Cloudera on Premises Data Services 1.5.5

Installing Cloudera Data Services on premises on the Cloudera Embedded Container Service

Date published: 2023-12-16
Date modified: 2025-06-06

https://docs.cloudera.com/

Legal Notice

Contents

Requirements

Software Support Matrix for Cloudera Embedded Container Service

This support matrix lists the supported software for the Cloudera on premises cluster and the Cloudera Data Services on premises containerized cluster when installing using the Cloudera Embedded Container Service.

Base Cluster	Version	• Cloudera Manager 7.13.1 CHF3 • 7.1.9, 7.1.9 SP1 CHF5, 7.1.7 SP3 CHF10
	Base OS	• See Private Cloud Base OS requirements
	TLS	• AutoTLS (Custom CMCA) • AutoTLS (Self-signed)
	Kerberos	• AD • FreeIPA
	JDK	• See Java Requirements
	Custom service principals	• Not supported
	Data Lake Storage	• HDFS (All) • Ozone • Iceberg v2 (with HDFS and Ozone)
	Base DB (HMS access from CDW Data Services)	• Oracle 19c • Oracle 19.9 • MySQL 8 • MySQL 5.7 • MariaDB 10.2 • MariaDB 10.3 • MariaDB 10.4 • MariaDB 10.5 • MariaDB 10.6 • Postgres 12 • Postgres 13 • Postgres 14 • Postgres 15 • Postgres 16
Containerized Cluster	ECS OS	• RHEL 8.10, 9.3, 9.4, 9.5 • OEL (RHCK Kernel Only) 8.10, 9.3, 9.4, 9.5 • Rocky Linux 8.10, 9.3, 9.4, 9.5
	Control Plane Metadata DB	• Embedded
	Vault	• Embedded
	Docker registry type	• Secure registry with self signed CA certs (pwd protected + self signed certs) • Embedded (Cloudera Embedded Container Service only. Not recommended)
	NFS	• Embedded • External

	IdP	• FreeIPA • ActiveDirectory (LDAP) • OpenLDAPs
	Network Access	• Airgap • Internet • HTTP proxy (Cloudera AI)
	TLS	• Manual - CA signed • ESC server signed (Cloudera Embedded Container Service only)
	Ingress Cert	ECS Default Non-default OCP Default
GPU Nodes	OS	• RHEL 8.8, 8.9, 8.10, 9.3, 9.4, 9.5

Cloudera Base on premises Software Requirements

The software requirements for the nodes on which Cloudera Data Services on premises are deployed are identical to Cloudera Base on premises.

Your Cloudera Base on premises cluster must have the operating system, JDK, database, Cloudera components, and Cloudera Runtime version compatible with Cloudera Data Services on premises. You must first set up the Cloudera Base on premises cluster, then you can install the on premises Containerized cluster.

For more information about the requirements for the Cloudera Base on premises cluster, see the Cloudera Base on premises section of the Requirements and Supported Versions and the Cloudera Base on premises section of the Software Support Matrix for Cloudera Embedded Container Service on page 4.

The following Cloudera Base on premises cluster services are required to fully access the Data Services:

- Zookeeper
- HBase
- Hive Metastore (HMS)
- Hive on Tez (needed for using compaction)
- Ranger
- Atlas
- HDFS
- Ozone
- YARN
- Kafka
- Solr

In addition to this, the hive user should be able to create and list an Ozone bucket. For information about creating and listing ozone bucket, see *Managing buckets*.

Note:

- Ozone is not a mandatory requirement for installing Cloudera AI without the model registry.
- Configure Atlas with Hive for Cloudera Data Engineering to work properly.

Related Information

Managing buckets

Cloudera Data Services on premises Hardware Requirements

Minimum and recommended hardware to successfully install and run Cloudera Data Services on premises.

In addition to the resources required for the Cloudera Control Plane, additional resources will be required depending on the Data Service(s) you intend to run. Minimum and recommended additional resource requirements for each of the Data Services can be found in the pages below. To calculate the total minimum or recommended resource requirements for your Cloudera Data Services on premises cluster, add the resources required for the Control Plane to the total minimum or recommended additional resources for your chosen Data Service(s).

You can also use the Cloudera Data Services on premises Spreadsheet to model the number and specification of hosts required for a deployment. See How to use the Cloudera Data Services on premises sizing spreadsheet on page 9.

Requirements for HA and Non-HA Cloudera Control Plane

Standard resource mode requirements for standalone HA and Non-HA Cloudera Control Plane.

Component	Minimum	Recommended
Node Count	1 (Non-HA)	3 (HA)
CPU	16 cores	32 cores (per node)
Memory	32 GB	64 GB (per node)
Storage	300 GB	1 TB (per node)
Network Bandwidth	1GB/s to all nodes and base cluster	1GB/s to all nodes and base cluster

Additional resource requirements for Cloudera Data Warehouse

Standard resource mode requirements for Cloudera Data Warehouse.

The following table lists the minimum and recommended compute (processor), memory, storage, and network bandwidth required for each OpenShift or worker node using the Standard Resource Mode for production use case. Note that the actual node still needs some extra resources to run the operating system, Kubernetes engine, and agent on .

Component	Minimum	Recommended
Node Count	4	10
CPU per worker	16 cores [or 8 cores or 16 threads that have Simultaneous Multithreading (SMT) enabled]	32+ cores (can also be achieved by enabling SMT)
Memory per worker	128 GB per node	384 GB* per node
FAST (Fully Automated Storage Tiering) Cache - Locally attached SCSI device(s) on every worker. Preferred: NVMe and SSD. OCP uses Local Storage Operator. ECS uses Local Path Provisioner.	1.2 TB* SATA, SSD per host	1.2 TB* NVMe/SSD per host
Network Bandwidth	1 GB/s guaranteed bandwidth to every	10 GB/s guaranteed bandwidth to every node

* Depending on the number of executors you want to run on each physical node, the per-node requirements change proportionally. For example, if you are running 3 executor pods per physical node, you require 384 GB of memory and approximately 1.8TB (600GB per executor) of locally attached SSD/NVMe storage for FAST Cache.

Important: When you add memory and storage, it is very important that you add it in the increments as follows:

- Increments of 128 GB of memory
- Increments of 600 GB of locally attached SSD/NVMe storage

If you add memory or storage that is not in the above increments, the memory and storage that exceeds these increments is not used for executor pods. Instead, the extra memory and storage can be used by other pods that require fewer resources.

For example, if you add 200 GB of memory, only 128 GB is used by the executor pods. If you add 2 TB of locally attached storage, only 1.8 TB is used by the executor pods.

Additional resource requirements for Cloudera Data Engineering

For standalone Cloudera Data Engineering, Cloudera recommends three nodes (one master and two workers) with the following minimum memory, storage, and hardware requirements for each node:

Component	Minimum	Recommended
Node Count	2	4
CPU	24 cores for CDE workspace (base and virtual cluster) and 12 cores for workload	24 cores for CDE workspace (base and virtual cluster) and 32 cores (you can extend this depending upon the workload size)
Memory	64 GB for CDE workspace (base and virtual cluster) and 32 GB (you can extend this depending upon the workload size)	64 GB for CDE workspace (base and virtual cluster) and 64 GB (you can extend this depending upon the workload size)
Storage	700 GB block storage	700 GB block storage
Network Bandwidth	1 GB/s to all nodes and base cluster	10 GB/s to all nodes and base cluster

Important: Optionally, if you want to use GPU in Spark, the Spark RAPIDS library is validated and certified by Nvidia for *NVIDIA P100, V100, T4 and A2/A10/A30/A100* GPU architecture.

Cloudera Data Engineering Service and Virtual Cluster requirements

- Cloudera Data Engineering Service requirements: Overall for a Cloudera Data Engineering service, it requires 110 GB Block PV or NFS PV, 10 CPU cores, and 30 GB memory.

Table 1: The following are the Cloudera Data Engineering Service requirements:

Component	vCPU	Memory	Block PV or NFS PV	Number of replicas
Embedded DB	4100 m	9 GB	100 GB	1
Admission Controller	250 m	512 MB	--	1
Config Manager	500 m	1 GB	--	2
Authz	1100 m	2 GB	--	1
Dex Downloads	350 m	1.5 GB	--	1
Knox	350 m	2 GB	--	1
Management API	1100 m	3 GB	--	1
NGINX Ingress Controller	200 m	1114 MB	--	1
Tgt Generator	100 m	1 GB	--	1
FluentD Forwarder	250 m	512 MB	--	1 to 5
Grafana	350 m	1.5 GB	10 GB	1
Keytab Management	350 m	512 MB	--	1

Component	vCPU	Memory	Block PV or NFS PV	Number of replicas
Data Connector	350 m	1.5 GB	--	1
Total	9350 m	28.71 GB	110 GB	

- Cloudera Data Engineering Virtual Cluster requirements:
  - For Spark 3: Overall storage of 400 GB Block PV or Shared Storage PV, 7 CPU cores, and 26 GB per virtual cluster.
  - For Spark 2: If you are using Spark 2, you need additional 600 m CPU, 5.5 GB memory and 100 GB storage, that is, the overall storage of 500 GB Block PV or Shared Storage PV, 8 CPU cores, and 32 GB per virtual cluster.

  Important: The Cloudera Data Engineering service and virtual cluster requirements does not include workloads. See the below workload information on the additional resources based on workload.

Table 2: The following are the Cloudera Data Engineering Virtual Cluster requirements for Spark 3:

Component	vCPU	Memory	Block PV or NFS PV	Number of replicas
Airflow API	450 m	1636 MB	100 GB	1
Airflow Scheduler	1100 m	2560 MB	100 GB	1
Airflow Web	350 m	1.5 GB	--	1
Runtime API	750 m	1.5 GB	100 GB	1
Livy	3100 m	14 GB	100 GB	1
SHS	350 m	1.5 GB	--	1
Pipelines	350 m	1.5 GB	--	1
Total	6450 m	25.1 GB	400 GB	

- Workloads: Depending upon the workload, you must configure resources.
  - The Spark Driver container uses resources based on the configured driver cores and driver memory and additional 40% memory overhead.
  - In addition to this, Spark Driver uses 110 m CPU and 232 MB for the sidecar container.
  - The Spark Executor container uses resources based on the configured executor cores and executor memory and additional 40 % memory overhead.
  - In addition to this, Spark Executor uses 10 m CPU and 32 MB for the sidecar container.
  - Minimal Airflow jobs need 200 m CPU and 328 MB memory per Airflow worker.

Additional resource requirements for Cloudera AI

Standard resource mode requirements for standalone Cloudera AI. Node count should not be a limiting factor assuming the other memory and CPU minimums are reached.

Component	Minimum	Recommended
Node Count	1	1 per workspace + additional nodes depending on expected user workloads
CPU	32 Cores Per Workspace+ additional Cores depending on expected user workloads	48 Cores Per workspace + additional Cores depending on expected user workloads
Memory	128 GB + additional memory depending on the expected workloads	256 GB Per Workspace + additional memory depending on the expected workloads

Component	Minimum	Recommended
Storage	Set up ECS/Longhorn with SSDs with the recommended cumulative 2600 GB of Block storage. For Production environments, it is strongly recommended to setup an External NFS environment with at least 1000 GB of NFS storage with additional Block storage based on project file sizing. The total (not per node) storage needed only for Cloudera AI in Cloudera Embedded Container Servicewithout disaster recovery (DRS) is 1300 Gi per workbench with the external NFS. If the Cloudera AI Workbench uses internal NFS, the total minimum storage needed per workbench is 3300Gi. Considering the DRS and single backup of the workbench, the total storage needed is 1300 Gi * 2 = 2600 Gi for the workbench with external NFS. If the workbench uses internal NFS, the total storage needed is 6600Gi.	
Network Bandwidth	1GB/s to all nodes and base cluster	1GB/s to all nodes and base cluster

Note:

The storage calculation accounts for a single backup of the workbench. If additional backups are required, the storage requirements will adjust accordingly.

Additional Resources for User Workloads:

Component	Minimum	Recommended
CPU	1 Core per concurrent workload	2–16 cores per concurrent workload (dependent on use cases)
Memory	2 GB per concurrent workload	4–64 GB per concurrent workload (dependent on use cases)

Additional resource requirements for Cloudera AI Inference service

Consider the following resource needs for Cloudera AI Inference service.

Table 3: Additional resource requirements for Cloudera AI Inference service

Component	Required resources
CPU	4 CPU cores
Memory	7 GB

Additional resource requirements for Cloudera AI Registry

Consider the following resource needs for Cloudera AI Registry.

Table 4: Additional resource requirements for Cloudera AI Registry

Component	Required resources
Memory	50 Gi

How to use the Cloudera Data Services on premises sizing spreadsheet

You can use the sizing spreadsheet to model the hardware requirements for a Cloudera Data Services on premises deployment.

Overview

The Cloudera Data Services on premises Sizing spreadsheet is a spreadsheet that you can use to model the quantity and specifications for worker hosts required in a Cloudera Data Services on premises deployment.

This spreadsheet is intended to use information about workloads you are planning to run and hardware specifications for worker nodes to arrive at an approximate number of worker nodes required for your deployment. Due to the

complexity of estimating workloads, Cloudera recommends you review any sizing or purchasing decisions with Cloudera Professional Services before committing to those decisions.

How to access the spreadsheet

You can access the spreadsheet here: Cloudera Private Cloud Data Services Sizing. The file is in Microsoft Excel format. You can open the file in Excel, or upload it to Google Sheets.

There are three tabs in the spreadsheet. You will make your inputs only on the Worker Node Totals tab. Do not modify the following tabs (these tabs contain data used to calculate values in the spreadsheet and should not be modified):

- Component Lookup
- K8s Resources

Important: Do not modify any cells except for the ones indicated below. Modifying the formulas in other cells will result in inaccurate calculations.

Workload inputs

The spreadsheet calculates the total amount vcores, RAM, and storage required based on information you enter about the combined workloads you intend to deploy. Then based on the hardware specifications entered, calculates the number of worker nodes required, which is displayed in cell F27.

The following sections describe values you must enter into the spreadsheet. Values are required for each Data Service you intend to deploy, and values to enter for the hardware specifications for your worker nodes.

Cloudera Control Plane monitoring

Label	Cell	Description
Cloudera Control Plane Monitoring	B3	Increment this number by one for each environment.

Cloudera Data Warehouse

If you will deploy Cloudera Data Warehouse, on the Worker Node Totals tab, enter the following information:

Label	Cell	Description
CDW Data Catalog (min 1 per env)	B5	Enter the number of Data Catalogs you will need in your deployment. You must have at least one Data Catalog.
CDW LLAP warehouses	B6	Enter the number of LLAP warehouses you will need for each Virtual Warehouse in your deployment.
-- LLAP Executors	B7	Enter the total number of LLAP Executors you will need in your deployment.
CDW Impala warehouses	B8	Enter the number of CDW Impala warehouses for each Virtual Warehouse you will need in your deployment.
-- Impala Coordinators (2 x for HA)	B9	Enter the number of Impala Warehouses you will need in your deployment. If you have enabled high availability, enter twice the number of Warehouses.
-- Impala Executors	B10	Enter the number of Impala Executors you will need in your deployment.
CDW Cache	B11	Enter the amount of CDW Cache space for each coordinator and executor (Default 600)

Label	Cell	Description
Data Viz - small instances	B12	Enter the size selected when creating a Data Visualization instance.
Data Viz - medium instances	B13	
Data Viz - large instances	B14	

For more information about sizing Cloudera Data Warehouse deployments, see:

- Standard resource mode requirements
- Low resource mode requirements

Cloudera AI

Sizing for a Cloudera AI deployment depends on the number of concurrent jobs you expect to run and the number of Workspaces you provision.

Label	Cell	Description
Cloudera AI Workbench (min of 1)	B16	Enter the number of workspaces you need in your deployment.
-- Cloudera AI small concurrent sessions	B17	Enter the number of concurrent small-sized sessions you intend to run.
-- Cloudera AI average concurrent sessions	B18	Enter the number of concurrent average-sized sessions you intend to run.

For more information about sizing the Cloudera AI service, see the following topics:

- Additional resource requirements for Cloudera AI.
- (OCP) Cloudera AI requirements
- (Cloudera Embedded Container Service) Cloudera AI requirements

Cloudera Data Engineering

Label	Cell	Description
CDE Service (min/max 1 per cluster)	B20	Enter the number of Cloudera Data Engineering clusters you will need in your deployment.
CDE Virtual Cluster	B21	Enter the number of Cloudera Data Engineering Virtual Clusters you will need in your deployment.
-- CDE Small concurrent jobs	B22	Enter the number of concurrent small-sized jobs you intend to run.
-- CDE Average concurrent jobs	B23	Enter the number of concurrent average-sized jobs you intend to run.

For more information about sizing the Cloudera Data Engineering service, see Additional resource requirements for Cloudera Data Engineering.

Worker node hardware specifications

Based on the inputs you supplied for your workloads, the spreadsheet totals the number of vcores, RAM, and storage required for the cluster in cells C20-C26. Then, based on the worker node hardware specifications you enter in cells B26-B29, divides the totals for vcores, RAM and storage by each of the worker node specifications to arrive at the required number of nodes for vcores, RAM and storage shown in cells D5-D29. The final number, in cell E27 chooses the higher value of these cells.

You may notice that the calculated values in cells D26 and D27 are different. This indicates that some nodes are oversubscribed for RAM or vcores. Adjust the hardware specifications for CPU and RAM until the two cells are closer together in value. Changing these values may also change the calculated number of worker nodes.

