CDP Private Cloud Data Services 1.5.3

Managing the Embedded Container Service (ECS)

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

https://docs.cloudera.com/

Legal Notice

Contents

The Embedded Container Service (ECS)

Cloudera Manager provides tools for managing and monitoring the CDP Private Cloud Embedded Container Service.

The Embedded Container Service (ECS) service enables you to run CDP Private Cloud Data Services by creating container-based clusters in your data center. In addition to the option to use OpenShift, which requires that you deploy and manage the Kubernetes infrastructure, you can also deploy a Embedded Container Service cluster, which creates and manages an embedded Kubernetes infrastructure for use with CDP Private Cloud Data Services. Installing, configuring, and managing OpenShift is not required. You only need to provide hosts on which to install the service and Cloudera Manager sets up the Embedded Container Service cluster and also provides management and monitoring of the cluster.

When you create an Embedded Container Service cluster, two new services are added to the cluster:

- Embedded Container Service (ECS) service. The ECS service has two roles:
  - ECS Server -- runs on a single host in the Embedded Container Service cluster.
  - ECS Agent -- runs on all hosts except the host running the Server role in the Embedded Container Service Cluster.
- Docker service. The Docker service has a single role:
  - Docker Server -- runs on all hosts in the Embedded Container Service Cluster.

Configuring the Embedded Container Service

You use Cloudera Manager to configure the Embedded Container Service and clusters.

Procedure

1. Open the Cloudera Manager Admin Console
2. From the Home page, Click on the Embedded Container Service Cluster
3. Click the Hosts, ECS service, or the Docker service links.
4. Click the Configuration tab.
5. Use the Filters or Search functions to locate the configuration property you are looking for.
6. Enter your change.
7. Click Save Changes.

Related Information

Modifying Configuration Properties Using Cloudera Manager

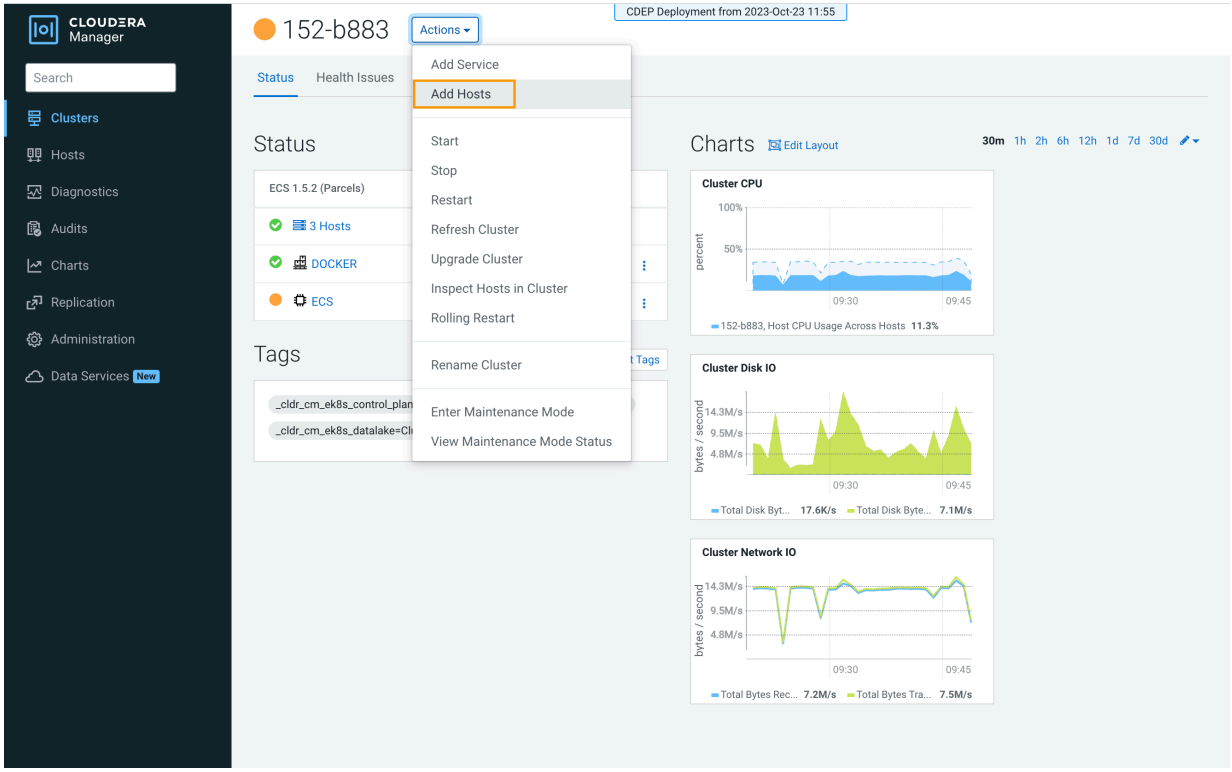
Adding hosts to a Embedded Container Service Cluster

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

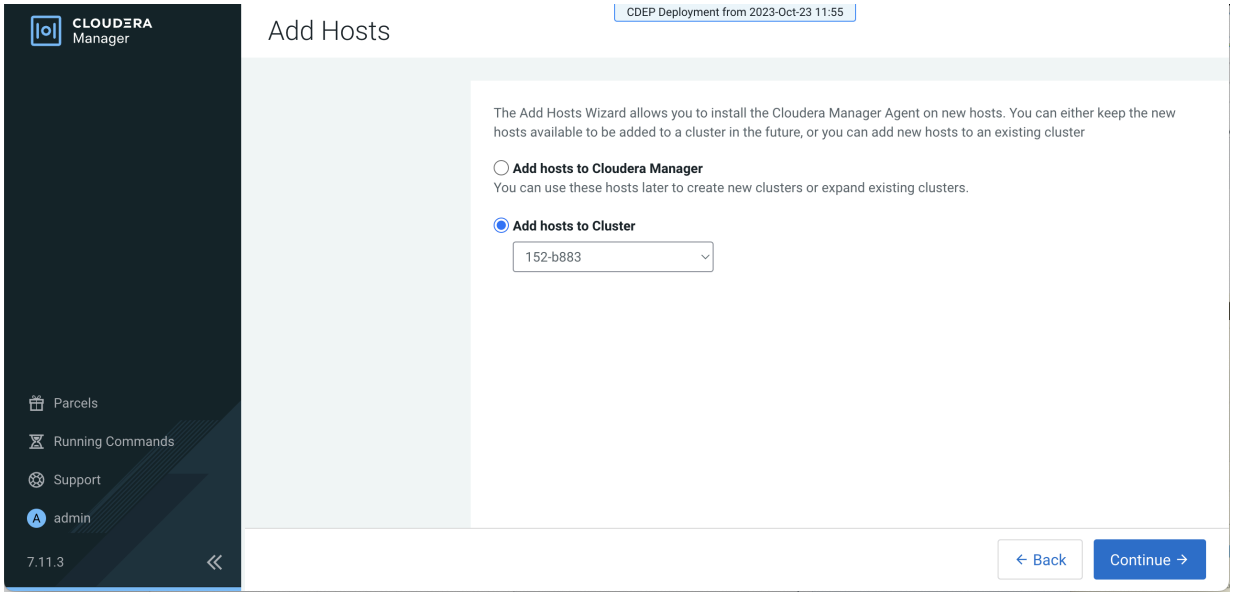
About this task

Procedure

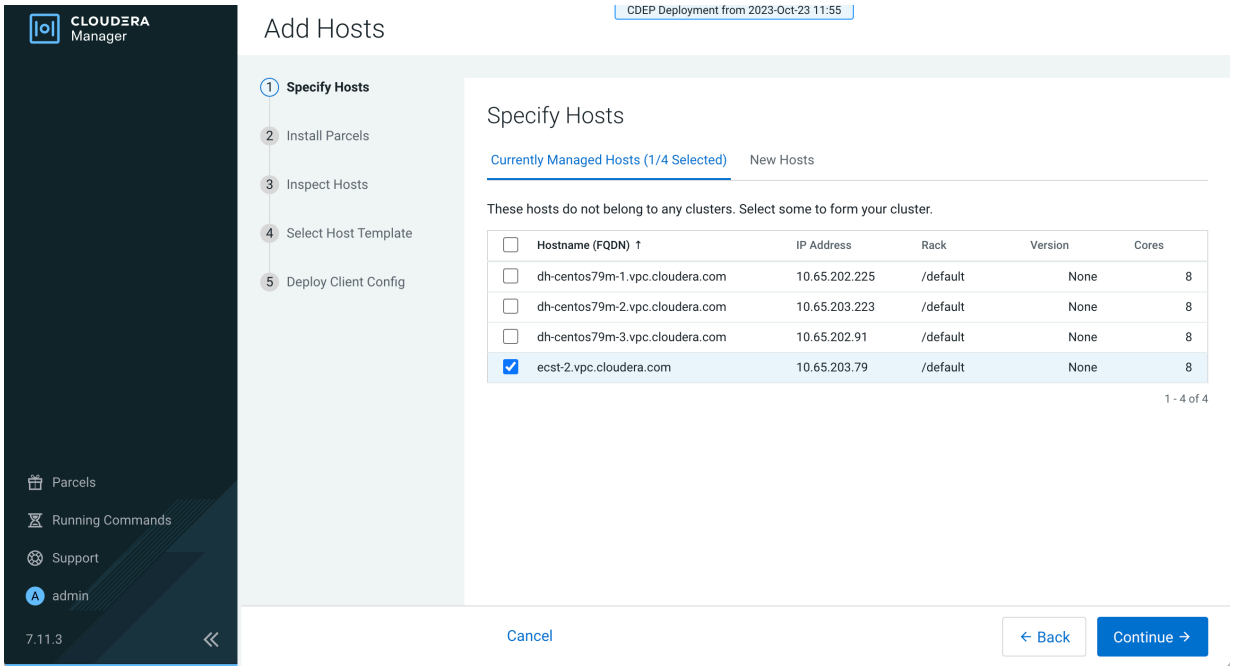
1. On the Cloudera Manager home page, click the ECS Cluster, then select Actions > Add Hosts.



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

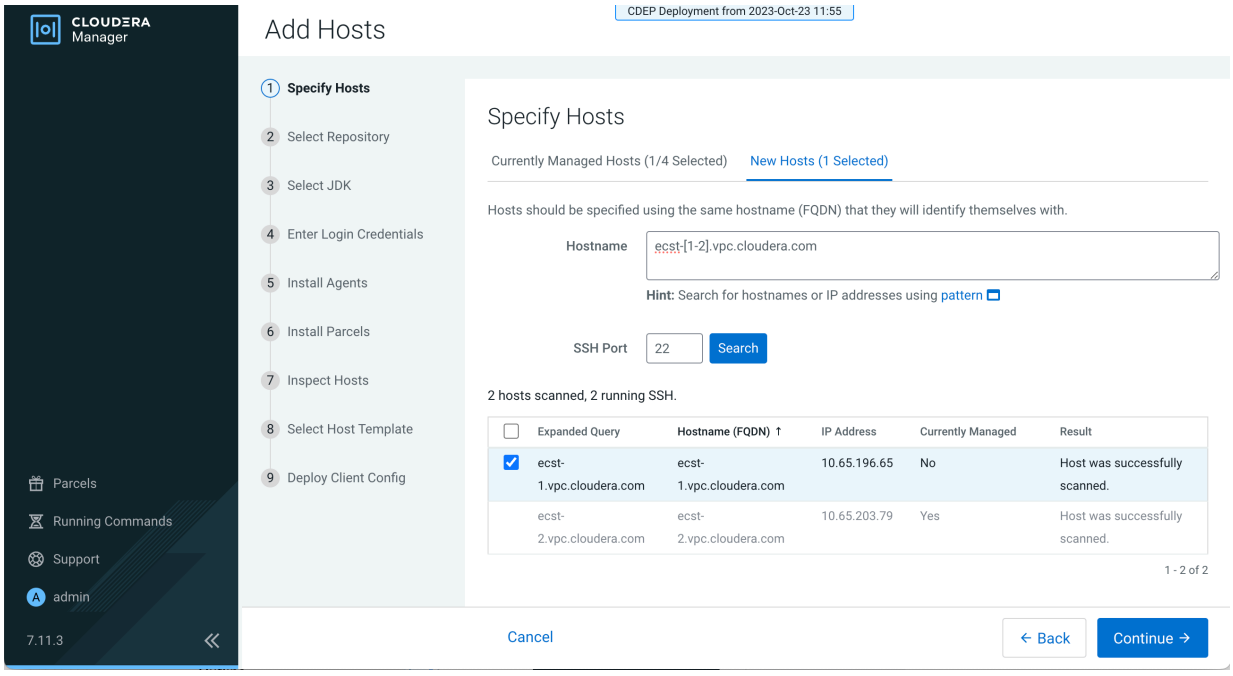


3. On the Specify Hosts page, hosts that have already been added to Cloudera Manager are listed on the Currently Managed Hosts tab. You can select one or more of these hosts to add to the ECS cluster.



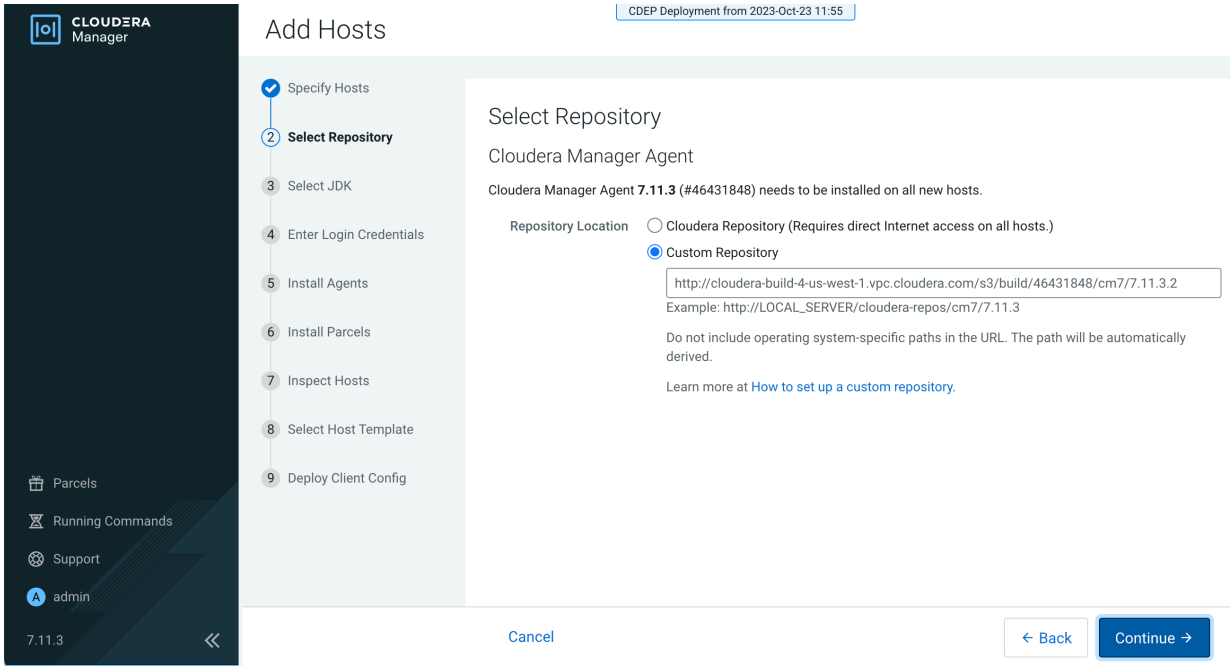
You can also click the New Hosts tab to specify one or more hosts that have not been added to Cloudera Manager. Enter a Fully Qualified Domain Name in the Hostname box, then click Search.

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

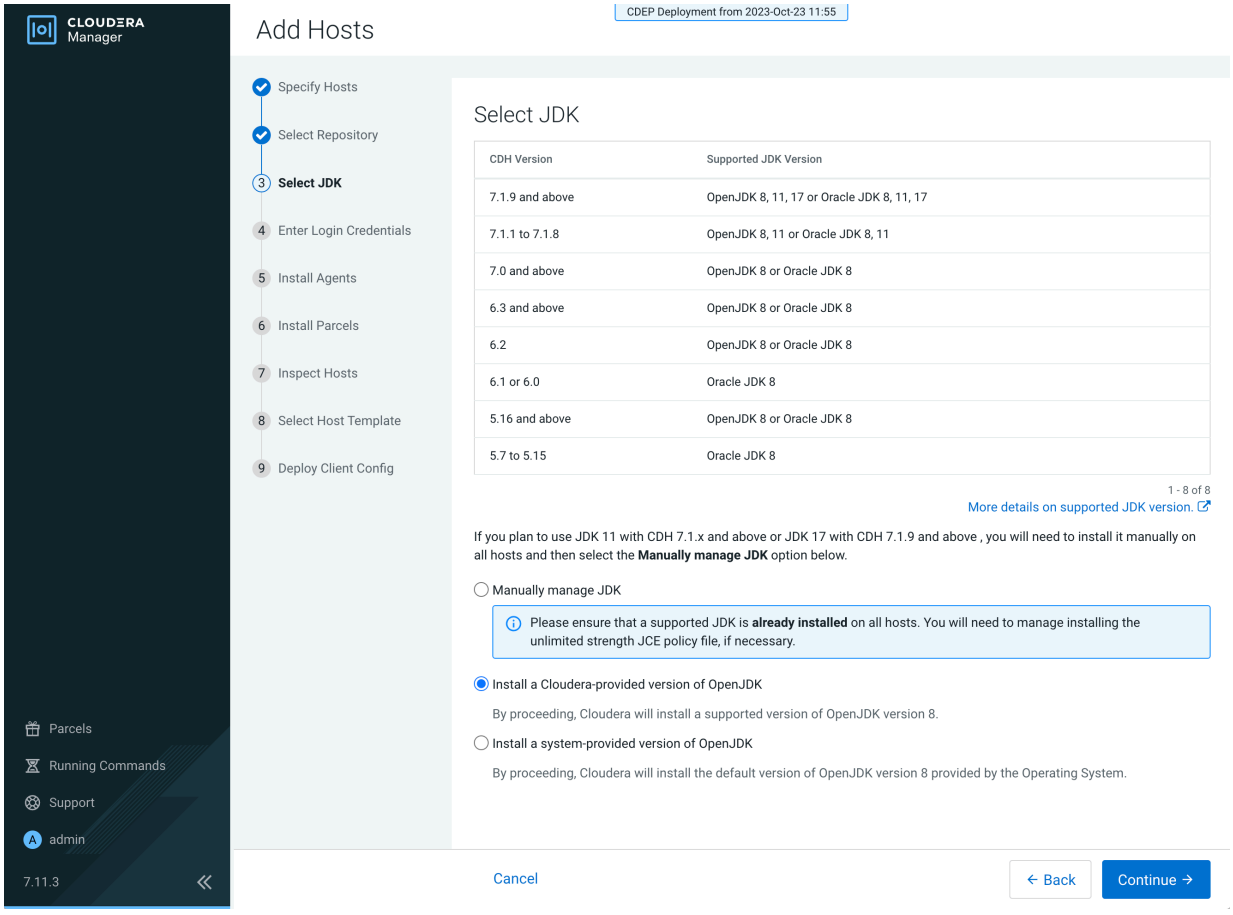


After you have finished specifying the ECS hosts, click Continue.

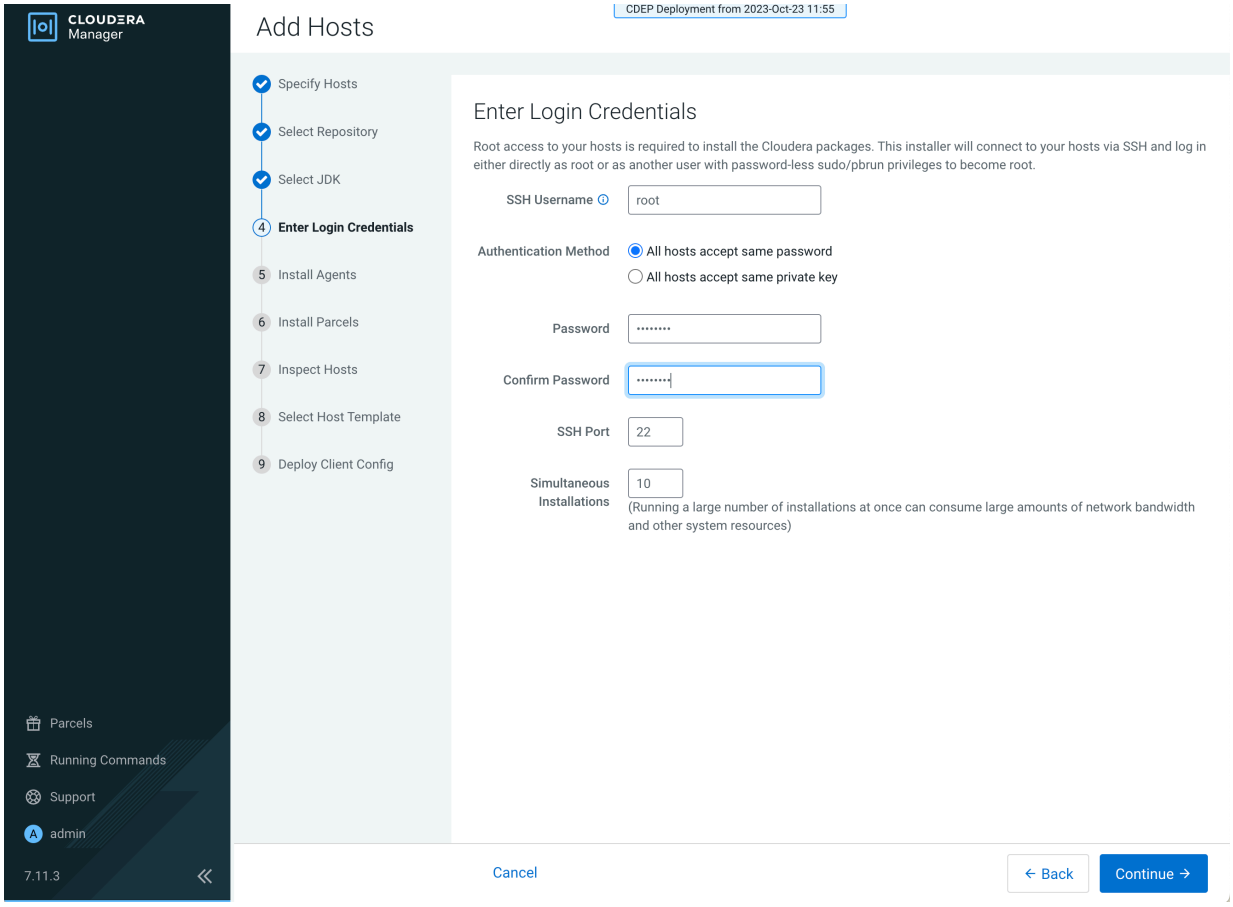
4. On the Select Repository page, the applicable Cloudera Manager Agent repository location is selected by default. Click Continue.



