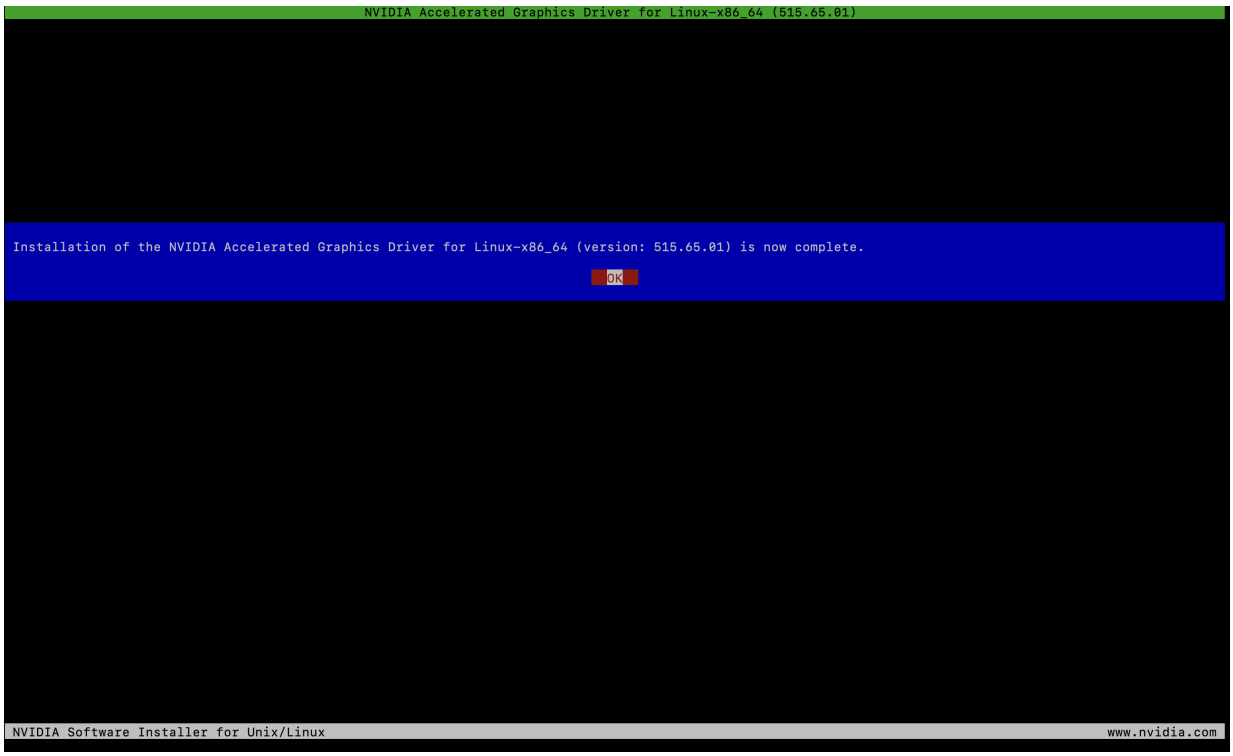
```
# yum update -y# yum install -y tar bzip2 make automake gcc gcc-c++ pciu
tils elfutils-libelf-devel libglvnd-devel vim bind-utils wget
# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest
-7.noarch.rpm
# yum -y group install "Development Tools"# yum install -y kernel-devel-$(
uname -r) kernel-headers-$(uname -r)
# reboot
```

4. Next, run the following commands to install the NVIDIA driver and nvidia-container-runtime software:

```
# BASE_URL=https://us.download.nvidia.com/tesla# DRIVER_VERSION=515.65.01
# curl -fSsl -O $BASE_URL/$DRIVER_VERSION/NVIDIA-Linux-x86_64-$DRIVER_VERS
ION.run
```

```
# sh NVIDIA-Linux-x86_64-$DRIVER_VERSION.run
```





5. After the installation is complete, run the nvidia-smi tool and ensure that the driver was successfully deployed. The output should look similar to the following example:

```
[root@ecsgpu ~]# nvidia-smi
 Wed Aug 24 13:03:46 2022
 +-----------------------------------------------------------------------------
--------+
 | NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7
     |
 |-------------------------------+----------------------+----------------
------+
 | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
 ECC |
 | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Com
pute M. |
 |                               |                      |               MI
G M. |
 |===============================+======================+================
======|
 |   0  NVIDIA A100-PCI...  Off  | 00000000:08:00.0 Off |
   0 |
 | N/A   32C    P0    37W / 250W |      0MiB / 40960MiB |      3%
Default |
 |                               |                      |             Disa
bled |
 +-------------------------------+----------------------+----------------
------+


 +-----------------------------------------------------------------------------
-------+
 | Processes:
      |
 |  GPU   GI   CI        PID   Type   Process name                  GPU Me
mory |
```

```
 |        ID   ID                                                    Usage
     |
 |=============================================================================
=======|
 |  No running processes found
      |
 +-----------------------------------------------------------------------------
-----+

 [root@ecsgpu ~]# lsmod | grep nvidia
 nvidia_drm             53212  0
 nvidia_modeset       1142094  1 nvidia_drm
 nvidia              40761292  1 nvidia_modeset
 drm_kms_helper        186531  3 qxl,nouveau,nvidia_drm
 drm                   468454  7 qxl,ttm,drm_kms_helper,nvidia,nouveau,n
vidia_drm

 [root@ecsgpu ~]# dmesg | grep nvidia
 [  123.588172] nvidia: loading out-of-tree module taints kernel.
 [  123.588182] nvidia: module license 'NVIDIA' taints kernel.
 [  123.704411] nvidia: module verification failed: signature and/or requi
red key missing - tainting kernel
 [  123.802826] nvidia-nvlink: Nvlink Core is being initialized, major dev
ice number 239
 [  123.925577] nvidia-uvm: Loaded the UVM driver, major device number 23
7.
 [  123.934813] nvidia-modeset: Loading NVIDIA Kernel Mode Setting Driver
 for UNIX platforms  515.65.01  Wed Jul 20 13:43:59 UTC 2022
 [  123.940999] [drm] [nvidia-drm] [GPU ID 0x00000800] Loading driver
 [  123.941018] [drm] Initialized nvidia-drm 0.0.0 20160202 for 0000:08:0
0.0 on minor 1
 [  123.958317] [drm] [nvidia-drm] [GPU ID 0x00000800] Unloading driver
 [  123.968642] nvidia-modeset: Unloading
 [  123.978362] nvidia-uvm: Unloaded the UVM driver.
 [  123.993831] nvidia-nvlink: Unregistered Nvlink Core, major device numb
er 239
 [  137.450679] nvidia-nvlink: Nvlink Core is being initialized, major de
vice number 240
 [  137.503657] nvidia-modeset: Loading NVIDIA Kernel Mode Setting Driver
 for UNIX platforms  515.65.01  Wed Jul 20 13:43:59 UTC 2022
 [  137.508187] [drm] [nvidia-drm] [GPU ID 0x00000800] Loading driver
 [  137.508190] [drm] Initialized nvidia-drm 0.0.0 20160202 for 0000:08:
00.0 on minor 1
 [  149.717193] nvidia 0000:08:00.0: irq 48 for MSI/MSI-X
 [  149.717222] nvidia 0000:08:00.0: irq 49 for MSI/MSI-X
 [  149.717248] nvidia 0000:08:00.0: irq 50 for MSI/MSI-X
 [  149.717275] nvidia 0000:08:00.0: irq 51 for MSI/MSI-X
 [  149.717301] nvidia 0000:08:00.0: irq 52 for MSI/MSI-X
 [  149.717330] nvidia 0000:08:00.0: irq 53 for MSI/MSI-X
```

6. Install the nvidia-container-runtime software package, and then reboot the server:

```
# curl -s -L https://nvidia.github.io/nvidia-container-runtime/$(. /etc/
os-release;echo $ID$VERSION_ID)/nvidia-container-runtime.repo |  sudo te
e /etc/yum.repos.d/nvidia-container-runtime.repo# yum -y install nvidia-
container-runtime# rpm -qa | grep nvidia
 libnvidia-container-tools-1.11.0-1.x86_64
 libnvidia-container1-1.11.0-1.x86_64
 nvidia-container-toolkit-base-1.11.0-1.x86_64
 nvidia-container-runtime-3.11.0-1.noarch
 nvidia-container-toolkit-1.11.0-1.x86_64

 # nvidia-container-toolkit -version
 NVIDIA Container Runtime Hook version 1.11.0
```

```
 commit: d9de4a0

 # reboot
```

7. Uncordon the GPU worker node:

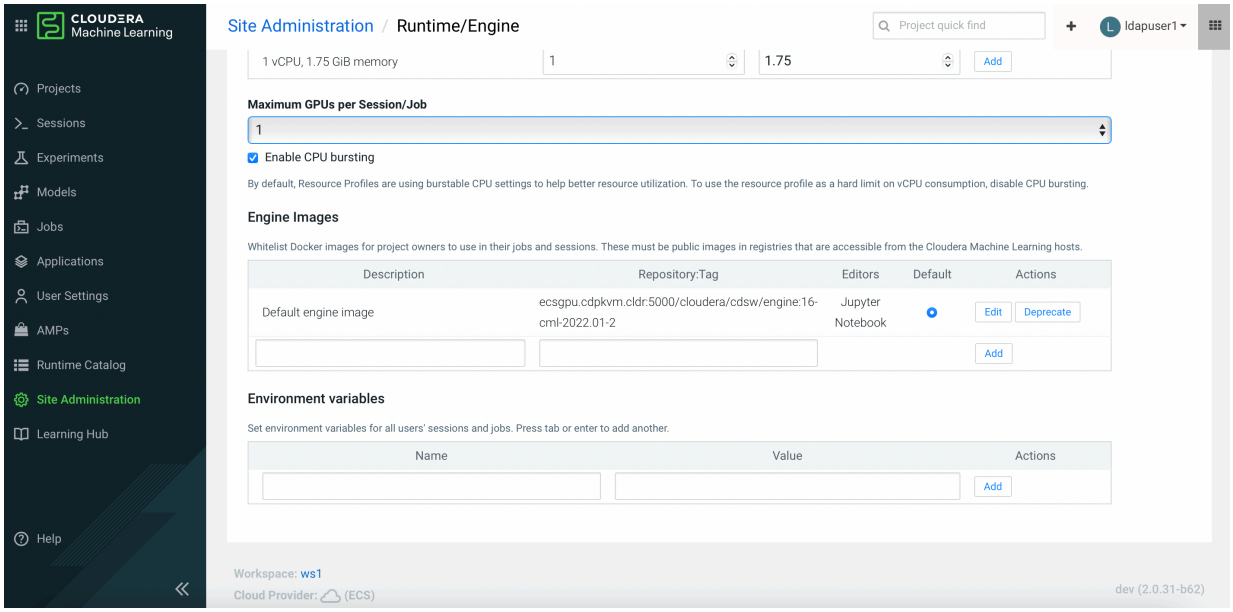
```
# kubectl uncordon ecsgpu.cdpkvm.cldr node/ecsgpu.cdpkvm.cldr cordoned
```

Testing the NVIDIA GPU card in CML

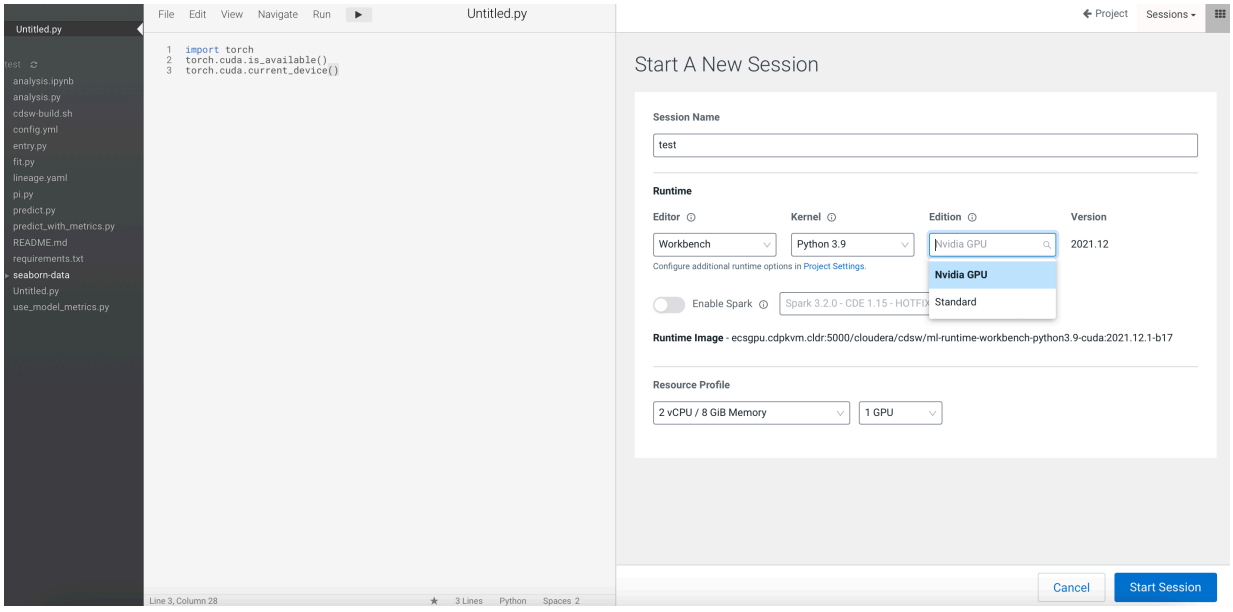
1. SSH into the ECS master node in the CDP Private Cloud Data Services cluster and run the following command to ensure that the ecsgpu.cdpkvm.cldr host has the nvidia.com/gpu: field in the node specification. Host ecsgpu.cdpkvm.cldr is a typical ECS worker node with the NVIDIA GPU card installed.

```
[root@ecsmaster1 ~]# kubectl describe node ecsgpu.cdpkvm.cldr | grep-A15
 Capacity:
 Capacity:
     cpu:                16
     ephemeral-storage:  209703916Ki
     hugepages-1Gi:      0
     hugepages-2Mi:      0
     memory:             263975200Ki
     nvidia.com/gpu:     1
     pods:               110
 Allocatable:
     cpu:                16
     ephemeral-storage:  203999969325
     hugepages-1Gi:      0
     hugepages-2Mi:      0
     memory:             263975200Ki
     nvidia.com/gpu:     1
     pods:               110

 [root@ecsmaster1 ~]# kubectl describe node ecsworker1.cdpkvm.cldr | grep-
A13 Capacity:
 Capacity:
     cpu:                16
     ephemeral-storage:  103797740Ki
     hugepages-1Gi:      0
     hugepages-2Mi:      0
     memory:             263974872Ki
     pods:               110
 Allocatable:
     cpu:                16
     ephemeral-storage:  100974441393
     hugepages-1Gi:      0
     hugepages-2Mi:      0
     memory:             263974872Ki
     pods:               110
```

2. In the CDP Private Cloud Data Services CML workspace, select Site Administration > Runtime/Engine. Specify a number for Maximum GPUs per Session/Job. This procedure effectively allows the CML session to consume the GPU card.

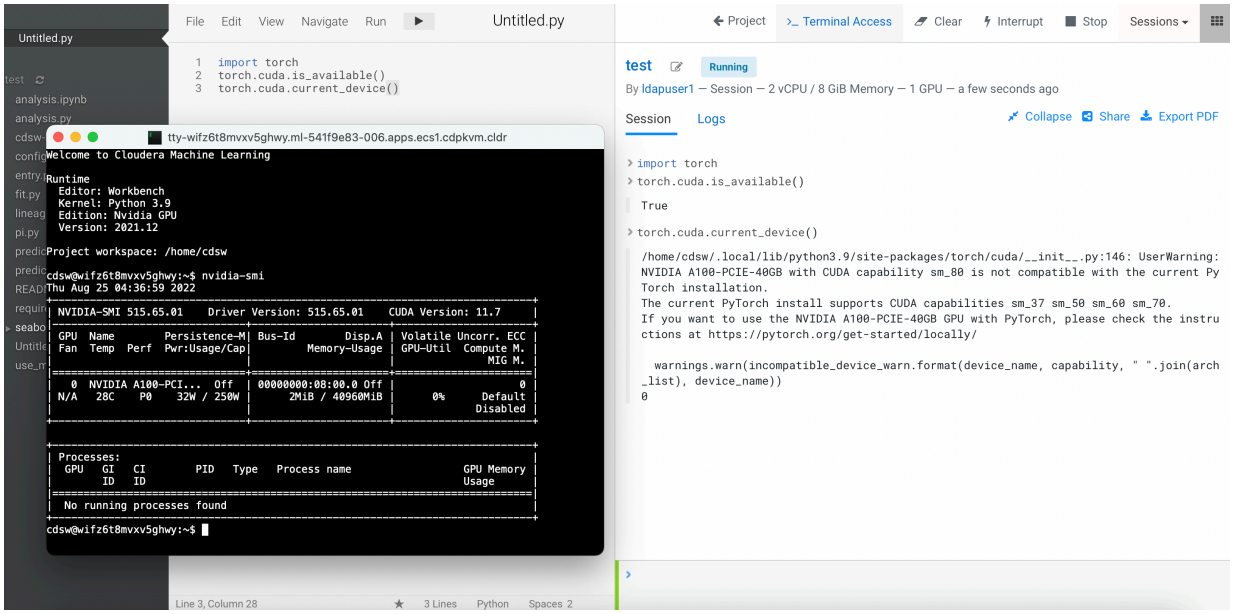


3. Create a CML project and start a new session by selecting the Workbench editor with a Python kernel and a NVIDIA GPU edition. Specify the number of GPUs to use – in this example, 1 GPU is specified.