Label	Cell	Description
CPU recommend 40+ cores (80 vcores)	B27	Enter the number of vcores for each worker node.
RAM (GB) recommend 415 GB RAM	B28	Enter the amount of RAM, in gigabytes, for each worker node.
Disk (GB) Block (OCP CSI block, Cloudera Embedded Container Service Longhorn)	B29	Enter the number of gigabytes Block required for: - OpenShift Container Platform: CSI block - Cloudera Embedded Container Service: Cloudera Embedded Container Service Longhorn
Disk (GB) Fast Cache for Cloudera Data Warehouse (nvme,ssd)	B30	Enter the number of gigabytes of Fast Cache used in Cloudera Data Warehouse.
Cloudera Control Plane Block Overhead per host (300 to 1024)	B31	Enter the Control Plane block overhead
NFS (GB) (choose 1 from below)	B33	Enter required storage in either cell B34 or cell B35
-- Embedded nfs - (subtract from Block provider) non-prod	B34	Enter the number of gigabytes storage for an embedded NFS.
-- External nfs	B35	Enter the number of gigabytes of storage for an External NFS.
Cloudera Embedded Container Service Master Node requires 1 for non HA - 3 for HA If you are using the Cloudera Embedded Container Service, you will also need to provision a host for the Cloudera Embedded Container Service Master Node (a node running the ECS Server component). The values described here contain Cloudera's recommendations for specifications for the Cloudera Embedded Container Service Master node.	B38	Minimum: 16 vcores Recommended: 32 vcores
	B39	Minimum: 32 GB RAM Recommended: 64 GB RAM
	B40	Minimum: 300 GB HDD (This amount is adequate for a proof-of-concept cluster.) Recommended: 1 TB HDD

Docker repository access

You must ensure that the cluster has access to the Docker Container Repository in order to retrieve the container images for deployment.

There are several types of Docker Repositories you can use:

Embedded Repository

During installation, a Docker daemon is provisioned to act as the Repository. Passwords and certificates are auto generated. No additional set up is needed. Images are copied to the repository during installation. During upgrades, only the new and changed images are copied. Copying images generally takes one to two hours.

It is important to note that the Embedded Repository can be a single point of failure. If the node that runs the Docker Repository fails or becomes unavailable, some cluster functionalities might become

unavailable. Moving the Docker Repository to another node is a complex process and will require engaging Cloudera Professional Services.

Cloudera Repository

Using the Cloudera Repository requires that the cluster have internet connectivity to the Cloudera public repository. Using the Cloudera Repository is the fastest option.

The Cloudera-hosted Docker Repository option may increase the time required to deploy or start the services in the cluster. Cloudera generates Docker Repository credentials that are identical to your paywall credentials. Refer to your welcome letter for the credentials or use the credential generator on cloudera.com to generate credentials from your license key.

This option is best suited for proof-of-concept, non-production deployments or deployments that do not have security requirements that disallow internet access.

Custom Repository

A Custom Repository is a repository that you manage in your environment and can be Enterprise grade and highly available.

During installation and upgrade, a custom script is generated that you use to copy the images. Copying images can take 4 - 5 hours.

Only TLS-enabled custom Docker Registry is supported. Ensure that you use a TLS certificate to secure the custom Docker Registry. The TLS certificate can be self-signed, or signed by a private or public trusted Certificate Authority (CA).

Important: When using an Cloudera Embedded Container Service cluster, passwords must not contain the $ character.

Cloudera Data Services on premises Software Requirements

This release ships with Cloudera Manager 7.13.1 CHF 3. If you have an existing Cloudera Base on premises cluster set up using an earlier version of Cloudera Manager, you must first upgrade Cloudera Manager to version 7.13.1 CHF3.

For more information about specific software requirments, see the Software Support Matrix for Cloudera Embedded Container Service on page 4.

Additionally, you must perform the following:

- For Cloudera AI, you must install nfs-utils in order to mount longhorn-nfs provisioned mounts. The nfs-utils package is required on every node of the Cloudera Embedded Container Service cluster. Run this command `yum install nfs-utils` to install nfs-utils.
- If you have nodes with GPU, ensure that the GPU hosts have nVidia Drivers and nvidia-container-runtime installed. You must confirm that drivers are properly loaded on the host by executing the command `nvidia-smi`. You must also install the nvidia-container-toolkit package.
- You must have a minimum of one agent node for Cloudera Embedded Container Service.
- Set up Kerberos on these clusters using an Active Directory.
- Enable TLS on the Cloudera Manager cluster for communication with components and services.
- If you do not have entitlements, contact your Cloudera account team to get the necessary entitlements.
- The default docker service uses /docker folder. Whether you wish to retain /docker or override /docker with any other folder, you must have a minimum of 300 GiB free space.
- Create the folder before the start of the installation. For example: mkdir /ecs/docker.
- Ensure that all of the hosts in the Cloudera Embedded Container Service cluster have more than 300 GiB of free space in the /var/lib directory at the time of installation.
- The cluster generates multiple hosts and host based routing is used in the cluster in order to route it to the right service. You must decide on a domain for the services which Cloudera Manager by default points to one of the host names on the cluster. However, during the installation, you should check the default domain and override the

default domain (only if necessary) with what you plan to use as the domain. To override, create an A record with a wildcard. For Example: *.apps.APPDOMAIN

- You must install `nvidia-container-toolkit`. (`nvidia-container-runtime` migrated to `nvidia-container-toolkit` , see Migration Notice.) The steps for this are shown in the NVIDIA Installation Guide. If using Red Hat Enterprise Linux (RHEL), use `dnf` to install the package. For an example with RHEL 8.7, see Installing the NVIDIA Container Toolkit.
- Python 3.8 is required for Cloudera Manager version 7.11.3.0 and higher versions. Cloudera Manager agents will not start unless Python 3.8 is installed on the cluster nodes.

Modifying Access Control Lists (ACLs) for any Rancher or Kubernetes-related directories is strictly prohibited as it can cause permission issues, service failures, or security vulnerabilities. Unauthorized ACL changes may lead to:

- Failure of Rancher services to start properly.
- Kubernetes components encountering permission errors.
- Issues with upgrades, backups, or cluster operations.

Affected Directories

Below are the key Rancher and Kubernetes directories that must not have their ACLs modified:

Rancher-Specific Directories:

- /var/lib/rancher/ – Contains Rancher cluster data, configurations, and metadata.
- /etc/rancher/ – Stores Rancher configuration files, certificates, and settings.
- /var/log/rancher/ – Logs generated by Rancher services.

Kubernetes-Related Directories:

- /var/lib/kubelet/ – Stores node-level Kubernetes configurations and data.
- /etc/kubernetes/ – Holds Kubernetes API server, controller manager, and scheduler configurations.
- /var/lib/etcd/ – Contains the etcd database, critical for cluster state management.
- /var/log/pods/ – Stores logs for Kubernetes pods.
- /var/run/secrets/kubernetes.io/ – Used for service account authentication and tokens.

Best Practices

- Ensure that these directories maintain default ownership and permissions as configured by Rancher/Kubernetes.
- For troubleshooting, rely on logs and built-in diagnostics rather than altering file permissions.

By following these guidelines, you can avoid unexpected permission issues and maintain a stable and secure Rancher/Kubernetes environment.

Related Information

Software Support Matrix for Cloudera Embedded Container Service

Requirements for Cloudera AI on Cloudera Embedded Container Service

There are minimal requirements when using Cloudera AI on Cloudera Embedded Container Service.

Cloudera Embedded Container Service requirements for NFS Storage

Note: The Cloudera Embedded Container Service installation wizard offers a one-time option to download Cloudera AI Docker images. You cannot revisit this option after the installation of the cluster.

Cloudera managed Cloudera Embedded Container Service deploys and manages an internal NFS server based on LongHorn which can be used for Cloudera AI.

Note:

The recommended option for Cloudera AI on Cloudera Embedded Container Service clusters is to use external NFS.

Cloudera AI requires the nfs-utils package to be installed in order to mount volumes provisioned by longhorn-nfs. The nfs-utils package is not available by default on every operating system. Check if nfs-utils is available, and ensure that it is present on all Cloudera Embedded Container Service cluster nodes.

Alternatively, the NFS server can be external to the cluster, such as a NetApp filer that is accessible from the on premises cluster nodes.

For further information, see *Installation using the Cloudera Embedded Container Service*.

Standard resource mode requirements

Review the memory, storage, and hardware requirements for getting started with the Cloudera Data Warehouse service in standard resource mode on Red Hat OpenShift and Cloudera Embedded Container Service.

To get started with the Cloudera Data Warehouse service on standard resource mode, make sure you have fulfilled the following requirements:

- must be installed and running.
- must be installed and running. See Installing on OpenShift and Installing on ECS for more details.
- An environment must have been registered with on the . See Environments for more details.
- In addition to the general requirements, also has the following minimum memory, storage, and hardware requirements for each worker node using the standard resource mode:

Depending on the number of executors you want to run on each physical node, the per-node requirements change proportionally. For example, if you are running 3 executor pods per physical node, you require 384 GB of memory and approximately 1.8 TB of locally attached SSD/NVMe storage.

The following table lists the minimum and recommended compute (processor), memory, storage, and network bandwidth required for each OpenShift or worker node using the Standard Resource Mode for production use case. Note that the actual node still needs some extra resources to run the operating system, Kubernetes engine, and agent on .

Component	Minimum	Recommended
Node Count	4	10
CPU per worker	16 cores [or 8 cores or 16 threads that have Simultaneous Multithreading (SMT) enabled]	32+ cores (can also be achieved by enabling SMT)
Memory per worker	128 GB per node	384 GB* per node
FAST (Fully Automated Storage Tiering) Cache - Locally attached SCSI device(s) on every worker. Preferred: NVMe and SSD. OCP uses Local Storage Operator. ECS uses Local Path Provisioner.	1.2 TB* SATA, SSD per host	1.2 TB* NVMe/SSD per host
Network Bandwidth	1 GB/s guaranteed bandwidth to every	10 GB/s guaranteed bandwidth to every node

* Depending on the number of executors you want to run on each physical node, the per-node requirements change proportionally. For example, if you are running 3 executor pods per physical node, you require 384 GB of memory and approximately 1.8TB (600GB per executor) of locally attached SSD/NVMe storage for FAST Cache.

Important: When you add memory and storage, it is very important that you add it in the increments as follows:

- Increments of 128 GB of memory
- Increments of 600 GB of locally attached SSD/NVMe storage

If you add memory or storage that is not in the above increments, the memory and storage that exceeds these increments is not used for executor pods. Instead, the extra memory and storage can be used by other pods that require fewer resources.

For example, if you add 200 GB of memory, only 128 GB is used by the executor pods. If you add 2 TB of locally attached storage, only 1.8 TB is used by the executor pods.

Prerequisites for Cloudera Data Engineering on premises

Prerequisites for Cloudera Data Engineering on premises.

Before deploying Cloudera Data Engineering, make sure you have reviewed and complied with the requirements in the installation guide for your environment:

- Installing on OpenShift
- Installing using the Cloudera Embedded Container Service

Cloudera Base on premises cluster requirements

The Cloudera Base on premises cluster that you are using for the Cloudera Data Engineering service must have the Apache Ozone service enabled before creating an environment.

Red Hat OpenShift Container Platform requirements

For Cloudera Data Engineering on premises running on Red Hat OpenShift Container Platform (OCP), you must configure a route admission policy.

You must configure the OpenShift cluster for running applications in multiple namespaces with the same domain name. Run the following commands. If you have not installed the oc command line utility, install it using the instructions in the OpenShift documentation. For instructions on downloading the OCP kubeconfig file, see Downloading the kubernetes Configuration.

```
export KUBECONFIG=</path/to/ocp-kubeconfig>

oc -n openshift-ingress-operator patch ingresscontroller/default --patch '{
"spec":{"routeAdmission":{"namespaceOwnership":"InterNamespaceAllowed"}}}' -
-type=merge
```

Installation using the Cloudera Embedded Container Service

Preparing Cloudera Base on premises

Use Cloudera Manager to configure your Cloudera Base on premises cluster in preparation for the Cloudera Data Services on premises installation.

1. Perform the steps from Configuring TLS Encryption for Cloudera Manager Using Auto-TLS to configure TLS encryption for CDP Private Cloud Base cluster.

2. Configure Cloudera Manager with a JKS-format (not PKCS12) TLS truststore. For more information, see Database requirements.
3. Configure Cloudera Manager to include a root certificate that trusts the certificate for all Cloudera Manager server hosts you use with the Cloudera Base on premises, LDAP server (if you are using LDAP), and the Postgres DB of all Hive Metastores that you use with Cloudera Base on premises. If you use a single certificate authority (CA) to sign the certificate for all Cloudera Manager server hosts, then you must import only that single CA.
   a. Import the necessary certificates into the truststore configured in Configure Administration > Settings > Security > Cloudera Manager TLS/SSL Client Trust Store File .
4. Enable Kerberos authentication for all the services in Cloudera Base on premises cluster. For more information, see the Enabling Kerberos for authentication.
5. Configure Ranger to use LDAP for user authentication. Ensure that you have set up Ranger user synchronization. For more information, see Configure Ranger authentication for LDAP and Ranger usersync.
6. Configure authentication using an LDAP in Cloudera Manager IPA, Microsoft Active Directory (AD), and OpenLDAP are currently supported. For more information, see Configure authentication using an LDAP-compliant identity service.
7. Verify if all the running services in the cluster are healthy. To verify the health issues of all the running services in the cluster do the following:

   On the Cloudera Manager UI, go to Clusters > [***CLUSTER NAME***] > All Health Issues . If there are no health issues, then Cloudera Manager displays the No Health Issues Found message.
8. If you want to reuse data from your legacy CDH or HDP deployment into your Cloudera Base on premises cluster, copy the data from your CDH or HDP deployments into the Cloudera Base on premises cluster that you can access by Cloudera Data Services on premises. For more information about data migration, see Data Migration Guide.
9. For installing Cloudera Base on premises, see the Install Cloudera Private Cloud Base

Cloudera Base on premises checklist

Use this checklist to ensure that your Cloudera Base on premises is configured and ready for installing Cloudera Data Services on premises.

Note: The Cloudera Manager mentioned in this checklist is the Cloudera Base on premises Cloudera Manager using which you want to install Cloudera Data Services on premises.

Table 5: Cloudera Base on premises checklist to install Cloudera Data Services on premises

Item	Summary	Documentation	Notes
Runtime components	Ensure that you have Ranger, Atlas, Hive, HDFS, and Ozone installed in your Cloudera Base on premises.	• Software Support Matrix for Cloudera Embedded Container Service on page 4 • Cloudera Private Cloud Base requirements	If you do not install these components, you see an error when creating an environment in Cloudera Data Services on premises.
Network requirement	Ensure that all the network routing hops in production. Cloudera recommends not to use more than 4:1 oversubscription between the spine-leaf switches.		
Cloudera Manager database requirement	Refer to the the Cloudera Base on premises database requirements.	• Database Requirements • Cloudera Support Matrix	N/A
Cloudera Manager TLS configuration	Ensure that Cloudera Manager in the Cloudera Base on premises cluster is configured to use TLS.	Configuring TLS Encryption for Cloudera Manager Using Auto-TLS	You can also manually configure TLS to complete this task. See Manually Configuring TLS Encryption for Cloudera Manager

Item	Summary	Documentation	Notes
Cloudera Manager JKS-format TLS truststore	Ensure that the Cloudera Manager is configured with a JKS-format (not PKCS12) TLS truststore.	Obtain and Deploy Keys and Certificates for TLS/SSL	N/A
Cloudera Manager truststore and root certificate	Ensure that the Cloudera Manager truststore contains a root certificate that trusts the certificate for all Cloudera Manager server hosts used with CDP Private Cloud Data Services.	How to Add Root and Intermediate CAs to Truststore for TLS/SSL	Import the necessary certificates into the truststore configured in Configure Administration > Settings > Security > Cloudera Manager TLS/SSL Client Trust Store File .
LDAP configuration	Ensure that you configure LDAP using Cloudera Manager.	N/A	Only Microsoft Active Directory (AD) and OpenLDAP are currently supported.
Apache Ranger configuration for LDAP	Ensure that the Cloudera Base on premises cluster is configured with Apache Ranger and LDAP for user authentication.	Configure Ranger authentication for LDAP	N/A
Apache Ranger usersync configuration	Ensure that you have configured Apache Ranger and Apache Ranger usersync.	Ranger usersync	Apache Ranger user synchronization is used to get users and groups from the corporate ActiveDirectory to use in policy definitions.
Kerberos configuration	Ensure that Kerberos is enabled for all services in the cluster.	Enabling Kerberos for authentication	Custom Kerberos principals are not currently supported.
Internet access or air gap installation	Ensure that Cloudera Base on premises and the Cloudera Embedded Container Service hosts have access to the Internet. If you do not have access to the Internet, you must do an air gap installation.	Install Cloudera Private Cloud Data Services in air gap environment	You need access to the Docker registries and the Cloudera repositories during the installation process.
Services health check	Ensure that all services running in the cluster are healthy.	Cloudera Manager Health Tests	N/A
Cloudera on premises entitlement	Ensure that you have the necessary Cloudera entitlement to access the on premises installation.	N/A	
Reuse data from CDH or HDP (Optional)	To reuse data from your legacy CDH or HDP deployment in your on premises, ensure that you have migrated that data into your Cloudera Base on premises. You must be using Cloudera Runtime 7.1.7 for migrating data from your CDH or HDP cluster.	Data Migration Guide	N/A
(Recommended) Configure HDFS properties to optimize logging	Cloudera uses “out_webhdfs” Fluentd output plugin to write records into HDFS, in the form of log files, which are then used by different data services to generate diagnostic bundles. To optimize the size of logs that are captured and stored on HDFS, you must update a few HDFS configurations in the hdfs-site.xml file using Cloudera Manager.	Configuring HDFS properties to optimize logging	N/A

checklist

Use this checklist to ensure that your Cloudera Embedded Container Service is configured and ready for installing Cloudera Data Services on premises.