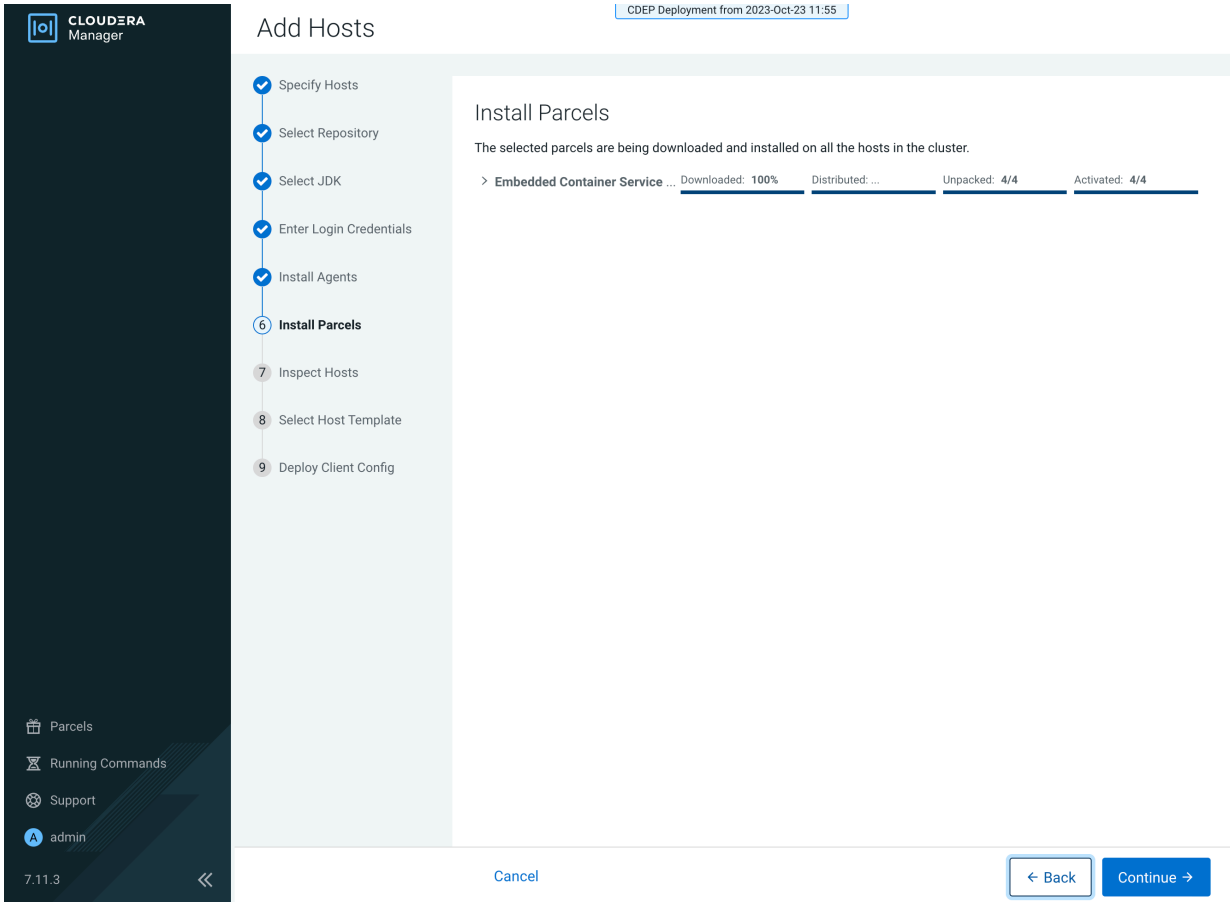
5. Select a JDK option on the Select JDK page, then click Continue.



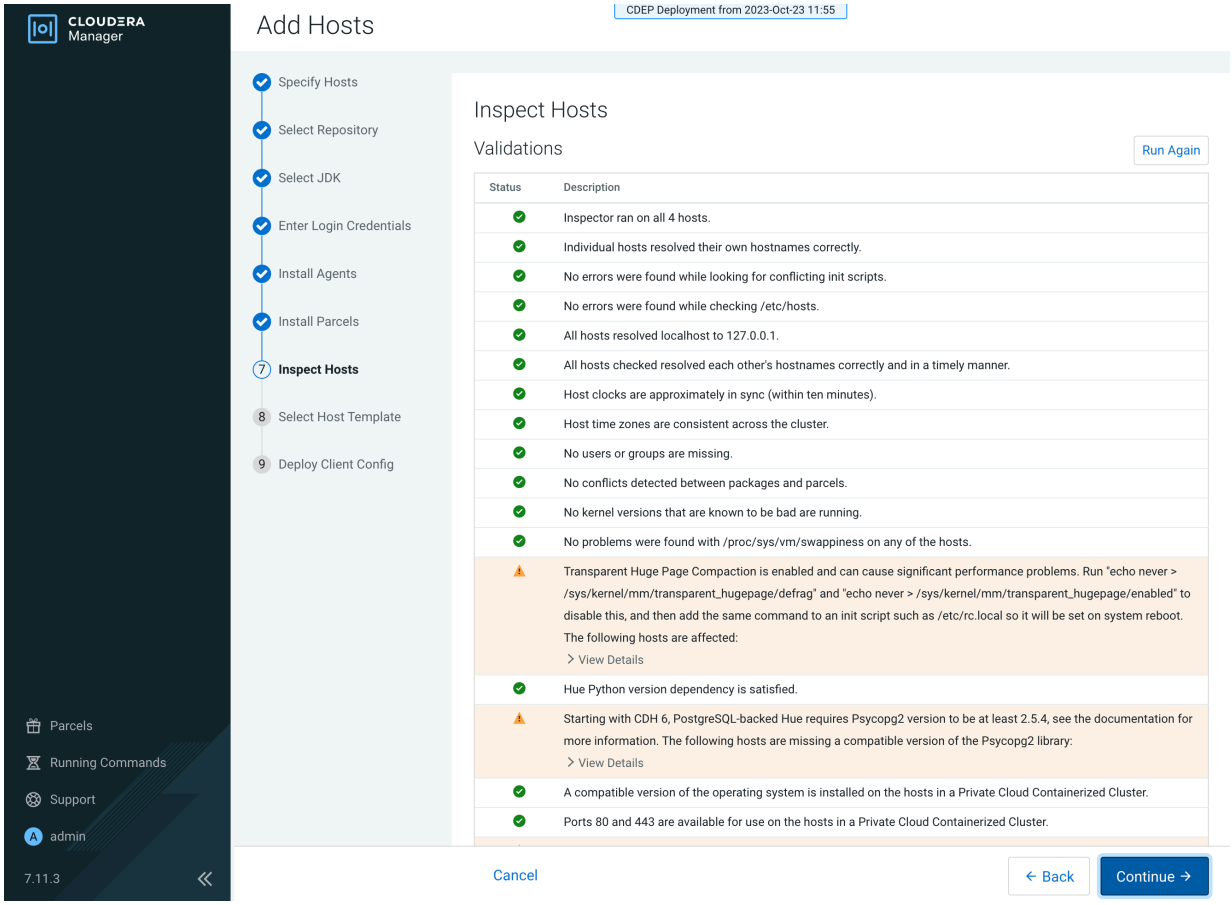
6. On the Enter Login Credentials page, All hosts accept the same password is selected by default. Enter the user name in the SSH Username box, and type in and confirm the password. You can also select the All hosts accept the same private key option and provide the Private Key and passphrase.



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.



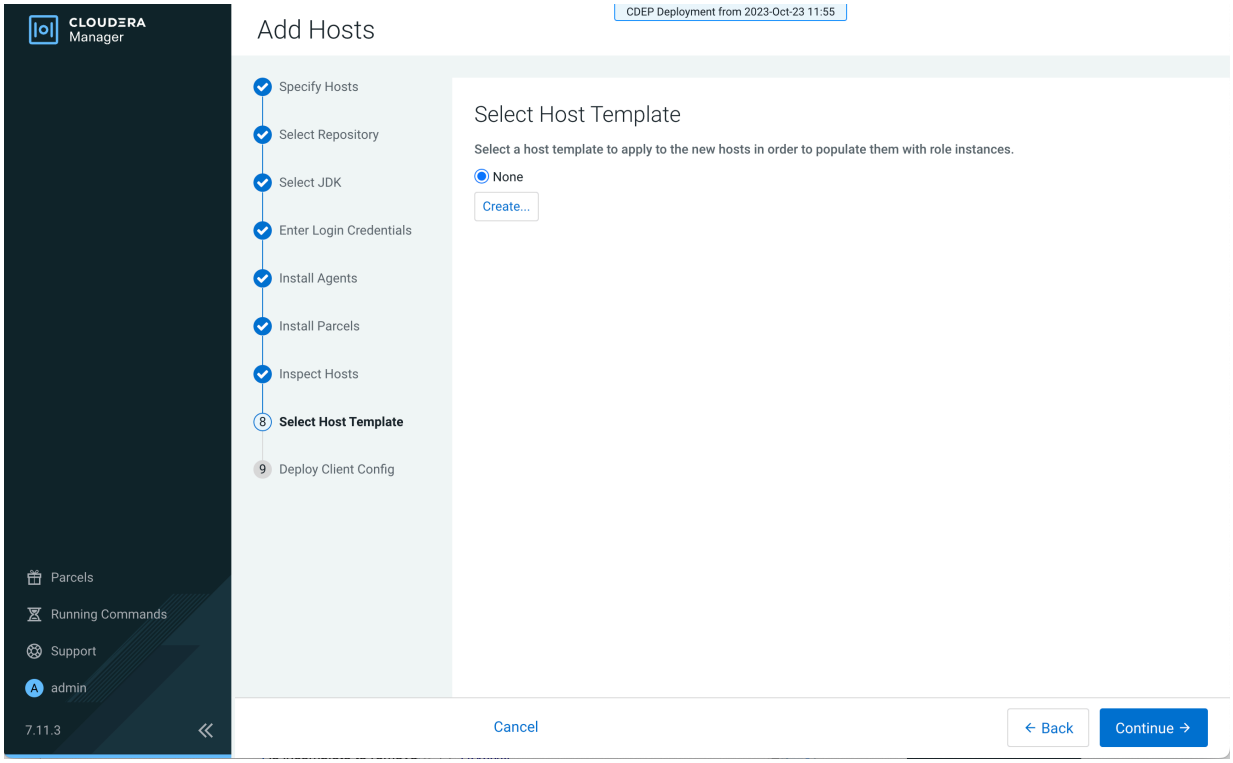
8. Review the Validations list on the Inspect Hosts page. If issues are detected, you can fix the issues, then click Run Again to repeat the host inspection. Click Continue.



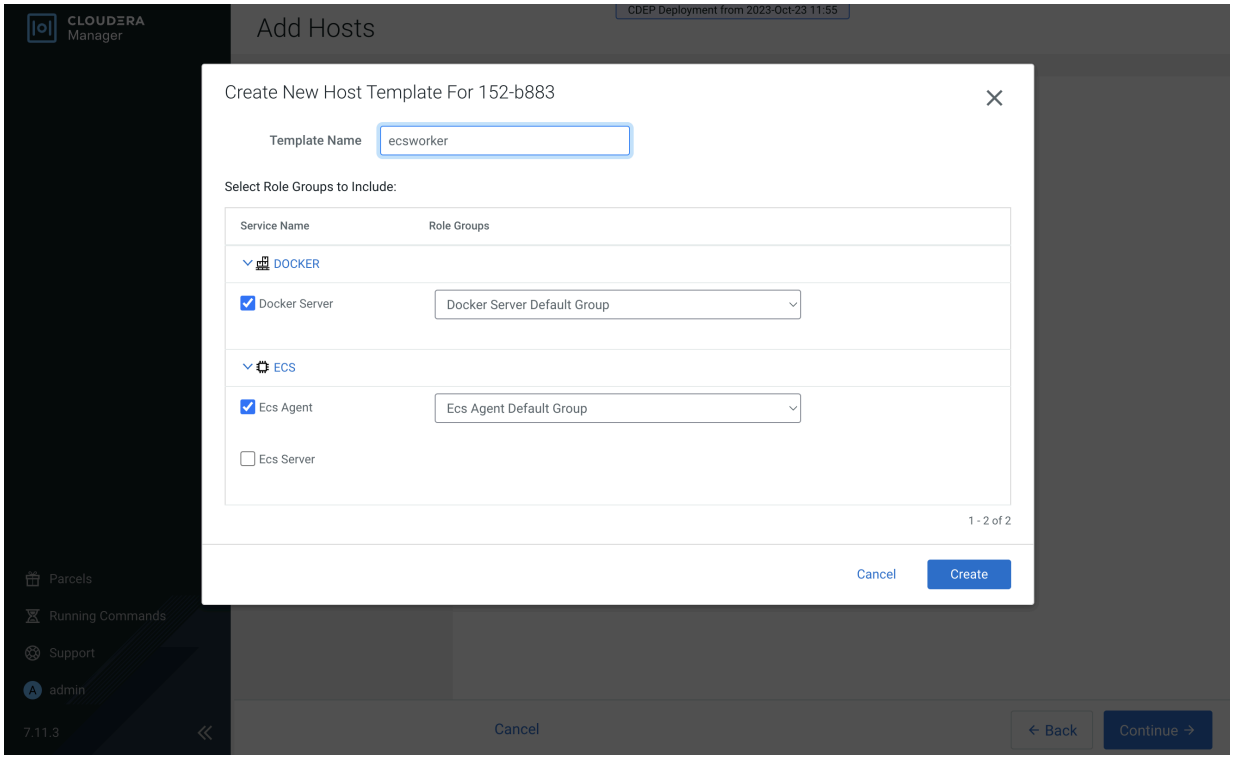
9. The Select Host Template page lists available host templates. Click Create.

Note:

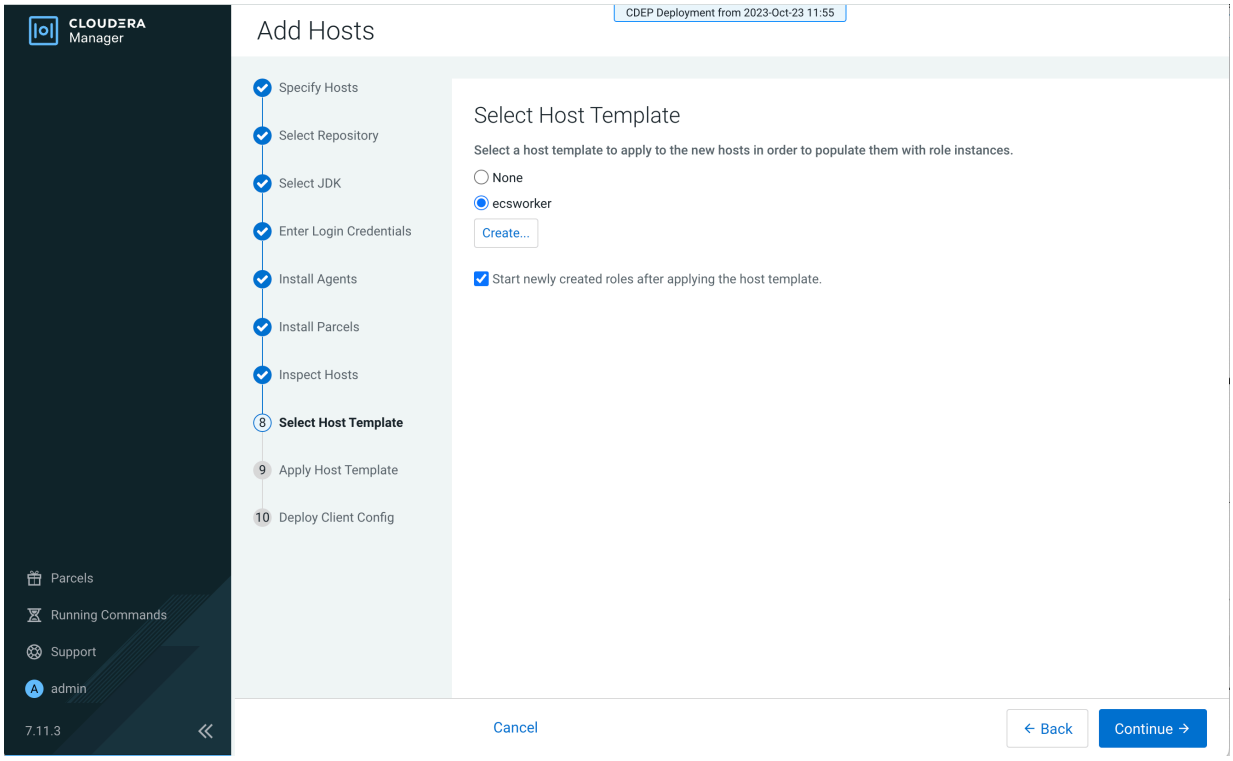
The following three steps describe how to create a host template to assign the Docker Server and Ecs Agent role groups to the new host. You can also select None and add these role instances after adding the new host to the cluster, as described at the end of this topic.



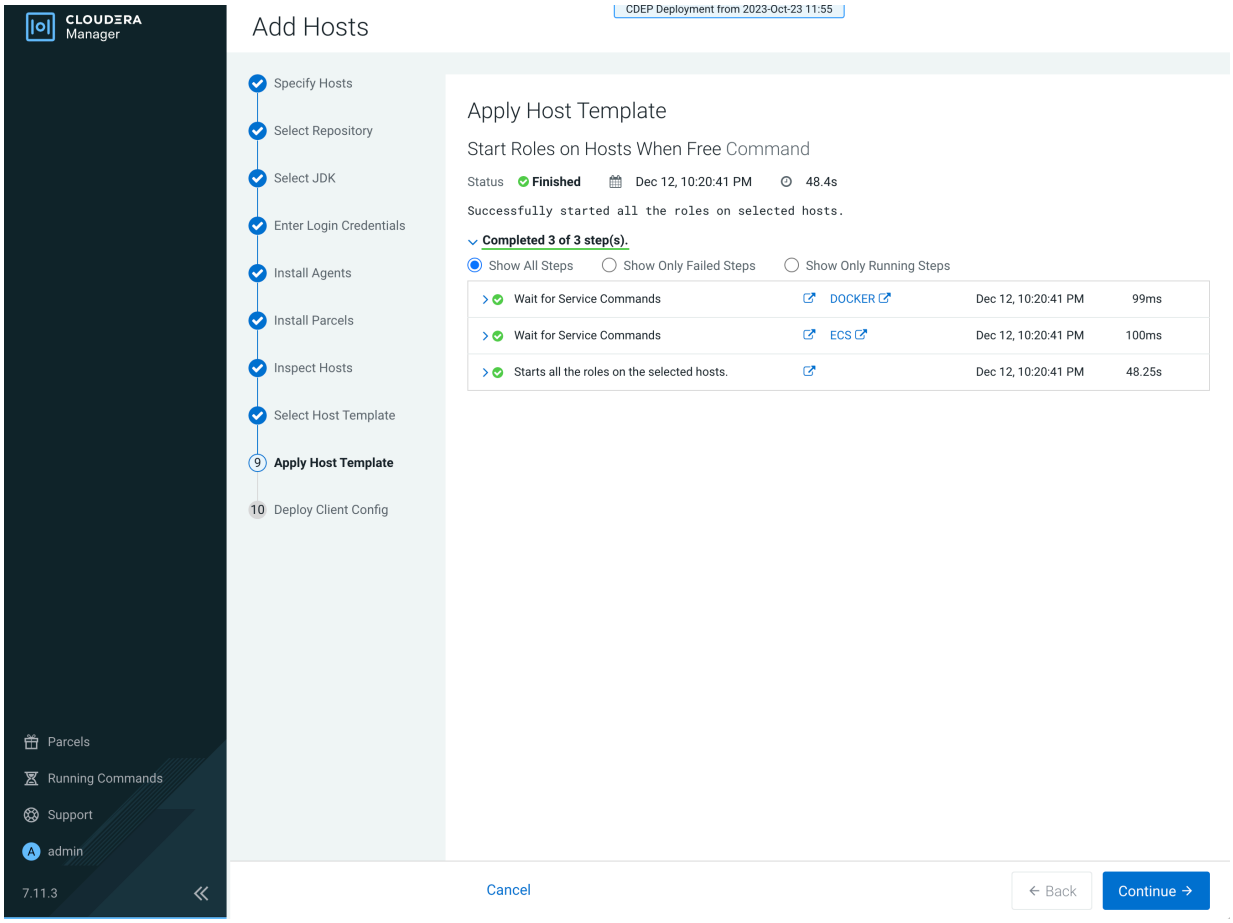
10. On the Create New Host Template pop-up, enter a template name and select the Docker Server and Ecs Agent role groups, then click Create.



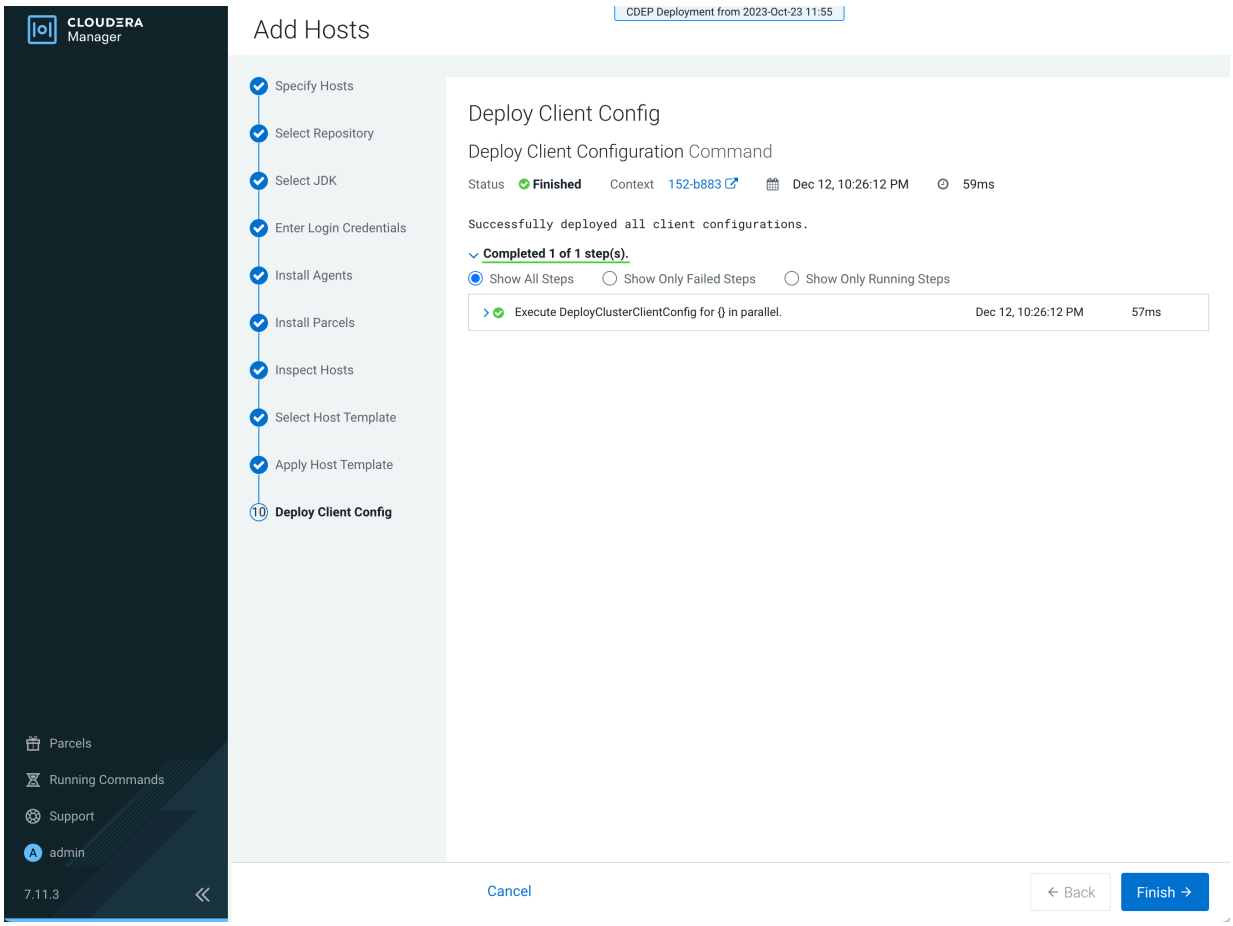
11. On the Select Host Template page, select the new template, then click Continue.