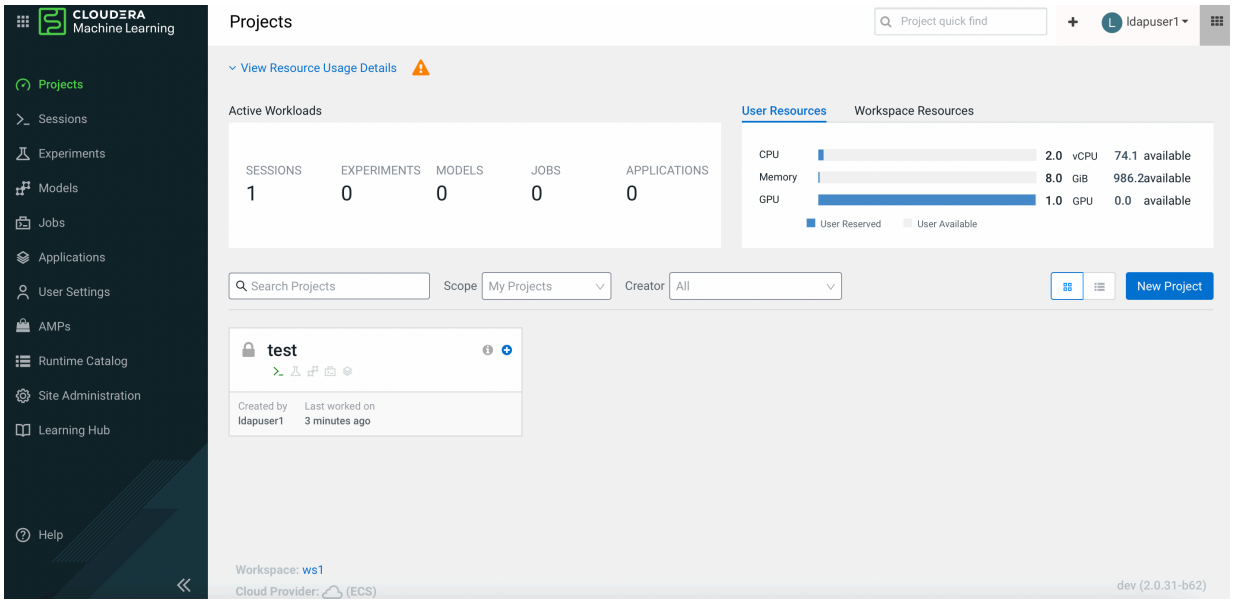


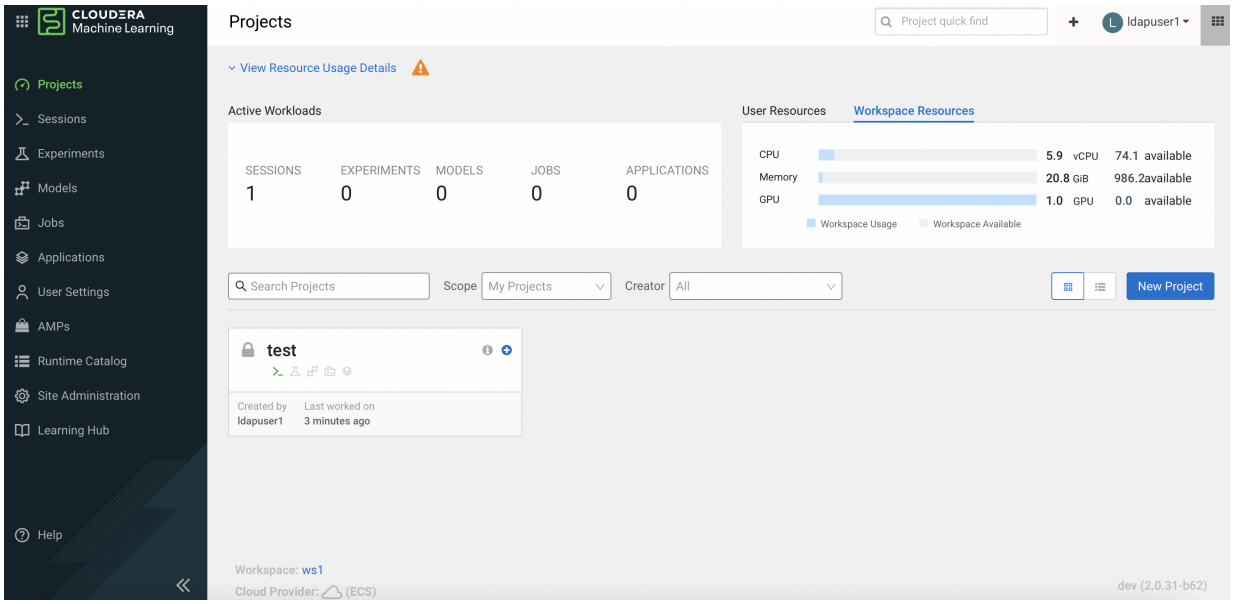
4. Create a new Python file and run the following script. Also, open the terminal session and run the nvidia-smi tool. Note that the output shows the NVIDIA GPU card details.

```
!pip3install torchimport torchtorch.cuda.is_available()torch.cuda.device
_count()torch.cuda.get_device_name(0)
```



5. Navigate to the CML Projects page and confirm that the User Resources dashboard displays the GPU card availability.





6. SSH into the ECS master node and run the following command to verify that the node that hosting the above CML project session pod is ecsgpu.cdpkvm.cldr.

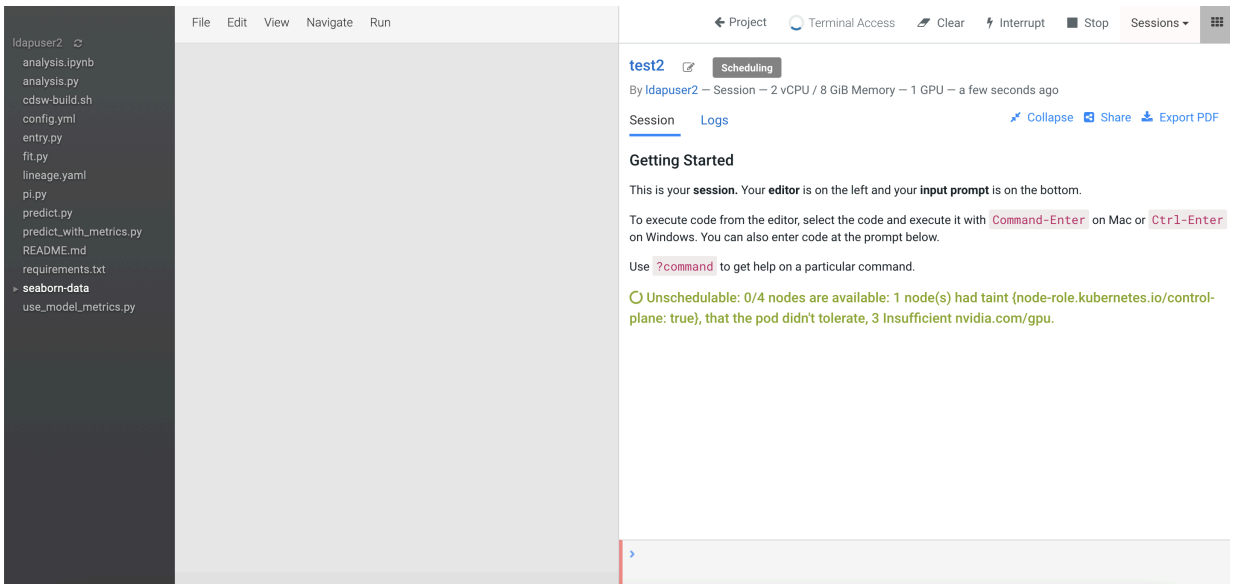
```
[root@ecsmaster1 ~]# oc -n workspace1-user-1 describe pod wifz6t8mvxv5ghwy
 | grep Node:
 Node:         ecsgpu.cdpkvm.cldr/10.15.4.185

 [root@ecsmaster1 ~]# oc -n workspace1-user-1 describe pod wifz6t8mvxv5gh
wy | grep-B2-i nvidia
     Limits:
         memory:          7714196Ki
         nvidia.com/gpu:  1
     --
         cpu:             1960m
         memory:          7714196Ki
         nvidia.com/gpu:  1
```

```
      --
```

7. When a process is consuming the NVIDIA GPU, the output of the nvidia-smi tool shows the PID of that process (in this case, the CML session pod).

```
[root@ecsgpu ~]# nvidia-smi
 Thu Aug 25 13:58:40 2022
 +-----------------------------------------------------------------------------
--------+
 | NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7
     |
 |-------------------------------+----------------------+----------------
------+
 | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
 ECC |
 | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Com
pute M. |
 |                               |                      |               MI
G M. |
 |===============================+======================+================
======|
 |   0  NVIDIA A100-PCI...  Off  | 00000000:08:00.0 Off |
   0 |
 | N/A   29C    P0    35W / 250W |  39185MiB / 40960MiB |      0%
Default |
 |                               |                      |             Disa
bled |
 +-------------------------------+----------------------+----------------
------+

 +-----------------------------------------------------------------------------
-------+
 | Processes:
      |
 |  GPU   GI   CI        PID   Type   Process name                  GPU Me
mory |
 |        ID   ID                                                   Usage
     |
 |=============================================================================
=======|
 |   0   N/A  N/A     29990      C   /usr/local/bin/python3.9       391
83MiB   |
 +-----------------------------------------------------------------------------
-----+
```

8. In the event that the ECS platform has no available worker node with a GPU card, provisioning a session with GPU will result in a Pending state as the system is looking for a worker node installed with at least one NVIDIA GPU card.



Decommissioning ECS Hosts

You can decommission ECS hosts and remove them from the cluster.

About this task

1. Cordon the node. Longhorn will automatically disable the node scheduling when a Kubernetes node is cordoned. Run the following command on any ECS Server host:

```
kubectl cordon [***node***]
```

2. Drain the node to move the workload to somewhere else. Run the following command on any ECS Server host:

```
kubectl drain [***node***] --ignore-daemonsets --pod-selector='app!=csi-attacher,app!=csi-provisioner' --delete-emptydir-data
```

3. Detach all the volumes on the node. Navigate to the ECS Service page on Cloudera Manager UI.

   a. In the Web UI dropdown, select Storage UI to open the Longhorn UI.

   b. Under the Volume tab in Longhorn UI, select the volumes on this node. Click Detach and select Yes on the screen prompt.

   If the node has been drained, all the workloads should be migrated to another node already.

   If there are any other volumes remaining attached, detach them before continuing.

4. Remove the node from Longhorn using the Delete in the Node tab. Or, remove the node from Kubernetes. Run the following command on any ECS Server host:

```
kubectl delete node [***node-name***]
```

   Longhorn will automatically remove the node from the cluster.

5. Uninstall ECS and Docker artifacts from the host. Run below commands on the host:

```
cd /opt/cloudera/parcels/ECS/bin
./rke2-killall.sh # usually 2 times is sufficient
```

```
./rke2-uninstall.sh
rm -rf /ecs/* # assumes the default defaultDataPath and lsoDataPath
rm -rf /var/lib/docker_server/* # deletes the auth and certs
rm -rf /etc/docker/certs.d/* # delete the ca.crt
rm -rf /docker # assumes the default defaultDataPath for docker
```

6. Go to the Hosts page for the ECS Cluster, select that host, and under Actions for Selected, click Begin Maintenance (Suppress Alerts/Decommission)

ECS Server High Availability

ECS Server High Availability (HA) is not enabled by default – you must enable it after installing ECS. If you do not wish to enable ECS HA, you can safely ignore this section. If you are enabling ECS HA, you should review the following notes and supported ECS Server scenarios before proceeding.

Note:

- Longhorn replication defaults to two replicas. This can be set only during the installation time. Three or more replicas potentially have performance issues.
- Kubectl delete node <host> permanently removes host from cluster and any data on the host is lost. You must reformat the host before reading to the cluster.
- Single node failure may cause the Control Plane or any other management service to be unavailable. In 1.3.4 or later, it will take several minutes to recover automatically.

ECS Server scenarios

Clusters with only two servers are not supported. This is only for the temporary transition from a single server cluster to a three server cluster.

1. Three or more servers
   - Redundancy requirements:
     - One failure requires three or more servers
     - Two failures require five or more servers
     - For more information see, Fault Tolerance
   - To recover, you must scale-up the ECS Server roles. For more information on adding ECS node to a cluster, see the following section.
2. Two servers to one server
   - Only after a double failure in a three server cluster
   - To recover:
     - Stop the ECS service
     - Remove both the failed ECS server roles and hosts from cluster
     - On the surviving server, run the following command `/opt/cloudera/parcels/ECS/bin/rke2 server --cluster-reset`
     - Start the ECS service
3. Single server
   - No failure supported

Enable ECS Server HA Post ECS Installation

If you want to enable ECS Server for High Availability after installing ECS, then you must proceed with this section. If you do not want to enable ECS HA, you can safely ignore this section.

As a prerequisite, during the installation, you must have installed ECS with 1 master (with app_domain as Load Balancer URL) + agents. When you are adding more masters, ensure that you add Docker server as well.

Install iptables on the new ECS master nodes

You must install iptables on all of the additional ECS master nodes.

If your ECS hosts are running the CentOS 8.4, OEL 8.4, or RHEL 8 operating systems, you must install iptables on all the ECS hosts. Run the following command on each additional ECS master node:

```
yum --setopt=tsflags=noscripts install -y iptables
```

Adding hosts to the containerized cluster

You must add hosts to the containerized cluster.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Click the Actions drop-down.
4. Click the Add Hosts button. The Add Hosts page appears.
5. Select the Add hosts to cluster option.
6. Select the cluster where you want to add the host from the drop-down list. Click Continue.
7. In the Specify Hosts page, provide a list of available hosts or you can add new hosts. You can provide the Fully Qualified Domain Name (FQDN) in the following patterns: You can specify multiple addresses and address ranges by separating them by commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges.

   For example, use host[1-3].network.com to specify these hosts: host1.network.com, host2.network.com, host3.network.com.

   Click Continue.
8. In the Select Repository page, you must specify the repository location. Choose any one of the following:
   a. Cloudera Repository (Requires direct internet access on all hosts)
   b. Custom Repository
9. In the Select JDK page, select any one from the below options:
   a. Manually manage JDK
   b. Install a Cloudera-provided version of OpenJDK
   c. Install a system-provided version of OpenJDK
10. In the Enter Login Credentials page select the SSH Username and provide the password.
11. The Install Agents page appears. Click Continue.
12. In the Install Parcels page, the selected parcels are downloaded and installed on the host cluster. Click Continue.
13. In the Inspect Hosts page, you can inspect your hosts. If the inspect tool displays any issues, you can fix those issues and run the inspect tool again. Click Continue.
14. In the Select Host Template page, select the hosts.
15. The Deploy Client Config page appears. Click Finish.

Adding Role Instances to Docker Server

You must add role instances to the docker server.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Open Docker Server.
4. Click the Actions drop-down.
5. Click the Add Role Instances button.
6. Select the hosts.
7. Click OK.

Adding Role Instances to Containerised Cluster

You must add the role instances to the containerised cluster.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Click the Actions drop-down.
4. Click the Add Role Instances button. The Add Role Instances page appears.
5. In the Assign Roles page, specify the role assignments for your new roles. Click Continue.
6. In the Review Changes page, click Finish.

Starting Docker Server on Nodes

You must start the Docker server on nodes.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Open Docker Server.
4. Click the Actions for Selected drop-down.
5. Click Start. Docker Server starts.

Starting ECS Server on Nodes

You must start the ECS server on nodes.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Click the Instances tab.
4. Select the nodes by clicking the checkbox
5. Click the Actions for Selected drop-down.
6. Click Start. ECS Server starts.

Refreshing ECS

You must refresh the ECS servers.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Click the Actions drop-down.
4. Click the Refresh button.

Checking Nodes and Pods in the UI

You must check the nodes and pods in the UI.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. Click the Web UI drop-down.
4. Click ECS Web UI. The Kubernetes web UI page opens in a new tab.
5. Check the Nodes and Pods on the Web UI.

Enable ECS Server HA and promote agents Post ECS Installation

If you want to enable ECS Server for High Availability after installing ECS, then you must proceed with this section. If you do not want to enable ECS HA, you can safely ignore this section.

As a prerequisite, during the installation, you must have installed ECS with 1 master (with app_domain as Load Balancer URL) + agents. This allows you to promote Agents as masters.

Enabling ECS Server deployment for High Availability

You can enable ECS Server deployment for High Availability by installing a Load Balancer and promoting the existing ECS Agents to ECS Server. By performing this procedure, you will be able to deploy HA on your existing ECS Server. You must have an ECS cluster installed and configured with a single ECS Server.

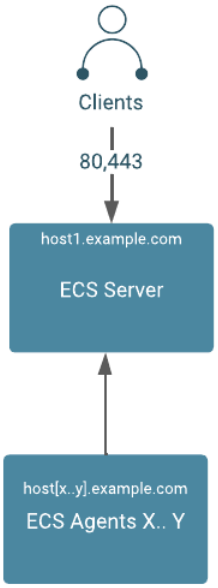
If you have a production quality ECS cluster, Cloudera recommends that you configure ECS Server High Availability. You can also consider having an ECS Server HA for any non-production ECS cluster that you expect to be available long-term.

Enabling ECS Server deployment for High Availability involves preparing your cluster, configuring a DNS wildcard entry, adding a Load Balancer into the topology, and promoting ECS Agents to the ECS Server. An ECS High Availability cluster must consist of:

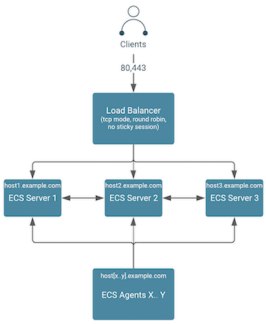
- An odd number of server nodes that will run etcd, the Kubernetes API, and other control plane services. Cloudera recommends a minimum of three ECS Server nodes.
- Two or more agent nodes that are designated to run CDP data services.
- A software or hardware Load balancer using TCP mode (non-terminating https).

Note: A Load Balancer is required for the ECS Server HA. This documentation uses HAProxy as an example. However, Cloudera recommends that you use your production quality Load Balancer technology from commercial vendors.

Architecture of CDP Private Cloud Data Services on a single ECS Server:



Architecture of CDP Private Cloud Data Services with High Availability:



Preparing the cluster for High Availability:

Review the table to understand the requirements for enabling the High Availability.

1. This process has been tested with a minimum of five ECS hosts. However, Cloudera recommends six or more hosts.
2. DNS requirements for ECS High Availability must be fulfilled.