Table 6: checklist to install Cloudera Data Services on premises

Item	Summary	Documentation	Notes
Network requirements			Cloudera Data Services on premises requires a single ethernet interface. Multihoming is currently not supported.
DNS configuration	Ensure that you have set up the DNS and Reverse DNS between Cloudera Embedded Container Service hosts and Cloudera Base on premises. This is required for obtaining Kerberos ticket-granting tickets.	N/A	A wildcard DNS entry is required for resolving the ingress route for applications. The ingress route is usually behind a load balancer.
Check that ECS Ingress can be resolved in DNS.	Ensure that Cloudera Embedded Container Service application hostnames can be accessed from outside the cluster. You can test this by creating an ingress point on the target cluster.	The cluster generates multiple hosts and host-based routing is used in the cluster in order to route it to the right service. You must decide on a domain for the services which Cloudera Manager, by default points to one of the hostnames on the cluster. However, during the installation, you should check the default domain and override the default domain (only if necessary) with what you plan to use as the domain. The default domain must have a wildcard DNS entry. For example, *.apps.myhostnam e.com.	Perform a DNS query on the ingress point, to check if you can access the hostnames outside the cluster.
Clock time from NTP source	Ensure that the NTP clock in Cloudera Base on premises is in sync with the time configured in the Cloudera Embedded Container Service cluster. This is an important step if your setup does not have access to the Internet.	Enable an NTP Service	Installing Cloudera on Premises Data Services (ECS)
Default subnet created for docker service	The default subnet used by docker is 172.17.0.0/16. Since the IP range within the subnet is used by docker, any internal applications running or using an IP in this range will not be accessible.	Docker documentation	Ensure 172.17.0.0/16 IP range is not used by other applications or services, to avoid conflicts with the docker subnet. You can use a different subnet for docker if needed.

Adding a Cloudera Data Services on premises cluster

Using Cloudera Manager, you can either install Cloudera Data Services on premises by downloading the repository from the Internet, or you can do an air gap installation if Cloudera Manager does not have access to the Internet.

Before you begin:

Important:

RHEL 7.x support on Cloudera Embedded Container Service has been dropped in Cloudera Data Services on premises 1.5.5 and higher versions. If you are running RHEL 7.x, you must upgrade to a higher version before installing Cloudera Data Services on premises.

Requirements for 1.5.5 release:

- Ensure that you have Cloudera Manager7.13.1 CHF3 is installed and you have the entitlements to the Cloudera Data Services on premises product.
- For more information about Python-OS support matrix, see Installing Python 3 and Cloudera Support Matrix .
- Only TLS 1.2 is supported for authentication with Active Directory/LDAP. You require TLS 1.2 to authenticate the Cloudera Control Plane with your LDAP directory service like Active Directory.
- If the installer fails, do not cancel the installation. For more information, see Manually uninstalling ECS from a cluster.
- Do not use any antivirus or other security tools on the Cloudera Embedded Container Service nodes. These third-party tools may cause issues with Cloudera Embedded Container Service functionality.

Installing Cloudera Data Services on premises using Cloudera Embedded Container Service

Follow the steps in this topic to install Cloudera Data Services on premises with the Cloudera Embedded Container Service.

About this task

Important:

RHEL 7.x support on Cloudera Embedded Container Service has been dropped in Cloudera Data Services on premises 1.5.5 and higher versions. If you are running RHEL 7.x, you must upgrade to a higher version before installing Cloudera Data Services on premises.

Note: When deploying an Cloudera Embedded Container Service cluster, the batch size limitation for adding Cloudera Embedded Container Service agent nodes to Cloudera Embedded Container Service cluster is under 50. If there is a requirement to deploy an Cloudera Embedded Container Service cluster with more than 50 nodes, it is recommended to start the initial deployment with less than 50 nodes and incrementally add nodes to the cluster after the first installation succeeds.

Note: Prior to configuring Cluster IP Range (cluster-cidr) and Service IP Range (service-cidr) in step 11, ensure to review best practices here. Once your cluster has been deployed, these values cannot change. Any misconfiguration will require decommissioning the cluster and redeploying it to correct the settings.

Procedure

1. If you are installing Cloudera Embedded Container Service on RHEL 8 or RHEL 9:
   a) Run the following command to check to see if the nm-cloud-setup.service and nm-cloud-setup.timer services are enabled:

   ```
   systemctl status nm-cloud-setup.service nm-cloud-setup.timer
   ```

   b) If the nm-cloud-setup.service and nm-cloud-setup.timer services are enabled, disable them by running the following command on each host you added:

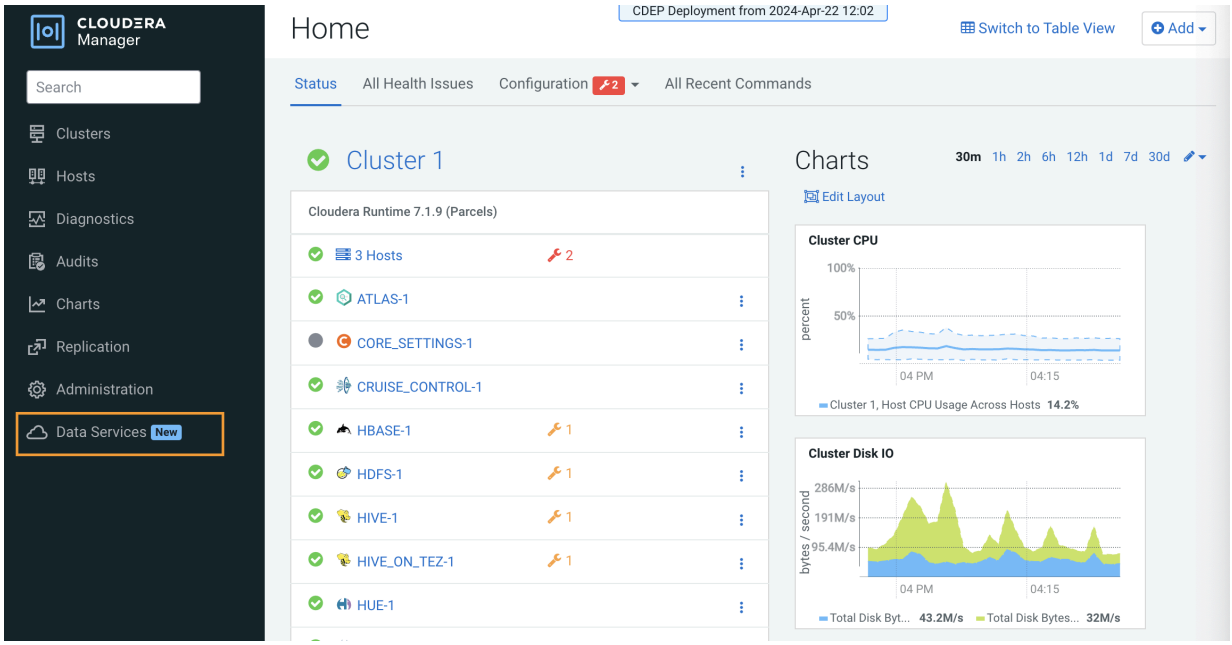
   ```
   systemctl disable nm-cloud-setup.service nm-cloud-setup.timer
   ```

   For more information, see Known issues and limitations.

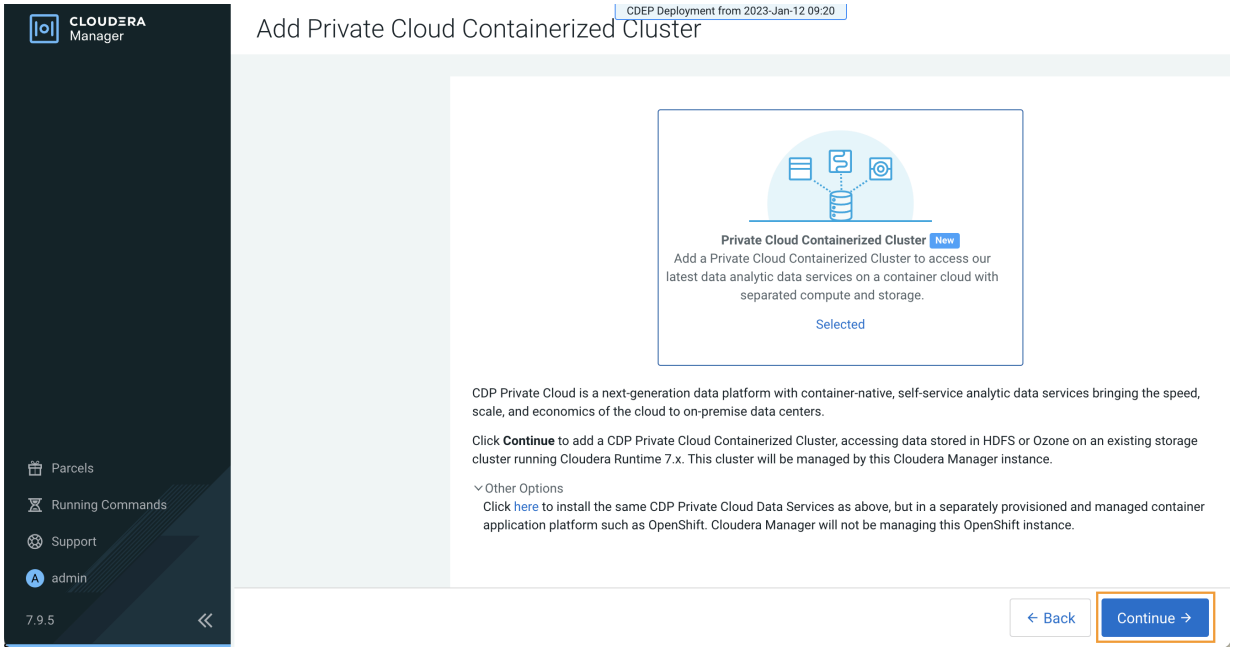
   Note: If the service is active, you have to first stop the service and disable the service.

   c) If you disabled the nm-cloud-setup.service and nm-cloud-setup.timer services, reboot the added hosts.

2. In Cloudera Manager, click Data Services in the left menu.



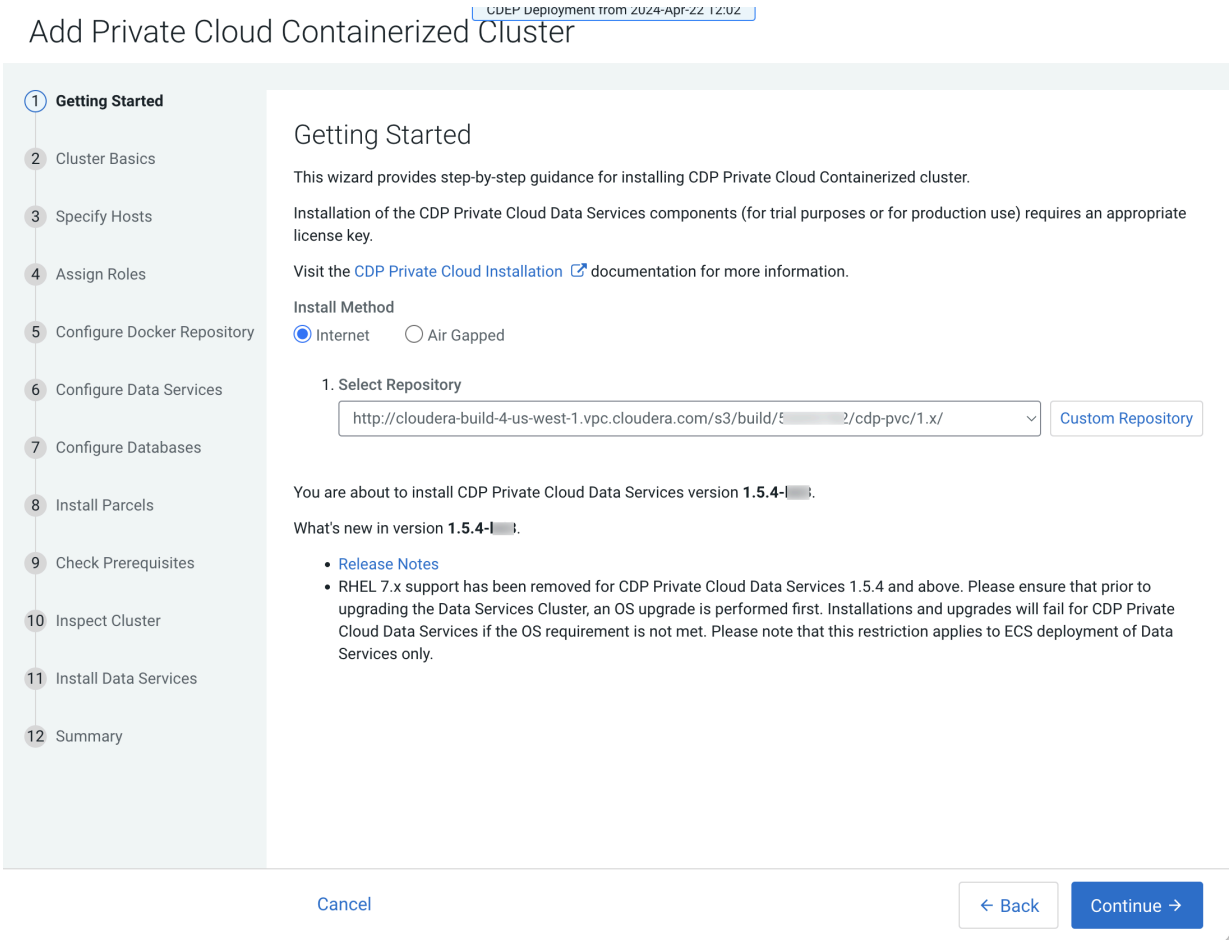
The Add Private Cloud Containerized Cluster page appears. Click Continue.



Note: You can also click Add Add Cluster at the top right in Cloudera Manager, then select Private Cloud Containerized Cluster as the cluster type.

3. On the Getting Started page of the installation wizard, select Internet or Air Gapped as the Install Method.

   Internet install method (To use a custom repository link provided to you by Cloudera, click Custom Repository) :



   If you select the Air Gapped install option, extra steps are displayed. Follow these steps to download and mirror the Cloudera archive URL using a local HTTP server.

   a. Download everything under: https://archive.cloudera.com/p/cdp-pvc-ds/latest

   ```
   wget -l 0 --recursive --no-parent -e robots=off -nH --cut-dirs=2 --rejec
   t="index.html*" -t 10 https://<username>:<password>@archive.cloudera.com
   /p/cdp-pvc-ds/latest/
   ```

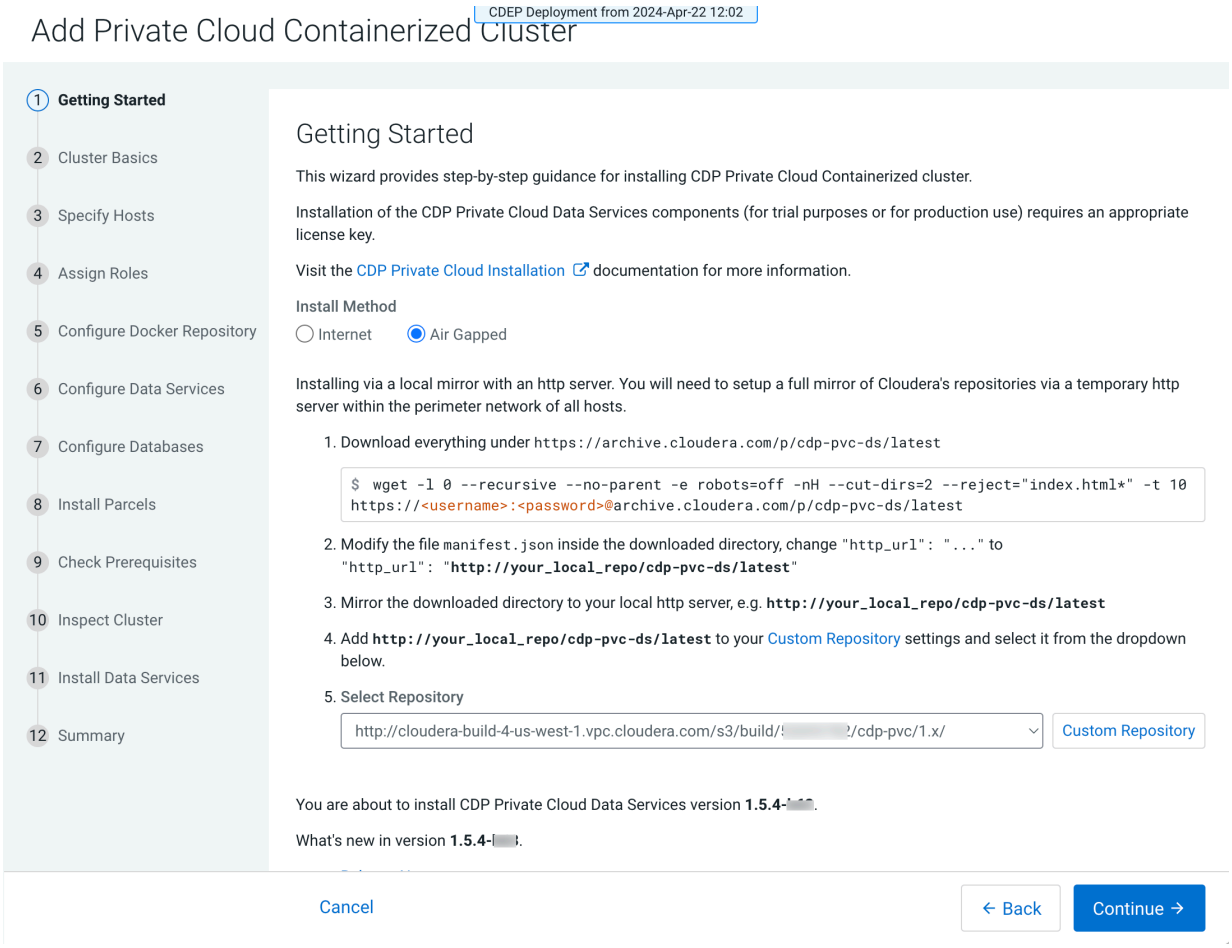
   b. Edit the manifest.json file in the downloaded directory. Change "http_url": "..." to