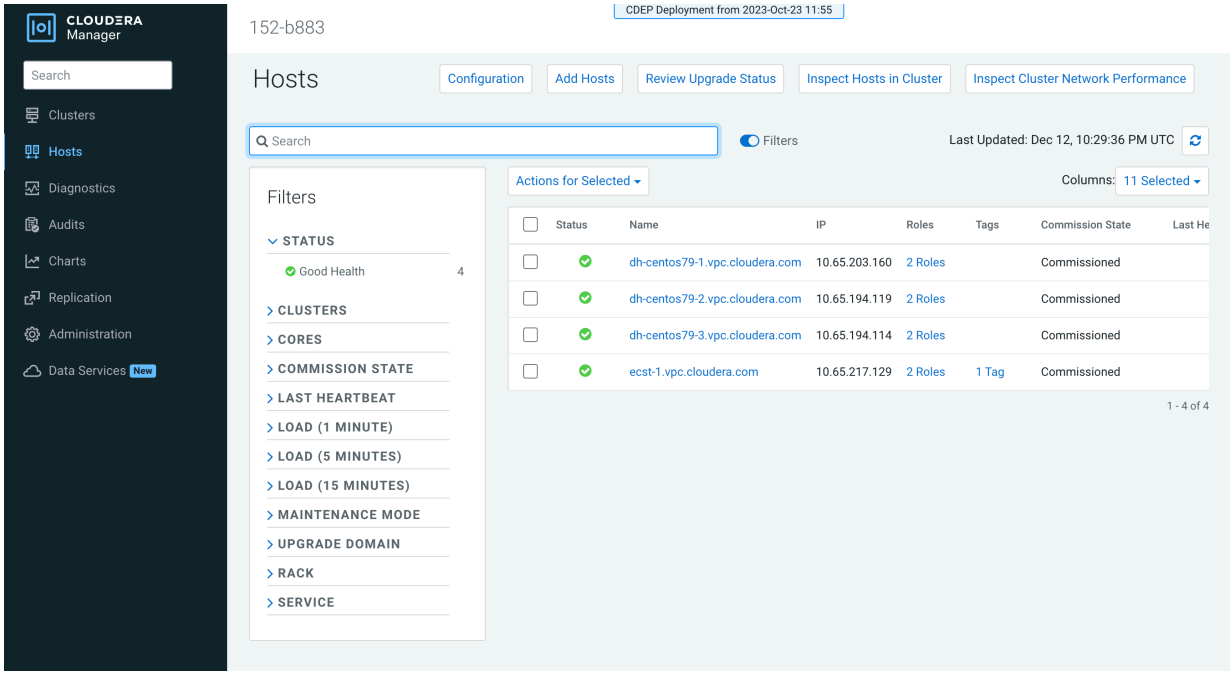
12. The Apply Host Template page appears. After the roles have successfully started, click Continue.



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
```

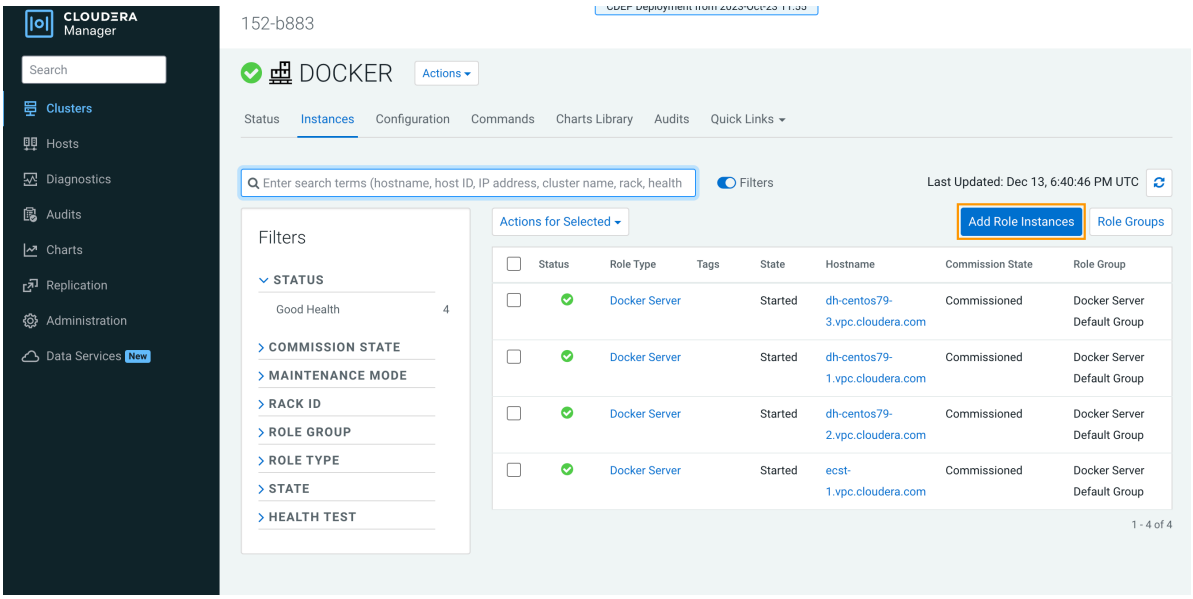
For RHEL 7.9, run the following command on each ECS host:

```
yum install -y iptables
```

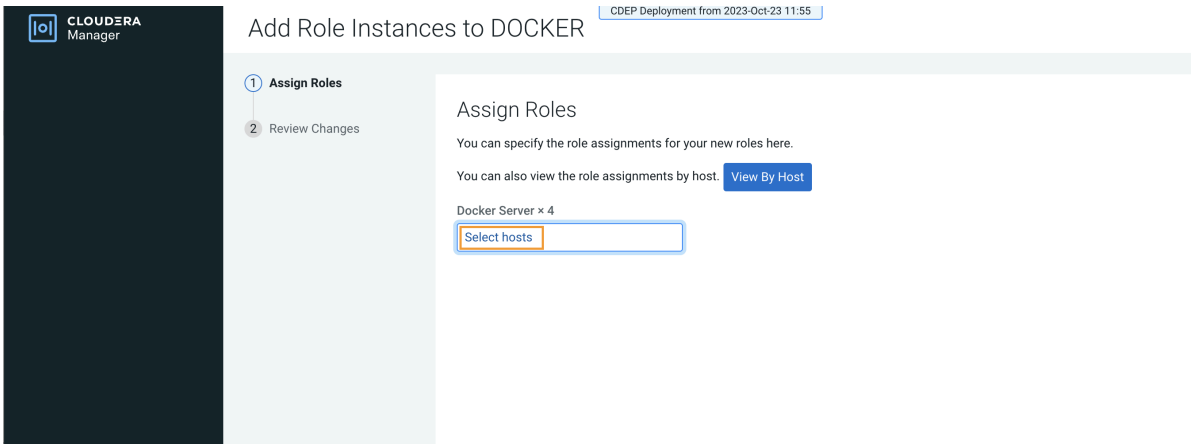
16. If you did not apply a host template to assign roles, perform the following steps to assign the Docker Server and Ecs Agent role groups to the new host.

To assign the Docker Server role group:

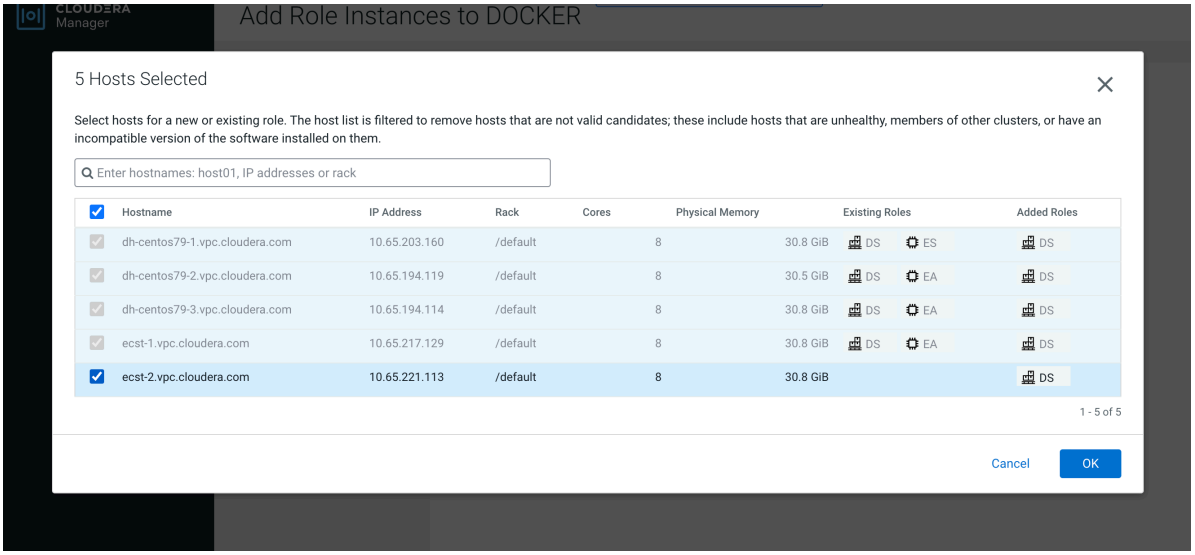
a. Click DOCKER on the ECS cluster home page, select Instances, then click Add Role Instances.



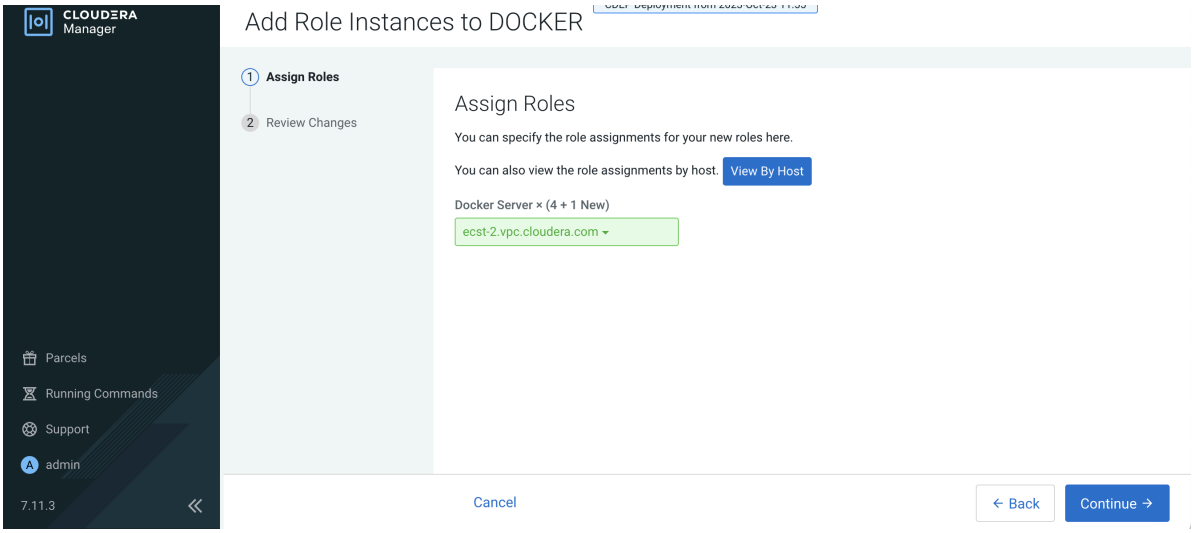
b. On the Add Role Instances to DOCKER page, click Select hosts.



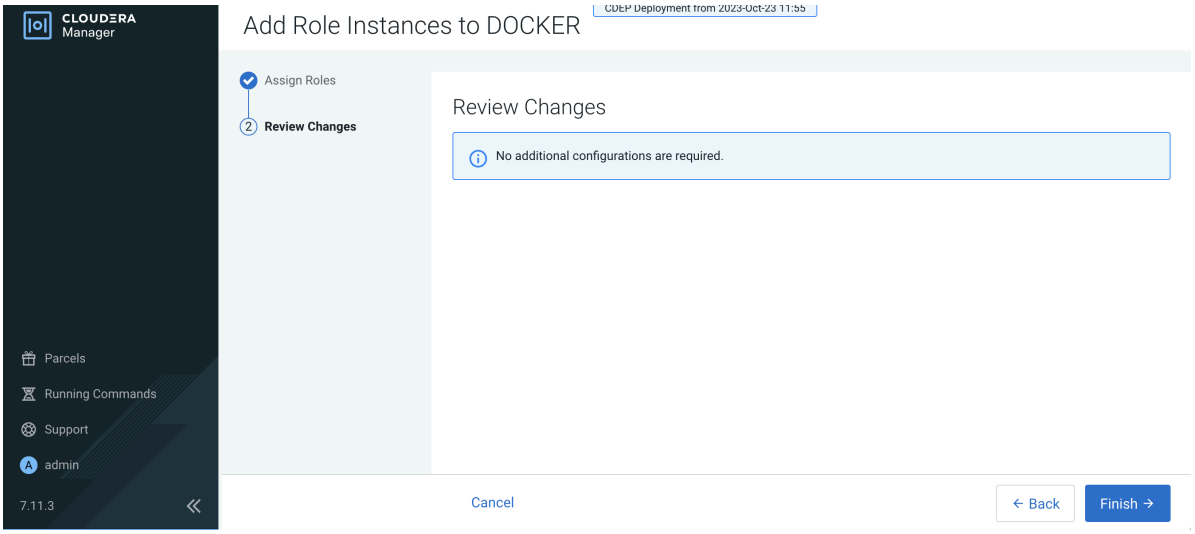
c. On the Hosts Selected pop-up, select the new host, then click OK.



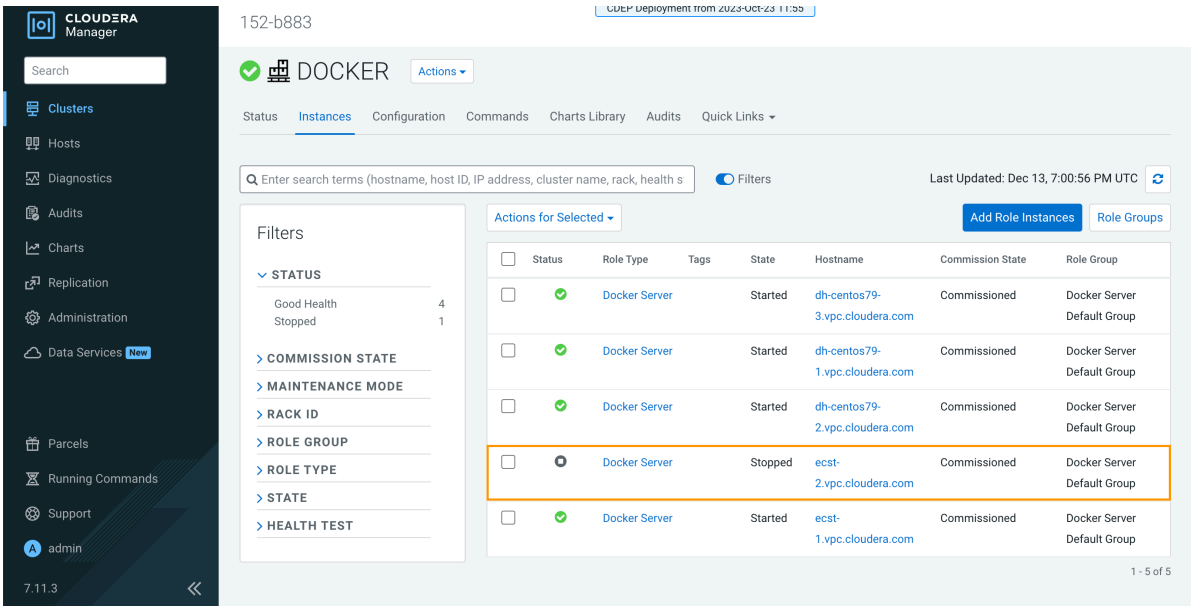
d. On the Assign Roles page, click Continue.



e. On the Review Changes page, click Finish.

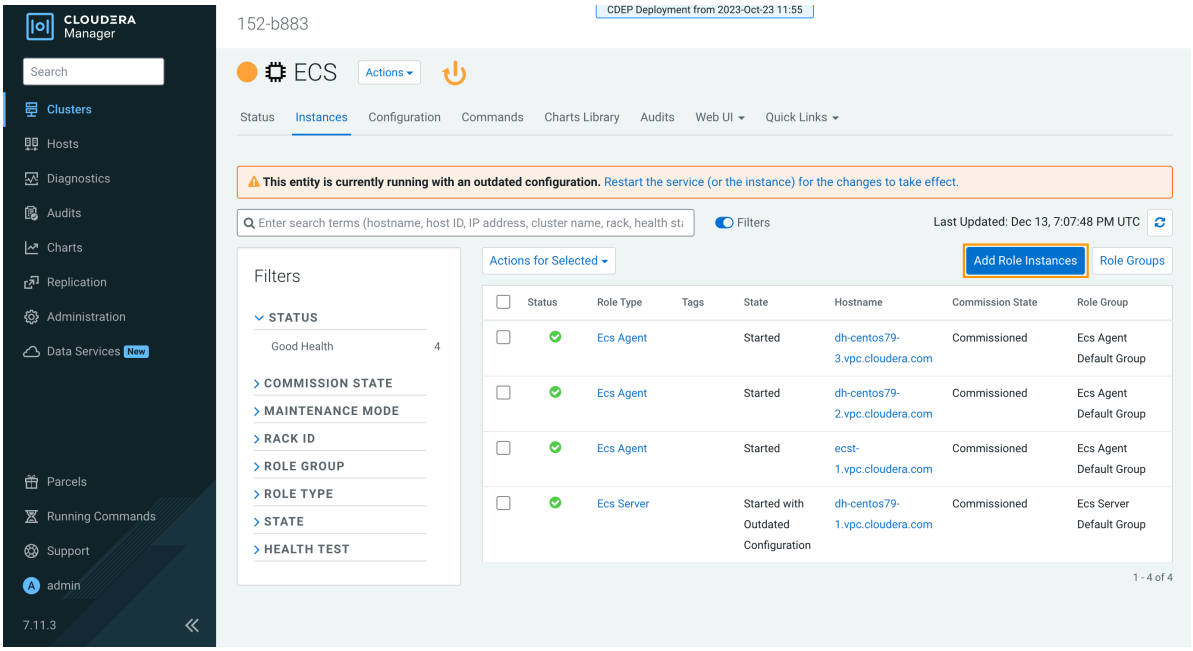


f. The new host is listed on the Docker Instances page.



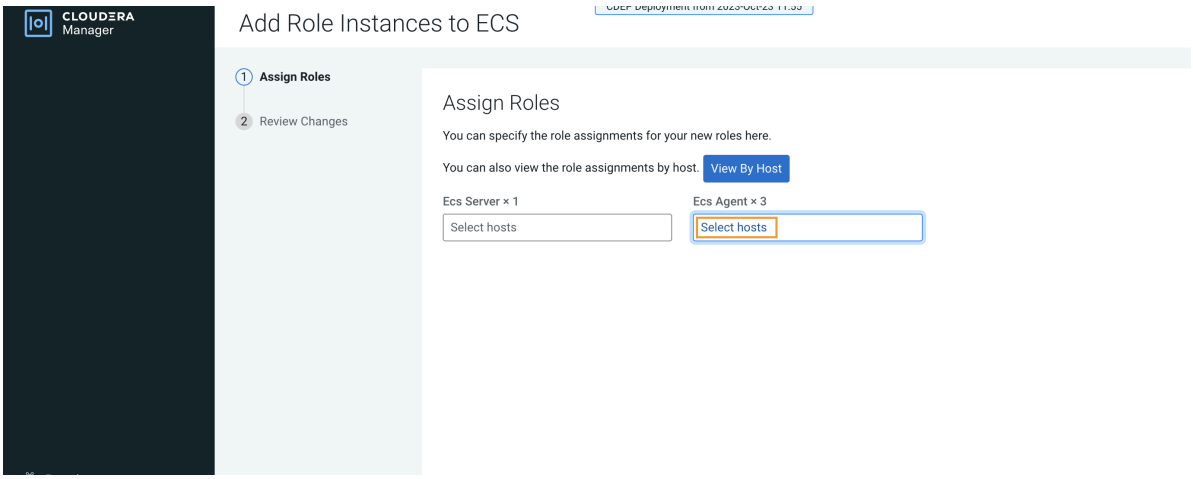
To assign the ECS Agent role group:

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

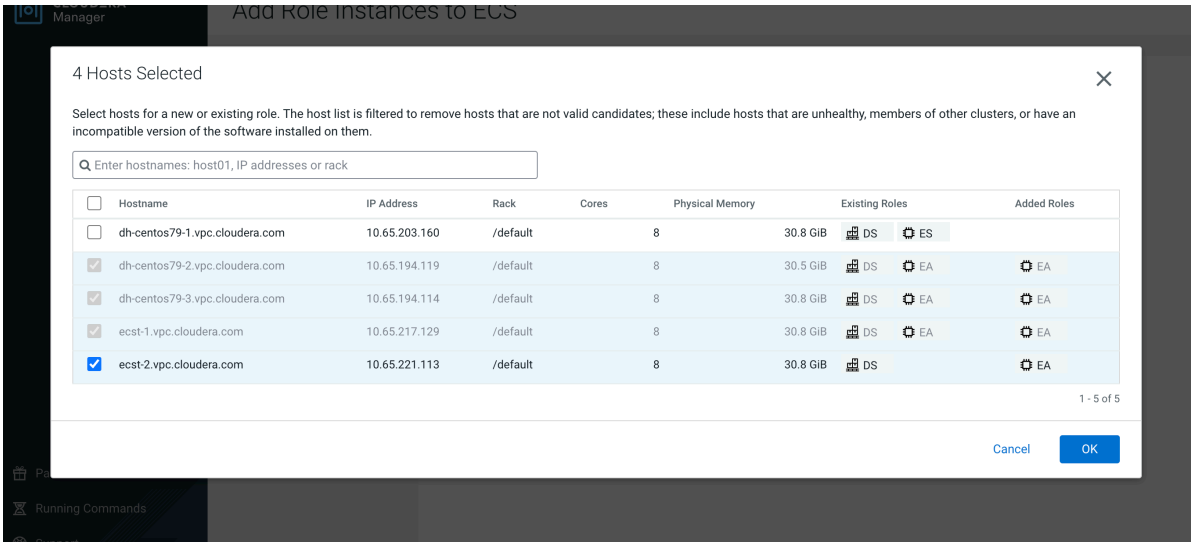


b. On the Add Role Instances to ECS page, in the Ecs Agent box, click Select hosts.

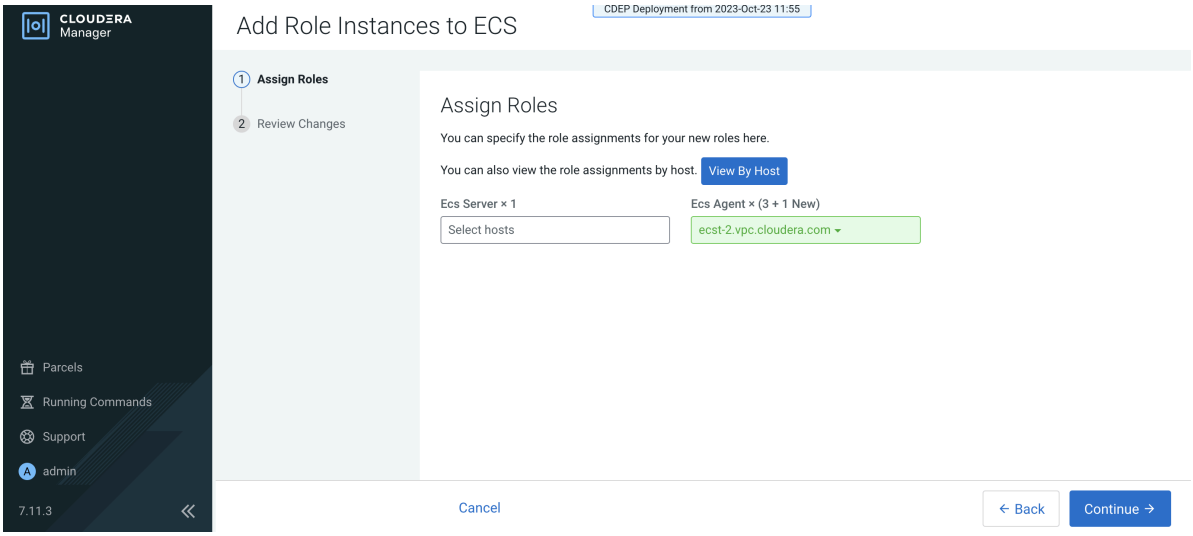
Important: Be sure to click Select hosts in the Ecs Agent box – do not click the link in the Ecs Server box.



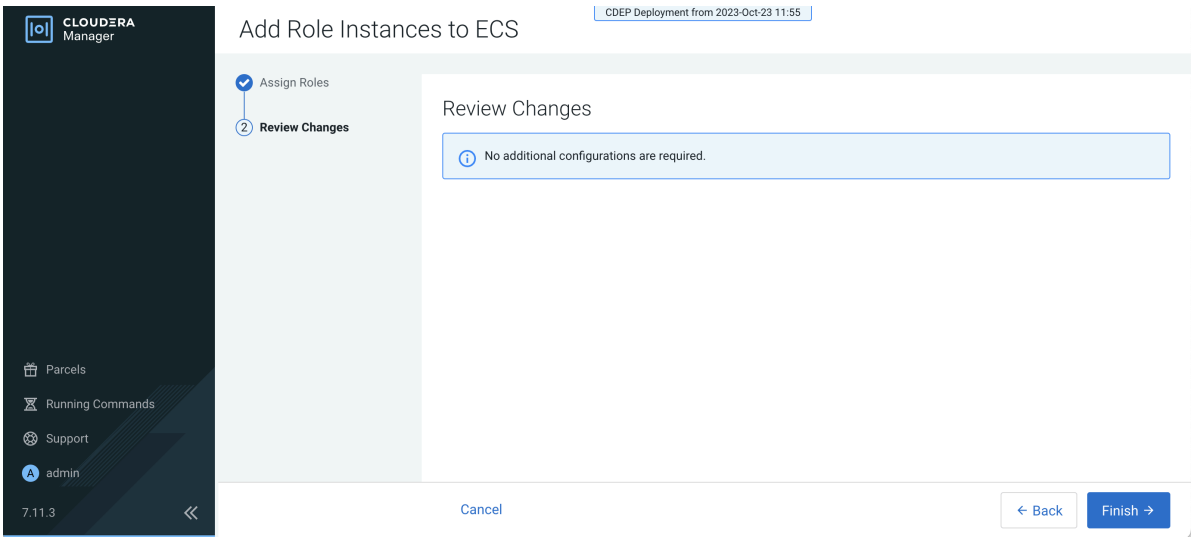
c. On the Hosts Selected pop-up, select the new host, then click OK.



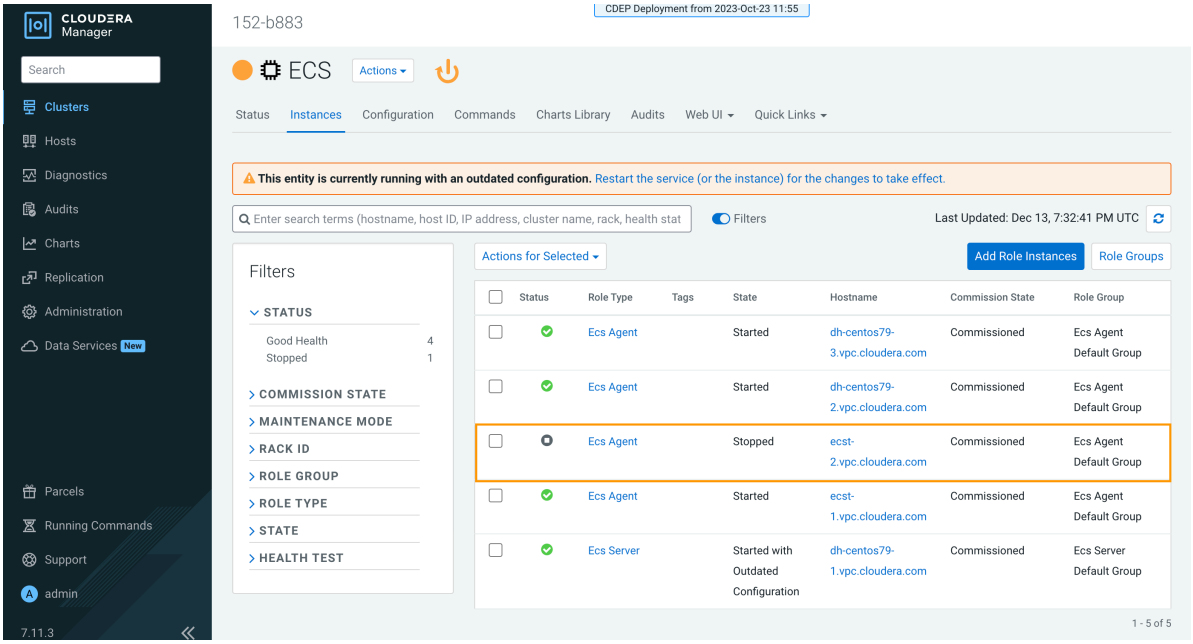
d. On the Assign Roles page, click Continue.



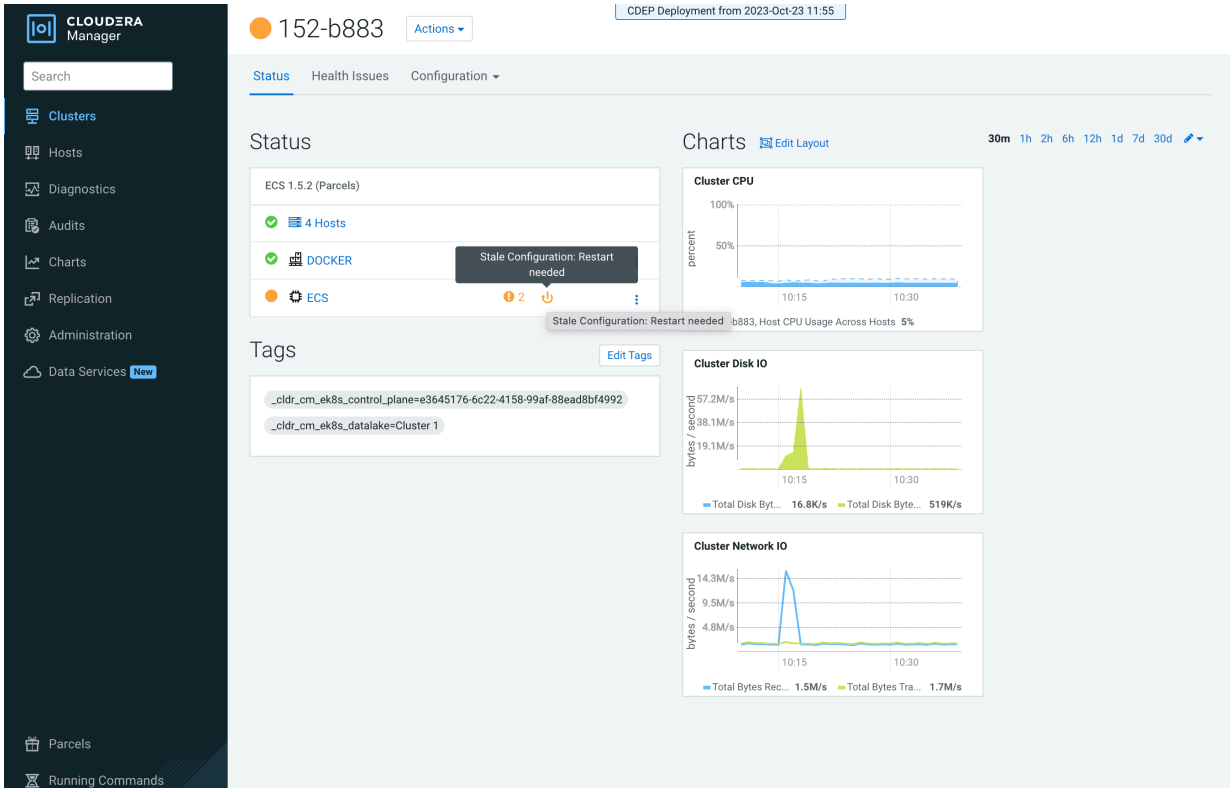
e. On the Review Changes page, click Finish.



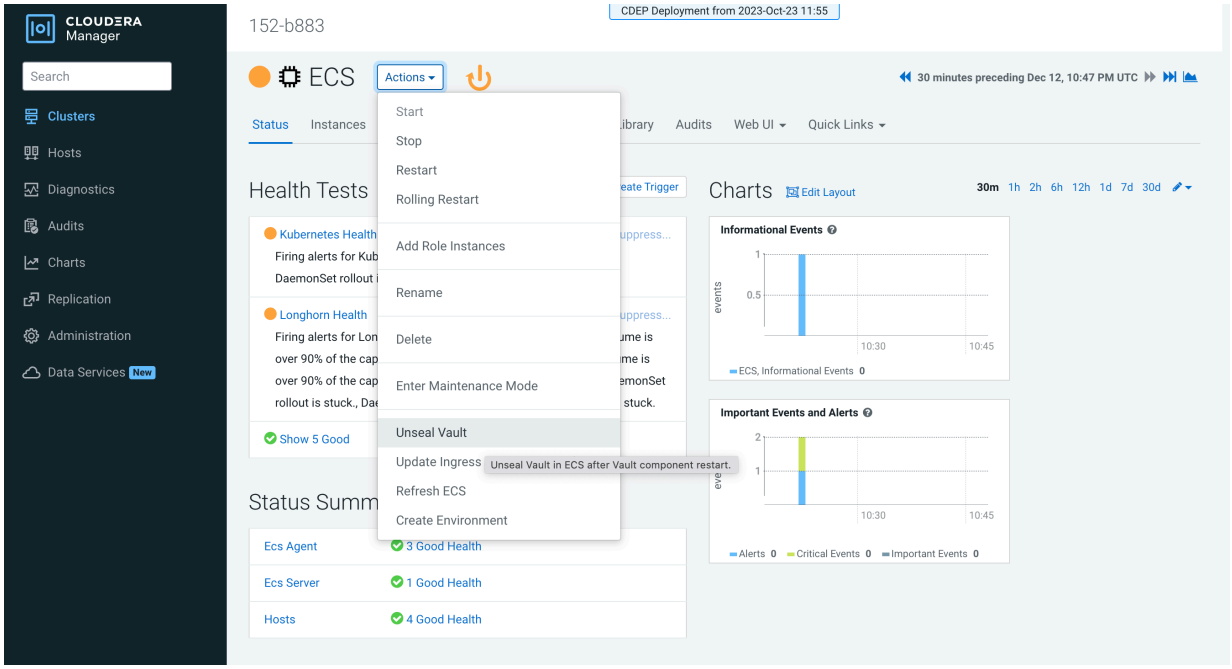
f. The new host is listed on the ECS Instances page.



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.



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.

Monitoring Embedded Container Service Clusters

Procedures to monitor Embedded Container Service clusters

Related Information

Monitoring Services

Monitoring Clusters

Docker Server Health Tests

ECS Health Tests
ECS Agent Health Tests
ECS Server Health Tests
Docker Server Metrics
ECS Agent Metrics
ECS Server Metrics

Viewing Health Status

Procedure

1. Open the Cloudera Manager Admin Console.
2. From the Home page, Click on the Embedded Container Service cluster.
3. Click on the ECS or Docker service.

Results

The Service status page displays the Health Test, Status Summary and Health History of the services.

Viewing the Kubernetes Dashboard

About this task

The Kubernetes Dashboard displays configuration and other information about the embedded Kubernetes infrastructure used in the Embedded Container Service cluster. Although you can make configuration changes using the dashboard (if you have the appropriate permissions), you should not make any changes using the dashboard. Cloudera Support may use the dashboard to diagnose problems with the cluster.

Procedure

1. In the Cloudera Manager Admin Console, go to the ECS service.
2. Click Web UIECS Web UI

Results

The Kubernetes Dashboard displays.

Viewing the Private Cloud Management Console

Procedure

1. In the Cloudera Manager Admin Console, go to the ECS service.
2. Click Web UIConsole

Results

The CDP Management Console displays.

Performing maintenance on an Embedded Container Service cluster

You can perform maintenance on the nodes in your ECS cluster by shutting down the nodes one at a time while keeping your Data Services running with slightly diminished capacity.

Before you begin

- The containerized cluster must be configured for ECS Server high availability. Contact Cloudera Professional Services for assistance in setting up high availability.
- You must be able to log into the nodes as root or have sudo privileges.
- The node to be maintained must have a status of Ready. A status of NotReady may suggest the node is having other complicating issues. Run the following command on an ECS server node to verify status of the nodes.

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

Procedure

1. Inform Kubernetes that it should no longer use this node for any new pods. This process is called cordon and Kubernetes tracks the node status as Ready,SchedulingDisabled.
   a) Run the following command to list the nodes:

   ```
   /var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.ya
   ml get nodes
   ```

   b) Run the following command for the node you are taking off line:

   ```
   /var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.ya
   ml cordon **node-name**
   ```

   c) Run the following command to verify the node status shows Ready,SchedulingDisabled:

   ```
   /var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.ya
   ml get nodes
   ```

2. Inform Kubernetes to evict this node's Data Services pods and cleanly detach any storage volumes. This allows the pods to be started up on other Ready nodes in the cluster and any replica volumes are migrated. The process is invoked by the drain command:

   ```
   /var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.yaml
    drain *node-name* --delete-emptydir-data --ignore-daemonsets --pod-select
   or='app!=csi-attacher,app!=csi-provisioner,app!=longhorn-admission-webho
   ok,app!=longhorn-conversion-webhook,app!=longhorn-driver-deployer'
   ```

   You will see a message

   ```
    “WARNING: ignoring DaemonSet-managed Pods:....
   ```

   You can ignore this warning.

   You will see repeating messages like this:

   ```
   error when evicting pods/"instance-manager-r-xxxxxxxx" -n "longhorn-syst
   em" (will retry after 5s): Cannot evict pod as it would violate the pod's
    disruption budget.
   ```

   This is normal, after several iterations those pods will be evicted and the drain is completed.
3. Log in to the Cloudera Manager Admin Console.
4. Go to the ECS service page and verify that the Vault is not sealed. This information displays in the Health Tests section.
5. If the Vault is sealed, click ActionsUnseal Vault.
6. Click the Action menu next to the ECS cluster and select Stop.

7. Shutdown ECS roles.
   a) Click the Instances tab.
   b) Select the hosts where the ECS Agent role is running and click ActionsStop.
   c) Select two of the hosts running the ECS Server role is running and click ActionsStop.
8. Perform the maintenance.
9. Reboot the hosts.
10. Log in to the Cloudera Manager Admin Console.
11. Click the Action menu next to the ECS cluster and select Start.
12. Uncordon the node to start the Data Services by running the following command:

```
/var/lib/rancher/rke2/bin/kubectl --kubeconfig=/etc/rancher/rke2/rke2.yaml
 uncordon **node-name**
```

13. Run the following command to verify that the node status is Ready:

```
/var/lib/rancher/rke2/bin/kubectl get nodes
```

14. Click ActionsRefresh ECS Cluster.

Configuring a containerized cluster with SELinux

You can configure a containerized cluster with SELinux to enable it to run the Embedded Container Service (ECS).

Procedure

1. Ensure that the hosts you use for the containerized cluster meet all hardware and software requirements for use with CDP Private Cloud Data Services.
2. Enable SELinux in Permissive mode by updating the /etc/selinux/config file on all ECS hosts by running the following commands:

```
sed -i 's/SELINUX=disabled/SELINUX=permissive/' /etc/selinux/config
reboot
```

3. Add the SELinux policies provided by RKE2 by installing the RPMs on all ECS hosts. Use the following commands:

```
yum localinstall -y http://mirror.centos.org/centos/7/extras/x86_64/Pack
ages/container-selinux-2.107-3.el7.noarch.rpm
wget https://github.com/rancher/rke2-selinux/releases/download/v0.8.stable
.2/rke2-selinux-0.8-2.el7.noarch.rpm
yum install -y rke2-selinux-0.8-2.el7.noarch.rpm
```

4. Uninstall the nscd service by running the following command on all ECS hosts :

```
yum erase -y nscd
```

5. Install a containerized cluster on all hosts. See Adding a CDP Private Cloud Data Services cluster.
6. Enable SELinux in Enforced mode by running the following commands on all ECS hosts:

```
setenforce 1
```

   You can confirm that SELinux is running in Enforced mode by running the following command:

```
getenforce
```

7. Verify that the ECS cluster hosts are sending heartbeats to the Cloudera Manager server.
   a) Open the Cloudera Manager Admin Console.
   b) Click Hosts All Hosts .
   c) Check the Last Heartbeat column for heartbeat status.
8. Verify that your workloads are functioning as expected.

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-at
tacher,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)

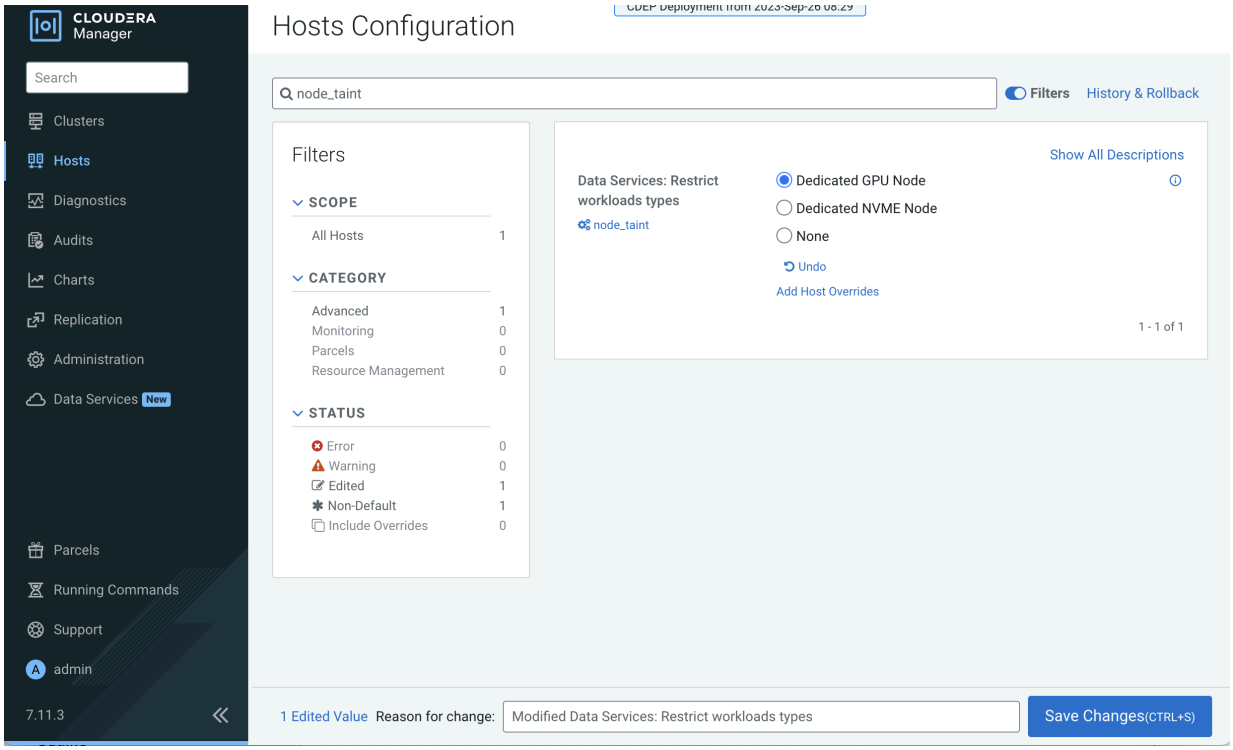
Dedicating ECS nodes for specific workloads

You use Cloudera Manager to dedicate Embedded Container Service (ECS) cluster nodes for specific workloads. You can dedicate GPU nodes for CML workloads, and NVME nodes for CDW workloads.

Dedicating ECS nodes when creating a new cluster

1. Check the ECS installation requirements.
2. Add the new hosts to Cloudera Manager.
3. In Cloudera Manager, click Hosts > All Hosts, then select one or more of the new ECS hosts.
4. Click the Configuration tab, then use the Search box to locate the node_taint configuration property.
5. Select Dedicated GPU Node to dedicate the node for CML workloads, or select Dedicated NVME node to dedicate the node for CDW workloads.

   When either of these options are selected, no other workload pods will be allowed to run on the dedicated node.



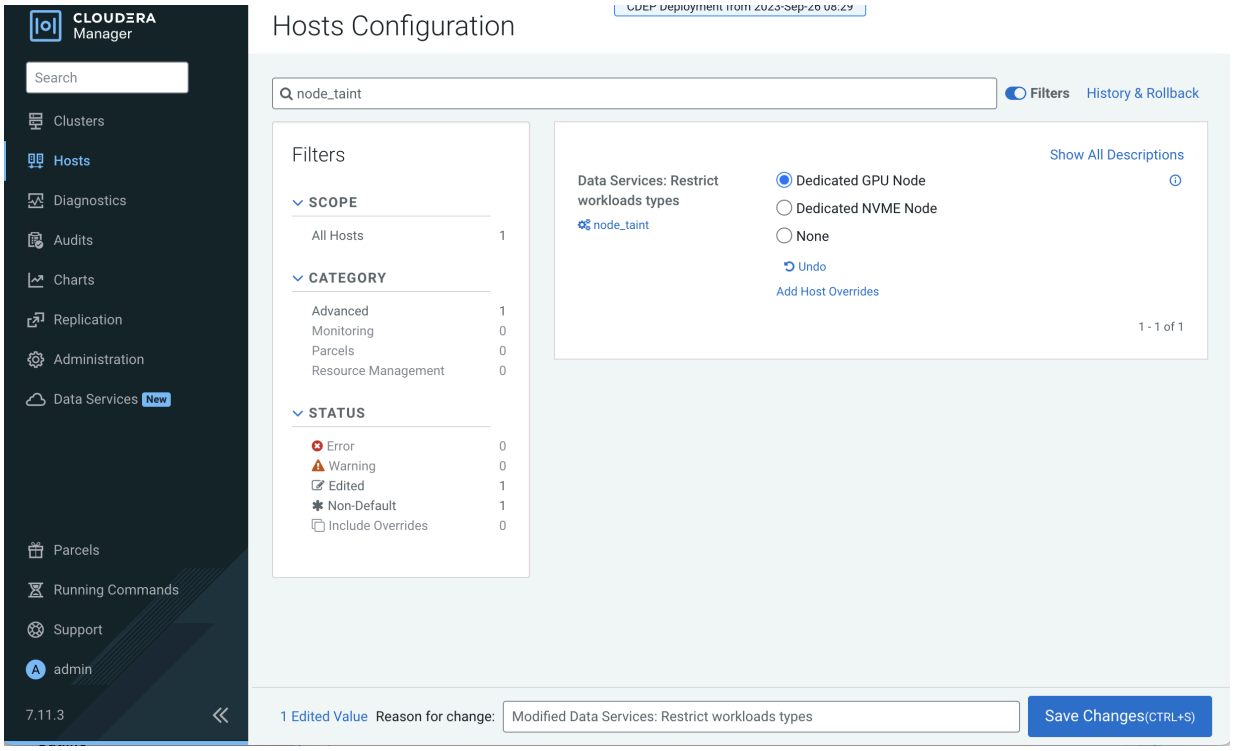
6. Click Save Changes.
7. Repeat the previous steps to add the other ECS hosts to Cloudera Manager and assign workload types.
8. Follow the ECS installation procedure. When you reach the Specify Hosts page in the installation wizard, the hosts you added to Cloudera Manager appear. Select the hosts, click Continue, then proceed through the rest of the installation wizard.
9. After the installation is complete, the applicable workloads will only run on the specified dedicated nodes.

Dedicating ECS nodes in an existing cluster

1. Open the Cloudera Manager Admin Console.
2. On the Home page, click the ECS Cluster.
3. Click Hosts, select one or more of the ECS hosts, then click the Configuration tab.
4. Click the Configuration tab, then use the Search box to locate the node_taint configuration property.

5. Select Dedicated GPU Node to dedicate the node for CML workloads, or select Dedicated NVME node to dedicate the node for CDW workloads.

   When either of these options are selected, no other workload pods will be allowed to run on the dedicated node.



6. Click Save Changes.
7. Repeat the previous steps to assign workload types to the other ECS hosts.
8. On the ECS Cluster landing page, click Actions > Refresh Cluster.
9. After the Refresh is complete, click Actions > Rolling Restart.

Specifying racks for ECS clusters

You use Cloudera Manager to assign Embedded Container Service (ECS) cluster hosts to a specific rack.

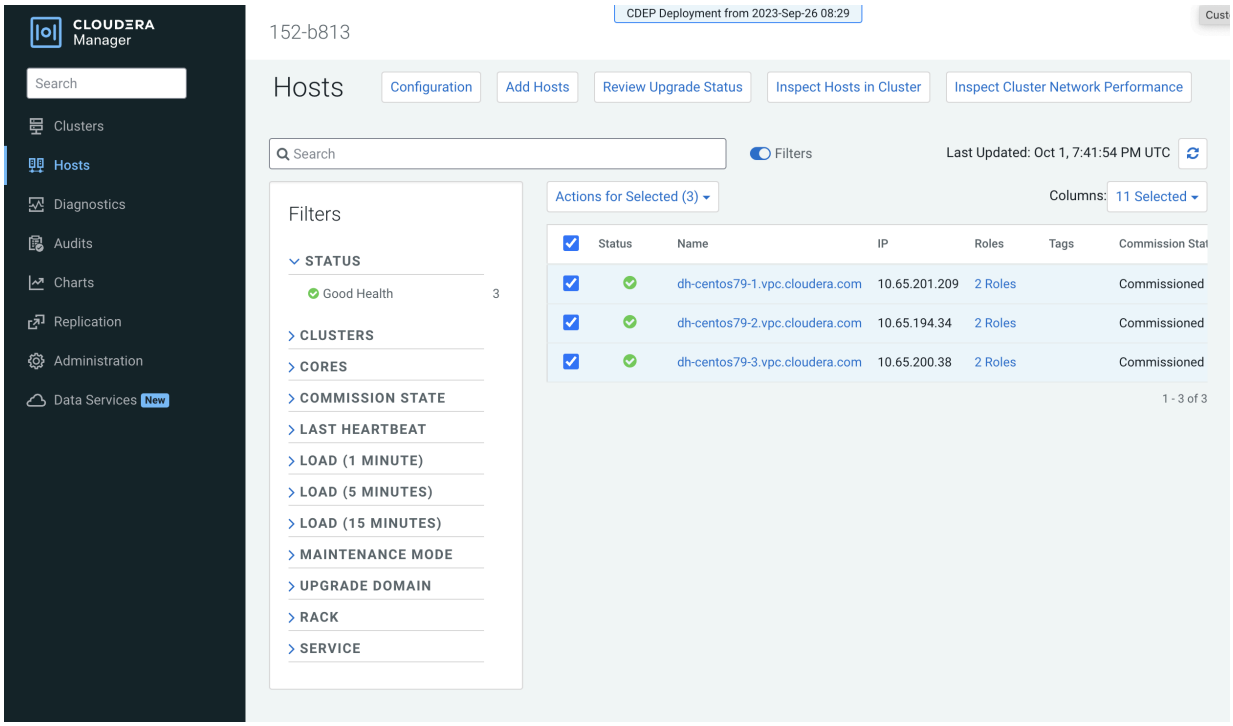
About this task

- All hosts in an ECS cluster must have the same assigned rack name and path structure. A configuration error will occur if the rack names do not match.
- ECS cluster hosts with no specified rack name are assigned the default rack name value. The default value means that no rack name has been specified for the ECS cluster hosts.

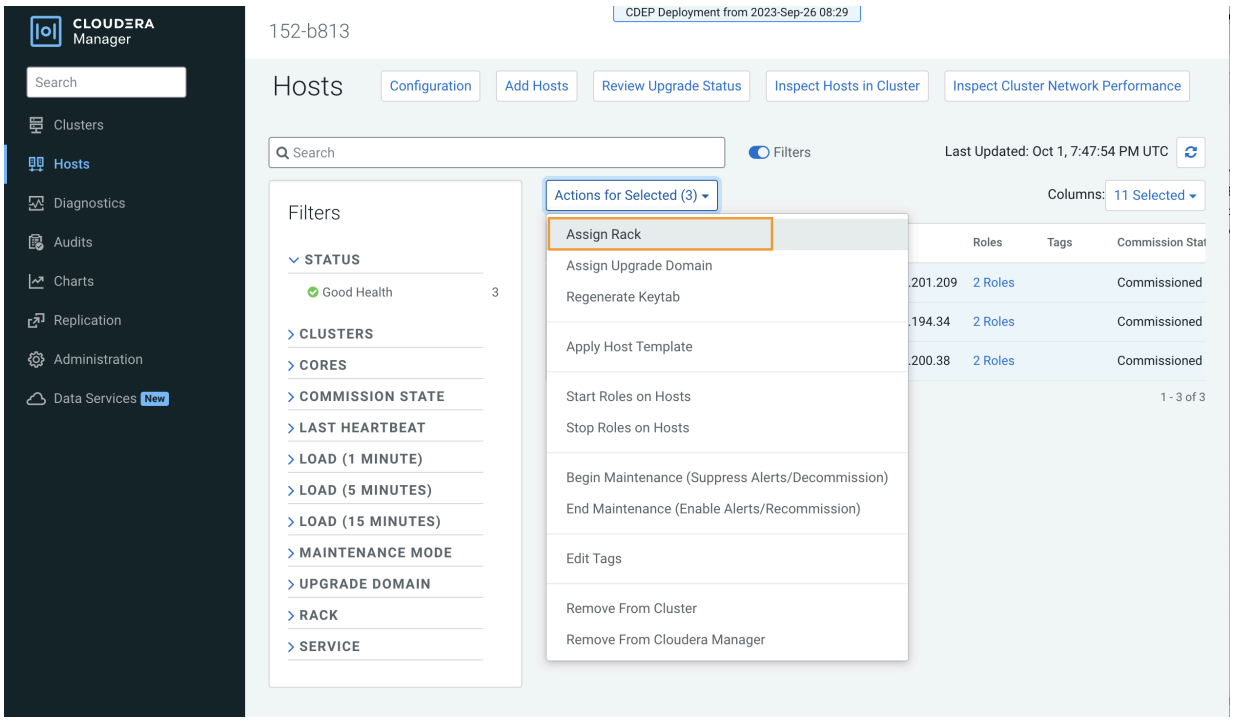
Specifying a rack name for an ECS cluster

1. In Cloudera Manager, select the ECS cluster, then click Hosts.

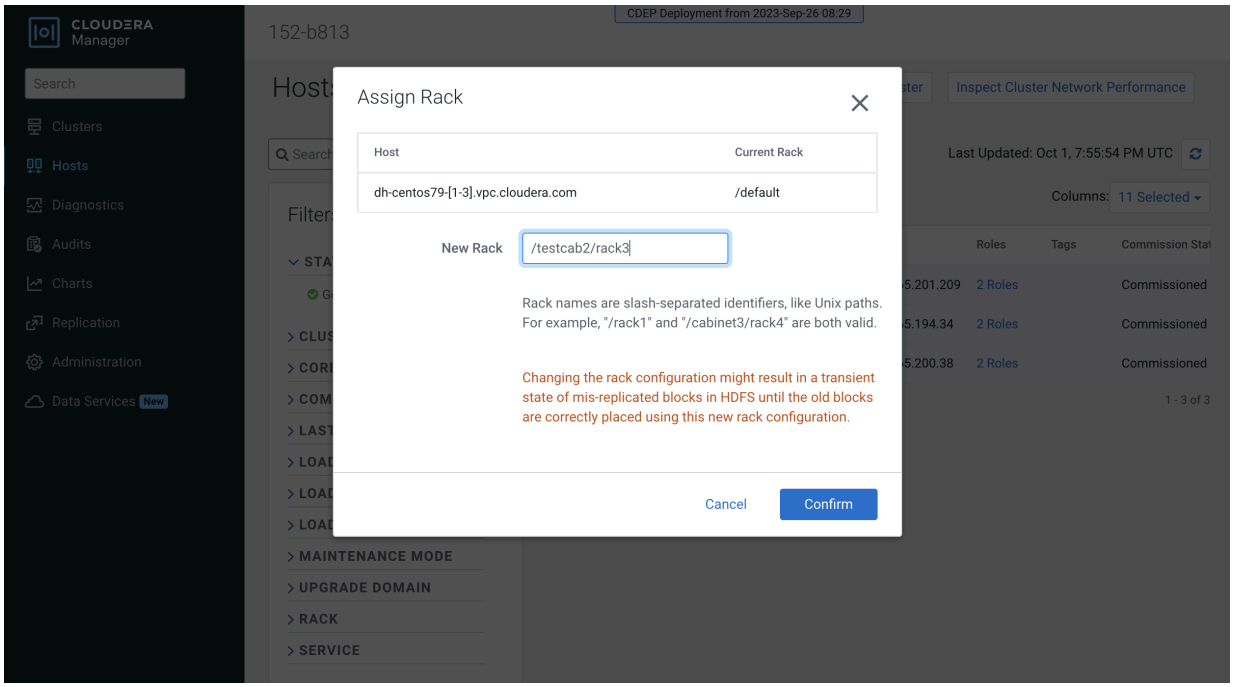
2. In the Hosts list, click the top checkbox to select all of the cluster hosts.



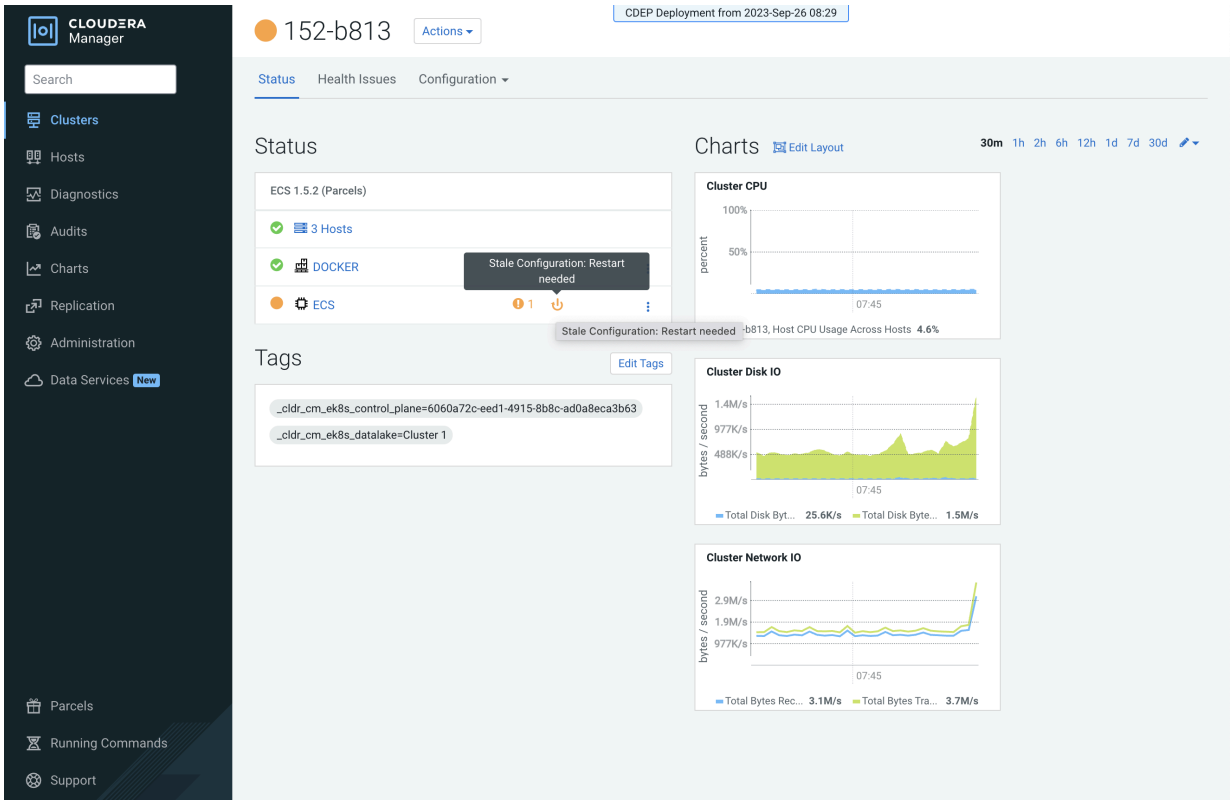
3. Click Actions for Selected, then click Assign Rack.



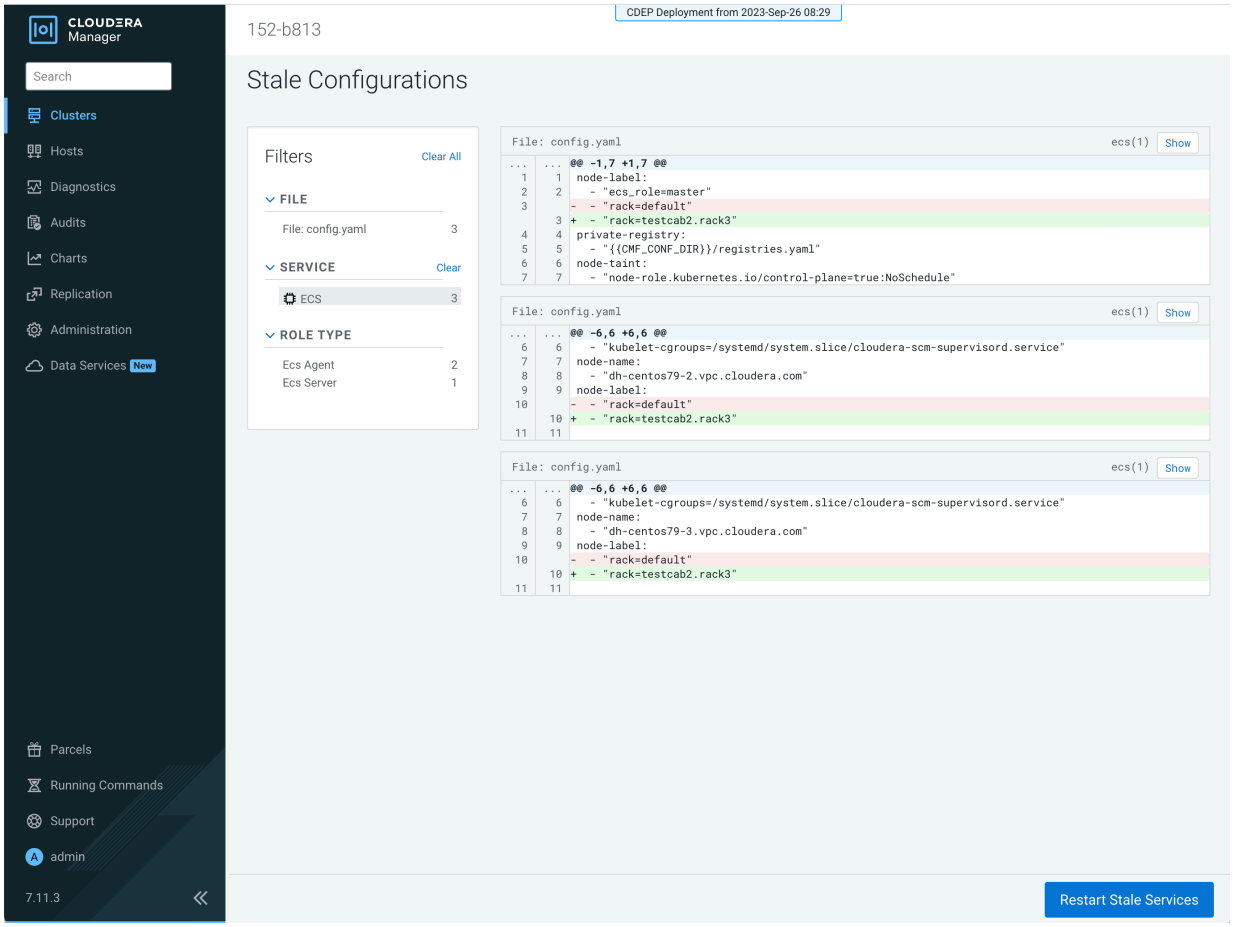
4. On the Assign Rack popup, enter a rack name in the New Rack box, then click Confirm.



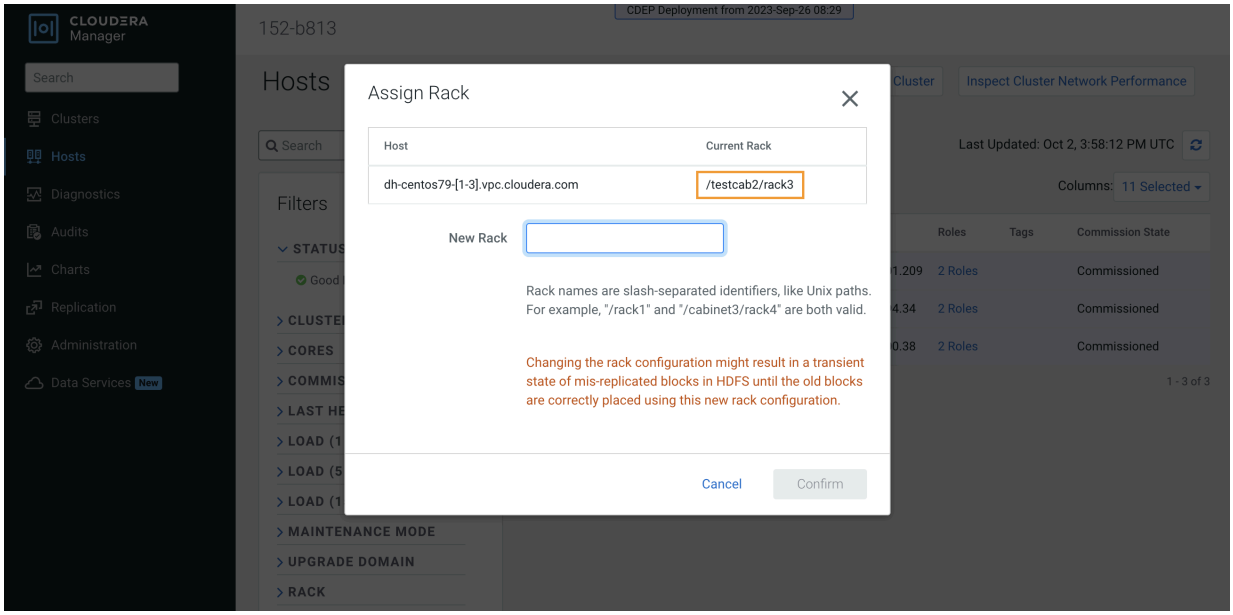
5. Cloudera Manager detects this configuration change, and displays a Stale Configuration warning. You must restart the cluster in order for the updated configuration to take effect.



6. Click the Stale Configuration icon, then click Restart Stale Services and click through the Restart wizard.

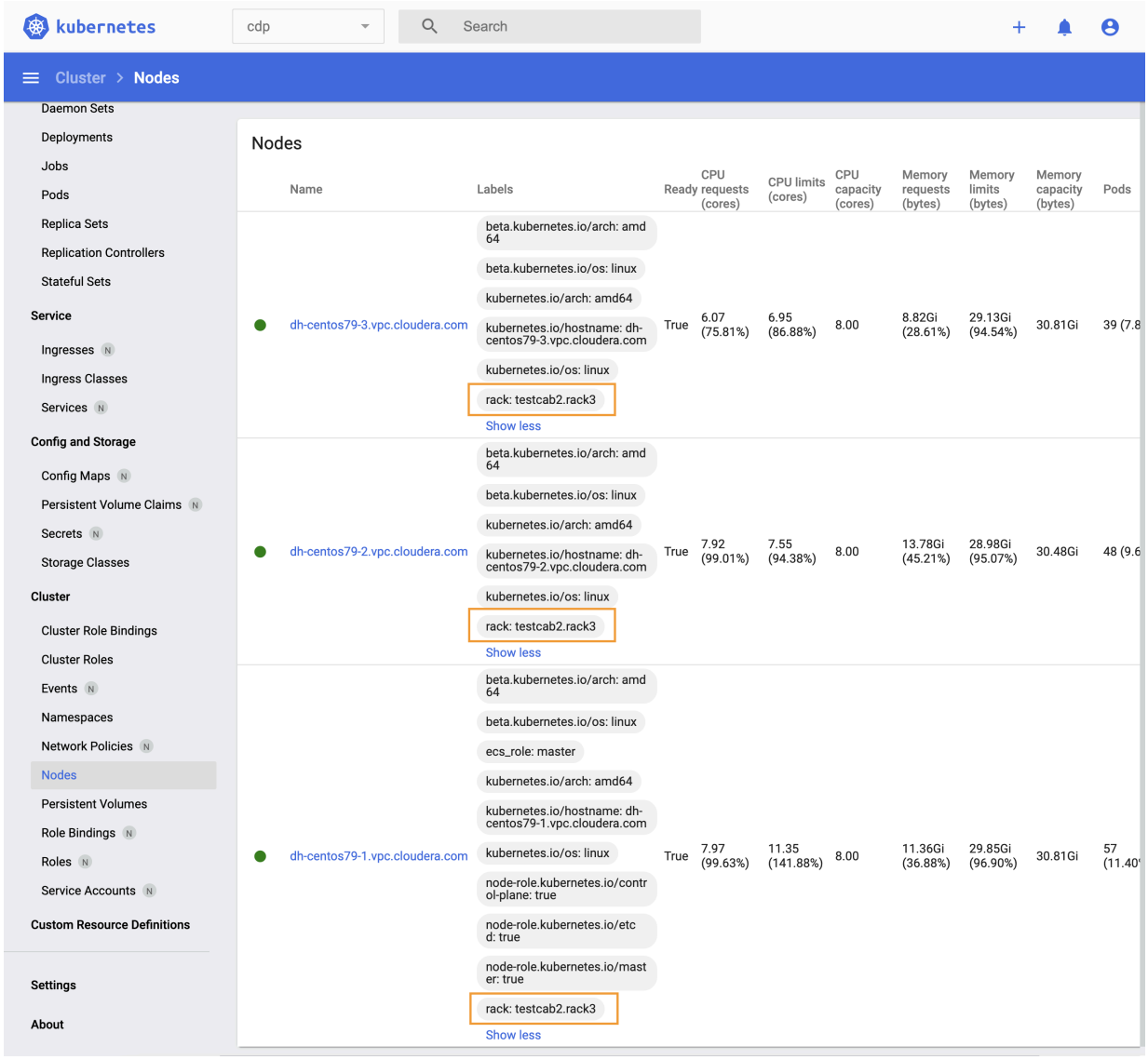


7. When the Restart is complete, you can use the Assign Rack popup to confirm that the new rack name has been applied to the ECS cluster hosts.



8. You can also use the ECS Web UI to view cluster hose rack assignments. Select the ECS cluster, click ECS, then click Web UI > ECS Web UI . In the Web UI, select the CDP namespace, then click Nodes.

   Note that in Kubernetes periods are used as separators (rather than slashes) in the rack name path. The leading slash is also not used in Kubernetes.

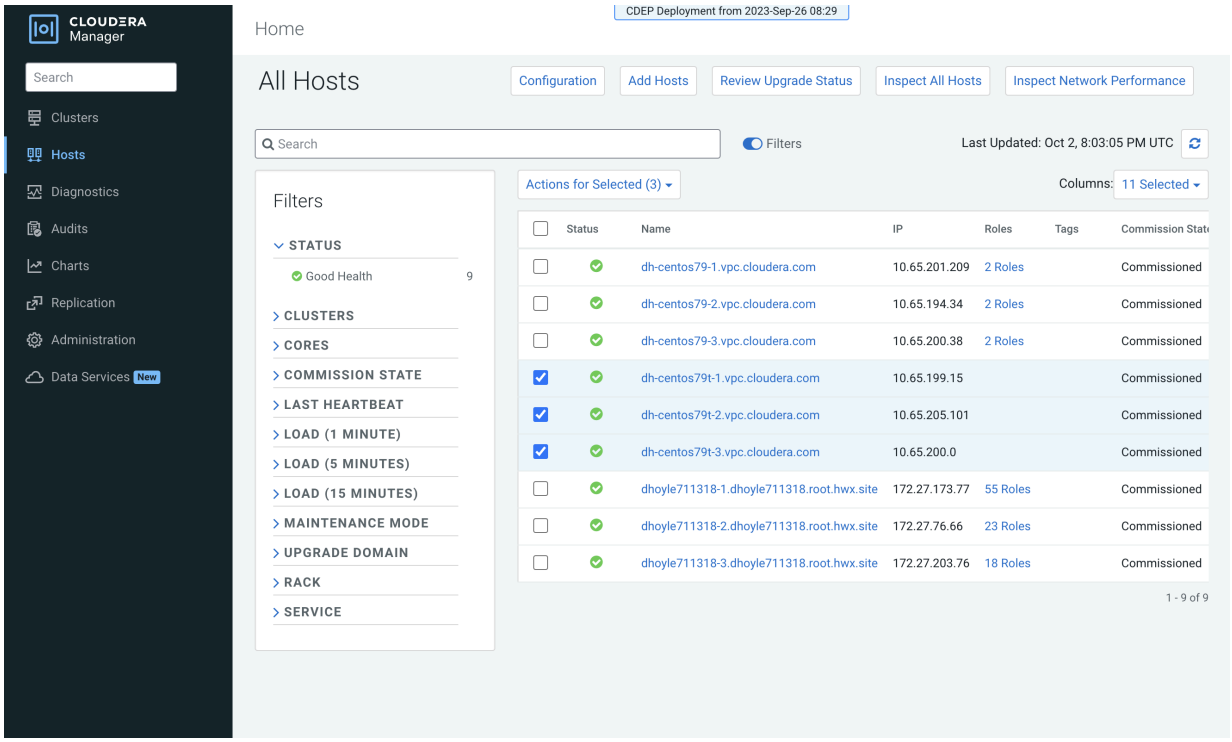


Specifying a rack name when creating a new ECS cluster

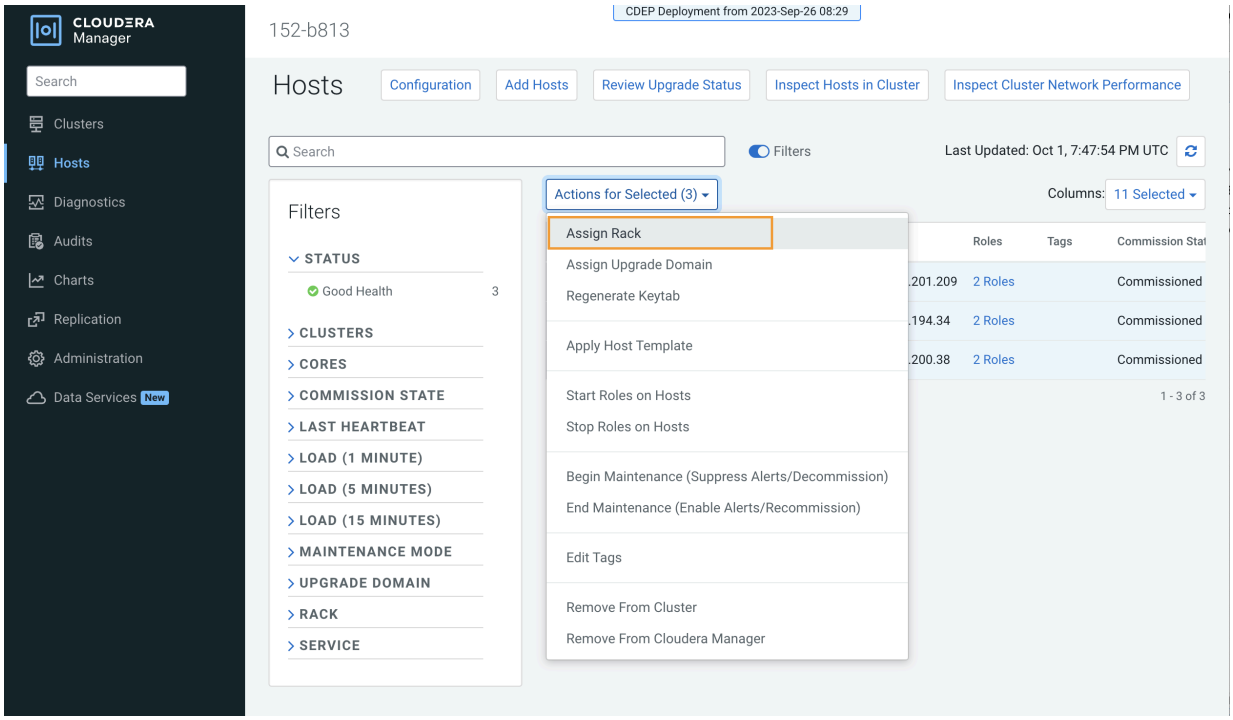
Currently the ECS installation wizard does not enable you to assign rack names when creating a new ECS cluster. Therefore, you should first add the new set of ECS hosts to Cloudera Manager, and then assign the rack name in Cloudera Manager. You can then use the ECS installation wizard to create a new ECS cluster using these hosts.

1. Check the ECS installation requirements.
2. Add the new hosts to Cloudera Manager.

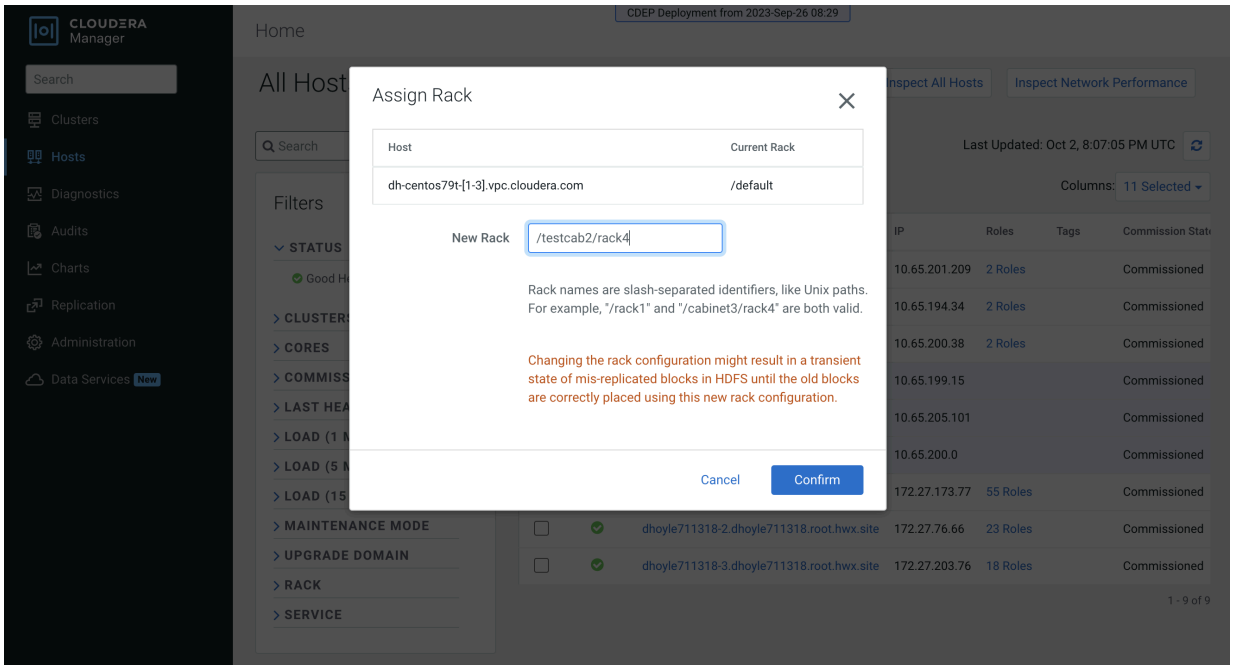
3. In Cloudera Manager, click Hosts > All Hosts, then select the hosts you just added.



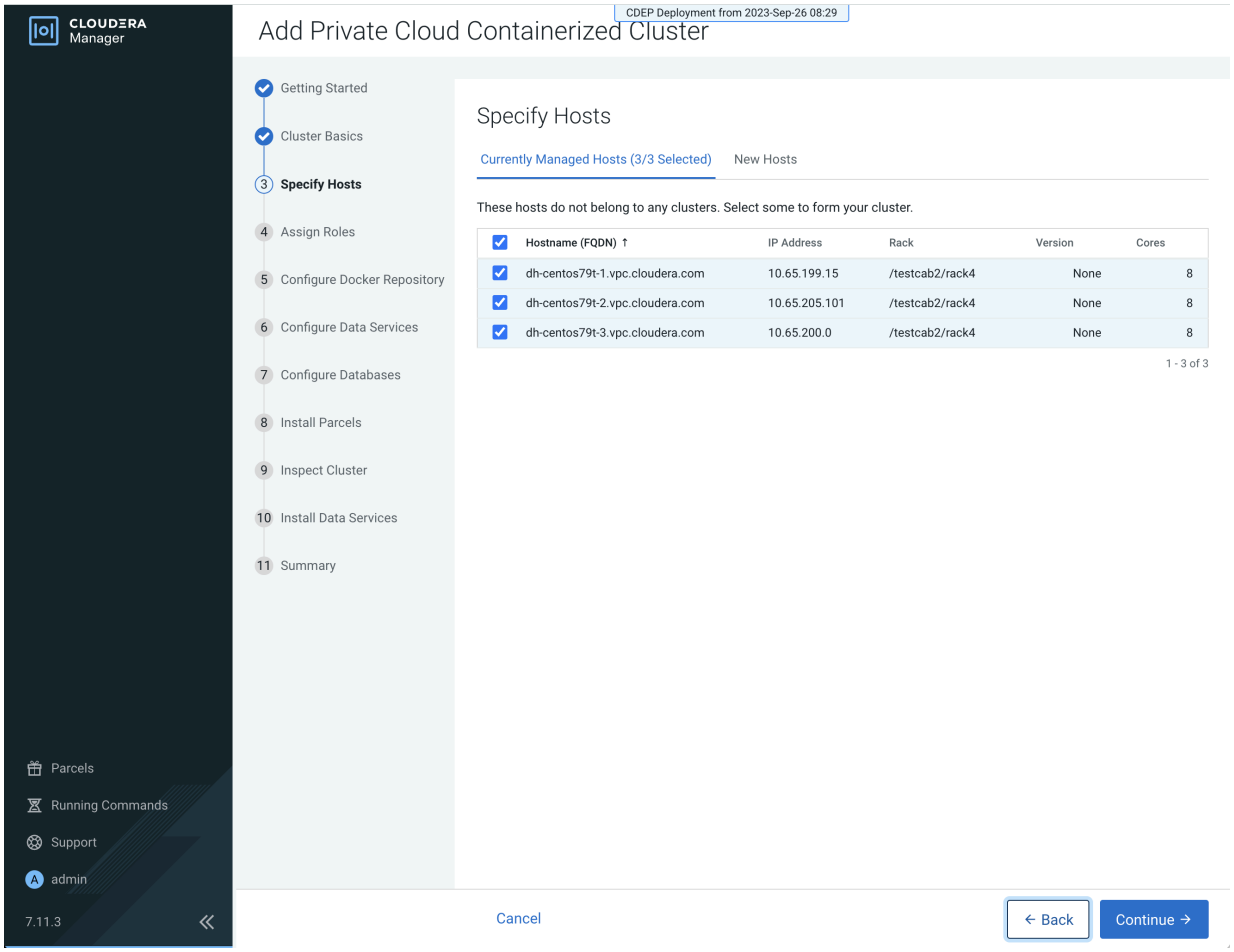
4. Click Actions for Selected, then click Assign Rack.



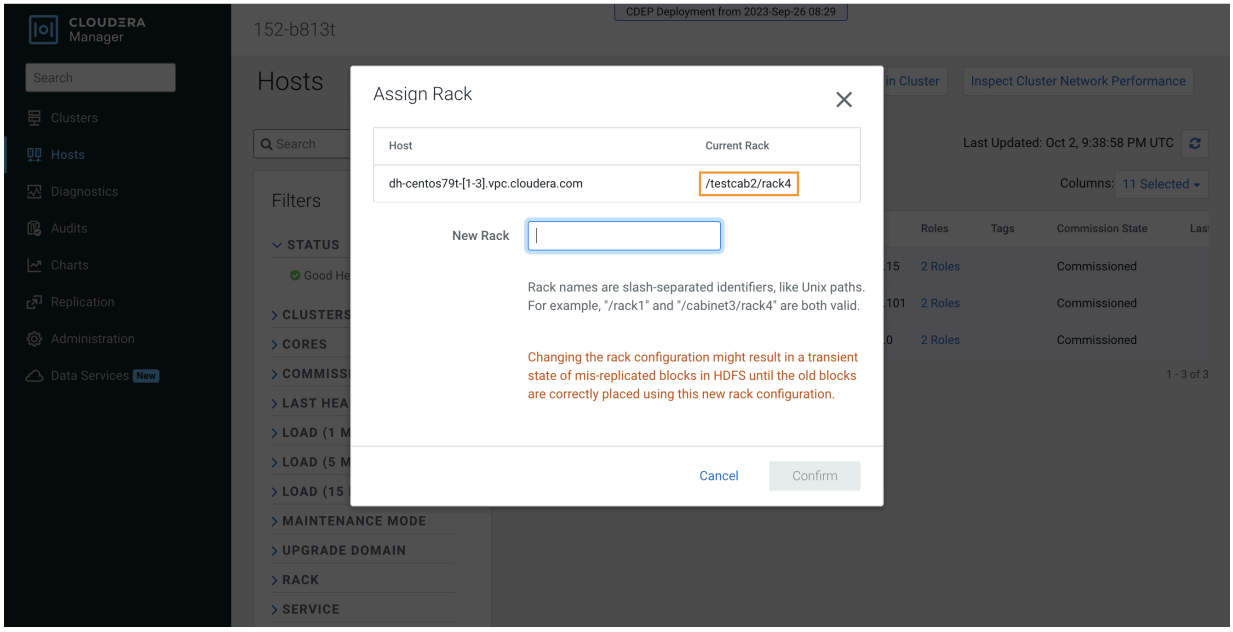
5. On the Assign Rack popup, enter a rack name in the New Rack box, then click Confirm.

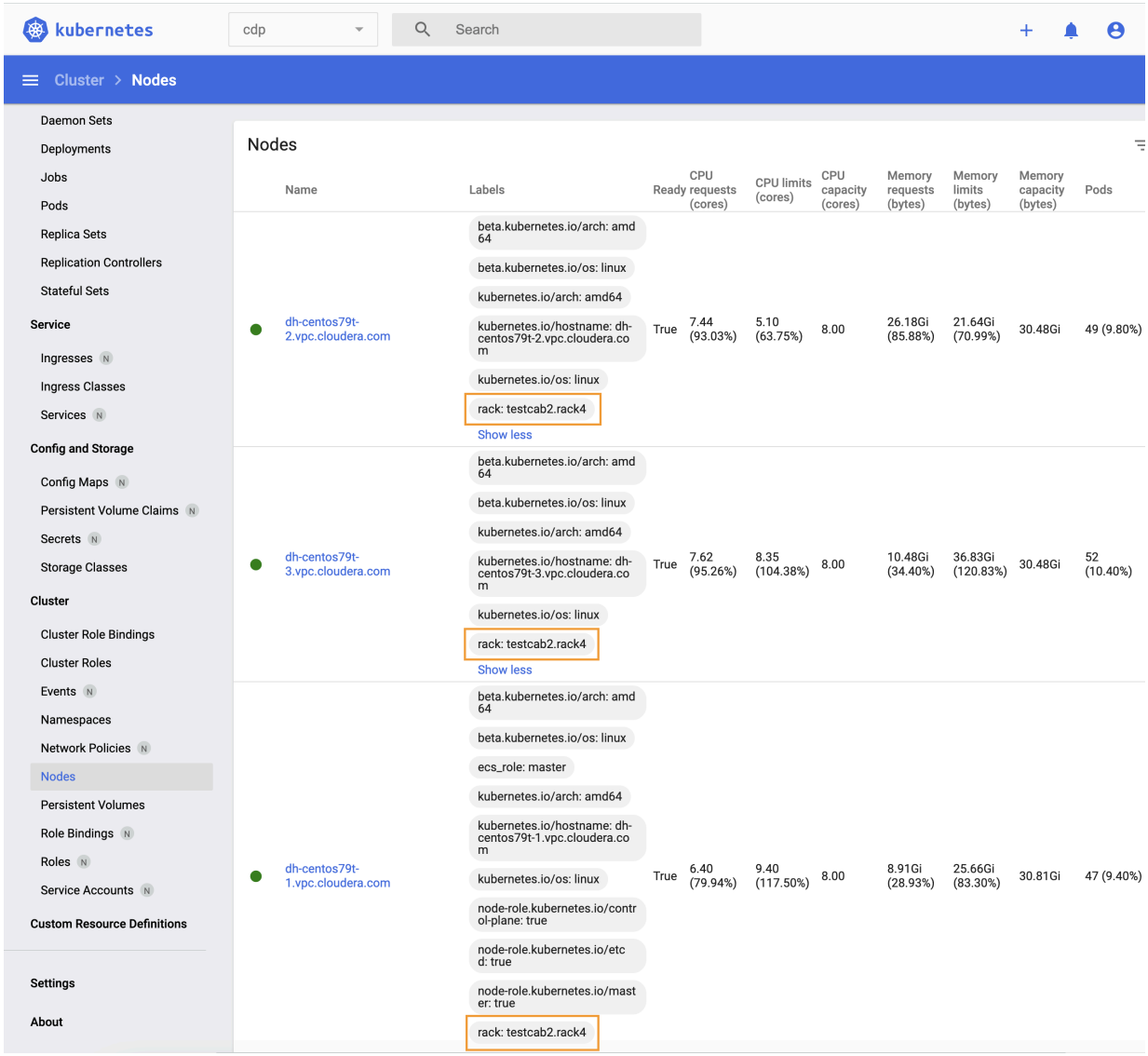


6. Follow the ECS installation procedure. When you reach the Specify Hosts page in the installation wizard, the hosts you added to Cloudera Manager appear. Select the hosts, click Continue, then proceed through the rest of the installation wizard.



7. After the installation is complete, you can use the Assign Rack popup or the ECS Web UI to view the rack assignments for the ECS cluster hosts.





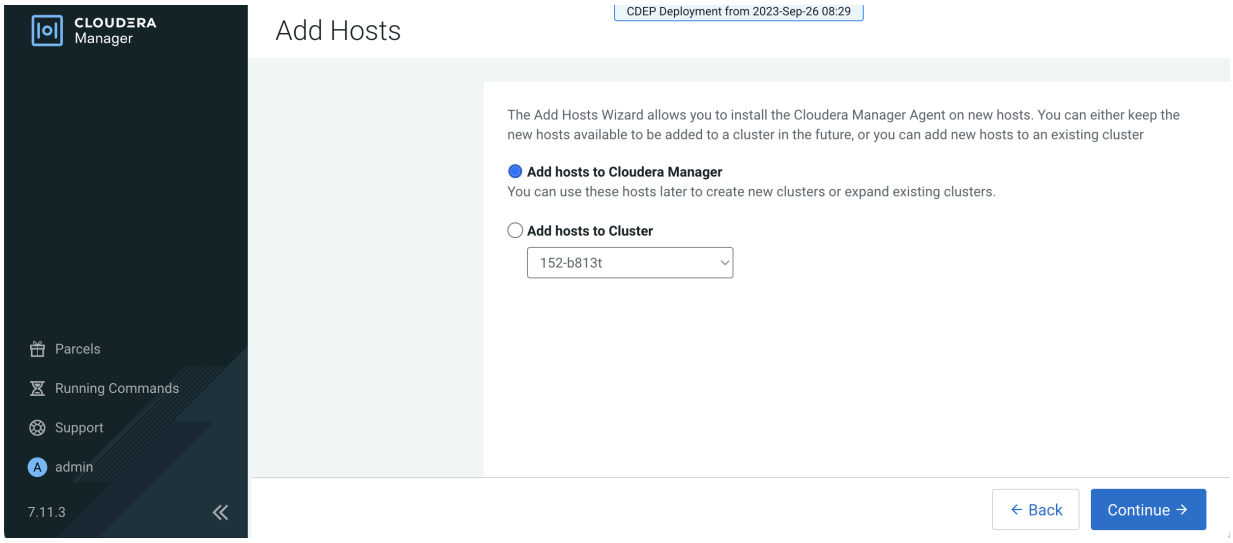
Adding a host to an ECS cluster with a previously specified rack name

When you add a host directly to an ECS cluster, there is no way to specify a rack name for the new host, so it will be assigned the default rack name. A configuration error will occur if you try to add a new host directly to an ECS cluster with a previously specified rack name, since the default rack name of the new host does not match the rack name previously assigned to the other cluster hosts.

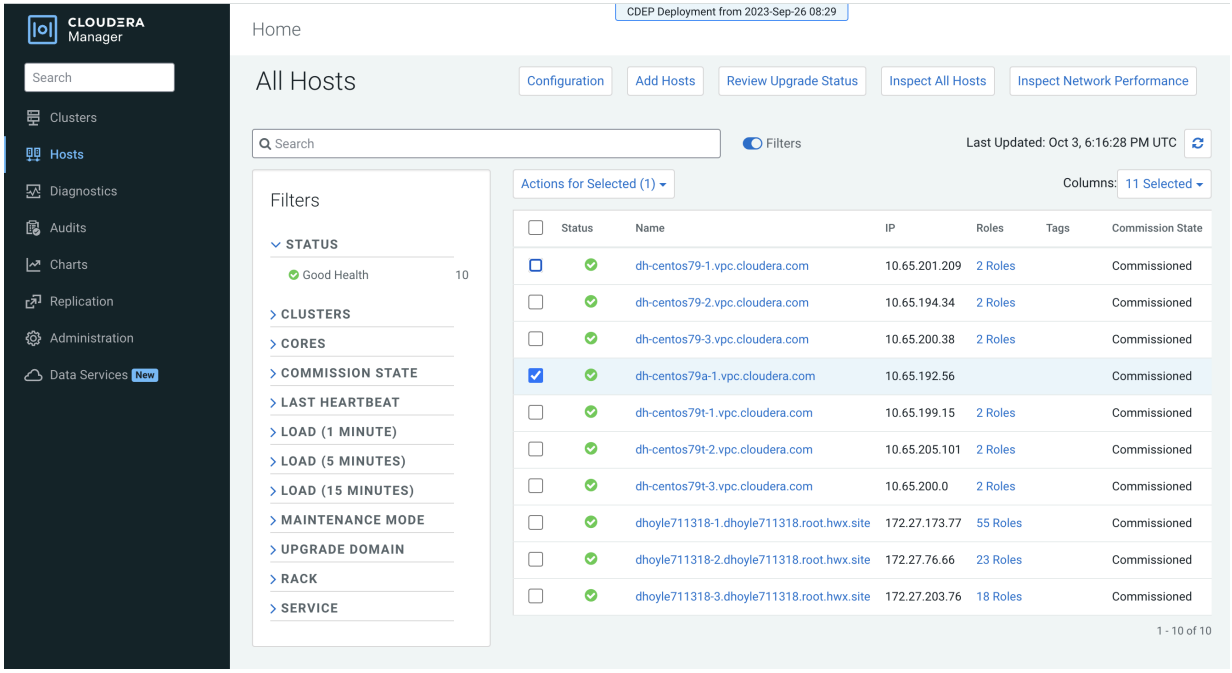
Therefore, you should first add the new ECS host to Cloudera Manager, and then use Cloudera Manager to assign the same rack name as the other ECS cluster hosts to the new host. You can then add the new host to the ECS cluster.

1. Check the ECS installation requirements.

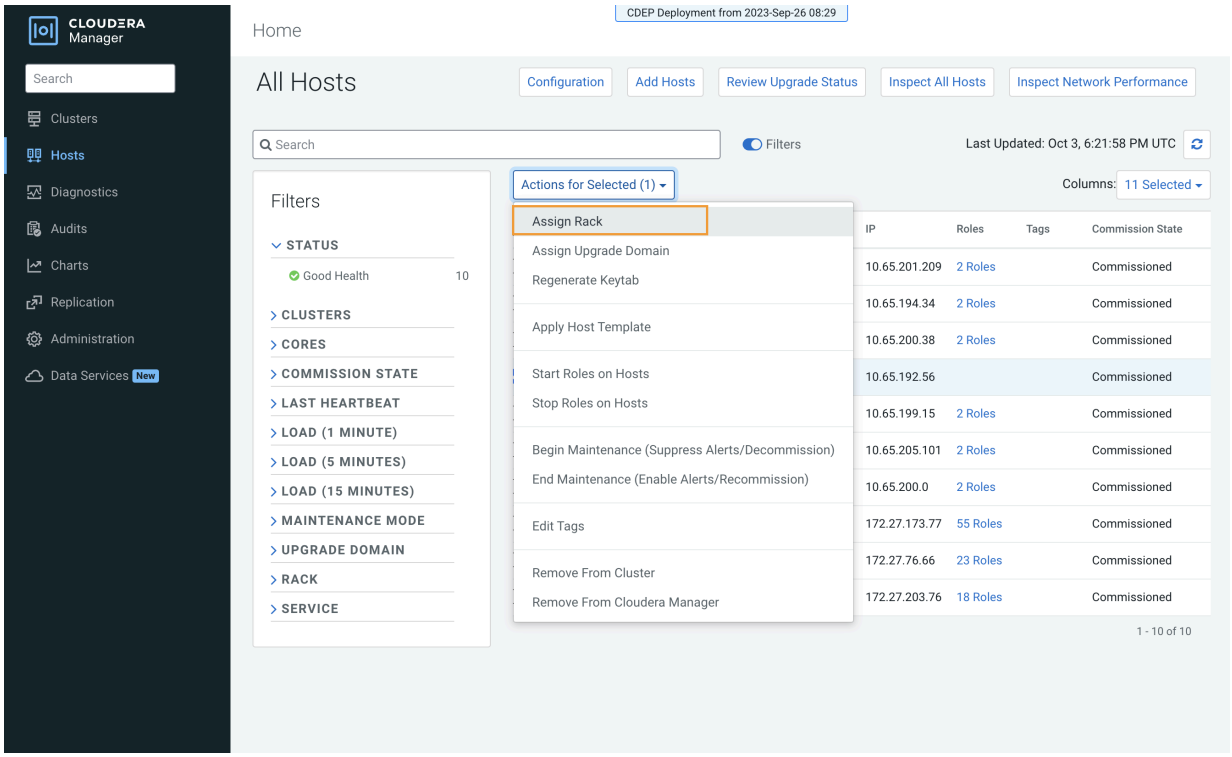
2. Add the new hosts to Cloudera Manager. You can also access the Add Hosts wizard by clicking Hosts in the ECS cluster, and then clicking Add Hosts. Select Add hosts to Cloudera Manager.



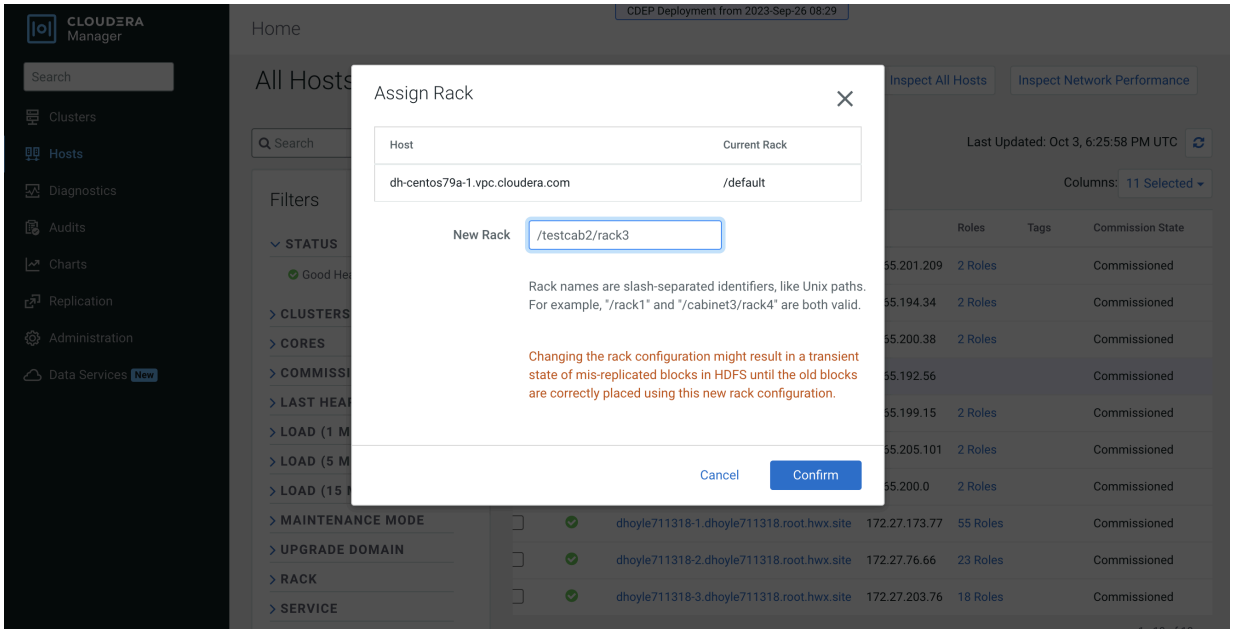
3. In Cloudera Manager, click Hosts, then select the host you just added.



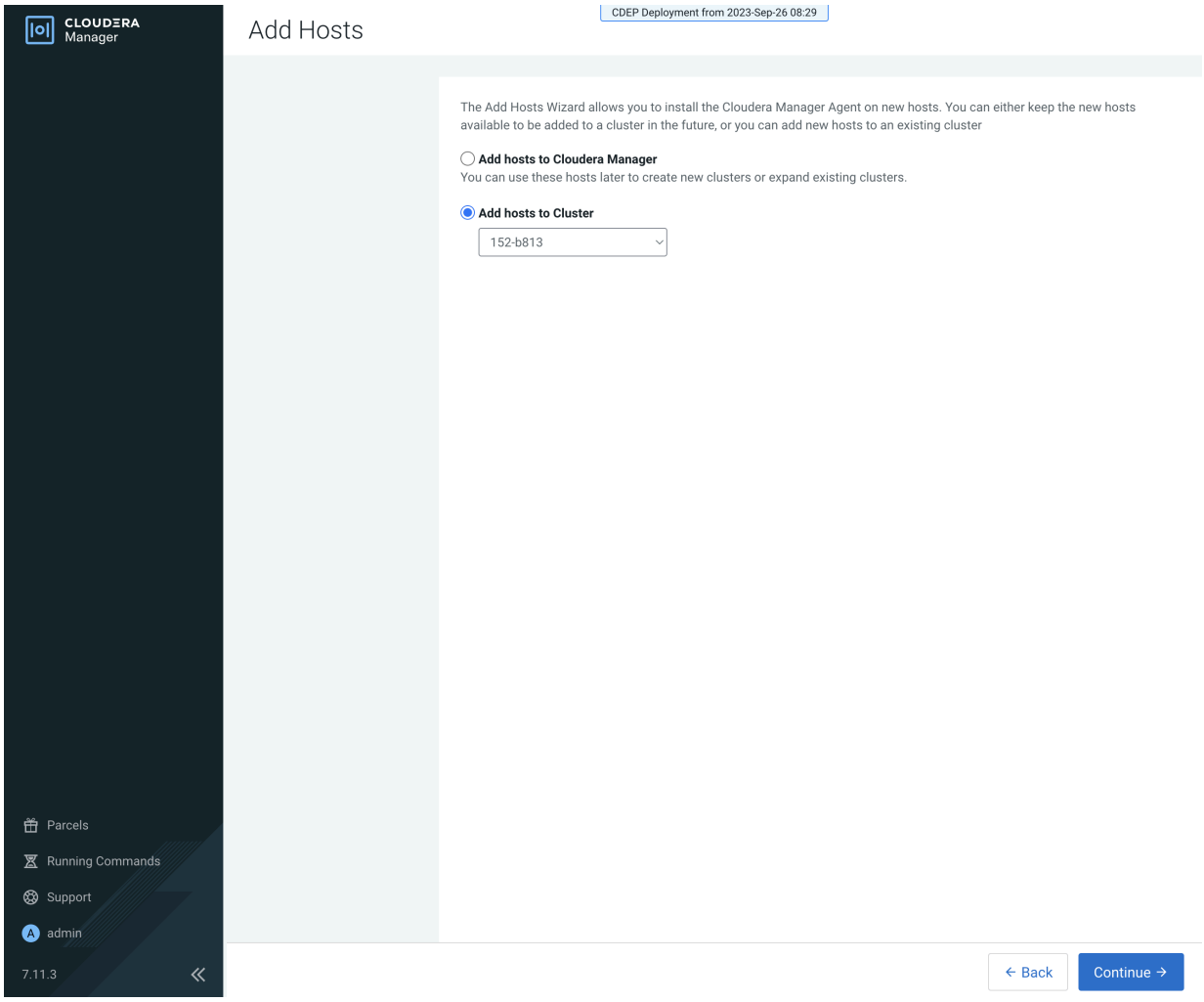
4. Click Actions for Selected, then click Assign Rack.



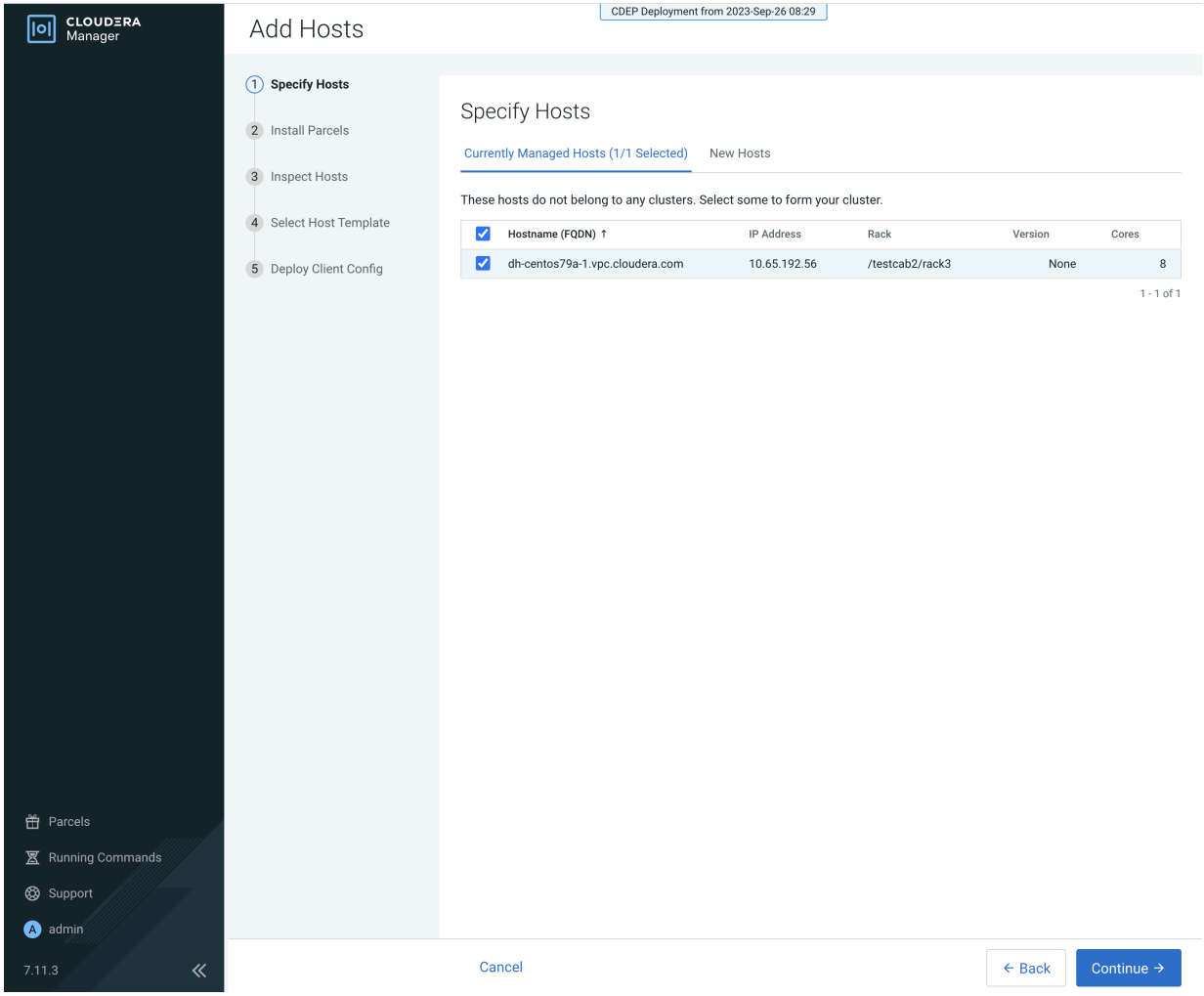
5. On the Assign Rack popup, enter the same rack name assigned to the other ECS cluster hosts in the New Rack box, then click Confirm.



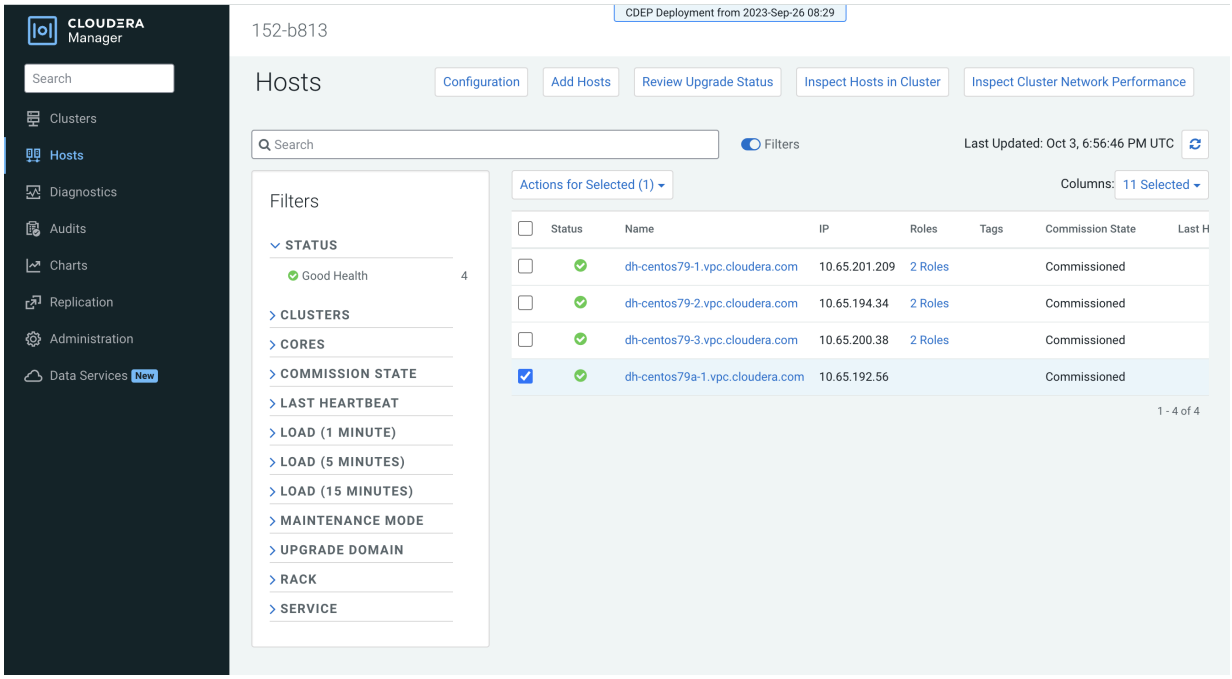
6. In the ECS cluster, click Hosts, then click Add Hosts. Select Add hosts to Cluster, then click Continue.



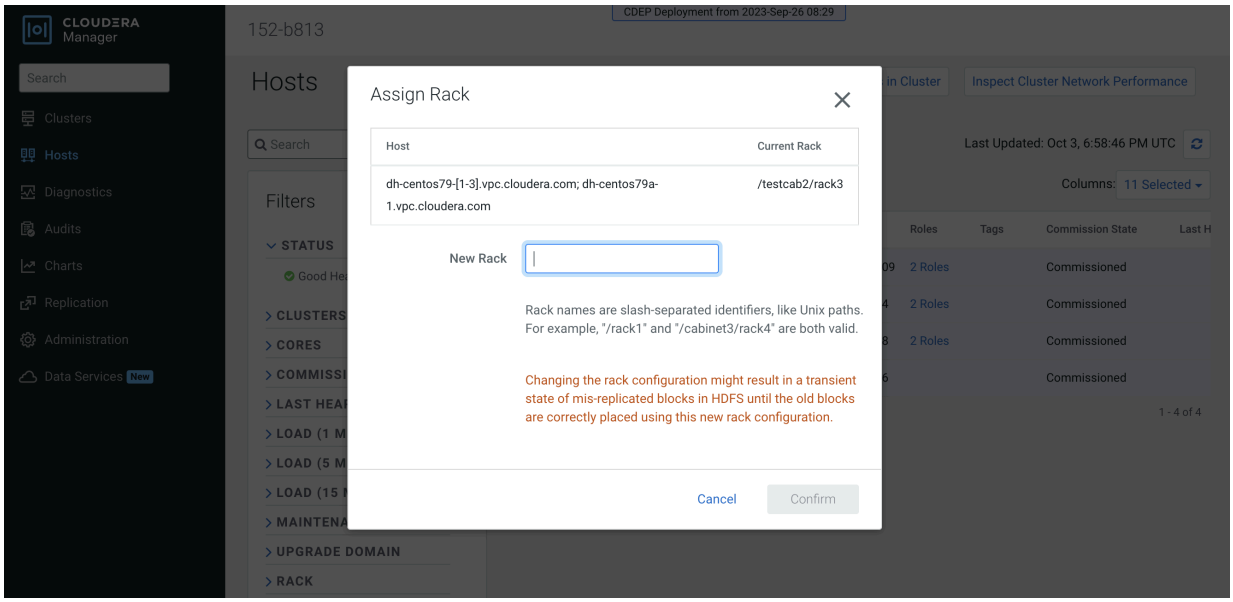
7. On the Specify Hosts page, select the new host, then click through the rest of the Add Hosts wizard.



8. After the Add Host wizard is completed, the new host appears on the ECS cluster Hosts page.



9. You can use the Assign Rack popup to view the rack assignments for the ECS cluster hosts and confirm that the rack name for the new host matches the rack name of the other cluster hosts.



ECS unified time zone

You can synchronize the Embedded Container Service (ECS) cluster time zone with the Cloudera Manager Base time zone.

In CDP Private Cloud Data Services versions earlier than 1.5.2, containers running on an ECS Kubernetes cluster did not inherit the time zone settings from the Cloudera Manager Base host. In most cases, Kubernetes containers use Coordinated Universal Time (UTC) by default.

In Private Cloud Data Services 1.5.2 and higher versions, you can unify the time zone in the ECS cluster with the Cloudera Manager Base time zone. All workload pods in the ECS cluster run under the Cloudera Manager time zone, and workload logs on the ECS cluster are correlated with the Cloudera Manager Base logs. Timestamp-related SQL queries are also correlated.

- Unified time zone is enabled by default for new CDP Private Cloud Data Services 1.5.2+ installs.
- When upgrading from earlier versions of CDP Private Cloud Data Services to 1.5.2+, unified time zone is disabled by default to avoid affecting timestamp-sensitive logic.

You can enable or disable unified time zone using the following script in the ECS parcel:

```
bash /opt/cloudera/parcels/ECS/k8tz-webhook/configure-k8tz-webhook.sh -h
```

This script modifies the k8tz webhook settings.

Syntax:

configure-k8tz-webhook.sh [-i|-h]

Options:

- i – This option enables the unified time zone feature
- No options – To disable the unified time zone feature, run the configure-k8tz-webhook.sh script without any options.
- Use the -h flag to print Help information

To complete the process of enabling the unified time zone feature:

- Restart the workload pods where you want the Cloudera Manager Server timezone to be applied.

-OR-

- Initiate an ECS cluster rolling restart. This will inject the time zone information into all workload pods.

When the unified time zone feature is disabled, all running pods are not affected. To apply the new disabled setting so they run with the default UTC time zone, a pod restart or a rolling restart is required.

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:

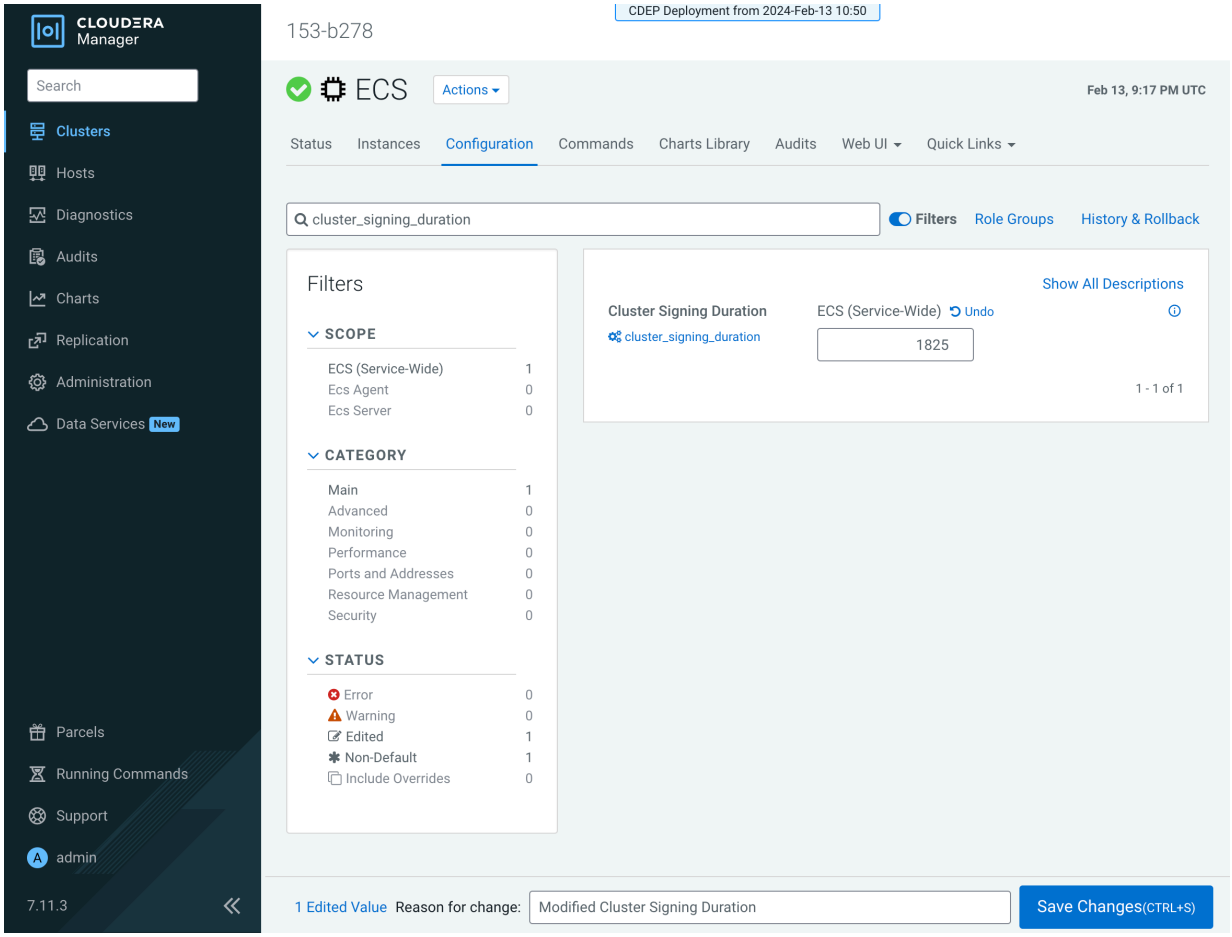
This topic only applies to internal certificates within ECS. It does not apply to the ingress controller certificate.

- 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
```