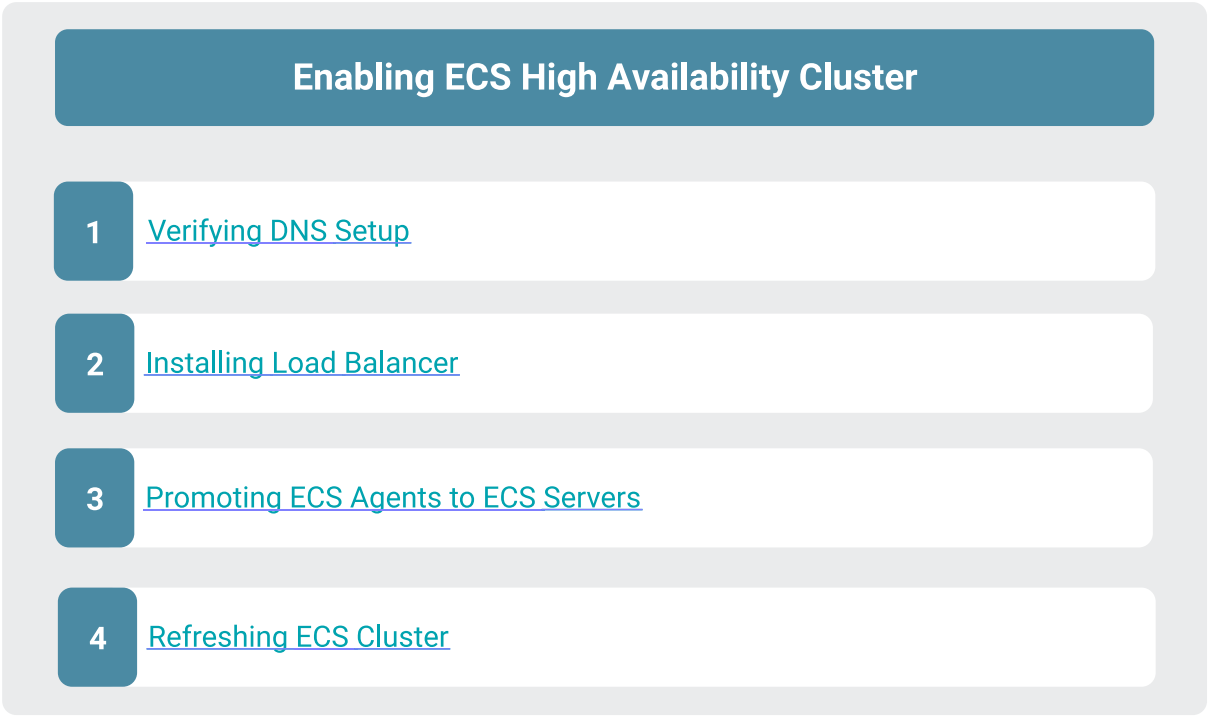
Hostname	Subdomain	Expected Roles	DNS ForwardZone	Reverse Zone PTR
“Wildcard” (hostname = *)	apps.ecs.example.com The string “apps” is required, “ecs” is up to user	Virtual app domain wildcard	“A Record” wildcard (hostname = *), may be a CNAME on certain DNS systems that use text-based config. Resolves to fixed IP of ha_proxy (or VIP of some commerical LB’s)	N
“apps alias”	apps.ecs.example.com	Virtual app domain alias	“CNAME” alias points to A Record of ha_proxy (or VIP). Alternatively, this can be an ARecord with IP of ha_proxy (or VIP)	N/A
HAProxy (or commercial LB)	<domain of your LB>	HA Load Balancer	Depends on vendor/ software	
ecs-master1	example.com	ECS Server 1 Docker server	“A Record” resolves to IP of ecs-master1	Y
ecs-master2	example.com	ECS Server 2 Docker server	“A Record” resolves to IP of ecs-master2	Y
ecs-master3	example.com	ECS Server 3 Docker server	“A Record” resolves to IP of ecs-master3	Y
ecs-agentN	example.com	ECS Agent N Docker server N	“A Record” resolves to IP of ecs-agentN	Y

Note:

1. The above table uses a consistent subdomain (“example.com”) but this is not mandatory. To support multiple domains, you must follow certain steps to ensure that the domains are forward and reverse resolvable using DNS, from all Base cluster and ECS cluster hosts (that is through forest/domain level trusts and/or hosts level /etc/resolv.conf config). You must avoid the use of /etc/hosts entries.
2. A predefined wildcard DNS record allows the resolution of *.apps.<app domain name> to the IP address of the Load Balancer. You cannot proceed further until this is in place.

High Level steps to enable an ECS High Availability cluster

Review the high level steps to understand the steps in enabling High Availability.



Note:

1. You must have installed an ECS with one ECS server and other nodes that are ECS Agents.
2. You must have a DNS wildcard record that has an IP address pointing to your Load Balancer (hostname or VIP). For more information, see the KB article.

Verifying DNS setup

You must verify the DNS setup to ensure that the app domain DNS hostname points to the Load Balancer.

Procedure

1. Verify that the app domain DNS hostname has moved from single non-HA ECS Server to the Load Balancer.

Hostname	Expected Roles	DNS
ecs-loadbalancer.example.com	Load Balancer	Resolves to IP of LB host (or VIP). The example uses 10.10.0.99. Both *.apps.ecs.example.com and apps.ecs.example.com resolve to 10.10.0.99.

2. Verify the DNS setup with nslookup.

   Note: You must verify that a random hostname resolves in the wildcard entry. In this example, Cloudera uses foobar.apps.ecs.example.com as the random name. Both entries should resolve to the same IP address.

   For example,

```
$ hosts="apps.ecs.example.com foobar.apps.ecs.example.com"
$ for target in $hosts; do nslookup $target; done

Server: 10.10.xx.xx
Address: 10.10.xx.xx#53

apps.ecs.example.com canonical name = ecs-loadbalancer.example.com.
Name: ecs-loadbalancer.example.com
```

```
Address: 10.10.0.99

Server: 10.10.xx.xx
Address: 10.10.xx.xx#53

Name: foobar.apps.ecs.example.com
Address: 10.10.0.99
```

Results
DNS setup is verified.

What to do next
You must now install the Load Balancer.

Installing Load Balancer
To install the HAProxy Load Balancer, Cloudera uses an example that uses a single instance of HAProxy, configured with round robin balancing and TCP mode. This allows for non-terminating https (https passthrough). The HAProxy service can be configured for High Availability using keepalived.

Before you begin

You must consult your operating system vendor's documentation for requirements and the install guide for configuring HAproxy with keepalived.

To install a HAProxy Load Balancer, you must ssh into the HAProxy host, install, and then configure HAProxy:

Procedure

1. `sudo su -`
2. `yum install haproxy -y`
3. `cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak`
4. `cat > /etc/haproxy/haproxy.cfg << EOF`

   global

log	127.0.0.1 local2
chroot	/var/lib/haproxy
pidfile	/var/run/haproxy.pid
user	haproxy
group	haproxy
daemon	

   defaults

mode	tcp
log	global
option	tcplog
option	dontlognull
option	redispatch
retries	3

maxconn	5000
timeout connect	5s
timeout client	50s
timeout server	50s

listen stats

```
bind *:8081
mode http
stats enable
stats refresh 30s
stats uri /stats
monitor-uri /healthz
```

frontend fe_k8s_80

```
bind *:80
default_backend be_k8s_80
```

backend be_k8s_80

```
balance roundrobin
mode tcp
server ecs-server1.example.com 10.10.0.1:80 check
server ecs-server2.example.com 10.10.0.2:80 check
server ecs-server3.example.com 10.10.0.3:80 check
```

frontend fe_k8s_443

```
bind *:443
default_backend be_k8s_443
```

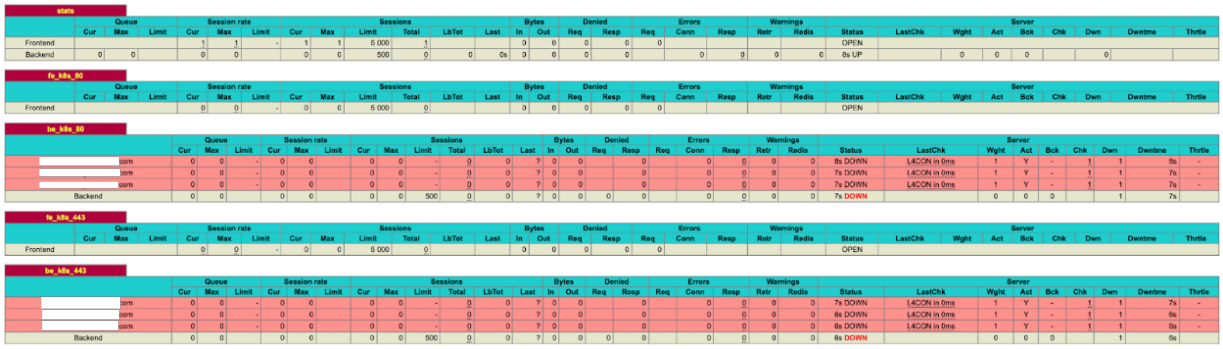
backend be_k8s_443

```
balance roundrobin
mode tcp
server ecs-server1.example.com 10.10.0.1:443 check
server ecs-server2.example.com 10.10.0.2:443 check
server ecs-server3.example.com 10.10.0.3:443 check
```

EOF

```
systemctl enable haproxy
systemctl restart haproxy
systemctl status haproxy
```

5. You can verify that all the hosts are shown from the HAproxy UI. However, at this point the hosts are not listening to the configured ports.



Important: Since you already have an ECS cluster running, you must alter your DNS wildcard to point to the IP address of the HAProxy server. You cannot change the Application Domain configured through the ECS wizard. So you must ensure that you send all ingress traffic to the HAProxy IP address by making that change in the IP address of your wildcard DNS Record.

Note:

a. Application Domain (app_domain property in Cloudera Manager) maps to your wildcard DNS record (For example, app_domain ecs.example.com maps to your DNS entry *.apps.ecs.example.com)

b. The resolved IP address must be the host IP (or VIP) of your Load Balancer. For more information, see the Verify DNS Step 5 above.

Results

Load Balancer is now installed.

Promoting ECS Agents to ECS Servers

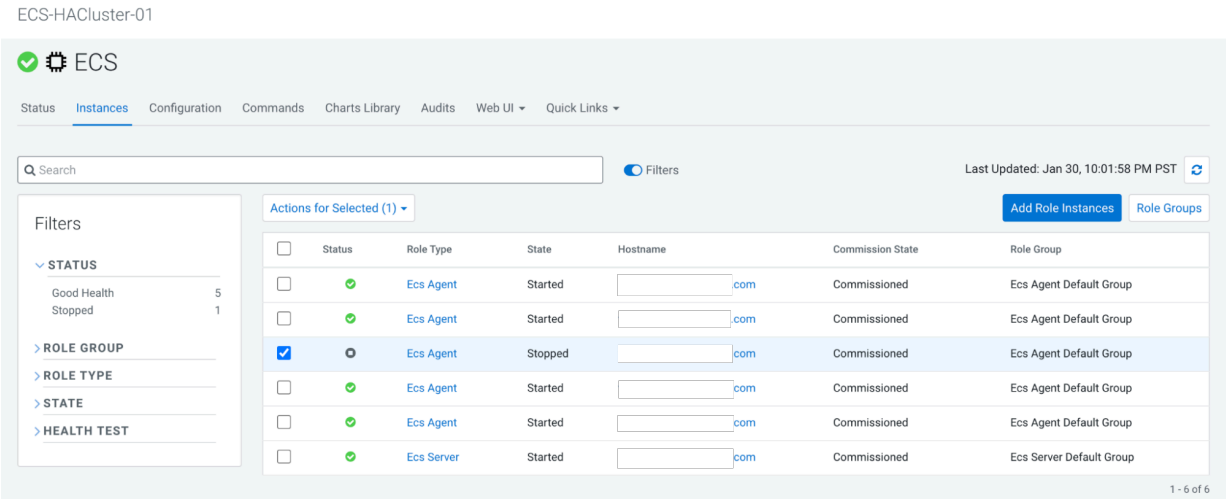
After installing the Load Balancer, you must reconfigure the existing Embedded Container Service (ECS) Agents to ECS Servers. This process is referred to as promoting the agents to servers. You must promote only one agent at a time.

About this task

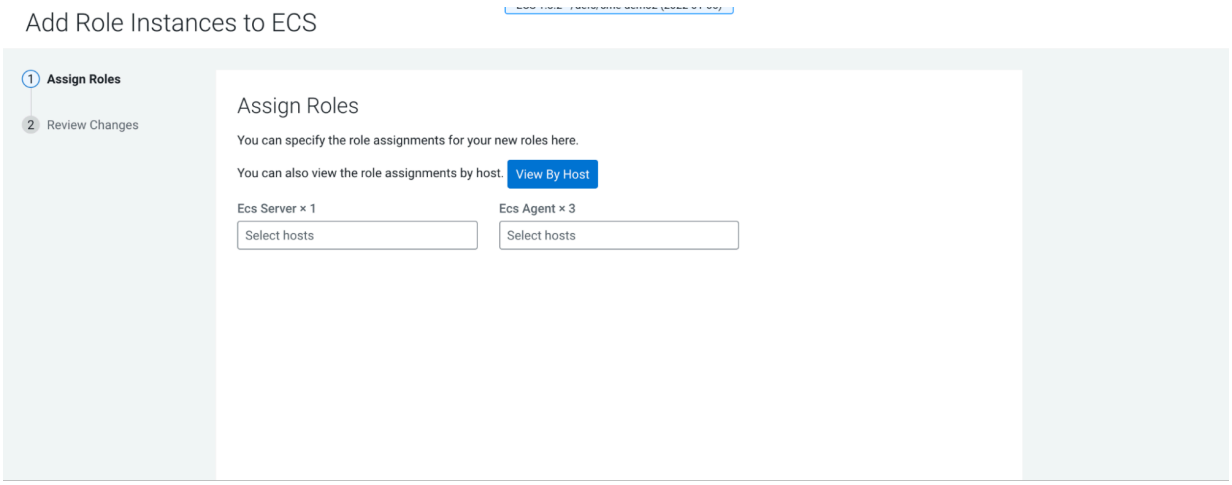
In this example we will promote the ECS agent on agent1.example.com and then promote the ECS agent on agent2.example.com.

Procedure

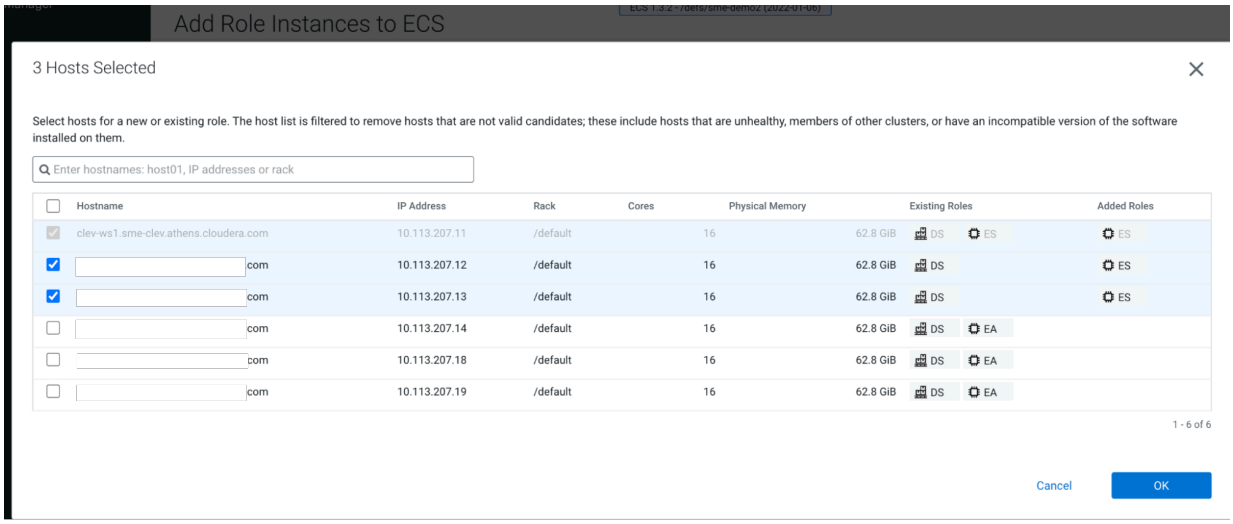
1. In Cloudera Manager, select the ECS cluster, then click ECS. Stop the ECS agent running on agent1 and then delete the agent.