   "http_url": "http://your_local_repo/cdp-pvc-ds/latest"

   c. Mirror the downloaded directory to your local http server, e.g. http://your_local_repo/cdp-pvc-ds/latest

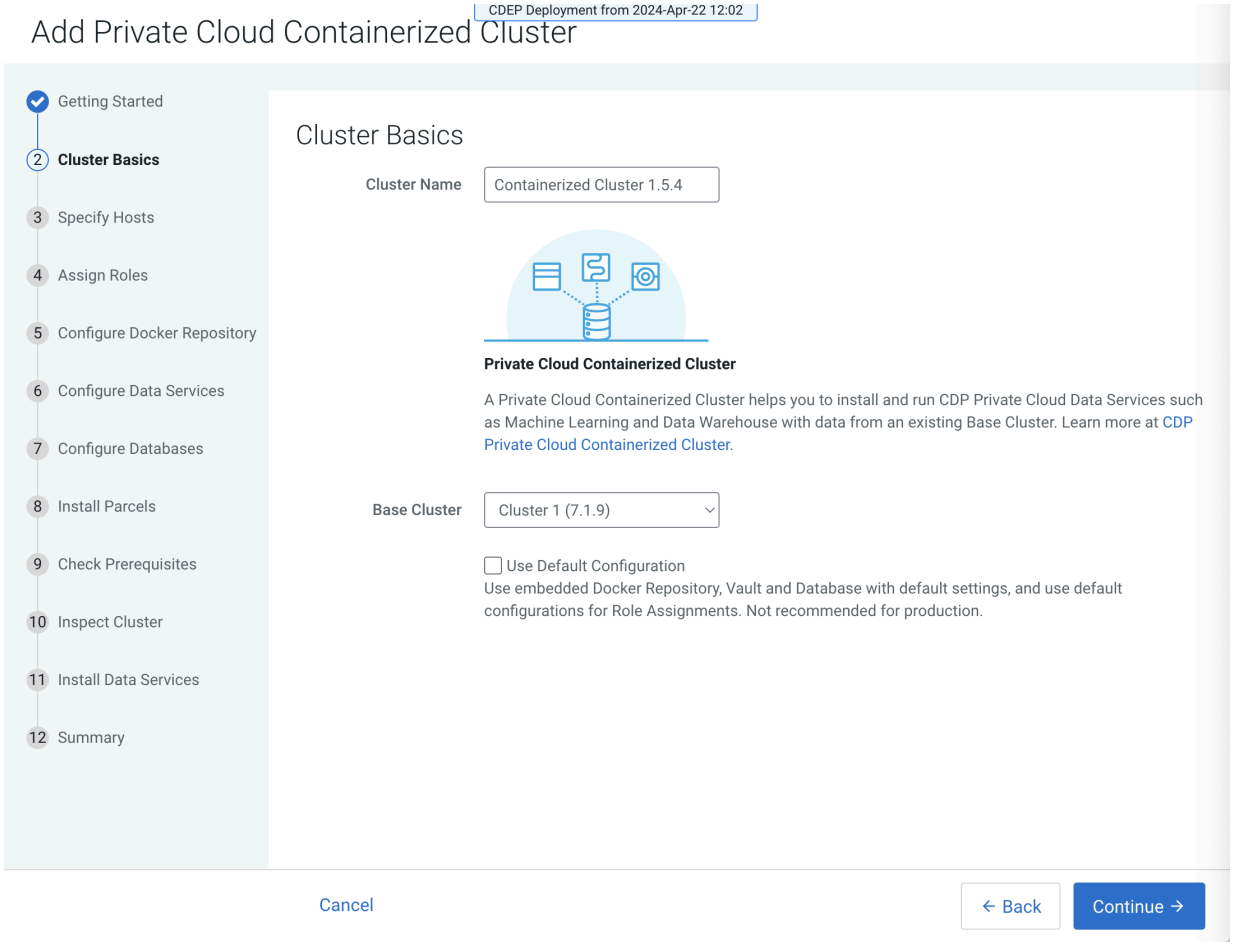
   d. Click Custom Repository and add http://your_local_repo/cdp-pvc-ds/latest as a custom repository.

   e. Click the Select Repository drop-down and select http://your_local_repo/cdp-pvc-ds/latest

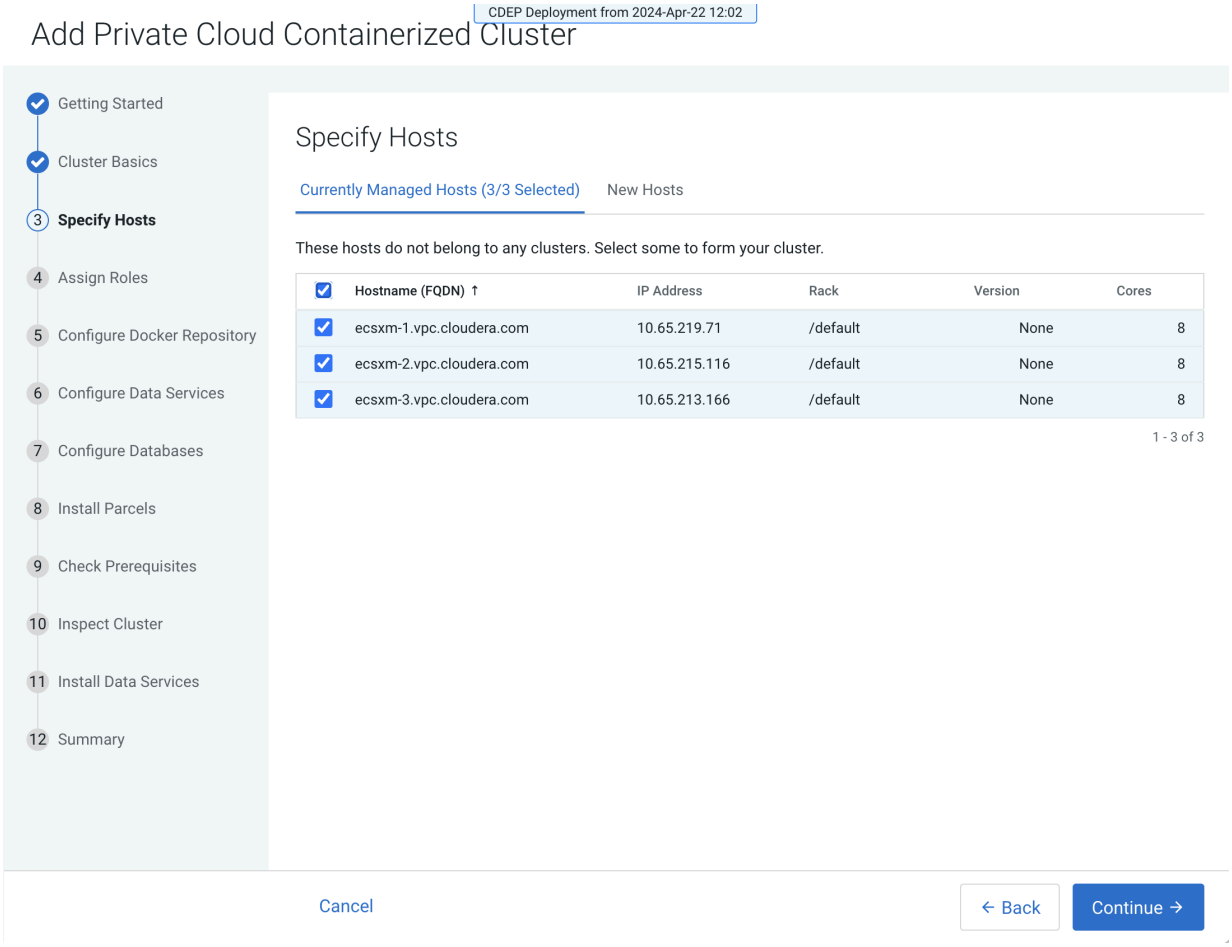


Click Continue.

4. On the Cluster Basics page, type a name for the Private Cloud cluster that you want to create in the Cluster Name field. From the Base Cluster drop-down list, select the cluster that has the storage and SDX services that you want this new Private Cloud Data Services instance to connect with. Click Continue.

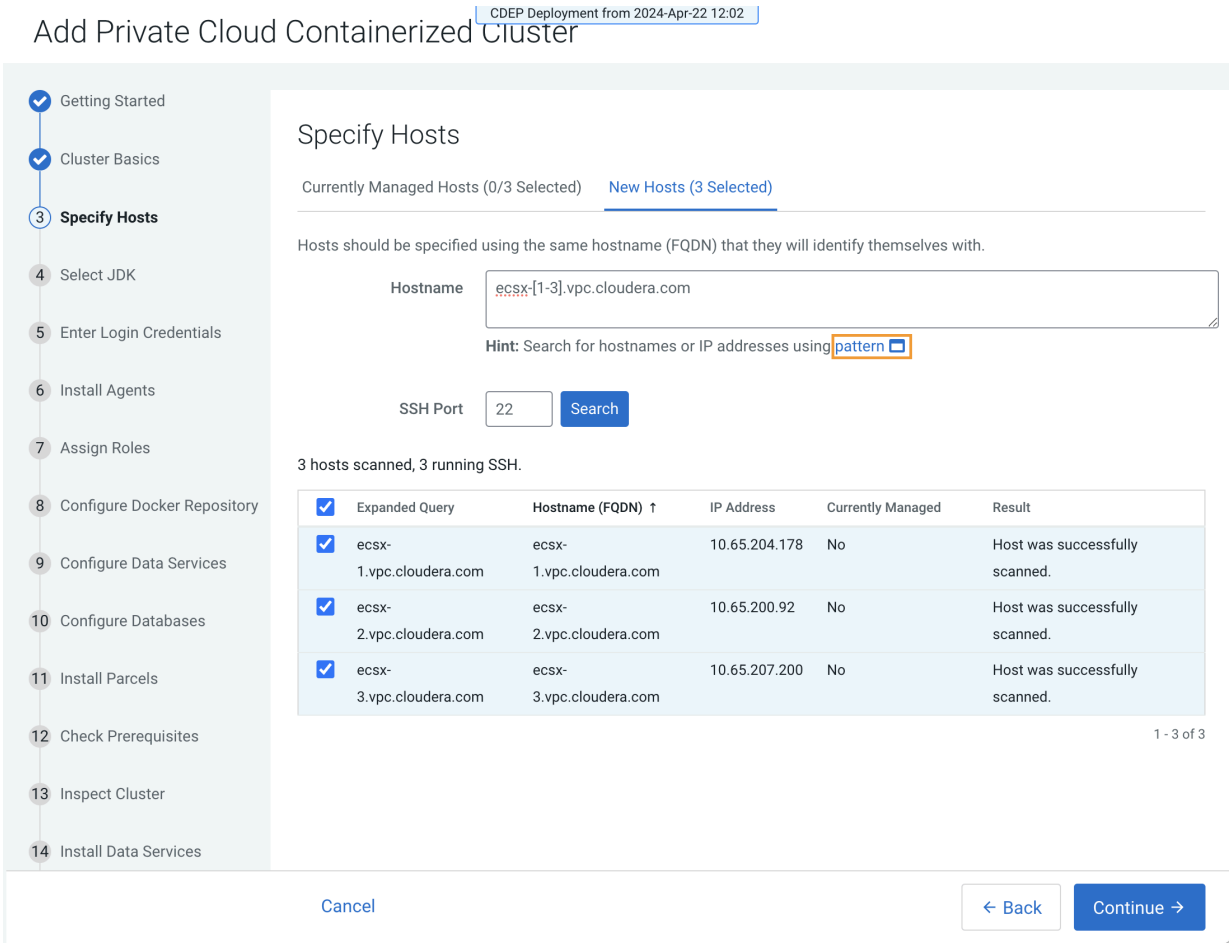


5. On the Specify Hosts page, hosts that have already been added to Cloudera Manager are listed on the Currently Managed Hosts tab. You can select one or more of these hosts to add to the ECS cluster.



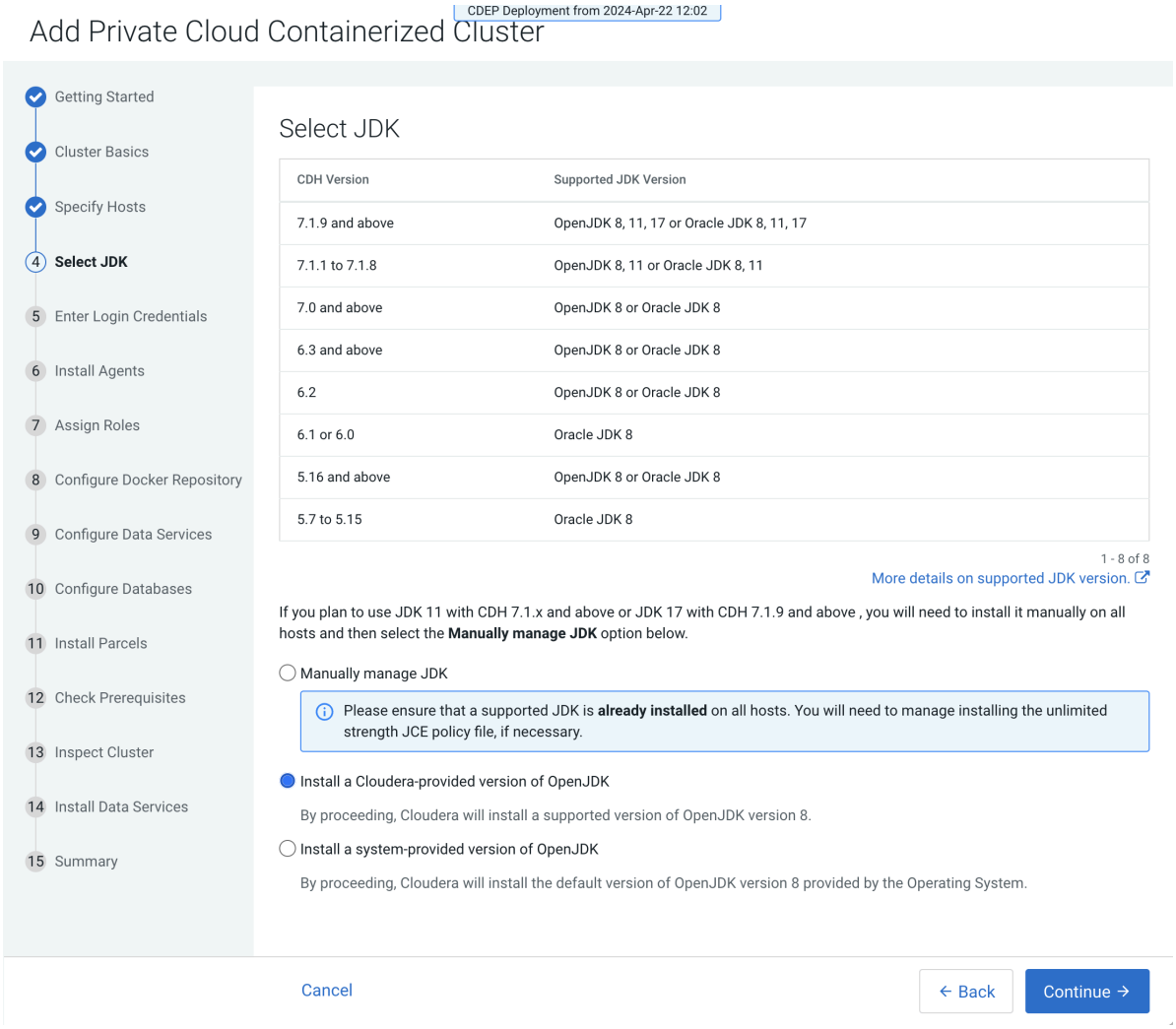
You can also click the New Hosts tab to specify one or more hosts that have not been added to Cloudera Manager. Enter a Fully Qualified Domain Name in the Hostname box, then click Search.