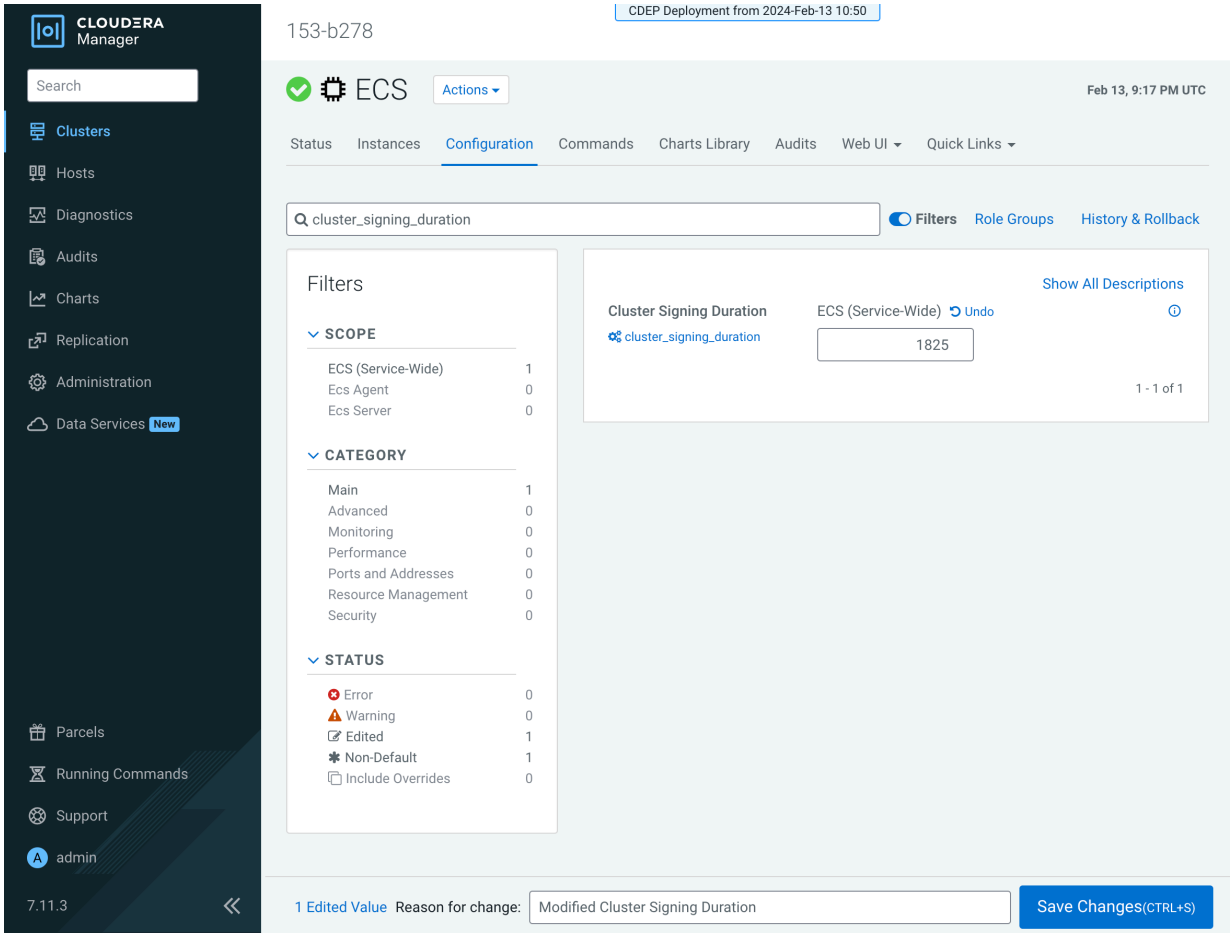
```
CGROUP_ROOT_BLKIO=/sys/fs/cgroup/blkioCGROUP_ROOT_CPU=/sys/fs/cgroup/cpu,c
puacctKEYTRUSTEE_KP_HOME=/usr/share/keytrustee-keyprovider
CLOUDERA_MYSQL_CONNECTOR_JAR=/usr/share/java/mysql-connector-java.jarCMF_
SERVER_ROOT=/opt/cloudera/cm
CGROUP_ROOT_CPUACCT=/sys/fs/cgroup/cpu,cpuacctCDH_FLUME_HOME=/usr/lib/f
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
```