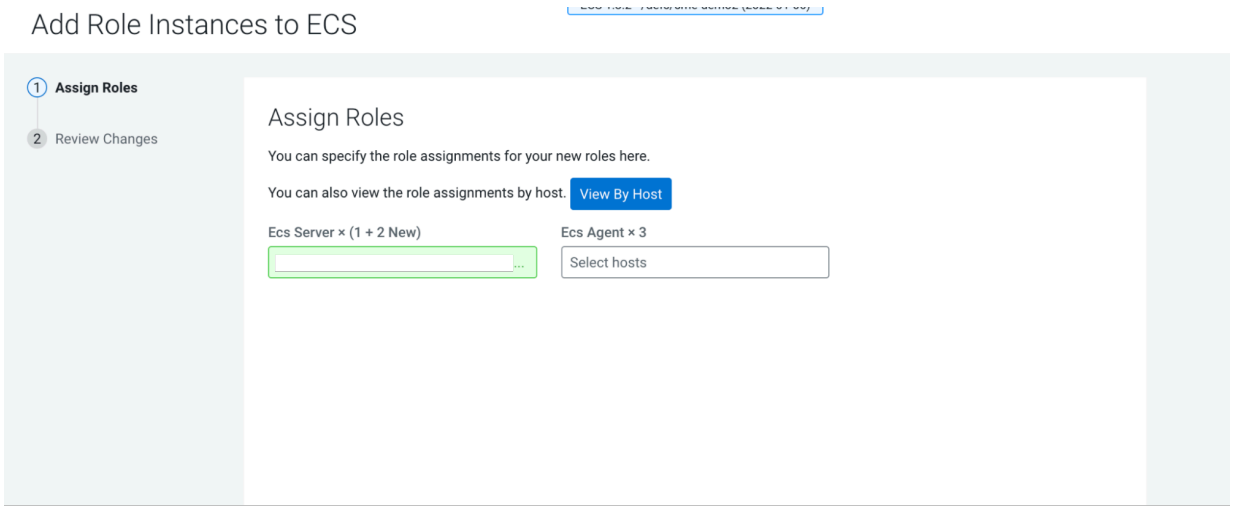


2. In ECS, click Add Role Instances.

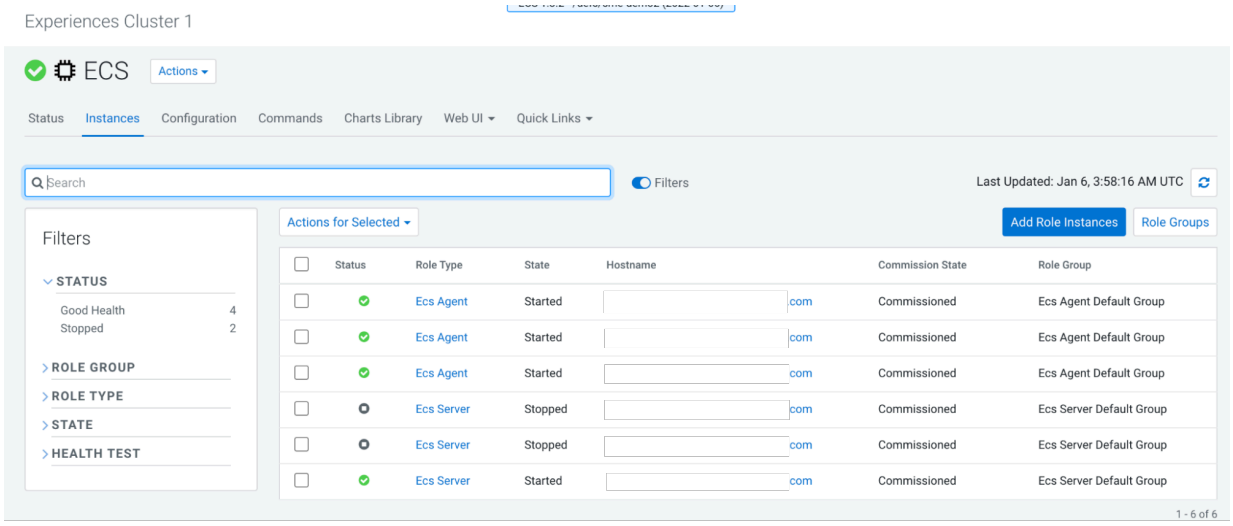


3. Add the available host agent1 as an ECS server in the Add Role Instances to ECS pop-up. Click OK.

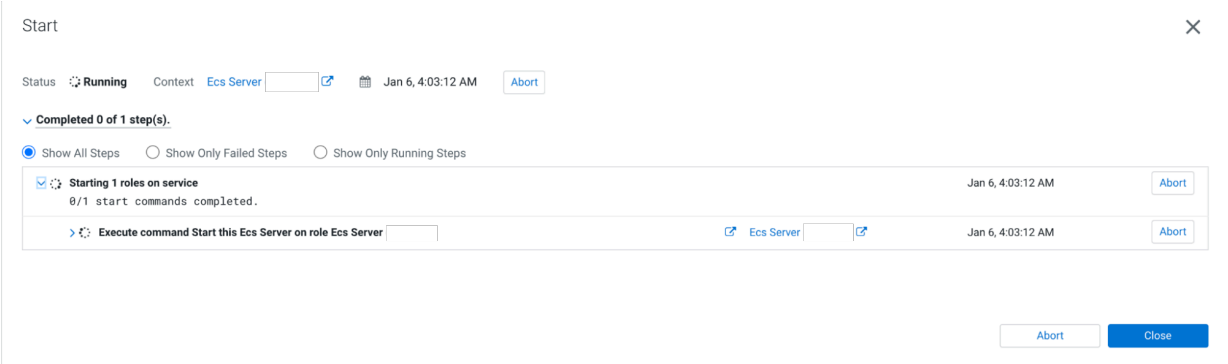




4. Click Continue.



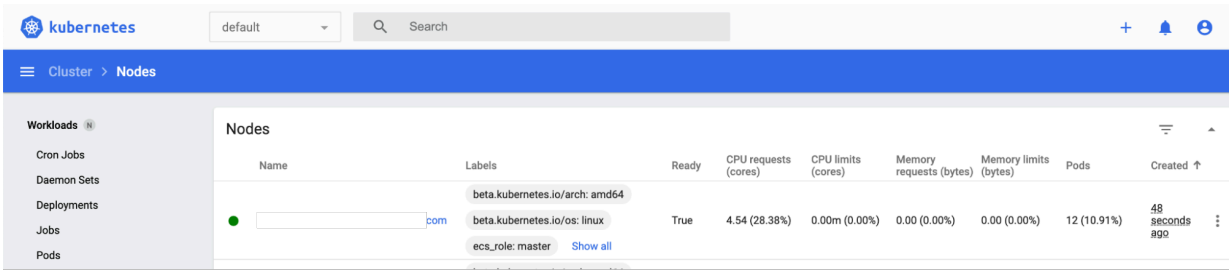
5. Start the new ECS server from the ECS Instances view. For example, start the ECS server on agent1.



6. Confirm the node's status from the Web UI or the command line by running the following command:

```
sudo /var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.yaml get nodes
```

Note: Do not proceed until the node status is Ready. This may take several minutes.



What to do next

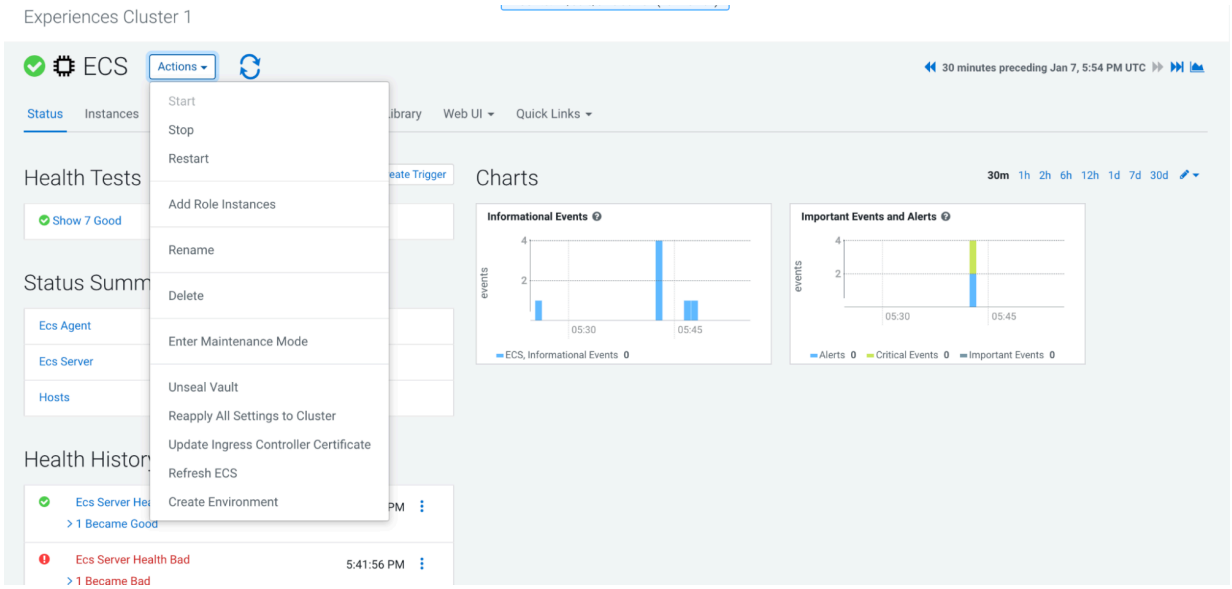
When agent1 is ready, you can promote agent2. To promote agent2, perform steps 1-8 again using agent2.example.com.

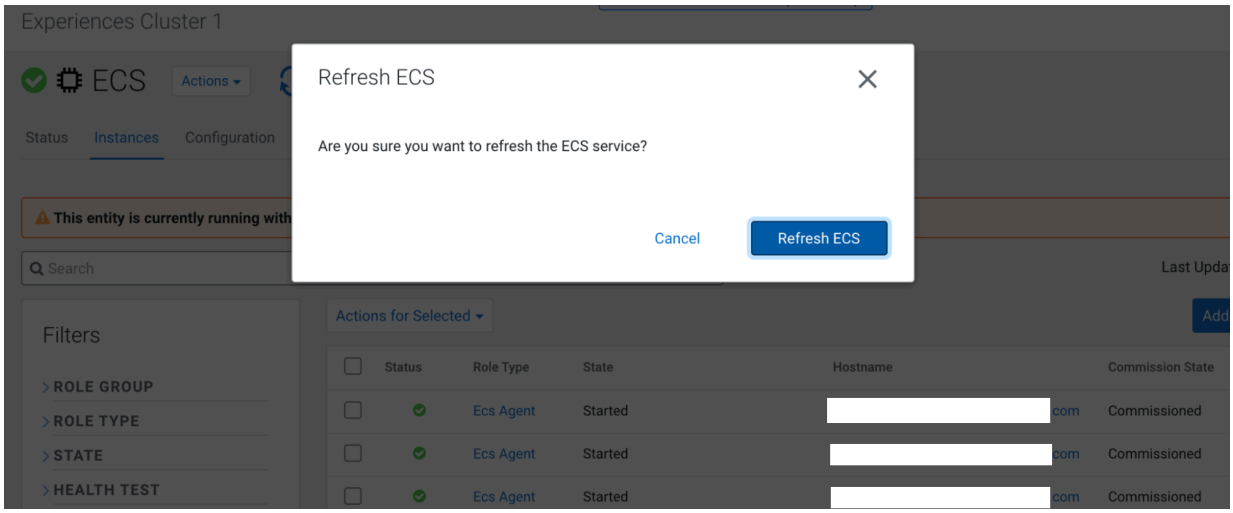
Refreshing ECS

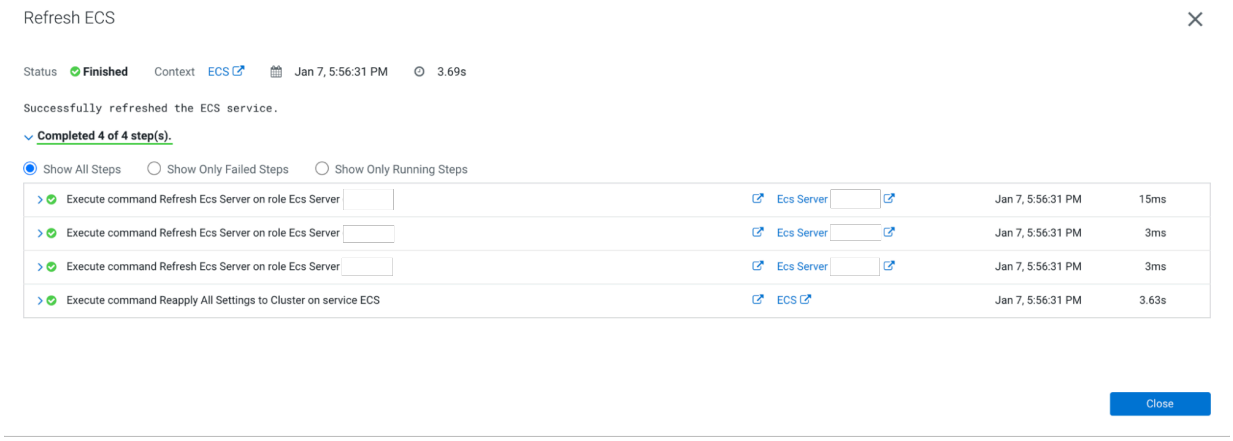
After all the ECS Agents are promoted to ECS Servers, you must log in to Cloudera Manager and refresh the ECS cluster.

Procedure

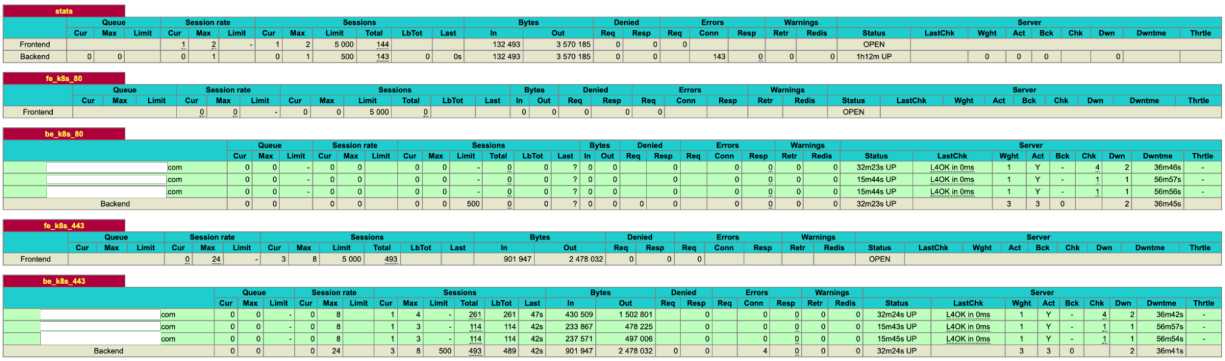
1. Navigate to ECS Cluster >> ECS view >> Actions >> Refresh ECS. This sets the ingress proxy so that all three servers are eligible to process incoming commands.







2. Confirm that all backends of HAProxy display the status UP. This may take several minutes.



stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	2	-	1	2	5 000	144			132 493	3 570 185	0	0	0					OPEN								
Backend	0	0		0	1		0	1	500	143	0	0s	132 493	3 570 185	0	0		143	0	0	0	1h12m UP		0	0	0		0		

fe_k8s_80

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	5 000	0			0	0	0	0	0					OPEN								

be_k8s_80

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
com	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	32m23s UP	L4OK in 0ms	1	Y	-	4	2	36m46s	-
com	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	15m44s UP	L4OK in 0ms	1	Y	-	1	1	56m57s	-
com	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	15m44s UP	L4OK in 0ms	1	Y	-	1	1	56m56s	-
Backend	0	0		0	0		0	0	500	0	0	?	0	0	0	0		0	0	0	0	32m23s UP		3	3	0		2	36m45s	

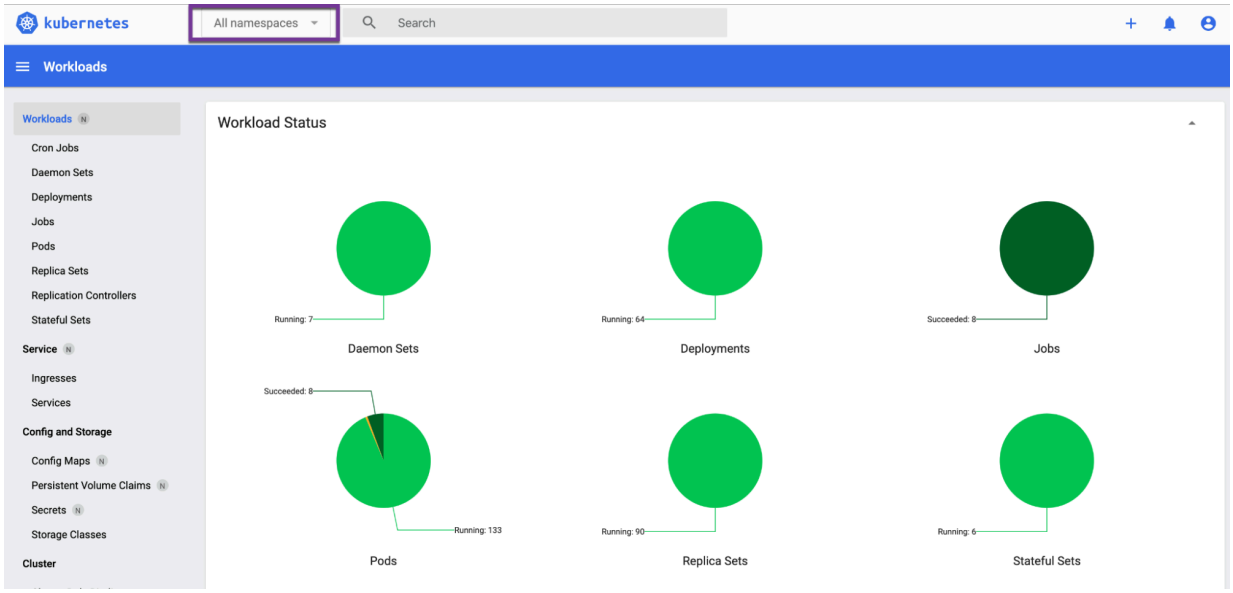
fe_k8s_443

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	24	-	3	8	5 000	493			901 947	2 478 032	0	0	0					OPEN								

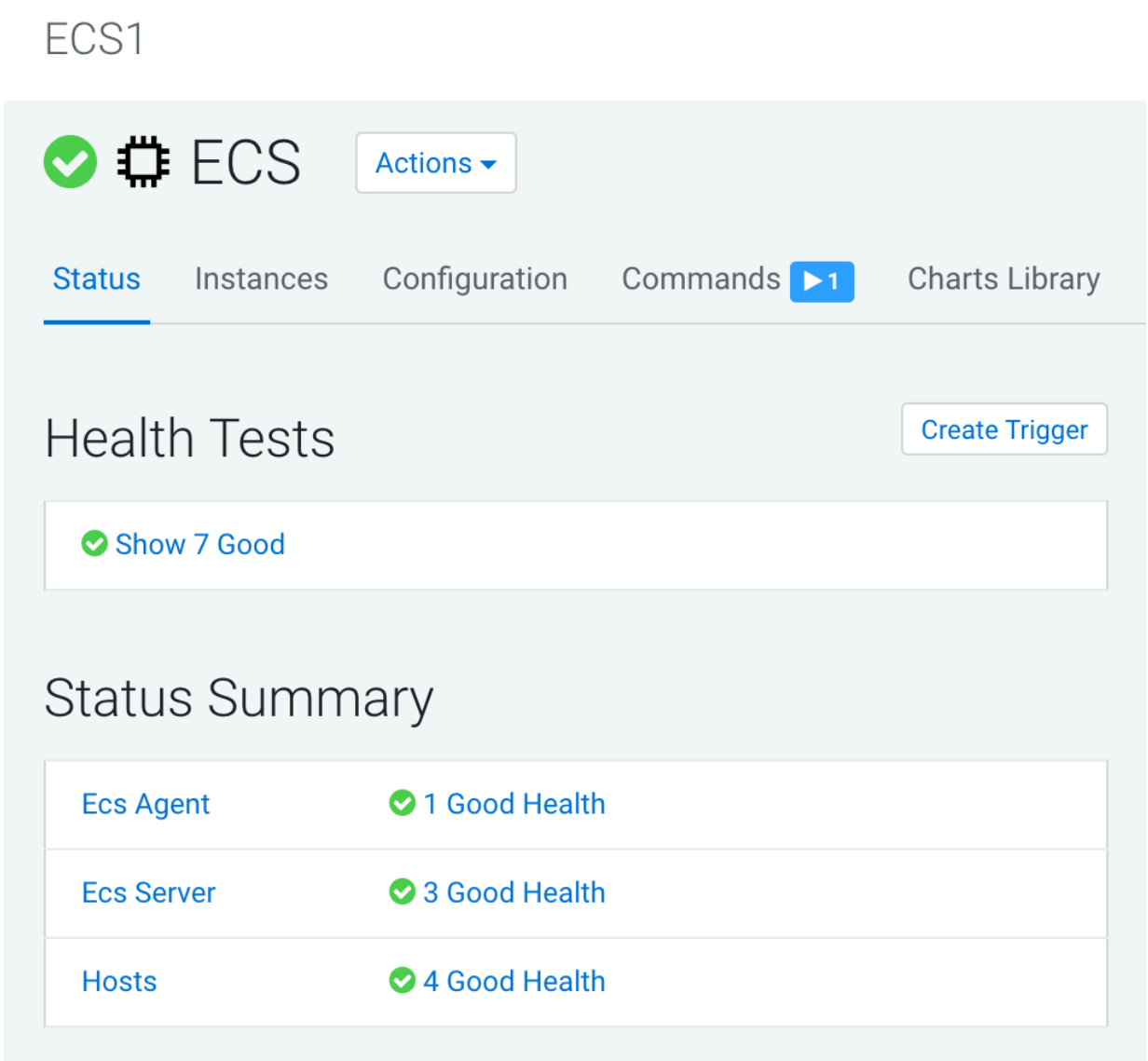
be_k8s_443

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
com	0	0	-	0	8		1	4	-	261	261	47s	430 509	1 502 801		0		0	0	0	0	32m24s UP	L4OK in 0ms	1	Y	-	4	2	36m42s	-
com	0	0	-	0	8		1	3	-	114	114	42s	233 867	478 225		0		0	0	0	0	15m43s UP	L4OK in 0ms	1	Y	-	1	1	56m57s	-
com	0	0	-	0	8		1	3	-	114	114	42s	237 571	497 006		0		0	0	0	0	15m45s UP	L4OK in 0ms	1	Y	-	1	1	56m54s	-
Backend	0	0		0	24		3	8	500	493	489	42s	901 947	2 478 032	0	0		4	0	0	0	32m24s UP		3	3	0		2	36m41s	

3. Confirm that all pods are green in the ECS webUI >> (All Namespaces) >> Workloads.



4. Confirm that there are no alerts in the ECS service.



Results
High Availability is now deployed on your ECS cluster.