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

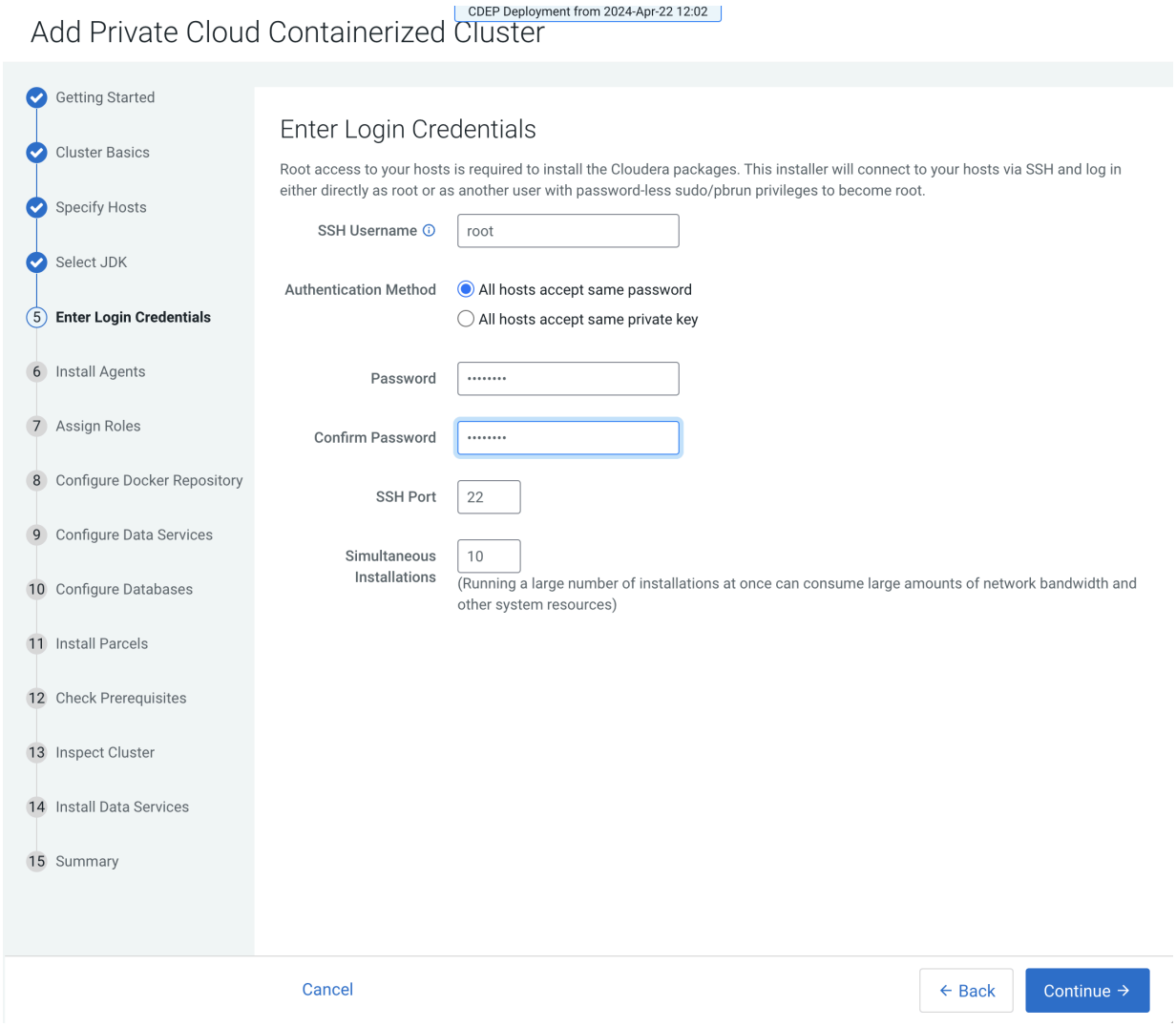


After you have finished specifying the ECS hosts, click Continue.

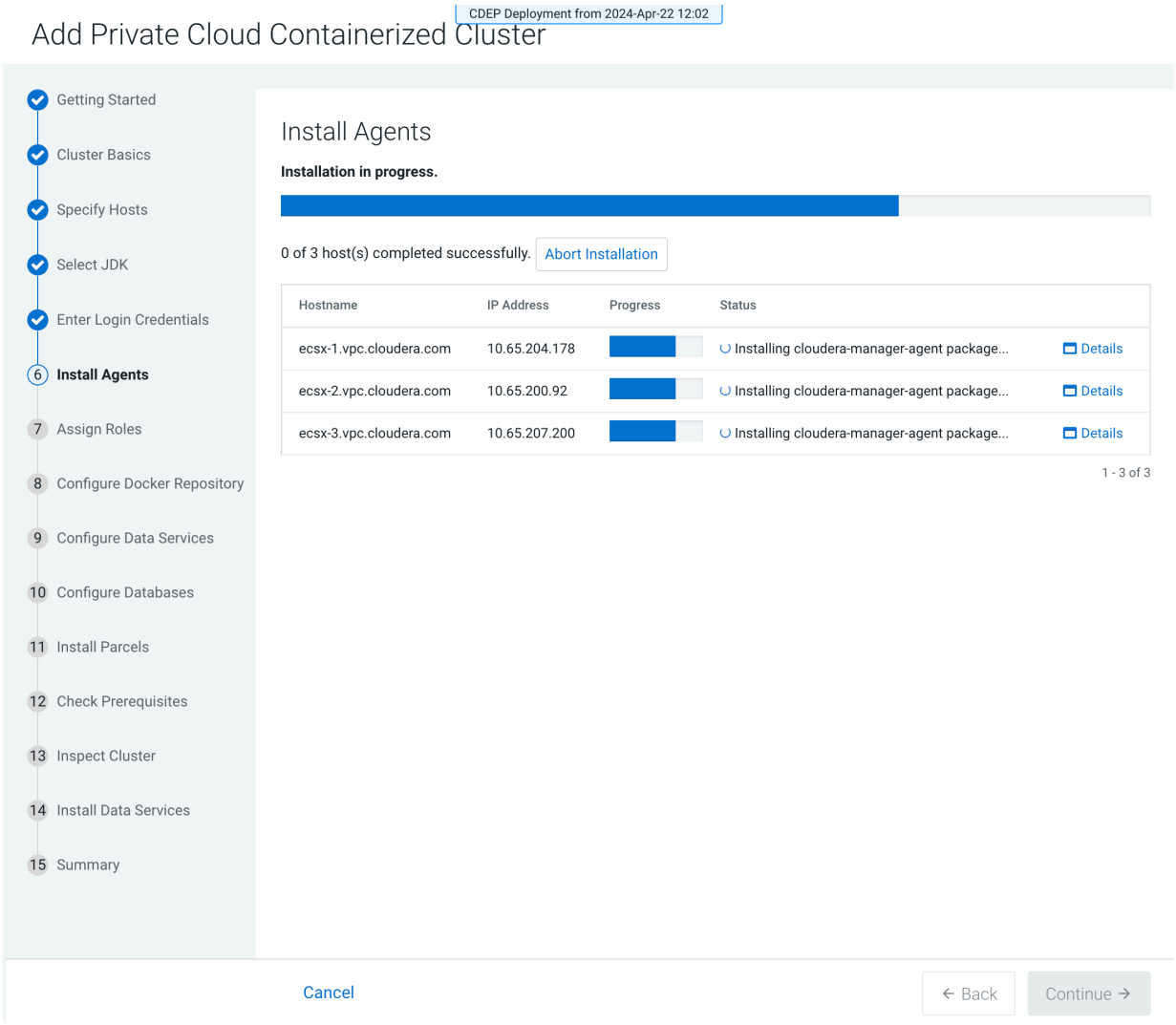
6. On the Select JDK page, select any one from the below options:
   a) Manually manage JDK
   b) Install a Cloudera-provided version of OpenJDK
   c) Install a system-provided version of OpenJDK



7. 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.

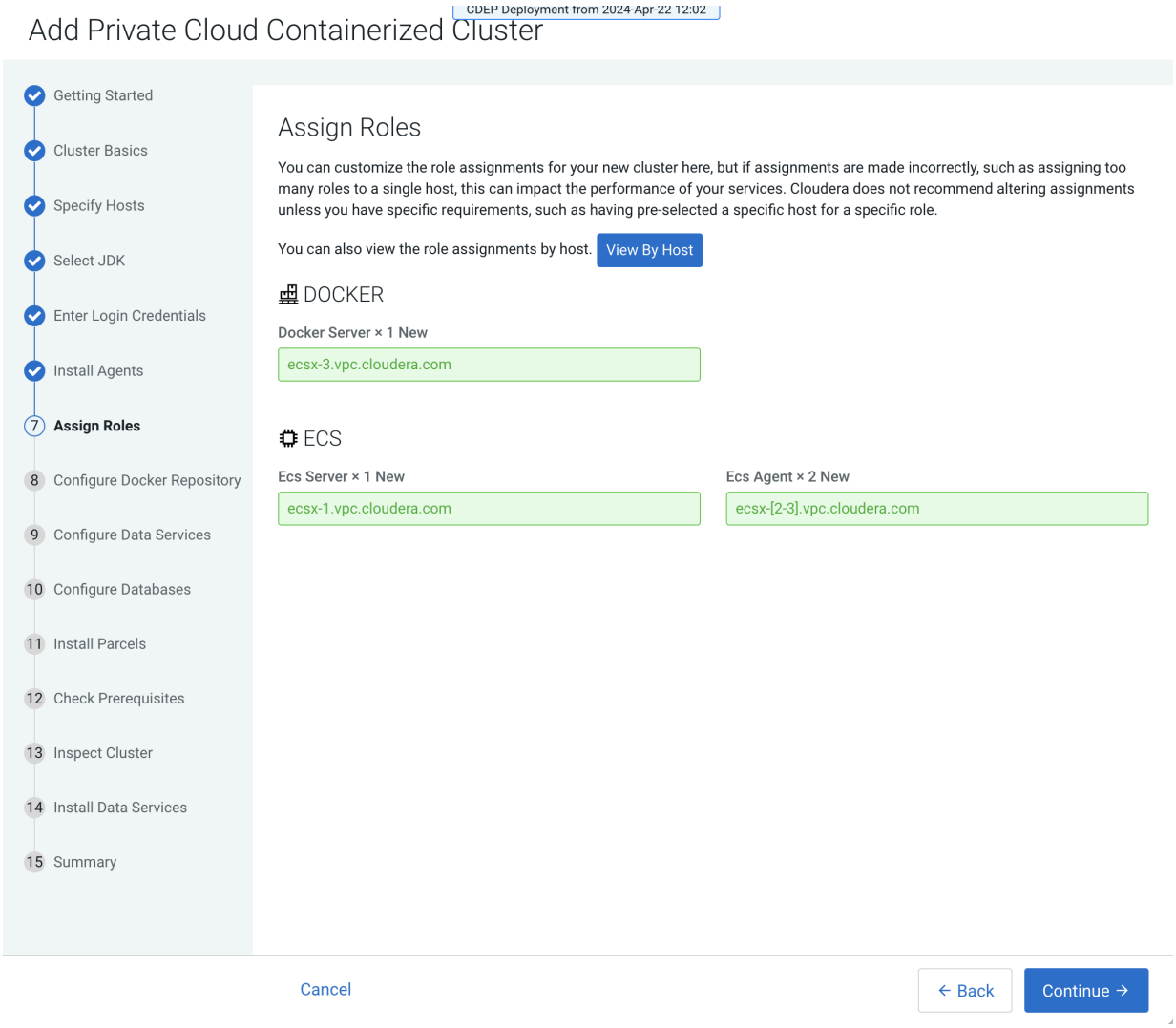


8. The Install Agents page appears and displays a progress indicator as the agent packages are installed.

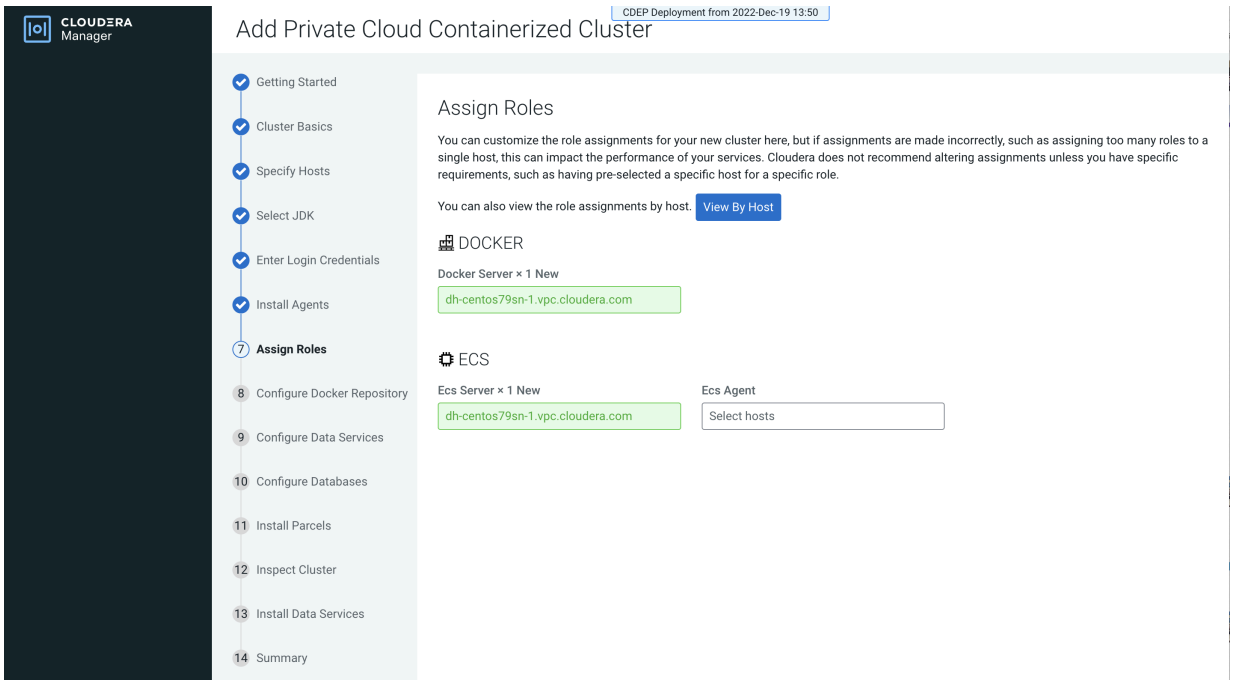


9. On the Assign Roles page, you can customize the roles assignment for your new Private Cloud Containerized cluster.

Important: Cloudera does not recommend altering assignments unless you have specific requirements such as having selected a specific host for a specific role.



Single node ECS installation is supported, but is only intended to enable CDSW to CML migration. If you are installing ECS on a single node, only the Docker and ECS Server roles are assigned. The ECS Agent role is not required for single node installation.



Click Continue.

10. Configure a Docker Repository.

There are several options for configuring a Docker Repository. For more information about these options, see Docker repository access on page 12.

Note: You need to get the Generate the copy-docker script button working to generate and download the script.

a. Ensure Cloudera Manager Server host's /tmp is mounted without " noexec " option, OR
b. Customize tmpdir to some other directory by adding flag -Djava.io.tmpdir=/opt/cloudera to CMF_JAVA_OPTS in the file /etc/default/cloudera-scm-server followed by Cloudera Manager Server restart AND
c. Ensure not to use podman service on the server hosting the images as it will fail to install.

The following ports must be opened and allowed no matter which Docker repository option you choose.