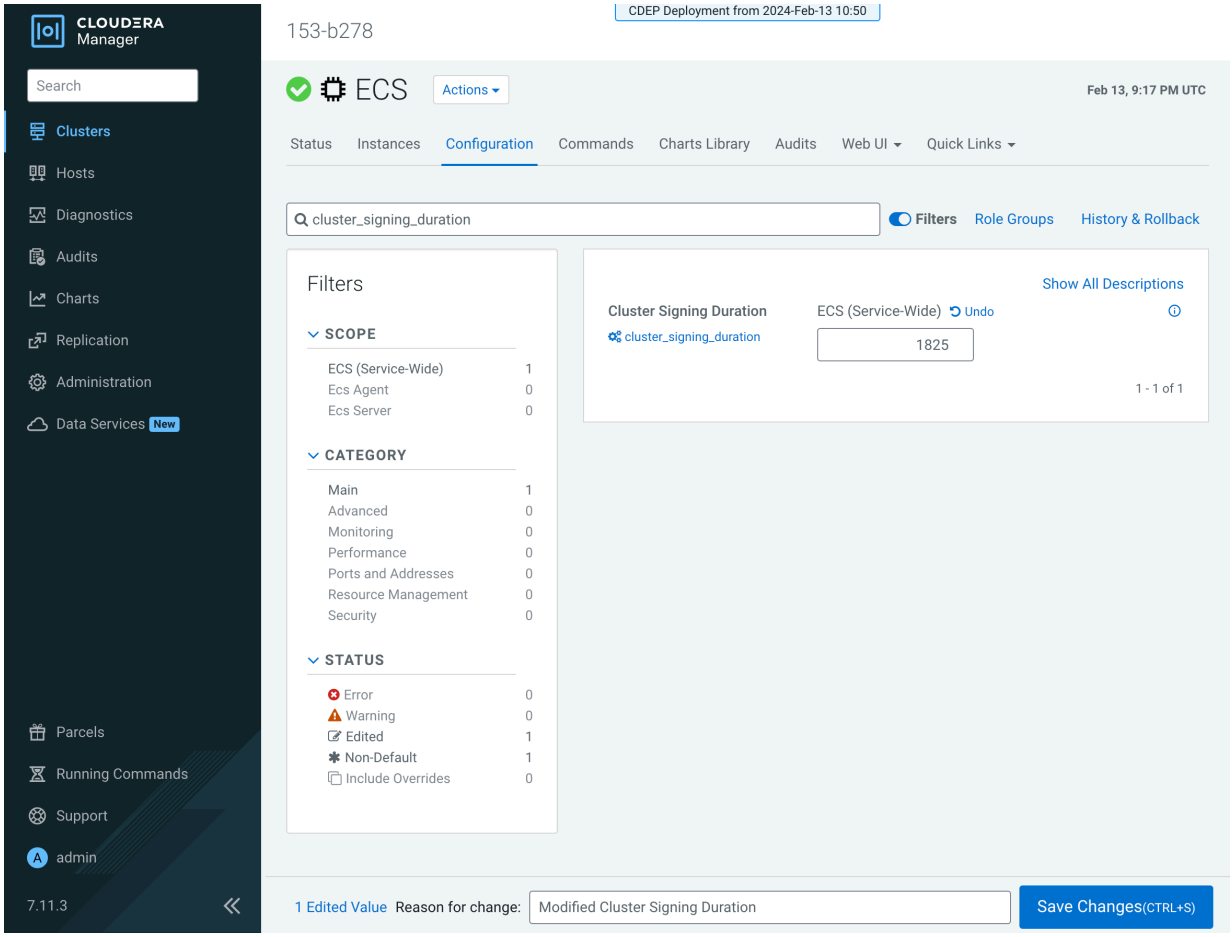
```
<snip!>
```

```
[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:
```

Configuring multiple Base clusters with one ECS cluster

You can configure one Embedded Container Service (ECS) cluster to work with multiple CDP Private Cloud Base clusters managed by separate instances of Cloudera Manager. In order to do this you must first create a combined truststore .pem file that contains the ECS Control Plane truststore .pem file appended with the certificate files of each of the CDP Private Cloud Base clusters.

About this task

Use the following steps to configure one ECS cluster to work with multiple CDP Private Cloud Base clusters:

1. Append the ECS Control Plane truststore .pem file with the certificate files from the additional CDP Private Cloud Base clusters.
2. Register an ECS environment with each of the additional CDP Private Cloud Base clusters.
3. Create data services within each environment.

Step 1: Append the ECS Control Plane truststore .pem file with the certificate files from the Base clusters

1. On the ECS Control Plane, run the following kubectl command to get the contents of the configmap:

```
kubectl get configmap cdp-private-installer-truststore -n cdp -o yaml >
cdp-private-installer-truststore.yaml
```

2. Copy the truststorePEM content, decode it, and store it in a file. For example:

```
echo LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURhakNDQWxJQ0NRRG5iNnhmK0d
QR1l6QU5CZ2txaGtpRzl3MEJBUXNGQURCWk1Rc3dDUV
lEVlFRR0V3SlYKVXpFTE1Ba0dBMVVFQ0F3Q1EwRXhDekFKQmdOVkJBY01BbE5ETVEwd0N3WU
RWUVFLREFSRFRFUlNNUXd3Q2dZRApWUVFMREFOQ1RGS
XhFekFSQmdOVkJBTU1DaW91YUhkNExuTnBkR1V3SGhjTk1qTXhNVEV3TVRRME1qUXdXaGNOC
k1qUXhNVEE1TVRRME1qUXdXakFWTVJNd0VRWURWUVFE
REFvcUxtaDNlQzV6YVhSbE1JSUJvakFOQmdrcWhraUcKOXcwQkFRUZBQU9DQVk4QU1JSUJp
Z0tDQVlFQS9lZkJtK05IQTdWUTF1M05qK3ZoRGFRV0p
```

```
JcUhFbVcxOFlpYgpBQUdiYmlvYi9YYnY0aTRINU81MXV3SjJ1cWowaktUM3dBU3l0UG0yS0p
1RE9vVXMveWhJc0xuK3VOWlMzd292CkNxSk5RcWpRT3
N2RUVITU5ZZ3JOWExMclhlbHZHTXl4aG16bVFlSEhHTkZhcldENVkwd1laMVVIaG00a0pUUT
UKTFhoZm1JVjJlTUJieE4ySVB2WU1TV1AvYmo4ekF3a
k50OHQvVUhhaFRTeWljUktEWitsMGxoeGt0cHpzdmxmcQo4eXNCVTBBQ2MvbWp2bGNWS0xyN
VVRSTRadVNFb2ZRK1QyaEpITEZNQ0N4bFJvcWN5aFo0
QmtlZmZwaUhIOGJHCm9kd2tSaHRRMVFJcFFxSklCLytCOWNZbkFjYlBFaHlXekh1TGlqakl5
VTZOYWZ3SmpoTG1SVmptRmpWNzNvZmgKanJ4V1BtVyt
FSDJZODRWK3RpOVdIZE5LQW9KNzU4bzZaSmJsc3ZBRVBNVytBVmw2clFMTTFPZXN1UTNtczc
xMwpWOENObFBWVEQ0UGdpaythOG1YV3FWZkVZN2F1V3
N1YnIwUkIyeFliWHBHd21WdWxrSjdYRURHOEpmN2hFNzRqCkRhMlJaeWN5YXdScGF3SXV2V1
kwWGtoSktOOTNBZ01CQUFFd0RRWUpLb1pJaHZjTkFRR
UxCUUFEZ2dFQkFDcTcKSDU5R2lnKy9iUVB3enhmUmF6d1hXM09mT3M1UjNnU0hGeDRmS1BXV
lN5TjEwaW5Obmdxejd4R2dYVnBpRDdWNApQRGVXZFRZ
MjdHN2w3ZHBjek1FS2ptN25XOUp3RW05S3dyRndWRWh0OWEzNjVvUnhqTzA3Y09VanZYaEwy
dkx1Cnk1eHRYZlJyZXlPalNmZDVxcnlKVlBoMDBHb0N
UWTViMy9wK25saWJUUmNkY29mQkFTU0VhbnhaVDJoc1B2V3kKSG9PVkVGSm1rTnVxRHJhS2Y
ySlFxRnR4aGs0MFIvUW9LVUpKUTgzUWIxZHBmWWVCdE
9lWXRVNExmQWV3Y0RuRwpFWUQvYVplblgwU2cxRTRoRS9NaUNFN2R6ZzY4TVVPeWVBV1pCel
JuMHBEZ1VtanpTOUNndi9GQ240MjV0QnR5Cis5anY1W
it3TVNkd1VZL2VudEE9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KLS0tLS1CRUdJTiBDR
VJUSUZJQ0FURS0tLS0tCk1JSURlekNDQW1PZ0F3SUJB
Z0lVQWRidE11Q3JycVRMYlUzRzhPakZRUW5YNGY4d0RRWUpLb1pJaHZjTkFRRUwKQlFBd1dU
RUxNQWtHQTFVRUJoTUNWVk14Q3pBSkJnTlZCQWdNQWt
OQk1Rc3dDUVlEVlFRSERBSlRRekVOTUFzRwpBMVVFQ2d3RVEweEVVakVNTUFvR0ExVUVDd3d
EUWt4U01STXdFUVlEVlFRRERBb3FMbWgzZUM1emFYUm
xNQjRYCkRUSXpNVEV4TURFek1UTXpOVm9YRFRJMU1URXdPVEV6TVRNek5Wb3dXVEVMTUFrR0
ExVUVCaE1DVlZNeEN6QUoKQmdOVkJBZ01Ba05CTVFzd
0NRWURWUVFIREFKVFF6RU5NQXNHQTFVRUNnd0VRMHhFVWpFTU1Bb0dBMVVFQ3d3RApRa3hTT
VJNd0VRWURWUVFEREFvcUxtaDNlQzV6YVhSbE1JSUJJ
akFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBCk1JSUJDZ0tDQVFFQXczQXBYeXg4dkxXSVZq
SlpLZzNpb29XcGdtNjZwN2gxWCtRWUVVZ0Q0VEc3dkZ
2OGNUckkKdzlaZ1VpcW1zUTVJRlZxRk5lcEFpSFBteUxscDl1d1RhTEthdm9IZ2pXU0p1K2d
waUdiMHJiR1hkM3ltYkw5Rwp2Sm1pNmtPZW9SeHpQbk
N5SVVEa3NmU3kzdE5pWlNRRFRubmhUWk9Zc2tmbDdZK1VYaVJVS2NBNExkWTBWSTVJCnpmRl
R0cW5qM0o4SnJ6d0dJd1NoK0ZNdHRyWFQ5WFI5bzVpL
0M2cWh0L1JwbEx3QTB6ZVlYSDhkNjl2Ykw4T1EKemREeXZlcmptRXZjS3F1bGo4NU1CSTZwc
VRGb21QcEp5VVlxS0cwN2U1WDN0QmZiVzk2QXdYT1BT
SFd0QlpndwpyeTVFbzRxWVRJMGZmYlFCS3ZIVElzYTd3T0xmRzAvK3J3SURBUUFCb3pzd09U
QUxCZ05WSFE4RUJBTUNCREF3CkV3WURWUjBsQkF3d0N
nWUlLd1lCQlFVSEF3RXdGUVlEVlIwUkJBNHdESUlLS2k1b2QzZ3VjMmwwWlRBTkJna3EKaGt
pRzl3MEJBUXNGQUFPQ0FRRUFtKzFZUlg5M2k1Q1FPQl
FIVVZ2Y2M1OWFMb2Y3SnJxcGNaN0NOaGJXMzc4Zgo3RTNpTjhBY1BNQ0dvZllleWFrblQxVl
kwdDNiVXhtSTFSdXdEUXNDU3U1MmlhYnhIVUhrOFBEQ
jk5NTRxL3RtCkh4MXpVR0VURkZaZHdkb0dDMk14Ui9WdU9wbExza2hEc0ZJZmpaZC81clVrL
1QvMUxUaC8zMExBbGhPVzNtek8KZFJWWC9LR2QyWGZ3
SFNzQ3FRTFk4WGZQM0d3WHgrTmVUY09vTEQycXYvYW1kMnY1dlVtdXpONzErZjR3bXVvbwpa
Z1JiYk9OSkMvdzVzV3MvWVRaODd1M1JNUWExd2gvckl
YMk1QMzNTMG1SeHJkSXlpeGMxamF6ZTYxWmRUUnk5Ck9NQ2RmZEpGNFE1RndmODdWSWpYZXd
PemdQVnFJVGVNVW1vcy9HR0p0UT09Ci0tLS0tRU5EIE
NFUlRJRklDQVRFLS0tLS0= | base64 -d > cdp-private-installer-truststore.pem
```

3. Obtain the truststore .pem file from the first additional Cloudera Manager host from /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem or /opt/cloudera/CMCA/trust-store/cm-auto-global_cacerts.pem and copy the contents.
4. Append the cdp-private-installer-truststore.pem file created previously with the contents of the Cloudera Manager .pem file.
5. Repeat the previous two steps for all additional Cloudera Manager hosts you would like to register environments with.

6. Log in to the ECS cluster Management Console and click Administration > CA Certificates. Select Datalake in the CA Certificate Type drop-down, click Choose File, then select the appended cdp-private-installer-truststore.pem file and click Upload. Click Save to save your changes.

   You can also use the following CLI commands to upload the cdp-private-installer-truststore.pem file and update the global truststore with the encoded certificate file content:

   ```
   cat cdp-private-installer-truststore.pem | base64
   cdp environments --set-environment-setting --settings truststorePEM=<base6
   4 encoded CM cert> --no-verify-tls
   ```

Step 2: Register an ECS environment with each of the additional Base clusters

1. Log in to the ECS cluster Management Console and Register an environment for the first additional Base cluster using the applicable Cloudera Manager URL and credentials.
2. Repeat the previous step for the rest of the additional Base clusters.

Step 3: Create data services within each environment

Refer to the following topics to create the data services of your choice in each environment:

- Adding a Cloudera Data Engineering service
- Activate ECS environments (CDW)
- Provision an ML Workspace