Create an environment-wide backup

Data Recovery Service (DRS), a microservice in CDP Private Cloud Data Services, enables you to create an environment-wide backup of Kubernetes namespaces and resources on Embedded Container Service (ECS) and OpenShift Container Platform (OCP) in CDP Private Cloud Data Services Management Console.

Cloudera recommends that you create a backup of your Kubernetes namespace before a maintenance activity, before you upgrade, or in general, as a best practice.

Role Required: *PowerUser*

When you initiate the backup event in the Backup and Restore Manager for Control Plane, the data recovery service takes a backup of the following resources and data:

- Kubernetes resources associated with the *cdp* namespace and the embedded vault namespaces of the Control Plane in CDP Private Cloud Data Services. The resources include deployment-related information, stateful sets, secrets, and configmaps.
- Data used by the stateful pods, such as the data in the embedded database and Kubernetes persistent volume claim.

The data recovery service can back up and restore Kubernetes namespaces behind Cloudera Data Warehouse (CDW) entities (for example, Database Catalogs and Virtual Warehouses) on demand.

By default, the data recovery service is located in the *[***CDP_INSTALLATION_NAMESPACE***]*-drs namespace. For example, if the CDP Private Cloud Data Services installation is located in the *cdp* namespace, the data recovery service namespace is automatically named *cdp-drs*. If you have multiple CDP Private Cloud Data Services installations (as in OCP), the data recovery service is named accordingly.

The Data Recovery Service requires CSI snapshots to back up and restore Kubernetes namespaces and resources. The CSI snapshots are enabled on ECS by default. You might require an additional license to enable CSI snapshots in Red Hat ODF storage on OCP.

You can choose one of the following methods to back up and restore namespaces:

- **Backup and Restore Manager** in the CDP Private Cloud Data Services Management Console
- CDP CLI options

Creating backup of Control Plane and restoring it

The Backup and Restore Manager in the CDP Private Cloud Data Services Management Console helps you to backup and restore Kubernetes namespaces and resources on Embedded Container Service (ECS) and OpenShift Container Platform (OCP). You can also restore and delete the backups.

Before you begin

Ensure that the following prerequisites are complete:

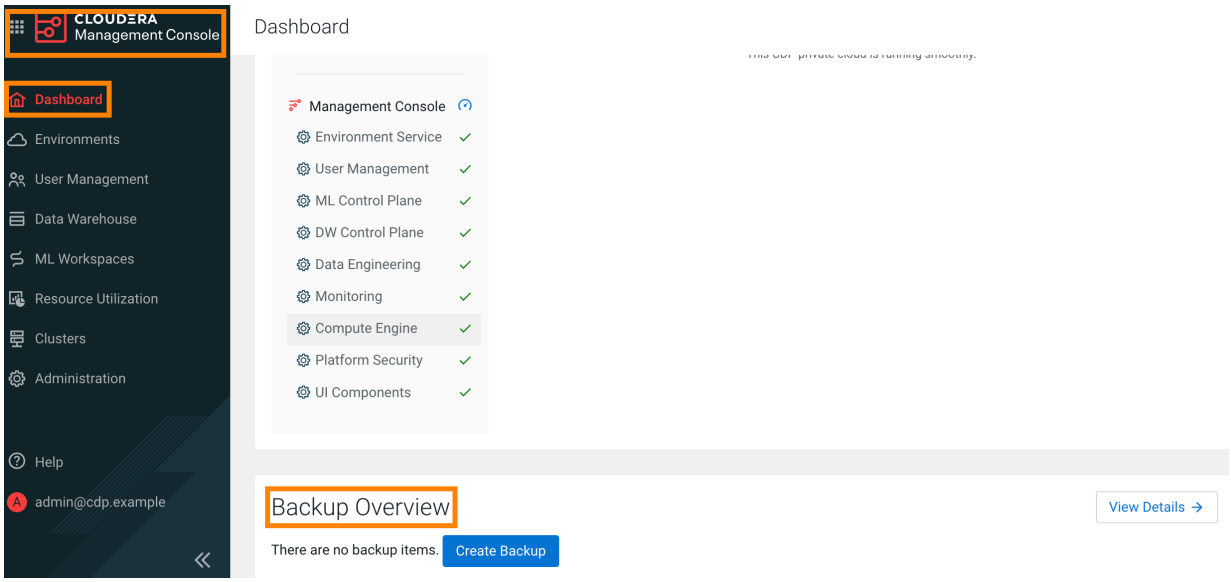
- You must have the *PowerUser* role.
- For OCP, ensure that a VolumeSnapshotClass is installed with a CSI driver that matches the CSI driver for the storage class used.

About this task

The following steps show how to create a backup of the Kubernetes namespaces and resources in the Control Plane, restore a backup, delete a backup, view logs for an event, and sample CDP CLI commands.

Procedure

1. Go to the CDP Private Cloud Data Services Management Console Dashboard Backup Overview section.

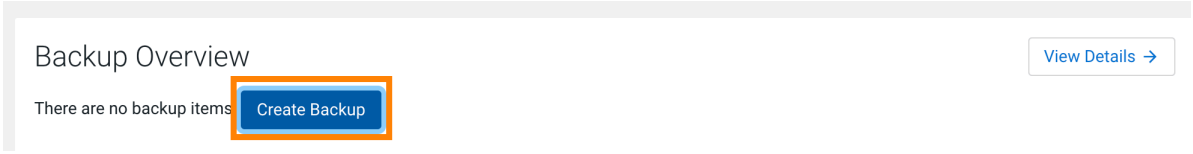


2. To create a backup, perform the following steps:

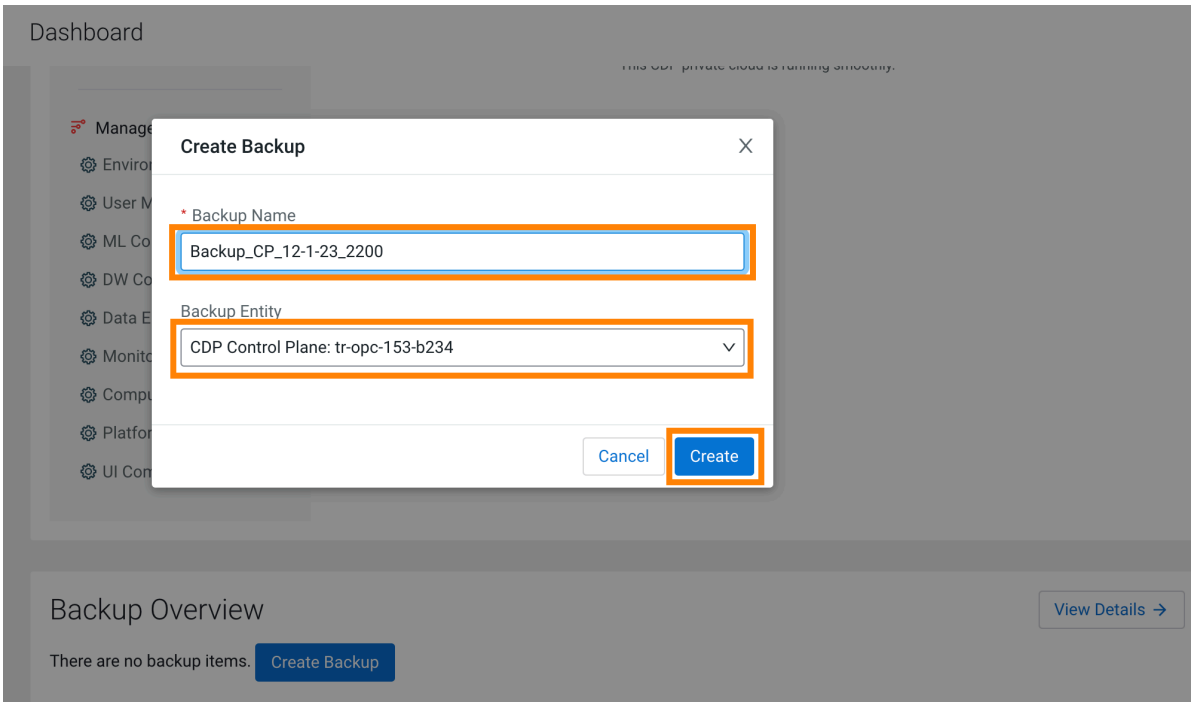
   When you create a backup of the Control Plane, the data recovery service initiates the backup event or job for the chosen backup entity, assigns an ID called backupCrn to the backup event, and creates a backup of the persistent

volume claim (PVC) snapshots of the Control Plane namespaces and the backup event's PVC. CRN or Customer Resource Number is the Cloudera-specific identifier provided for an event or job.

a) Click Create Backup in the Backup Overview section to create the first backup.

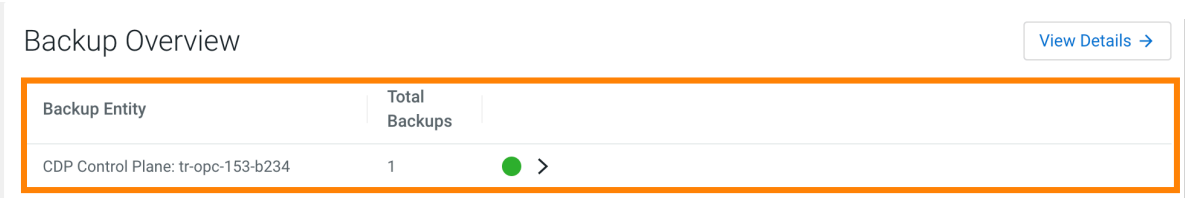


b) Enter a unique Backup Name and choose the Backup Entity that you want to back up in the **Create Backup** modal window, and then click Create.



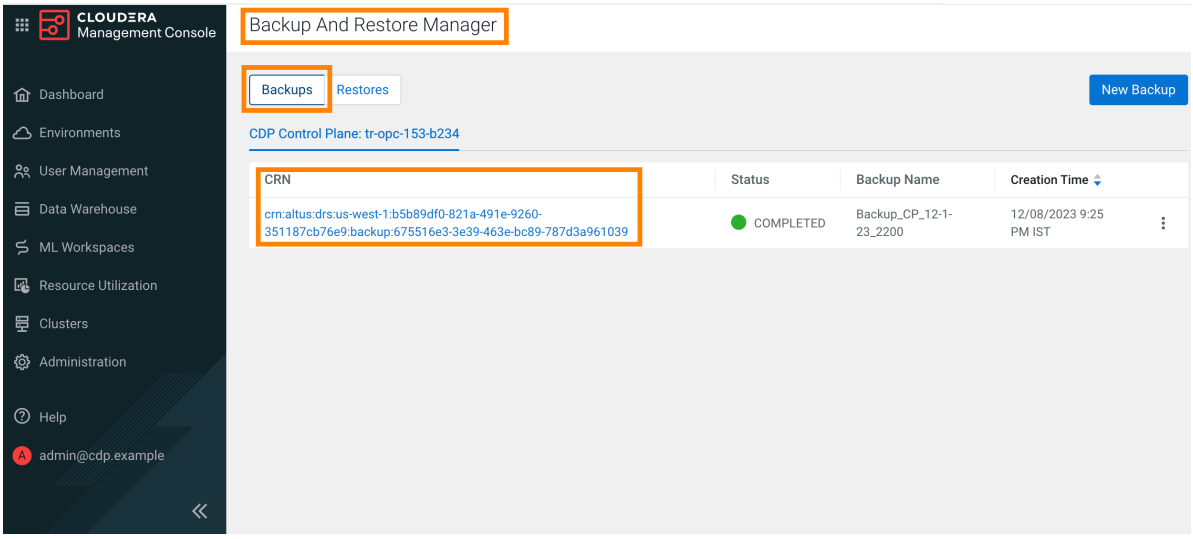
c) The data recovery service initiates the backup event and generates a backupCRN which is an automatically assigned ID for the backup event.

The backup event appears in the **Backup Overview** section.

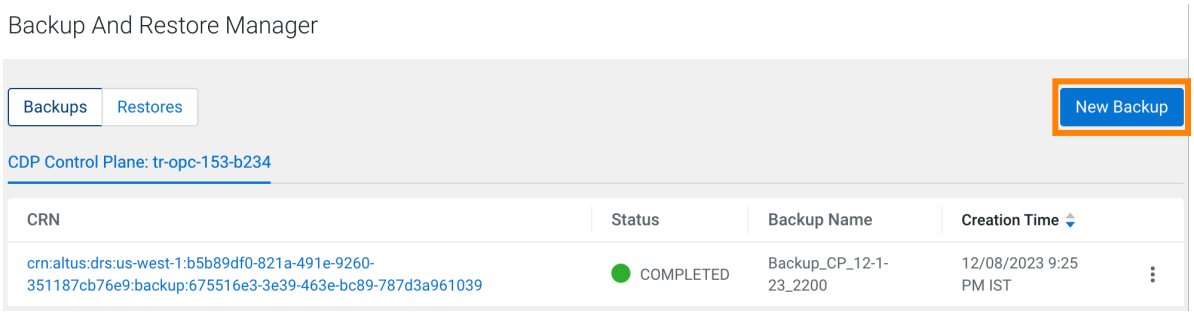


d) Click View Details. The **Backup and Restore Manager** page appears.

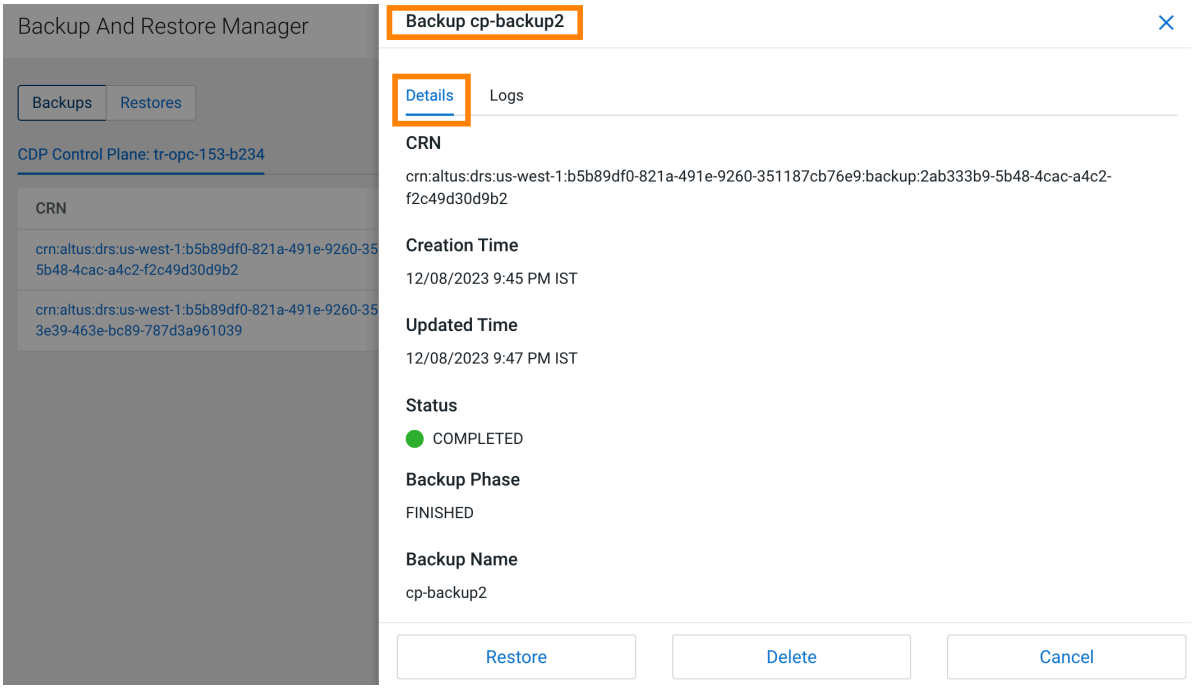
e) The backupCRN appears as a CRN on the **Backup and Restore Manager** page that you can click to view the backup event details.



f) For subsequent backups, click New Backup on the Backup and Restore Manager page.



g) Click the CRN to view more details about it on the Backup *[***backup name***]* modal window. For example, the following image shows the **Backup cp-backup2** modal window.