- Ports required for Cloudera Manager/Cloudera Manager agent (port 5000 is required for Cloudera Machine Learning):

Protocol	Port
TCP	7180-7192
TCP	19001
TCP	5000
TCP	9000

- Inbound rules for ECS Server nodes (Kubernetes/RKE2):

Protocol	Port
TCP	9345
TCP	6443
UDP	8472

Protocol	Port
TCP	10250
TCP	2379
TCP	2380

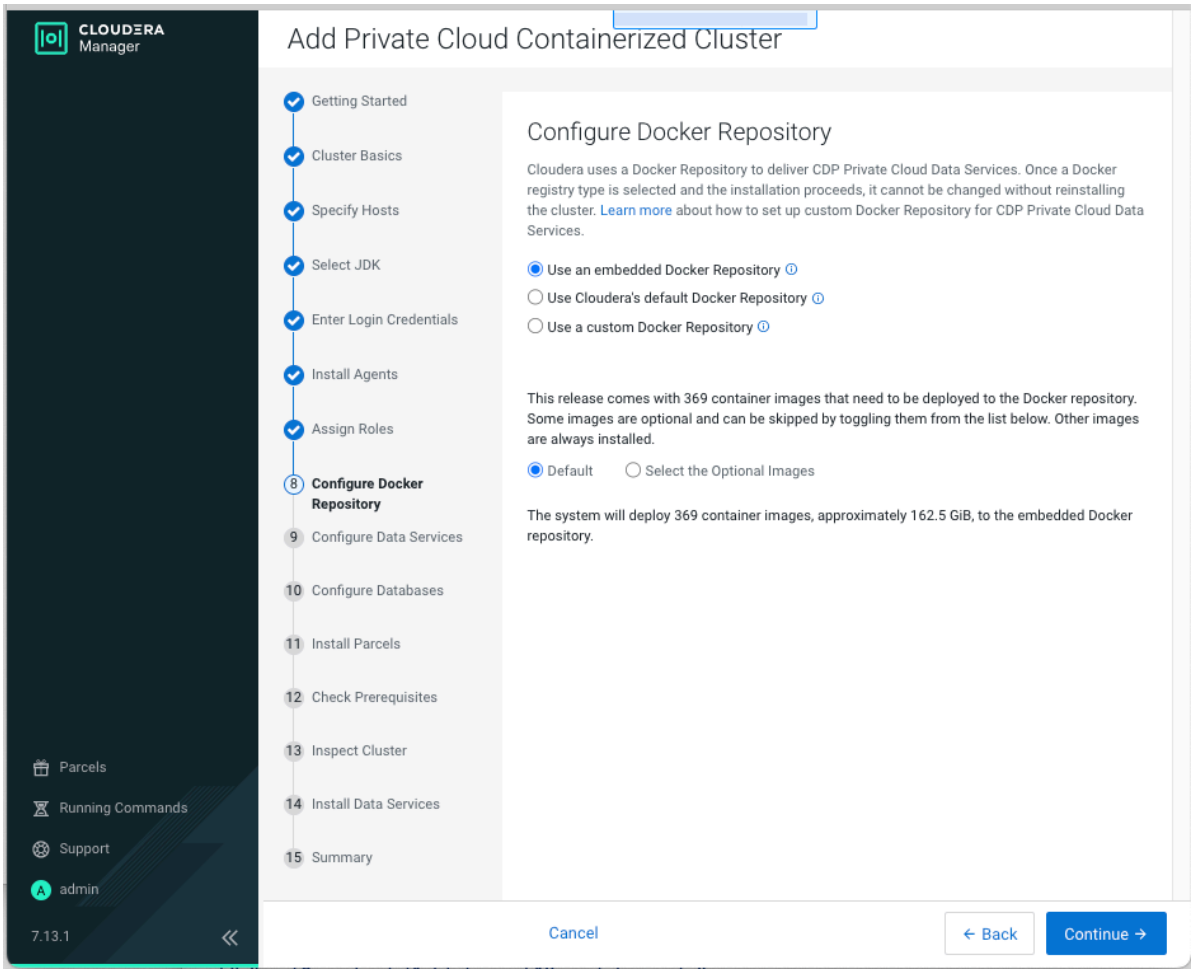
Protocol	Port
TCP	30000-32767

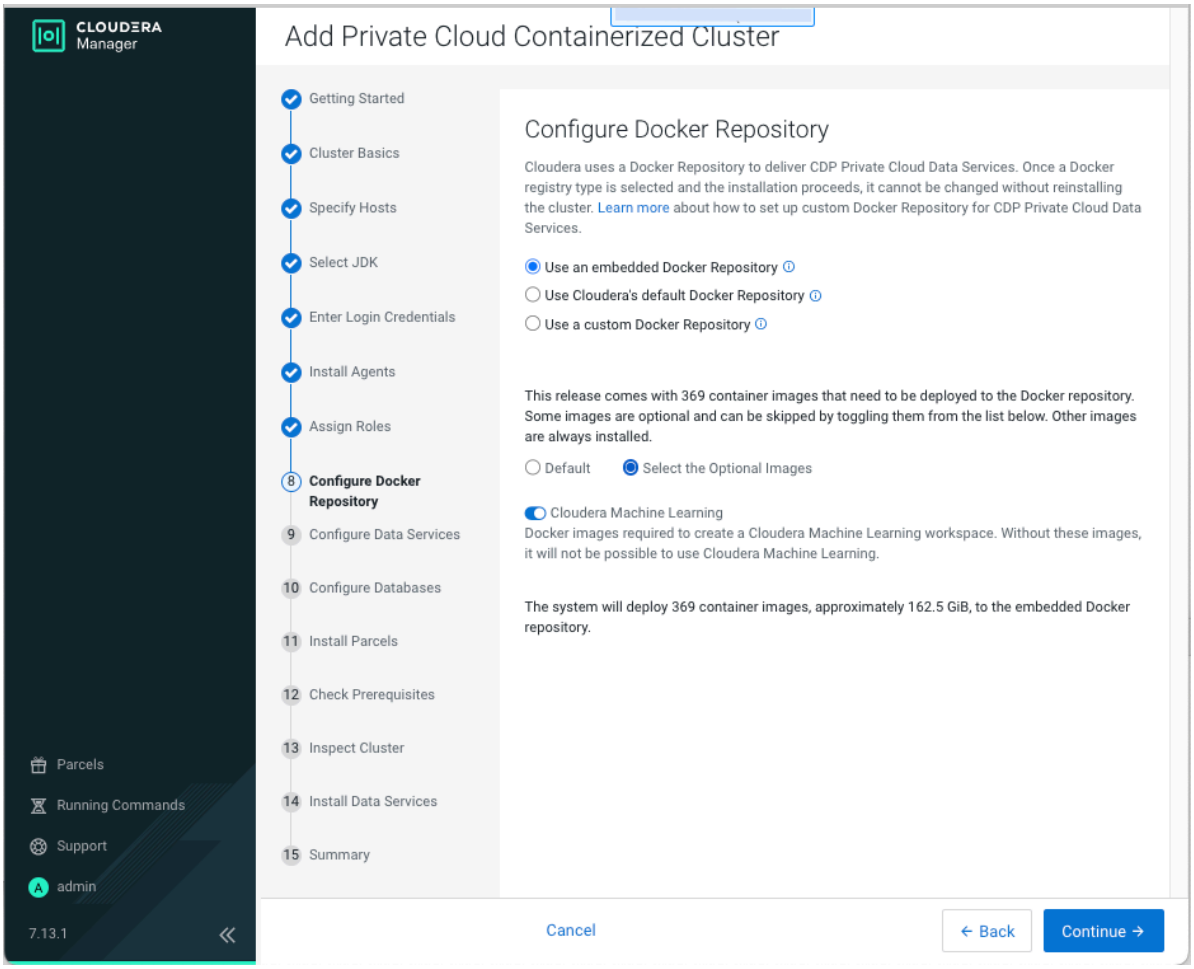
- Inbound Rules for the ECS Agent (Kubernetes/RKE2):

Protocol	Port
UDP	4789

On the Configure Docker Repository page, select one of these options:

- Embedded Docker Repository



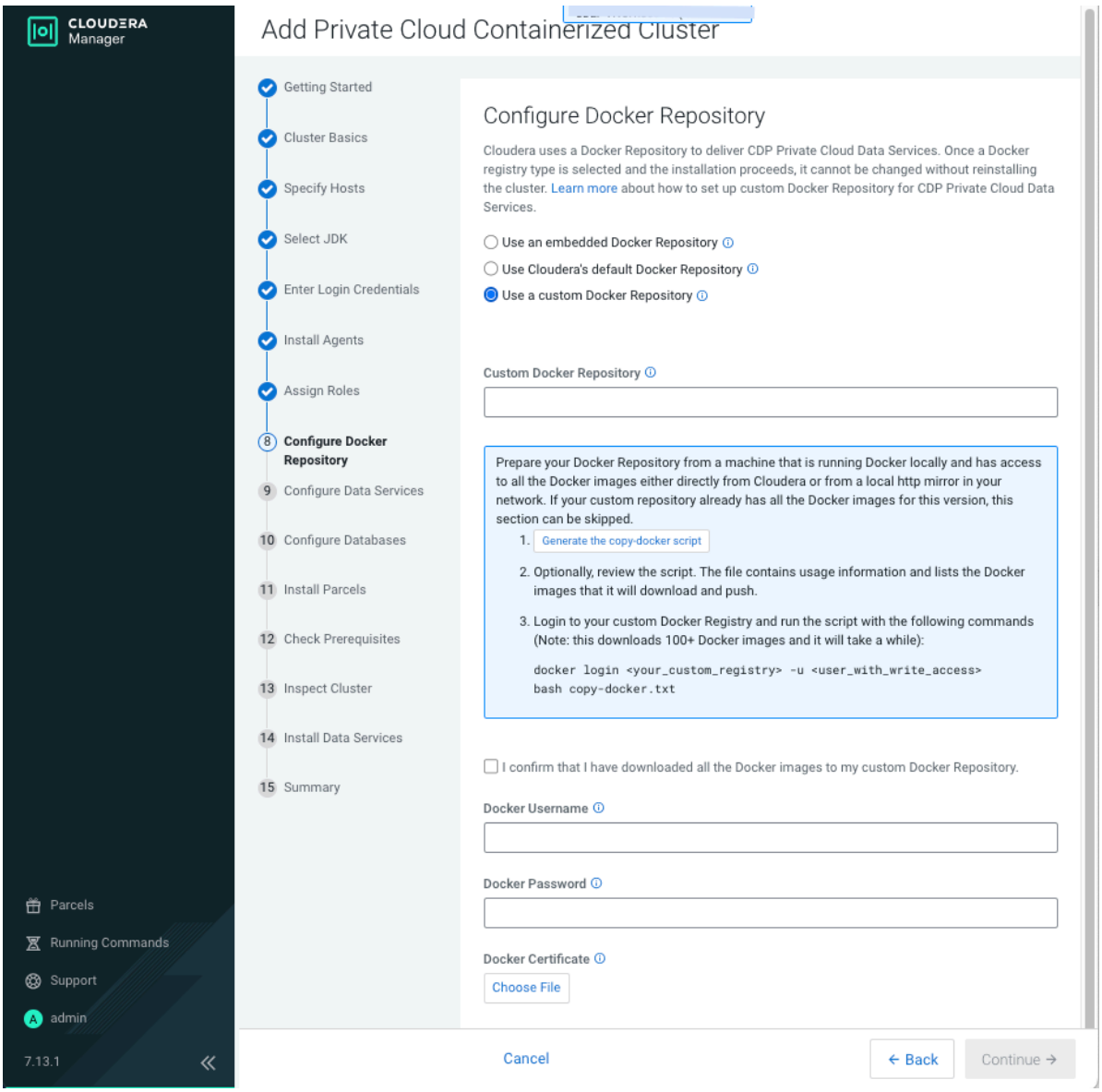


If you select the Internet Install Method option on the Getting Started page, images are copied over the internet from the Cloudera repository.

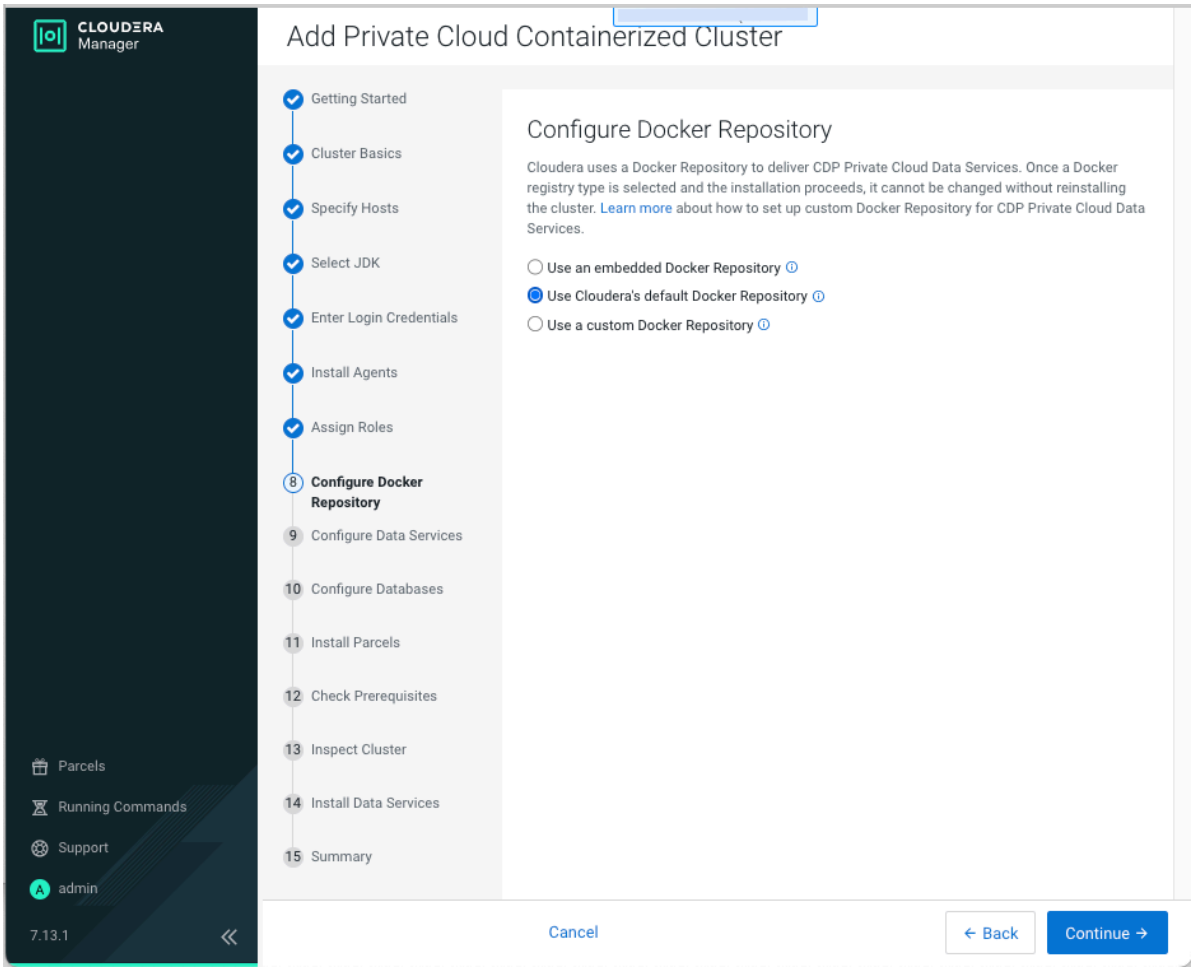
If you select the Air Gapped option, images are copied from a local http mirror you have set up in your environment.

Select Default to deploy all of the default Docker images to the repository, or select Select the Optional Images to choose which images to deploy. If you will be deploying Cloudera Machine Learning (CML), toggle the Cloudera Machine Learning switch on to copy the images for CML.

- Custom Docker Repository



- Cloudera default Docker RepositoryThis option requires that cluster hosts have access to the internet and you have selected Internet as the install method.



This option requires that you set up a Docker Repository in your environment and that all cluster hosts have connectivity to the repository.

Note: If you are installing ECS on a single node, you should select the Use a Custom Docker Repository option. Single node ECS installation is supported, but is only intended to enable CDSW to CML migration.

You must enter the following options:

- Custom Docker Repository – Enter the URL for your Docker Repository
- Docker Username – Enter the username for the Docker Repository.
- Docker Password – Enter the password for the Docker Repository.

  Important: Do not use the $ character for this password.

- Docker Certificate – Click the Choose File button to upload a TLS certificate to secure communications with the Docker Repository.

Click the Generate the copy-docker script button to generate and download a script that copies the Docker images from Cloudera, or (for air-gapped installation) from a local http mirror in your network.

Run the script from a machine that is running Docker locally and has access to the Docker images using the following commands:

```
docker login [***URL FOR DOCKER REPOSITORY***] -u [***USERNAME OF USER
 WITH WRITE ACCESS***]
```

```
bash copy-docker.txt
```

The copying operation may take 4 - 5 hours.

11. On the Configure Data Services page, you can modify configuration settings such as the data storage directory, number of replicas, and so on. If there are multiple disks mounted on each host with different characteristics (HDD and SSD), then Local Path Storage Directory must point to the path belonging to the optimal storage. Ensure that you have reviewed your changes. If you want to specify a custom certificate, place the certificate and the private key in a specific location on the Cloudera Manager server host and specify the paths in the input boxes

labelled as Ingress Controller TLS/SSL Server Certificate/Private Key File below. This certificate will be copied to the Control Plane during the installation process.

Note:

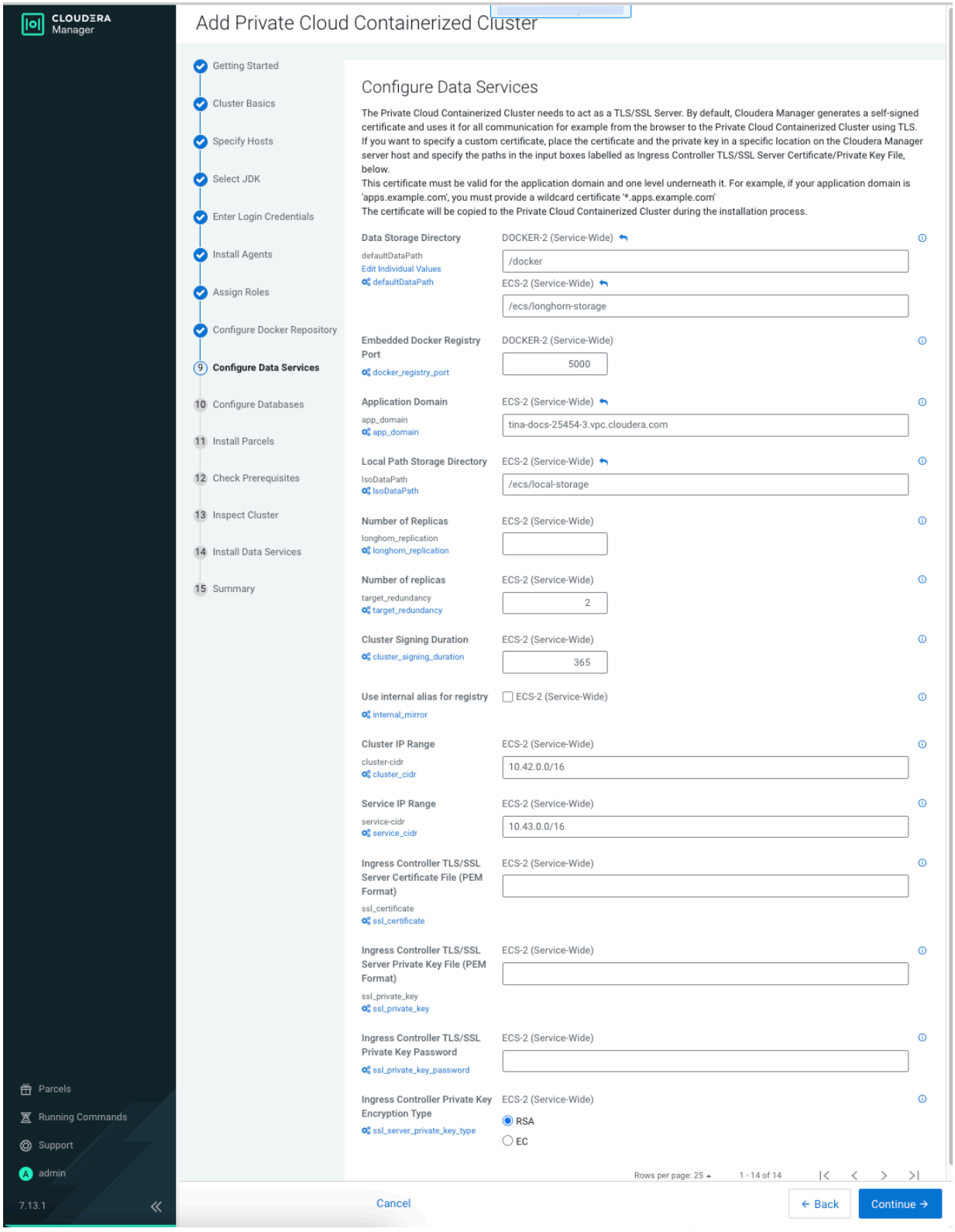
The "Ingress Controller TLS/SSL Server Certificate File (PEM Format)" must only contain -----BEGIN C ERTIFICATE----- through -----END CERTIFICATE----- (inclusive) for the server certs. It cannot include any preamble text and, and must not include a private key.