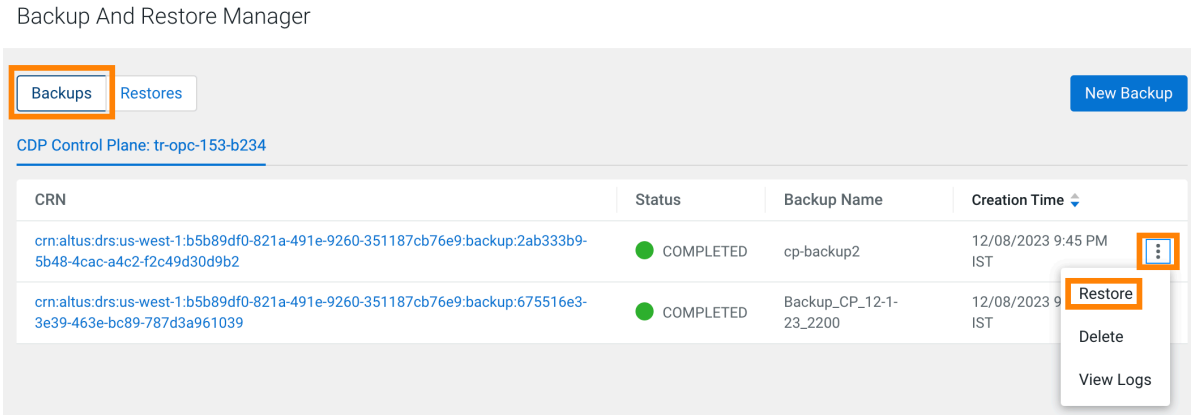
3. To restore a backup, perform the following steps:

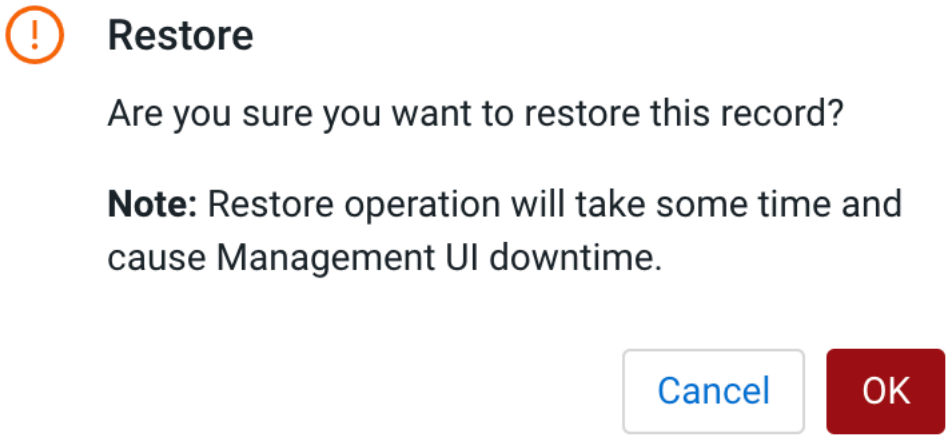
   When you start the restore a backup, the data recovery service initiates the restore event based on the chosen backup, assigns an ID called restoreCrn to the restore event, deletes the existing resources and data, and restores the resources and data from the backup.

   Note: The restore event has a downtime impact because the PODs and data are recreated. During the restore event, the ECS restore vault is sealed and the POD is down which might appear as a failure in the control plane environment. After the restore event is complete, the vault and POD are auto-recovered and restored. Depending on the number of resources and data, this step might take a maximum of 10 minutes to complete. If the environment does not come up, see the logs to troubleshoot. You can also contact your Cloudera account team.

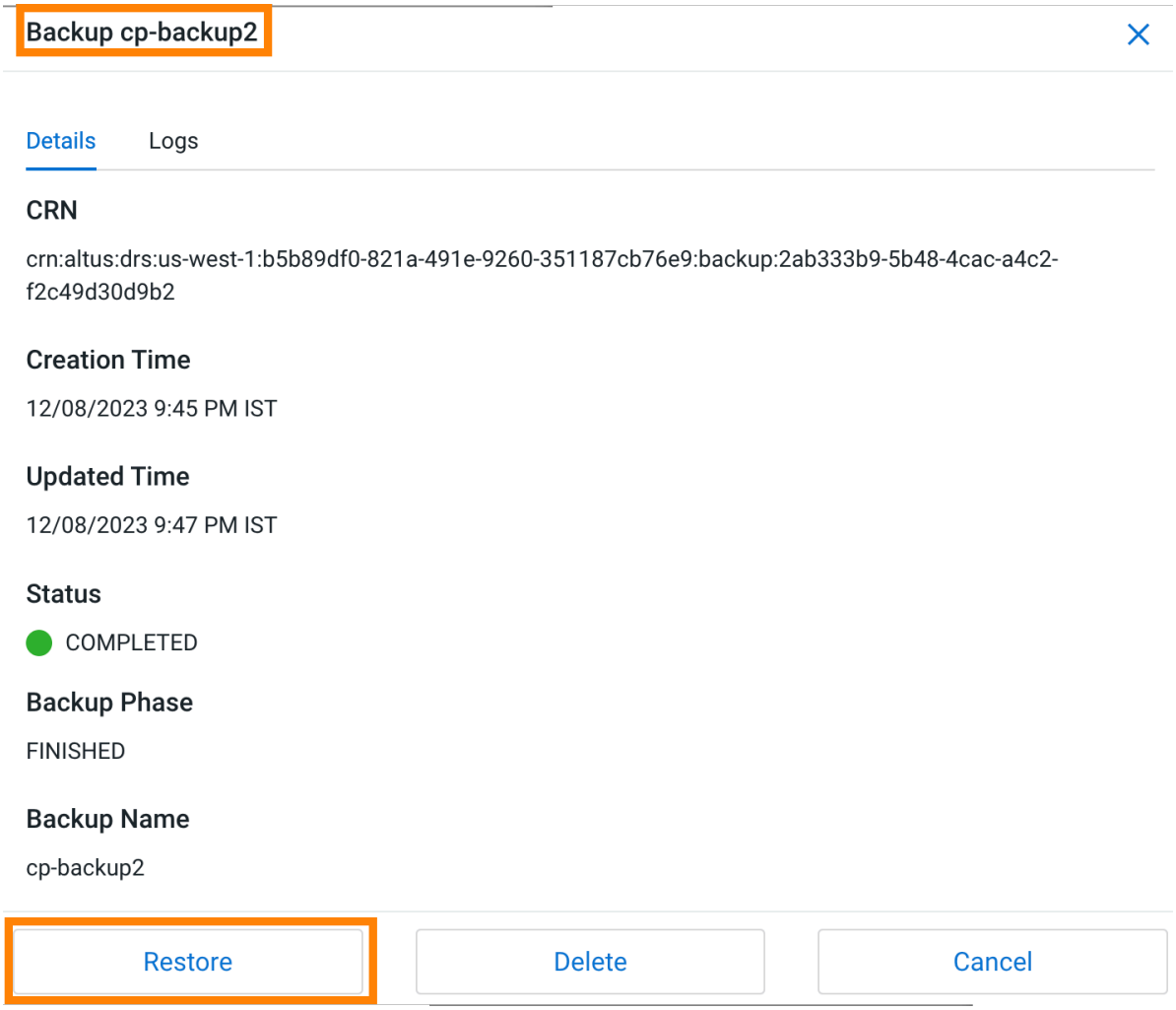
   a) Go to the Backup and Restore Manager Backups tab.
   b) Click Actions Restore , and then click OK in the **Restore** modal window to acknowledge that you want to restore the backup.

      Important: Do not delete the *[***CDP_INSTALLATION_NAMESPACE***]*-drs namespace while the restore event is in progress. For example, if the CDP Private Cloud Data Services installation is located in the *cdp* namespace, the data recovery service namespace is automatically named *cdp-drs*.

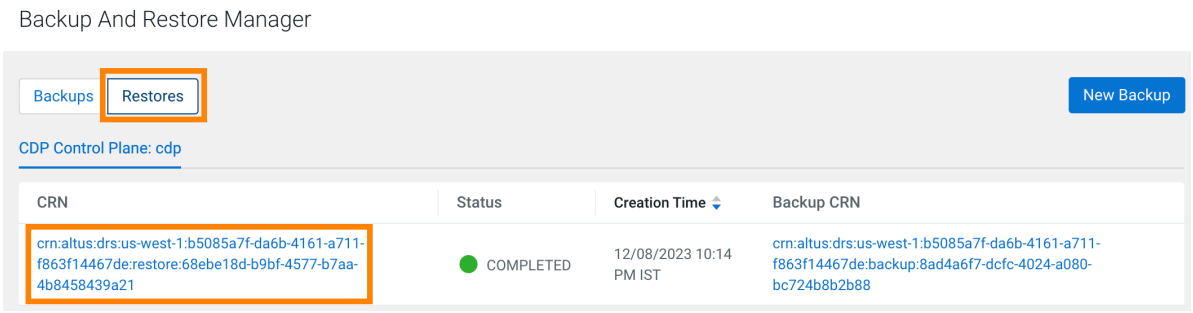




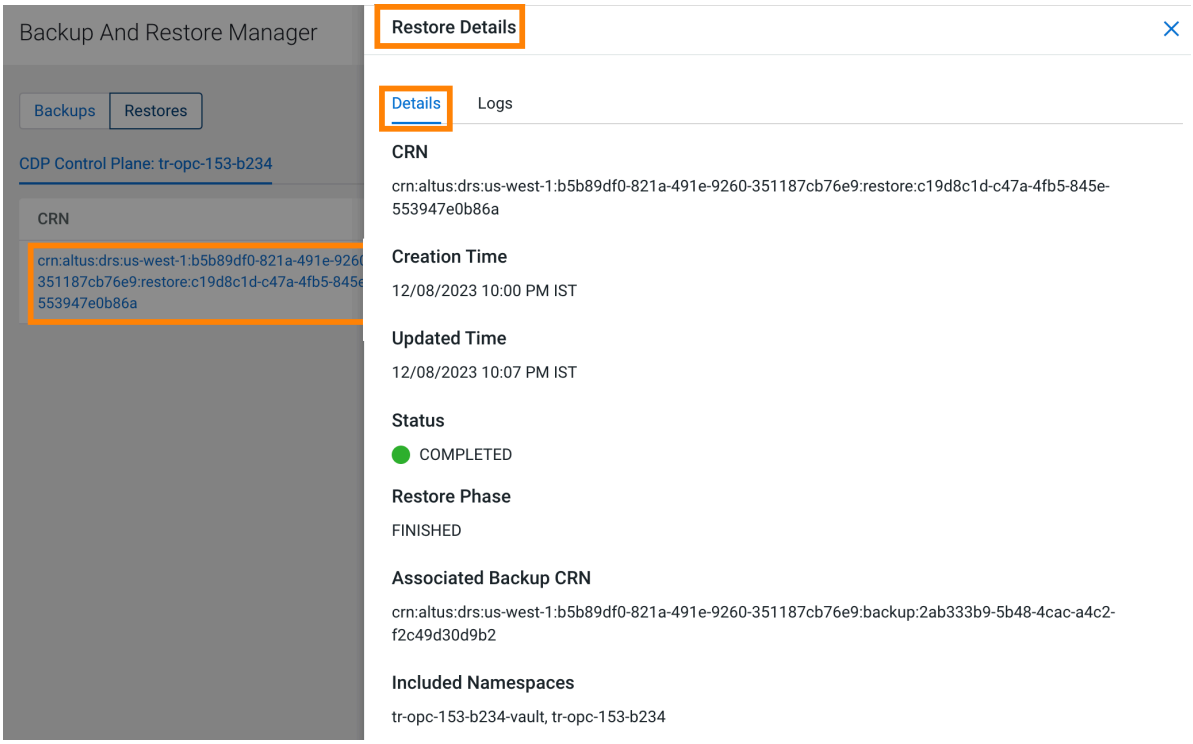
   c) Alternatively, click the CRN of the required backup, click Restore on the Backup *[***name of backup***]* modal window, and then click OK to acknowledge that you want to restore the backup.



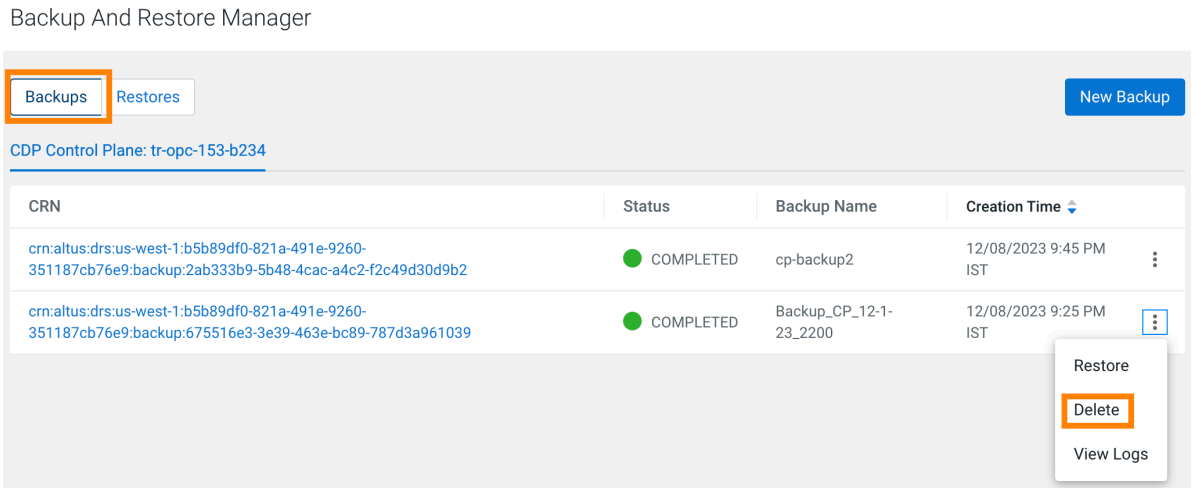
d) Go to the **Restores** tab to view the CRN for the restore event and other details about the restore event.

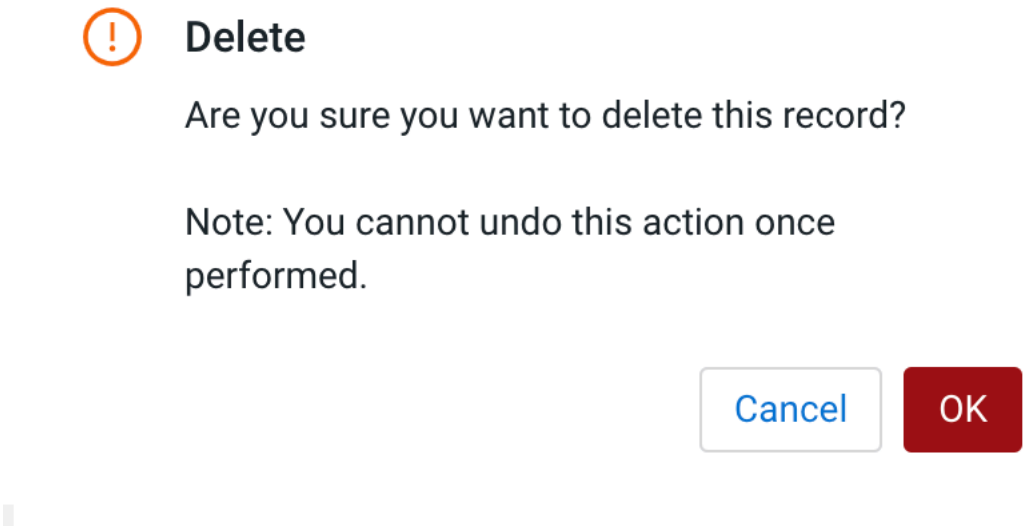


e) Click the CRN for a restore event to see its details on the **Restore Details** modal window.

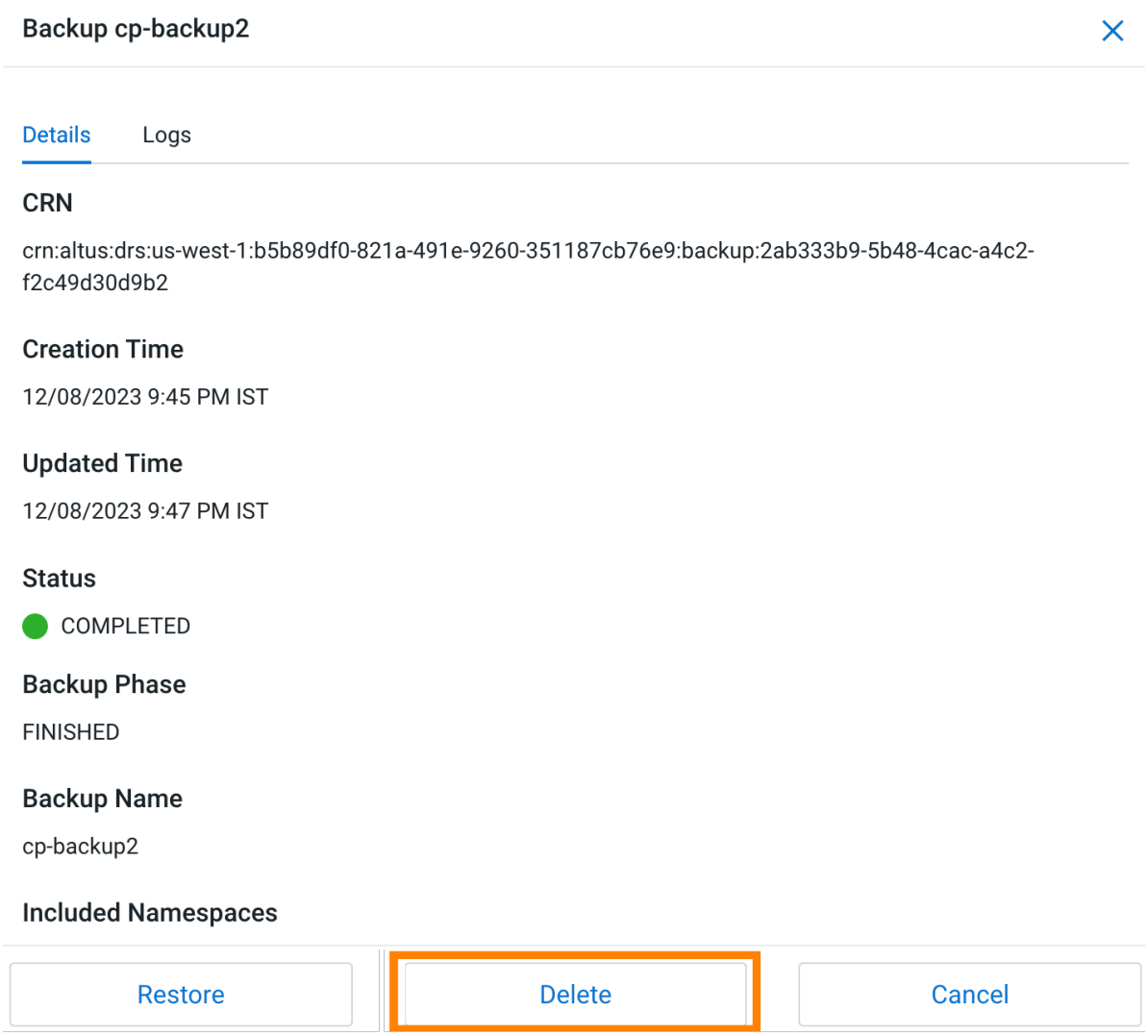

Backup And Restore Manager
Backups
Restores
CDP Control Plane: tr-opc-153-b234
CRN
crn:altus:drs:us-west-1:b5b89df0-821a-491e-926
351187cb76e9:restore:c19d8c1d-c47a-4fb5-845e
553947e0b86a
Restore Details
Details
Logs
CRN
crn:altus:drs:us-west-1:b5b89df0-821a-491e-9260-351187cb76e9:restore:c19d8c1d-c47a-4fb5-845e-553947e0b86a
Creation Time
12/08/2023 10:00 PM IST
Updated Time
12/08/2023 10:07 PM IST
Status
COMPLETED
Restore Phase
FINISHED
Associated Backup CRN
crn:altus:drs:us-west-1:b5b89df0-821a-491e-9260-351187cb76e9:backup:2ab333b9-5b48-4cac-a4c2-f2c49d30d9b2
Included Namespaces
tr-opc-153-b234-vault, tr-opc-153-b234

4. To delete a backup, perform the following steps:
   a) Go to the Backup and Restore Manager Backups tab.
   b) Click Actions Delete , and then click OK in the Delete modal window to acknowledge that you want to delete the backup.

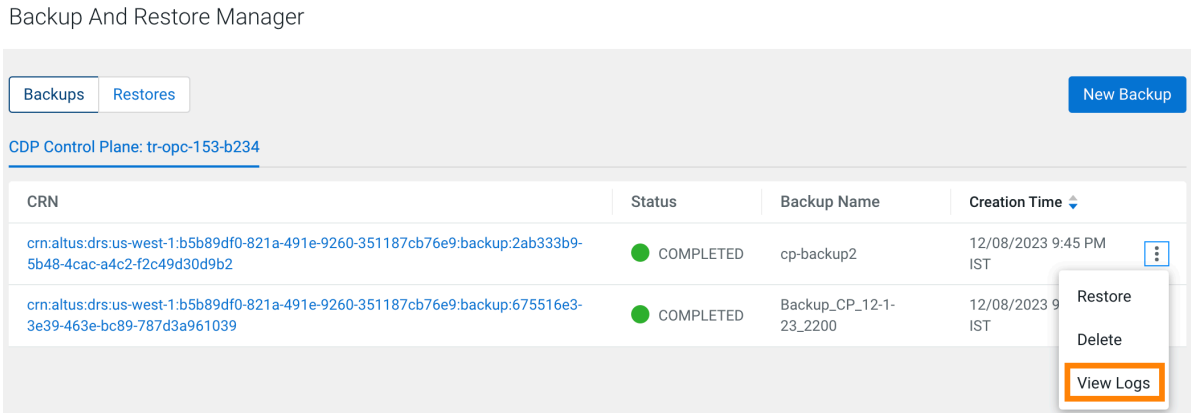




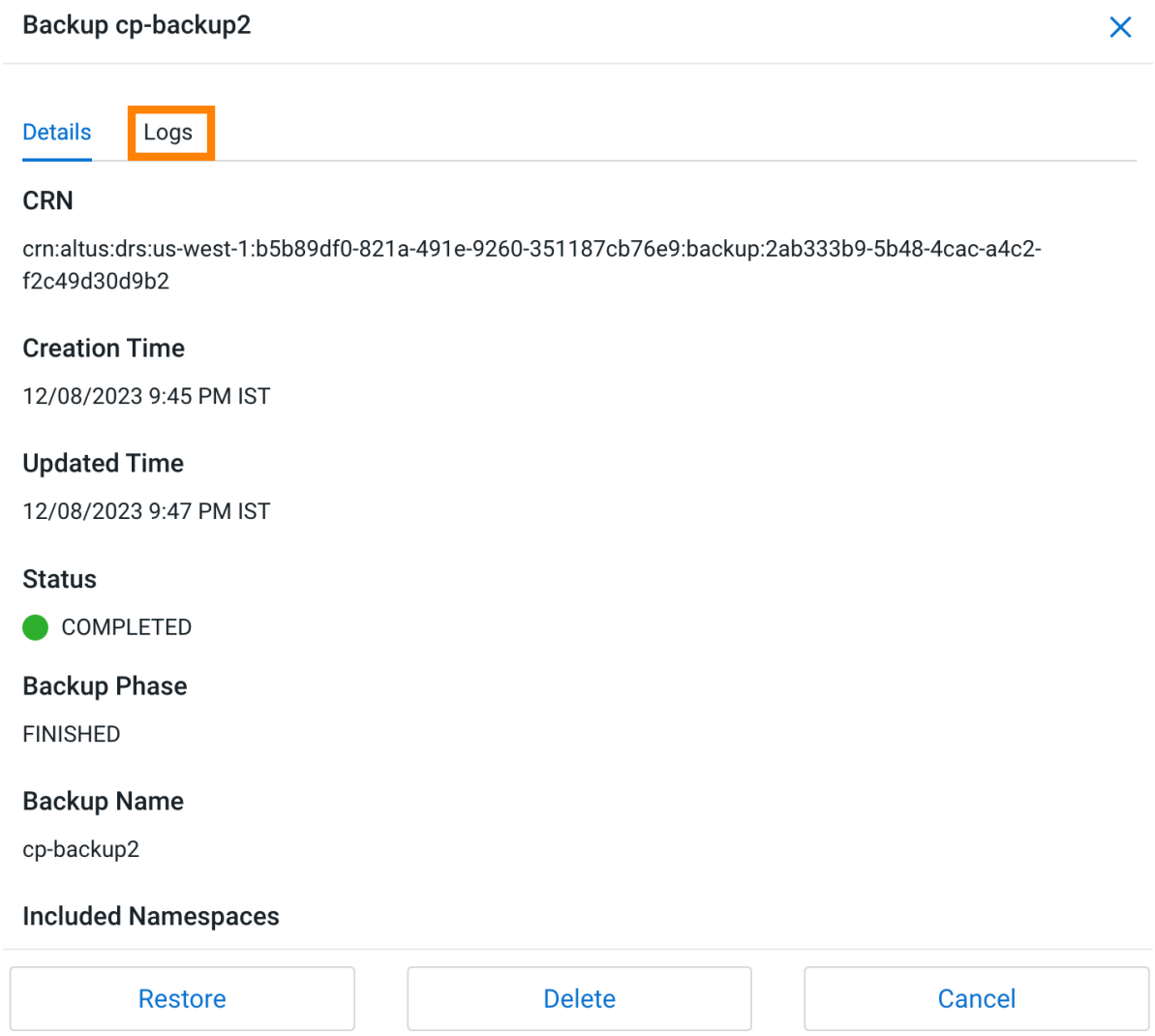
   c) Alternatively, click the CRN of the required backup. Click Delete on the Backup *[***name of backup***]* modal window, and then click OK to acknowledge that you want to delete the backup.


Backup cp-backup2
Details
Logs
CRN
crn:altus:drs:us-west-1:b5b89df0-821a-491e-9260-351187cb76e9:backup:2ab333b9-5b48-4cac-a4c2-f2c49d30d9b2
Creation Time
12/08/2023 9:45 PM IST
Updated Time
12/08/2023 9:47 PM IST
Status
COMPLETED
Backup Phase
FINISHED
Backup Name
cp-backup2
Included Namespaces
Restore
Delete
Cancel

5. To view the logs for a backup or restore event, perform the following steps:
   a) Go to the Backup and Restore Manager Backups tab.
   b) Click Actions Logs for the required backup.



   c) Click the Logs tab on the modal window.



   d) Alternatively, you can click the CRN for a backup event on the **Backups** tab, or click the CRN for a restore event on the **Restores** tab to open the modal window to view the logs for the event.

6. The following sample CDP CLI options show how to create a backup, restore or delete it, and monitor the progress of the events:
   a) Create a backup using the create-backup CDP CLI option.

      The following sample snippet creates a backup named *Backup 2*.

      ```
      cdp.sh --form-factor private --endpoint-url https://console-cp1.apps.sha
      red-os-01.kcloud.cloudera.com drscp create-backup --backup-name "Backup
       2"
      ```

   b) Track the progress of the current status of the specified backupCrn (backup event) using the describe-backup CDP CLI option.

      The following sample snippet output shows the current status of the *crn:altus:drs:us-west-1:18be-4c75-8c7f-f32e697dba4a:backup:91193c4f-45f0-949c-13e232f14c9e* backupCrn.

      ```
      cdp.sh --no-verify-tls --endpoint-url https://console-cdp.apps.drs31-1.v
      pc.cloudera.com --no-verify-tls --form-factor private drscp describe-
      backup --backup-crn crn:altus:drs:us-west-1:18be-4c75-8c7f-f32e697dba4
      a:backup:91193c4f-45f0-949c-13e232f14c9e
      ```

   c) List all the backups using the list-backup CDP CLI option.

      The following sample snippet output lists all the available backups.

      ```
      cdp.sh --no-verify-tls --endpoint-url https://console-cdp.apps.lh-lp1-1.
      vpc.cloudera.com --no-verify-tls --form-factor private drscp list-backup
      ```

   d) Restore a specific backup, using its CRN, with the restore-backup CDP CLI option.

      The following sample snippet restores the backup of *crn:altus:drs:us-west-1:88d84e3c-4c3e-9903-6c388a689690:backup:aebe-96d7-b79d10b64183* CRN.

      ```
      cdp.sh --form-factor private --no-verify-tls --endpoint-url https://cons
      ole-ocp1.apps.shared-os-01.kcloud.cloudera.com drscp restore-backup --b
      ackup-crn crn:altus:drs:us-west-1:88d84e3c-4c3e-9903-6c388a689690:backup
      :aebe-96d7-b79d10b64183
      ```

   e) Track the current status of the specified restoreCrn (restore event) using the describe-restore CDP CLI option.

      The following sample snippet output shows the current status of the *crn:altus:drs:us-west-1:a70c917a-4be8-927c-d36f3f7db2de:restore:c3b34532-4391-b62d-3f471fae5a40* restoreCrn:

      ```
      cdp.sh --form-factor private --no-verify-tls --endpoint-url https://cons
      ole-cp1.apps.shared-os-01.kcloud.cloudera.com drscp describe-restore --r
      estore-crn crn:altus:drs:us-west-1:a70c917a-4be8-927c-d36f3f7db2de:resto
      re:c3b34532-4391-b62d-3f471fae5a40
      ```

      For information about all the available CDP CLI options to backup and restore Control Plane and CDW, see CDP CLI options for Control Plane namespaces and CDP CLI options for Cloudera Data Warehouse (CDW).

      To set up a CDP client to run the CDP CLI commands, see CDP Private Cloud CLI.

Troubleshooting Backup and Restore Manager

The troubleshooting scenarios in this topic help you to troubleshoot issues that might appear for DRS in the Control Plane. The “Backup and Restore Manager” in CDP Private Cloud Data Services Management Console leverages the data recovery service capabilities to backup and restore Kubernetes namespaces and resources.

CDP Control Plane UI or the Backup and Restore Manager becomes inaccessible after a failed restore event?

Condition

What to do if the CDP Control Plane UI does not come up or the Backup and Restore Manager (or drscp options) becomes inaccessible after a failed restore event?

Cause

Sometimes, some configurations take more time to restore. For example, in a shared cluster (OCP) that is heavily loaded, the restore event might surpass the set timeout limit. In this scenario, you can either wait or rerun the restore event again.

Tip: Run the restore event for such scenarios during non-peak hours.

Solution

You can perform one of the following steps after a failed restore event:

- Wait for a minimum of 15 minutes. This might resolve the issue automatically if the issue was caused due to timeout. You can verify this in the logs.
- Run restore again. This might resolve the issue if it was temporary such as, restore event during cluster maintenance.

If the Control Plane is not restored successfully even after you follow the steps, contact Cloudera Support for further assistance.

Timeout error appears in Backup and Restore Manager?

Condition

What to do if a timeout error appears in the Backup and Restore Manager (or drscp options) during a restore event?

Solution

When the restore event crosses the time set in the POD_CREATION_TIMEOUT environment property of the cdp-release-thunderhead-drsprovider deployment in the drs namespace, a timeout error appears. By default, the property is set to 900 seconds. In this scenario, you must manually verify whether the pods are up or not.

Stale configurations in Cloudera Manager after a restore event?

Condition

Why are stale configurations in Cloudera Manager found after a restore event?

Cause

This scenario appears when you take a backup of the CDP Private Cloud Data Services Control Plane, upgrade Data Services, and then perform a restore. During the upgrade process, new parcels are activated and configurations in Cloudera Manager might have changed.

Solution

It is recommended that you restart Cloudera Manager after the upgrade process is complete and then initiate the restore event.

Timeout error during backup of OCP clusters

Condition

What to do when the “The execution of the sync command has timed out" error appears during a backup event for OCP clusters?

Cause

This scenario is observed when the cluster is heavily used and the backup event is initiated during peak hours.

Solution

You can restart the nodes, this causes the disk to unmount and forces the operating system to write any data in its cache to the disk. After the restart is complete, initiate another backup. If any warnings appear, scrutinize to verify whether there are any dire warnings, otherwise the generated backup is safe to use. The only drawback in this scenario is the downtime impact, that is the time taken to back up the OCP clusters is longer than usual. Therefore, it is recommended that you back up the clusters during non-peak hours.

If the sync errors continue to appear, contact your IT department to check whether there is an issue with the storage infrastructure which might be preventing the sync command from completing on time.

Managing certificates

Adjusting the expiration time of ECS cluster certificates

The RKE Kubernetes, Vault, and ECS webhook certificate expiration times are set to one year by default. To avoid certificate expiration errors, you may want to extend the expiration times.

About this task

Note:

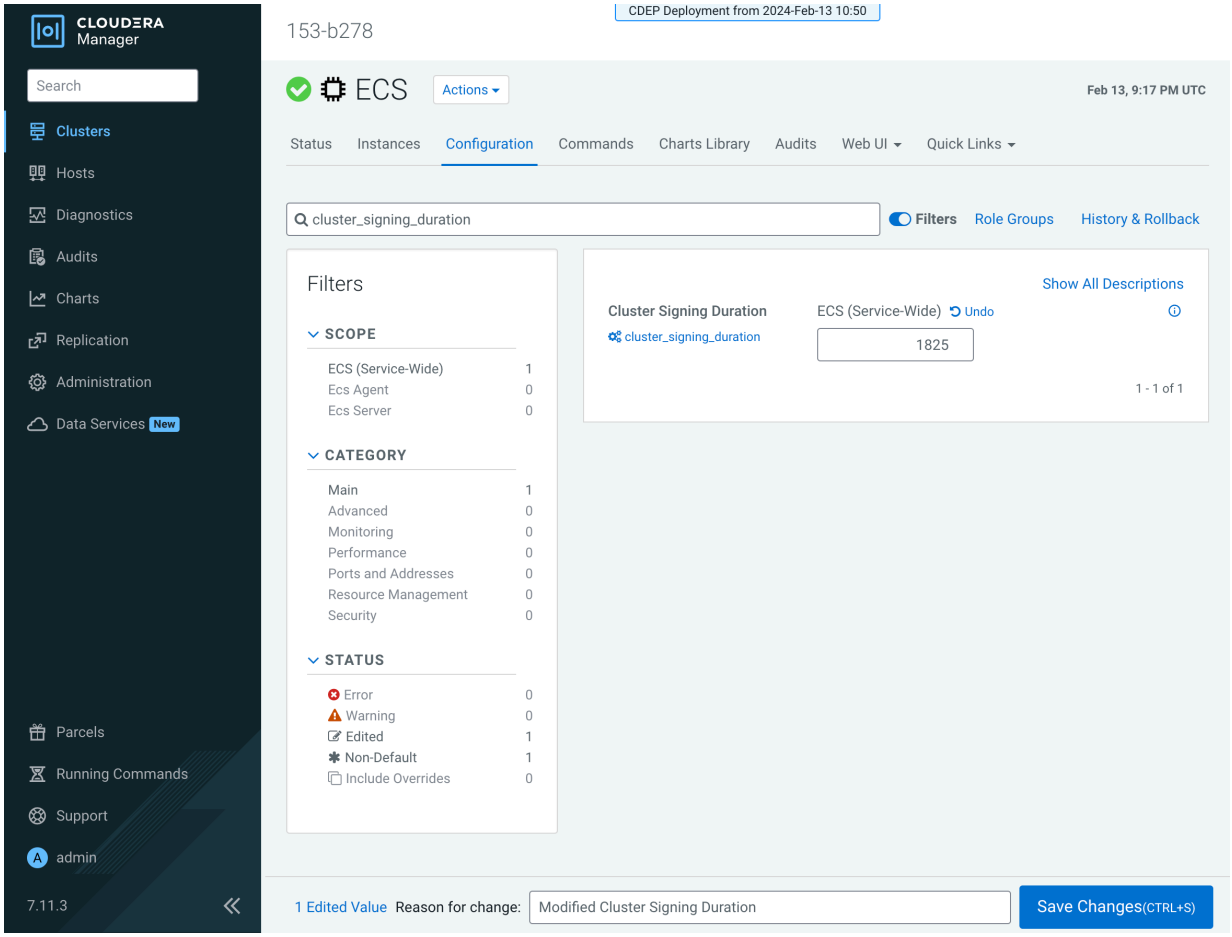
This topic only applies to internal certificates within ECS. It does not apply to the ingress controller certificate.

- These steps describe how to adjust the expiration time of internal cluster certificates in an existing ECS cluster.
- For a new cluster, if the nodes have been added to Cloudera Manager before creating the ECS cluster, you can edit the cluster_signing_duration configuration property in Cloudera Manager before creating the ECS cluster.

Adjusting the expiration time of the RKE Kubernetes cluster certificate

1. In Cloudera Manager, select the ECS cluster, then click ECS.
2. Click the Configuration tab, then use the Search box to locate the cluster_signing_duration configuration property.