The "Ingress Controller TLS/SSL Server Private Key File (PEM Format)" must only contain the unencrypted key, and only the header through the footer, with no preamble text.

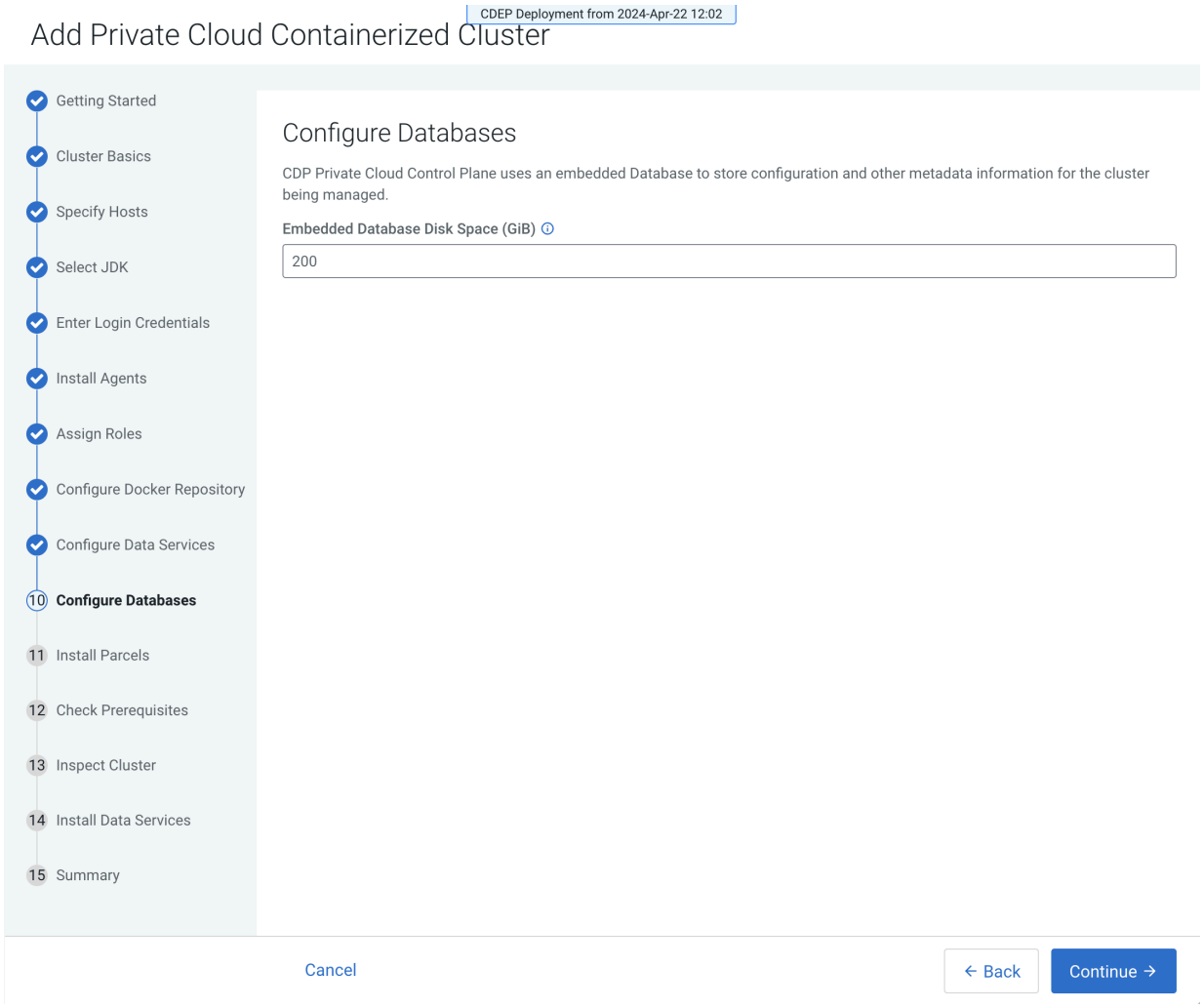
Both of these files must be readable by the "cloudera-scm" account.

For information on the required entries that must be present in DNS and TLS certificates when not using wildcards, refer to 'No Wildcard DNS/TLS Setup'

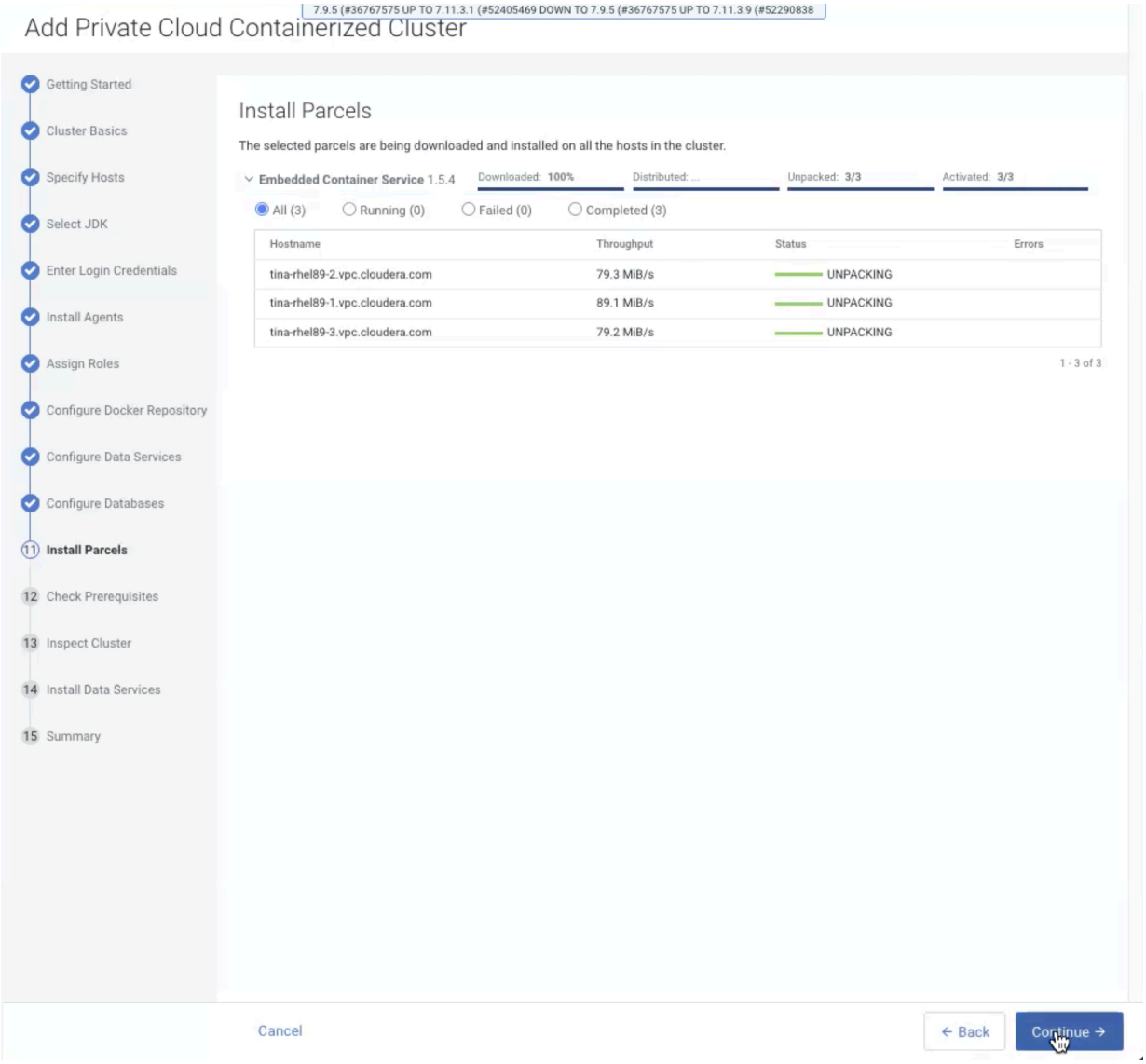
Click Continue.


CLOUDERA Manager
Add Private Cloud Containerized Cluster
Getting Started
Cluster Basics
Specify Hosts
Select JDK
Enter Login Credentials
Install Agents
Assign Roles
Configure Docker Repository
9 Configure Data Services
10 Configure Databases
11 Install Parcels
12 Check Prerequisites
13 Inspect Cluster
14 Install Data Services
15 Summary
Configure Data Services
The Private Cloud Containerized Cluster needs to act as a TLS/SSL Server. By default, Cloudera Manager generates a self-signed certificate and uses it for all communication for example from the browser to the Private Cloud Containerized Cluster using TLS. If you want to specify a custom certificate, place the certificate and the private key in a specific location on the Cloudera Manager server host and specify the paths in the input boxes labelled as Ingress Controller TLS/SSL Server Certificate/Private Key File, below.
This certificate must be valid for the application domain and one level underneath it. For example, if your application domain is 'apps.example.com', you must provide a wildcard certificate '*.apps.example.com'
The certificate will be copied to the Private Cloud Containerized Cluster during the installation process.
Data Storage Directory
defaultDataPath
Edit Individual Values
defaultDataPath
DOCKER-2 (Service-Wide)
/docker
ECS-2 (Service-Wide)
/ecs/longhorn-storage
Embedded Docker Registry Port
docker_registry_port
DOCKER-2 (Service-Wide)
5000
Application Domain
app_domain
app_domain
ECS-2 (Service-Wide)
tina-docs-25454-3.vpc.cloudera.com
Local Path Storage Directory
lsoDataPath
lsoDataPath
ECS-2 (Service-Wide)
/ecs/local-storage
Number of Replicas
longhorn_replication
longhorn_replication
ECS-2 (Service-Wide)
Number of replicas
target_redundancy
target_redundancy
ECS-2 (Service-Wide)
2
Cluster Signing Duration
cluster_signing_duration
ECS-2 (Service-Wide)
365
Use internal alias for registry
internal_mirror
ECS-2 (Service-Wide)
Cluster IP Range
cluster-cidr
cluster_cidr
ECS-2 (Service-Wide)
10.42.0.0/16
Service IP Range
service-cidr
service_cidr
ECS-2 (Service-Wide)
10.43.0.0/16
Ingress Controller TLS/SSL Server Certificate File (PEM Format)
ssl_certificate
ssl_certificate
ECS-2 (Service-Wide)
Ingress Controller TLS/SSL Server Private Key File (PEM Format)
ssl_private_key
ssl_private_key
ECS-2 (Service-Wide)
Ingress Controller TLS/SSL Private Key Password
ssl_private_key_password
ECS-2 (Service-Wide)
Ingress Controller Private Key Encryption Type
ssl_server_private_key_type
ECS-2 (Service-Wide)
RSA
EC
Rows per page: 25
1 - 14 of 14
Parcels
Running Commands
Support
admin
7.13.1
Cancel
← Back
Continue →

12. On the Configure Databases page, click Continue.



13. On the Install Parcels page, the selected parcel is downloaded to the Cloudera Manager server host, distributed, unpacked, and activated on the ECS cluster hosts. Click Continue.



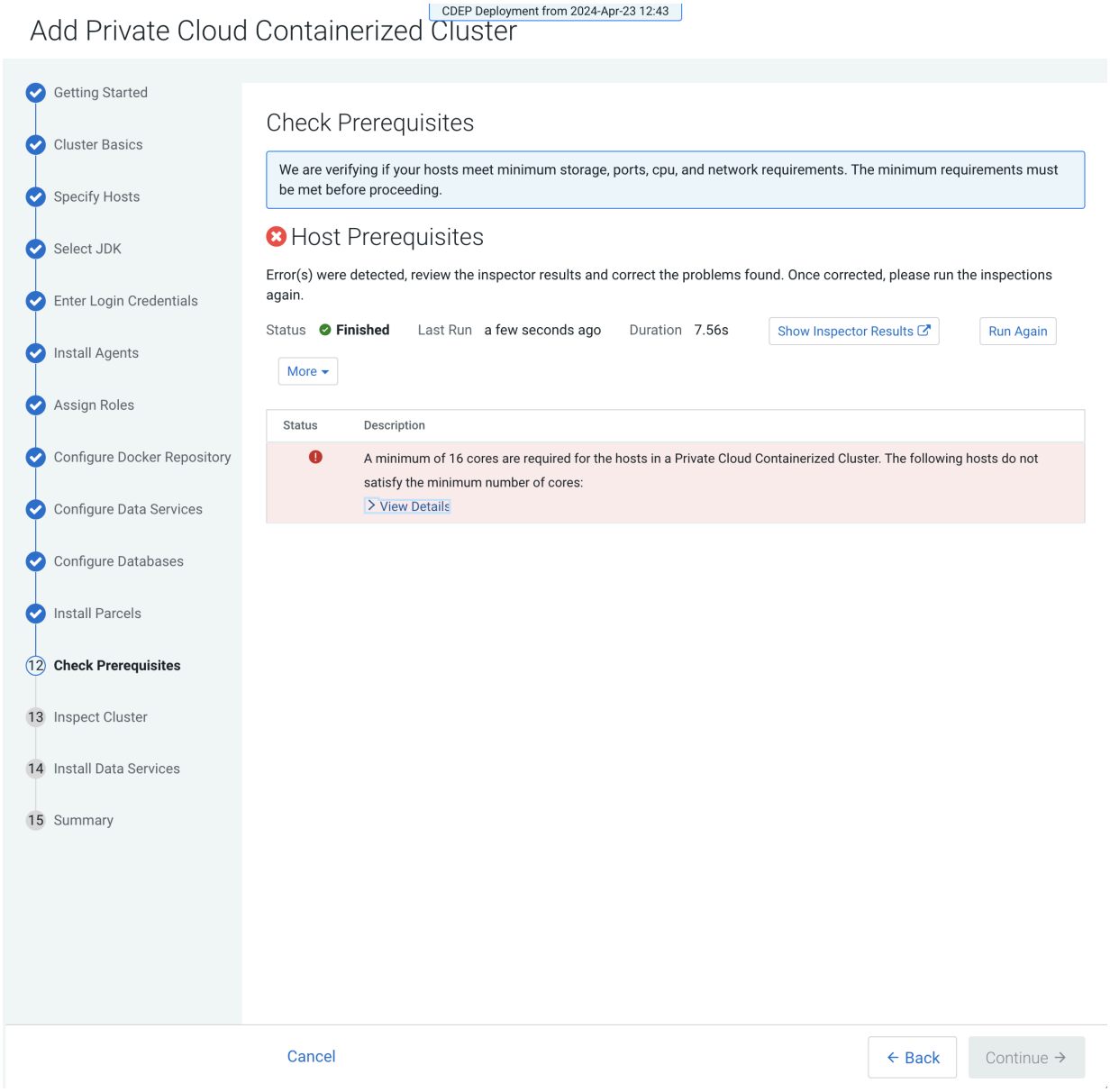
14. If the hosts do not meet the prerequisites, the Check Prerequisites page displays the applicable issues. Correct the issues, then click Run Again. After all of the issues have been resolved, click Continue.