3. The the cluster_signing_duration configuration property sets the expiration time for the RKE Kubernetes, Vault, and ECS webhook certificates, and is set to 1 year (365 days) by default. In the example below, the certificate expiration has been reset to 5 years (1825 days):



4. Click Save Changes.
5. On the ECS Cluster landing page, click Actions > Refresh Cluster.
6. After the Refresh is complete, click Actions > Rolling Restart.
7. After the restart is complete, the certificate expiration time is reset to the new value. You can also use the CLI to verify the new certificate expiration setting:

```
[root@host-1 ~]# cat /proc/47803/environ
CDH_PIG_HOME=/usr/lib/pigLD_LIBRARY_PATH=:/opt/cloudera/cm-agent/libCMF
_AGENT_ARGS=CDH_KAFKA_HOME=/usr/lib/kafka
CONF_DIR=/var/run/cloudera-scm-agent/process/1546342871-ecs-ECS_SERVERCDH_
PARQUET_HOME=/usr/lib/parquet
PARCELS_ROOT=/opt/cloudera/parcelsPARCEL_DIRNAMES=ECS-1.5.2-b866-ecs-1.5.2
-b866.p0.46395126LANG=en_US.UTF-8
CDH_HADOOP_BIN=/usr/bin/hadoopCDH_KMS_HOME=/usr/lib/hadoop-kmsCGROUP_GROUP
_CPU=CMF_PACKAGE_DIR=/opt/cloudera/cm-agent/service
ORACLE_HOME=/usr/share/oracle/instantclientMGMT_HOME=/opt/cloudera/cmINV
OCATION_ID=04c94a229a2b4684a95f8ec63783c81e
JSVC_HOME=/usr/libexec/bigtop-utilsCDH_IMPALA_HOME=/usr/lib/impalaKRB5_C
ONFIG=/etc/krb5.conf
CDH_YARN_HOME=/usr/lib/hadoop-yarnCLOUDERA_POSTGRESQL_JDBC_JAR=/opt/clo
udera/cm/lib/postgresql-42.5.1.jar
CDH_SOLR_HOME=/usr/lib/solrHIVE_DEFAULT_XML=/etc/hive/conf.dist/hive-defa
ult.xml
CLOUDERA_ORACLE_CONNECTOR_JAR=/usr/share/java/oracle-connector-java.jarC
GROUP_GROUP_BLKIO=system.slice/cloudera-scm-agent.service
```

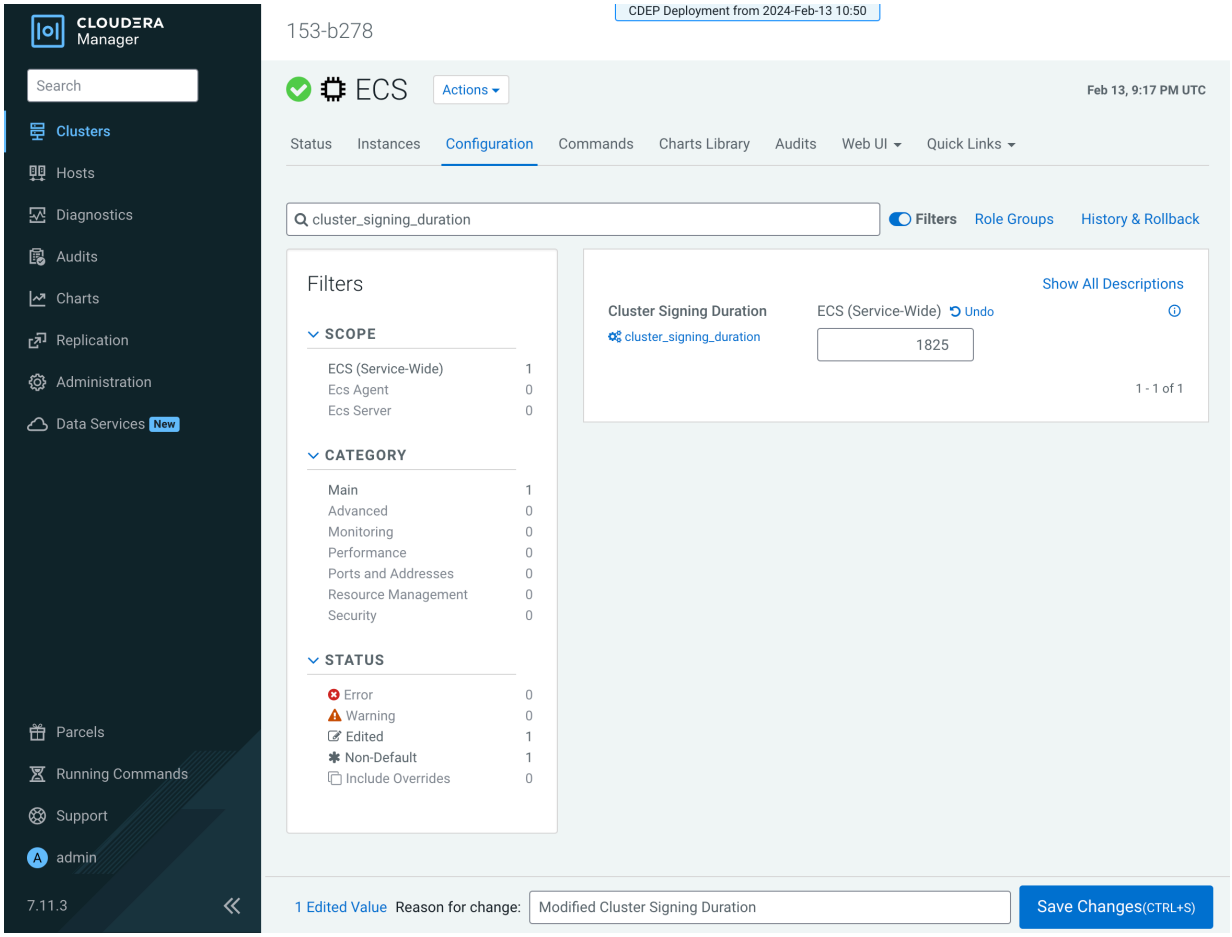
```
CGROUP_ROOT_BLKIO=/sys/fs/cgroup/blkioCGROUP_ROOT_CPU=/sys/fs/cgroup/cpu,c
puacctKEYTRUSTEE_KP_HOME=/usr/share/keytrustee-keyprovider
CLOUDERA_MYSQL_CONNECTOR_JAR=/usr/share/java/mysql-connector-java.jarCMF_
SERVER_ROOT=/opt/cloudera/cm
CGROUP_ROOT_CPUACCT=/sys/fs/cgroup/cpu,cpuacctCDH_FLUME_HOME=/usr/lib/f
lume-ng
CATTLE_NEW_SIGNED_CERT_EXPIRATION_DAYS=1825
<snip!>
```

```
[root@host-1 ~]# openssl x509 -in /var/lib/rancher/rke2/agent/serving-kubele
t.crt -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4005696761303552502 (0x379717fb376e51f6)
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN = rke2-server-ca@1697759349
        Validity
            Not Before: Oct 19 23:49:09 2023 GMT
            Not After : Oct 17 23:49:10 2028 GMT
        Subject: CN = host-1.rke-1019.kcloud.cloudera.com
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:92:81:74:b8:fb:aa:6c:c5:9a:40:2c:5f:91:60:
                    35:16:9a:d5:41:b2:bf:d8:29:f4:ed:68:ed:cd:3d:
                    87:0e:59:db:27:26:c5:d8:a7:79:c7:23:8f:0b:71:
                    c2:f5:d4:36:fe:97:a9:b5:62:ee:9d:9b:6d:ed:25:
                    60:fd:26:3a:08
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Authority Key Identifier:
                keyid:26:8F:9F:A1:04:CE:2D:04:3A:03:11:87:9D:DF:5A:B7:5C:0
6:72:32
            X509v3 Subject Alternative Name:
                DNS:host-1.rke-1019.kcloud.cloudera.com, DNS:localhost, IP
 Address:127.0.0.1, IP Address:10.17.130.15
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:fc:5c:89:ab:99:a6:79:33:a9:28:da:a8:47:
         52:cf:1f:43:13:8c:06:2e:23:67:4c:b4:b0:d6:e3:f9:b6:ad:
         50:02:21:00:c7:64:aa:86:97:5a:f3:12:7e:3f:a2:f1:ab:93:
         17:6c:3a:37:34:01:ef:ba:7f:08:85:70:2c:c9:40:e0:30:f5
```

Adjusting the expiration time of the Vault certificate

1. In Cloudera Manager, select the ECS cluster, then click ECS.
2. Click the Configuration tab, then use the Search box to locate the cluster_signing_duration configuration property.

3. The the cluster_signing_duration configuration property sets the expiration time for the RKE Kubernetes, Vault, and ECS webhook certificates, and is set to 1 year (365 days) by default. In the example below, the certificate expiration has been reset to 5 years (1825 days):



4. Click Save Changes.
5. Contact Cloudera support and ask them to provide you with a copy of the rotate-vault-cert.sh file.
6. Copy the rotate-vault-cert.sh file to the ECS master host. Set JAVA_HOME if needed.
7. Run the following command:

   ./rotate-vault-cert.sh APP_DOMAIN
8. Unseal Vault.
9. Restart all of the pods in the CDP namespace.
10. If you are using a default self-signed ingress controller certificate, update the ingress controller certificate (follow the steps in the script output).
11. You can use the CLI to verify the new certificate expiration setting:

```
root       49076   48970  2 16:49 ?        00:00:10 kube-controller-mana
ger
--flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --terminated-pod-
gc-threshold=1000 --permit-port-sharing=true
--allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/rancher/
rke2/server/cred/controller.kubeconfig
--authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/controller.
kubeconfig --bind-address=127.0.0.1
--cluster-cidr=10.42.0.0/16 --cluster-signing-duration=43800h
```

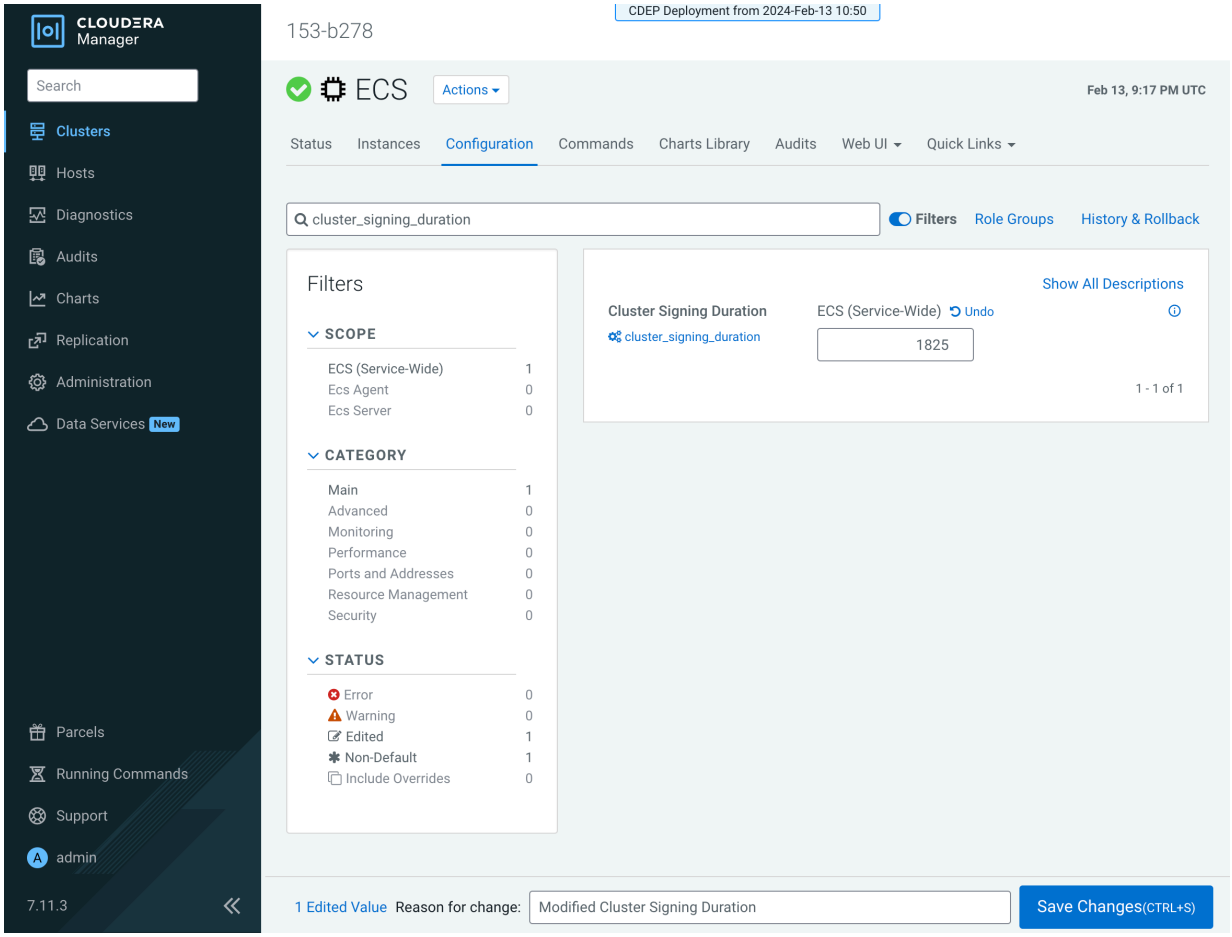
```
<snip!>
```

```
[root@host-1 ~]# openssl x509 -in vault.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            db:b7:a7:c3:79:86:4c:54:e8:97:49:bf:99:3d:df:a9
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN = rke2-server-ca@1697759349
        Validity
           Not Before: Oct 19 23:46:38 2023 GMT
            Not After : Oct 17 23:46:38 2028 GMT
        Subject: O = system:nodes, CN = "system:node:vault.vault-system.svc
;"
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:94:93:2e:9d:5c:01:5a:95:46:b2:9d:aa:23:c4:
                    4e:0f:92:07:7e:0e:3a:21:7d:ef:95:e8:09:d3:88:
                    38:ac:e9:9f:c2:36:37:04:56:43:87:3a:6f:34:08:
                    09:8f:3f:df:31:79:d6:12:db:78:f6:1c:9b:0e:c2:
                    d0:f5:25:50:86:37:d5:ff:f7:a0:82:6f:55:d1:ff:
                    03:54:f8:ce:8b:02:87:2d:af:3f:71:f8:c4:a9:f0:
                    24:50:7b:07:70:3d:7a:be:9d:41:f0:15:2f:56:c3:
                    d3:0d:1a:e1:87:8e:69:89:ff:bf:1b:f2:84:87:6c:
                    5e:f9:13:8b:2c:5c:de:64:9e:ae:de:6a:f0:7c:ae:
                    d9:01:41:aa:39:00:b3:2d:4f:5c:db:fb:2b:80:31:
                    88:b5:40:24:e1:06:08:c4:ad:82:70:a1:9e:4c:3e:
                    00:0d:61:d9:1a:5c:c7:11:a7:79:68:66:34:b2:c2:
                    e9:63:a8:5d:d1:13:be:e6:f1:8f:03:87:3d:be:eb:
                    b7:ce:a5:eb:56:81:37:5b:9d:ce:82:34:15:99:16:
                    4c:65:20:d9:df:e6:63:56:c2:49:79:e8:66:ce:c1:
                    01:9d:87:a2:ba:02:c0:7c:2b:e5:37:30:c5:23:bd:
                    87:a1:c8:2b:a9:49:be:67:31:22:8d:a4:68:f9:bd:
                    be:23
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Authority Key Identifier:
                keyid:26:8F:9F:A1:04:CE:2D:04:3A:03:11:87:9D:DF:5A:B7:5C:0
6:72:32
            X509v3 Subject Alternative Name:
                DNS:vault, DNS:vault.vault-system, DNS:vault.vault-system.
svc, DNS:vault.vault-system.svc.cluster.local, DNS:vault.localhost.localdoma
in, DNS:*.apps.host-1.rke-1019.kcloud.cloudera.com, IP Address:127.0.0.1
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:d9:5e:38:fc:31:9b:5a:eb:fc:7d:c2:8f:b3:
         54:5e:28:f0:8f:00:eb:36:65:9f:d3:70:ae:a2:79:77:ee:b5:
         f7:02:21:00:f4:e8:6f:c9:bd:bb:92:9d:63:81:69:55:67:8b:
         8a:f3:a4:5d:c1:67:66:b0:40:ff:22:a6:c3:6f:4f:8e:b2:8e
```

Adjusting the expiration time of the ECS webhook certificate

1. In Cloudera Manager, select the ECS cluster, then click ECS.
2. Click the Configuration tab, then use the Search box to locate the cluster_signing_duration configuration property.

3. The the cluster_signing_duration configuration property sets the expiration time for the RKE Kubernetes, Vault, and ECS webhook certificates, and is set to 1 year (365 days) by default. In the example below, the certificate expiration has been reset to 5 years (1825 days):



4. Click Save Changes.
5. Contact Cloudera support and ask them to provide you with a copy of the rotate-webhook-cert.sh file.
6. Copy the rotate-webhook-cert.sh file to the ECS master host.
7. Run the following command:

   ./rotate-webhook-cert.sh APP_DOMAIN
8. Check for any pods in the Pending state whose status shows that they cannot tolerate the node-role.kubernetes.io/control-plane toleration. Restart those pods.
9. You can use the CLI to verify the new certificate expiration setting:

```
root       49076   48970  2 16:49 ?        00:00:10 kube-controller-mana
ger
--flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --terminated-pod-
gc-threshold=1000 --permit-port-sharing=true
--allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/rancher/
rke2/server/cred/controller.kubeconfig
--authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/controller.
kubeconfig --bind-address=127.0.0.1
--cluster-cidr=10.42.0.0/16 --cluster-signing-duration=43800h
<snip!>
```

```
[root@host-1 ~]# openssl x509 -in ecs-tolerations-webhook-cert.pem -noout -t
ext
Certificate:
```

```
    Data:
        Version: 3 (0x2)
        Serial Number:
            a5:31:94:f4:84:bb:3b:a2:a4:63:8d:ec:de:b5:37:53
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN = rke2-server-ca@1697759349
        Validity
            Not Before: Oct 19 23:45:48 2023 GMT
            Not After : Oct 17 23:45:48 2028 GMT
        Subject: O = system:nodes, CN = "system:node:ecs-tolerations-webhook
.ecs-webhooks.svc;"
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:cc:12:e1:54:b8:aa:42:94:aa:11:a5:f7:35:0e:
                    0c:de:76:5b:d5:c6:c1:34:0b:b8:b7:2b:15:08:1d:
                    02:44:0f:2e:e1:17:dc:73:6a:e4:6c:df:5b:ac:43:
                    97:2e:34:73:f7:c9:6f:cf:c2:a8:52:79:b1:89:ea:
                    51:22:e1:41:b8:6a:ba:fd:22:a2:bf:a2:46:a4:8e:
                    f5:c6:2d:05:c3:a5:1d:6b:60:da:e8:40:a5:e1:e1:
                    5a:55:0e:94:2d:91:dd:71:d1:e9:aa:27:5d:e6:fc:
                    ea:5f:ea:c6:8e:52:71:27:ce:c2:a7:1b:10:ca:db:
                    db:27:c8:46:6d:14:d1:d0:b3:f5:ab:74:a9:63:8b:
                    71:83:31:eb:ad:87:1b:3b:8d:ff:ce:d0:7f:d1:1b:
```