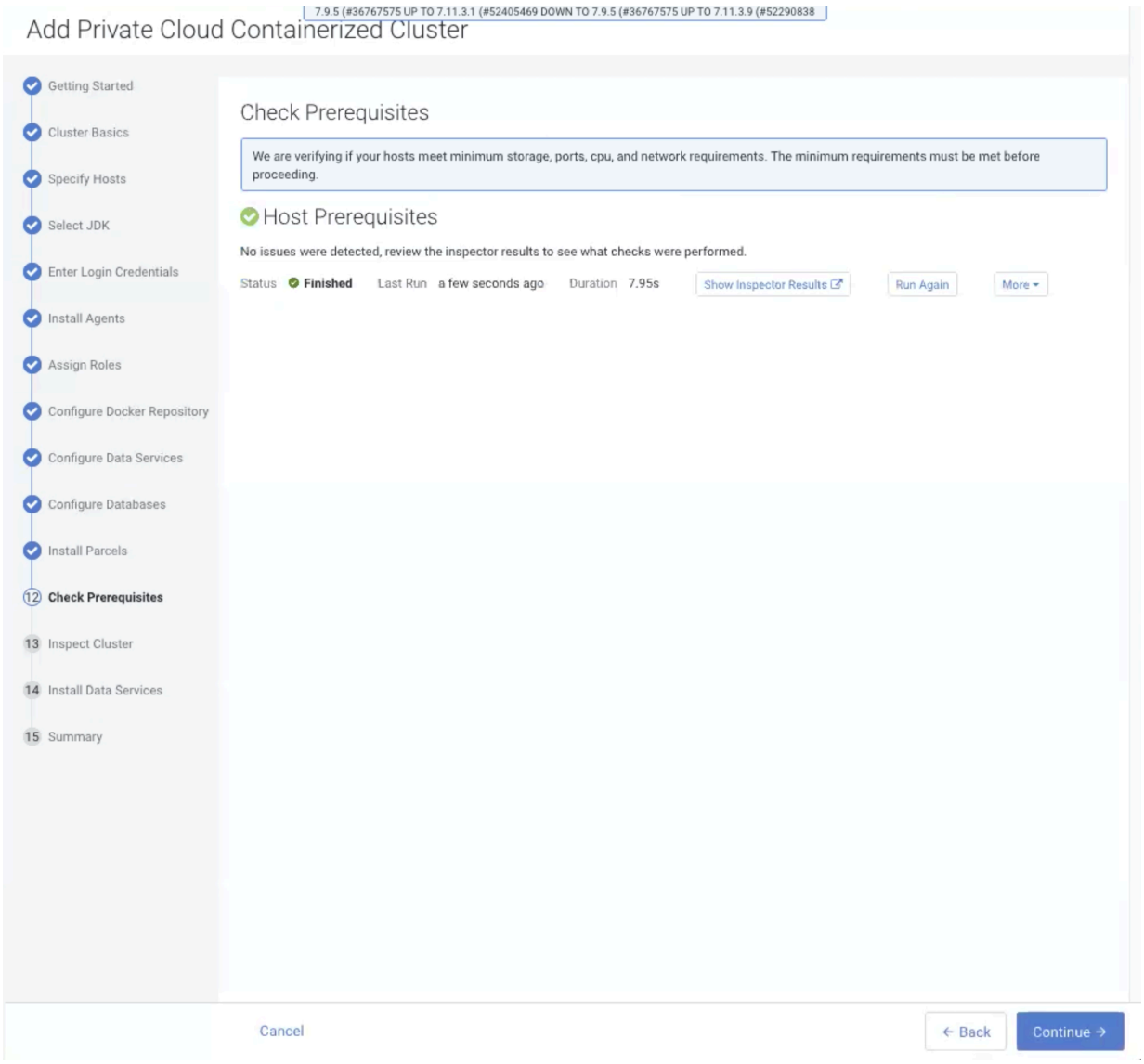
    The following prerequisites are checked:

Host Prerequisite Inspection	Validation
StorageInspection	Checks for a minimum of 300 GiB space in the /var/lib and docker data directories respectively. Checks if /var/lib/longhorn or its parent directories are symlinked. If they are, this inspection will fail.
CPUInspection	Checks to make sure the hosts have 16 virtual cores.
PortsInspection	Checks for the availability of ports 443 and 80.

EcsHostDnsInspection	Checks to make sure there are less than 3 nameserver entries in the /etc/resolv.conf file, and checks the connections to the Cloudera Manager cluster and the CDP console. It also checks to see if vault.localhost.localdomain's ping can be resolved. If not, it is likely that the host /etc/nsswitch.conf file is misconfigured. If this inspection fails: • Check the /etc/resolv.conf and /etc/nsswitch.conf files and ensure that /etc/resolv.conf does not contain 3 or more nameservers, and that /etc/nsswitch.conf must contain myhostname under the hosts field. • Check to see if the connections were resolved correctly. If connection to the CDP console fails, check to see if your DNS wildcard is configured properly.
VersionInspection	Checks that Java is installed and consistent among all ECS hosts.
IPTablesInspection	Checks that if the iptables command exists, rules are cleared. If the iptables command does not exist, iptables gets installed during FirstRun so this inspection passes. If iptables are installed and the rules are not cleared, this inspection will fail. For information on installing iptables, see Install iptables on the new Cloudera Embedded Container Service master nodes on page 52.

EcsCleanUpHostInspection	Checks to make sure that the /var/lib/rancher and docker data directories do not contain any files.

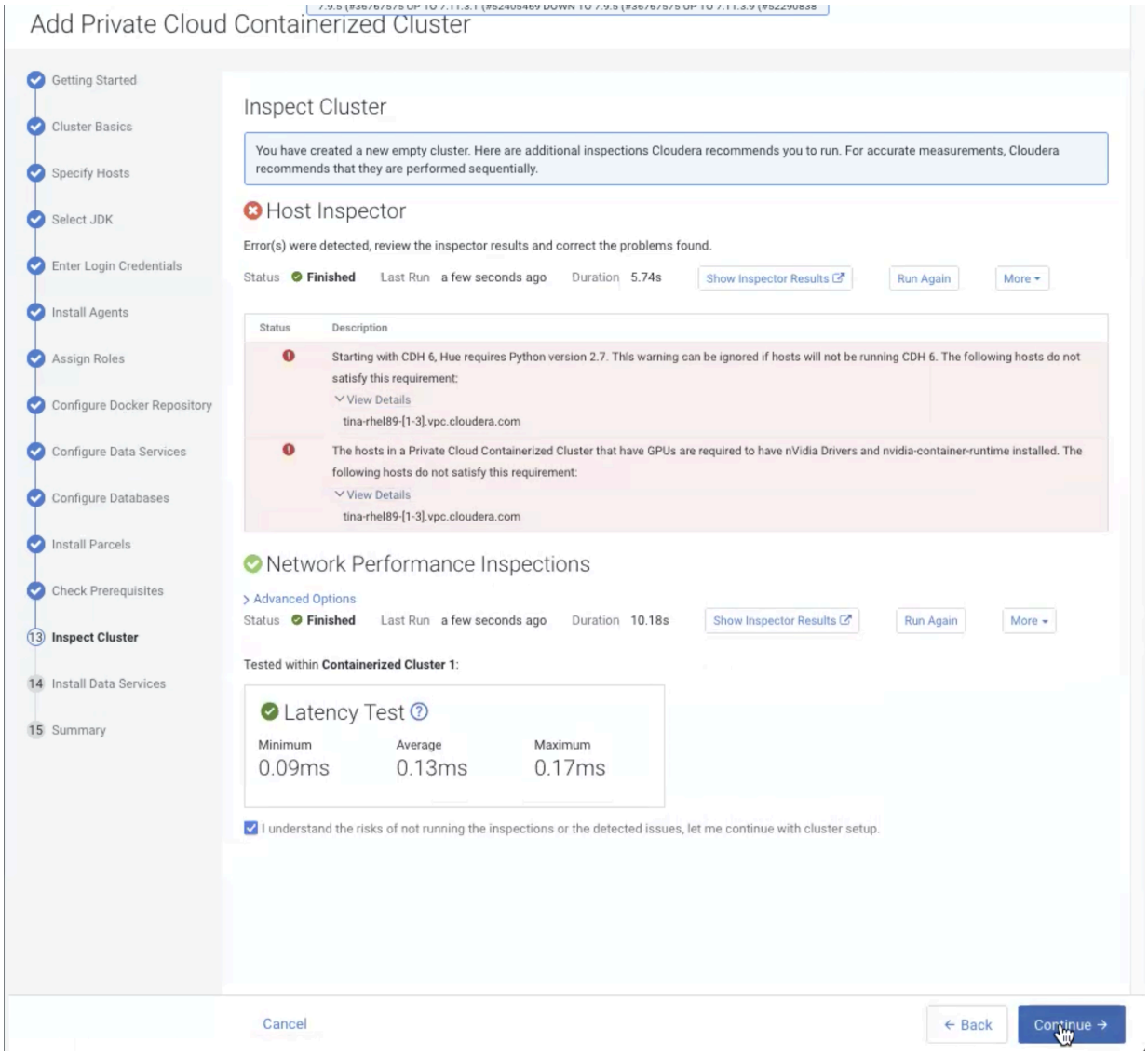



7.9.5 (#36767575 UP TO 7.11.3.1 (#52405469 DOWN TO 7.9.5 (#36767575 UP TO 7.11.3.9 (#52290838
Add Private Cloud Containerized Cluster
Getting Started
Cluster Basics
Specify Hosts
Select JDK
Enter Login Credentials
Install Agents
Assign Roles
Configure Docker Repository
Configure Data Services
Configure Databases
Install Parcels
12 Check Prerequisites
13 Inspect Cluster
14 Install Data Services
15 Summary
Check Prerequisites
We are verifying if your hosts meet minimum storage, ports, cpu, and network requirements. The minimum requirements must be met before proceeding.
Host Prerequisites
No issues were detected, review the inspector results to see what checks were performed.
Status Finished Last Run a few seconds ago Duration 7.95s
Show Inspector Results
Run Again
More
Cancel
← Back
Continue →

15. On the Inspect Cluster page, click Inspect Hosts and Inspect Network Performance to inspect your hosts and network performance . If the Inspect tool displays any issues, you can fix those issues and click Run Again to rerun the inspections. After all of the issues have been resolved, click Continue.

Note:

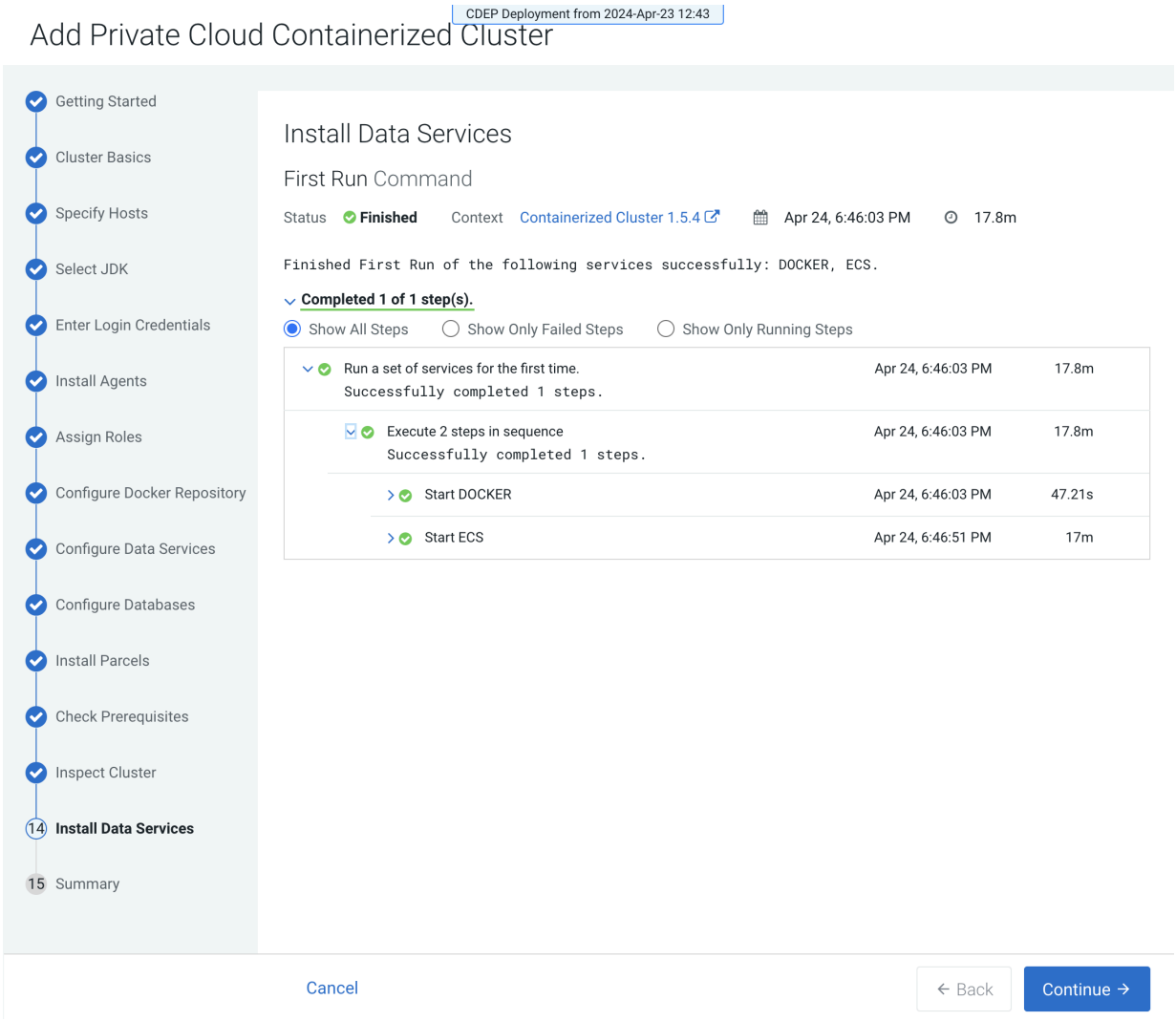
These inspections are more comprehensive host and network tests that you can optionally run. To skip these tests, select the I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup checkbox.



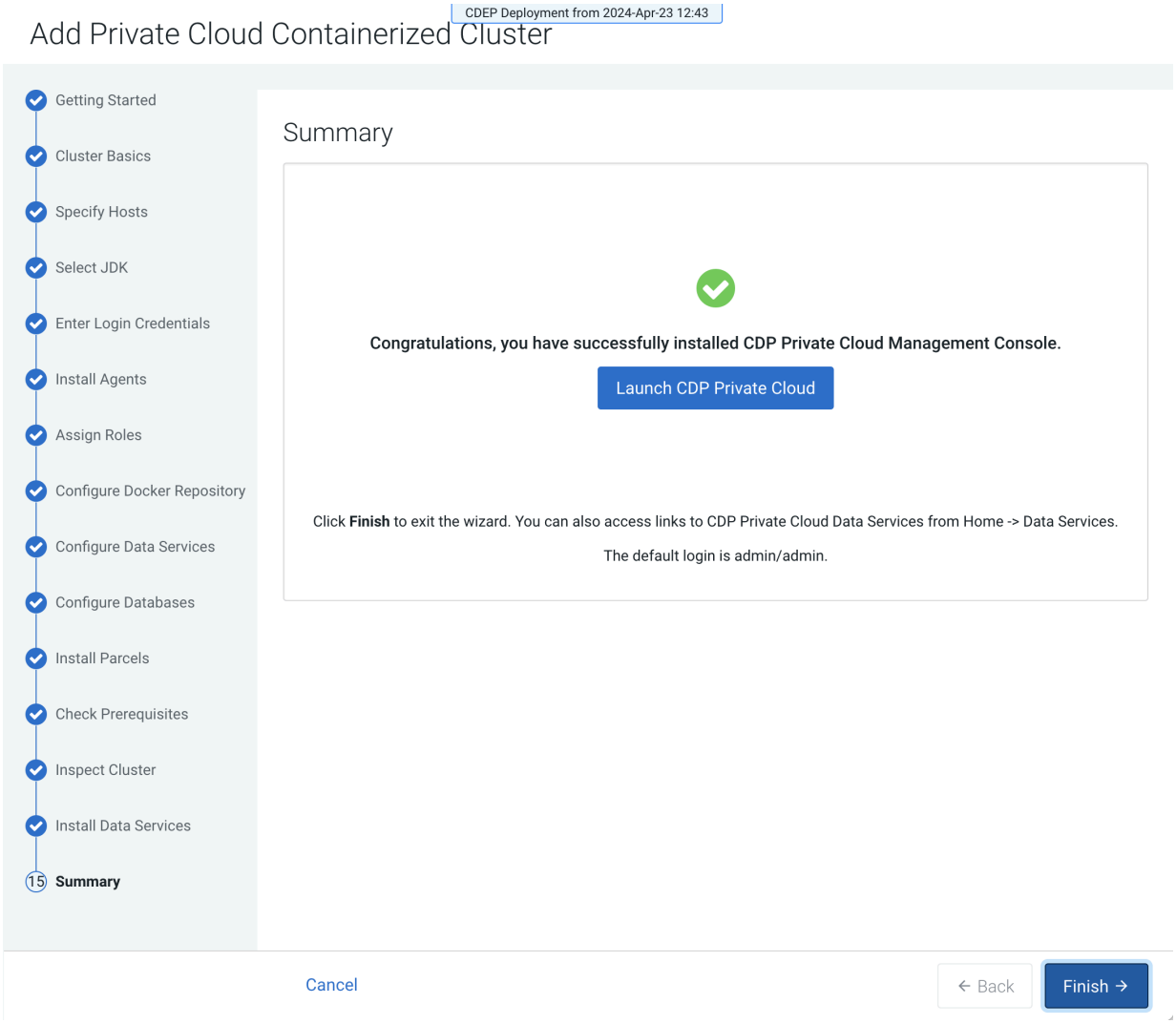
16. The installation progress is displayed on the Install Data Services page. When the installation is complete, click Continue.

Note: After the installation is complete, `Certification Manager` is installed by default from Cloudera Data Services on premises1.5.5 release.

For setting up Cerification Manager using Venafi TPP, see Setting up Certification Manager using Venafi TPP on page 48.



17. When the installation is complete, the Summary page appears. Click Launch CDP Private Cloud. You can also click Finish and then access the Data Services cluster from Cloudera Manager.



18. When the installation is complete, you can access your Private Cloud Data Services instance from Cloudera Manager. Click Data Services, then click Open Private Cloud Data Services for the applicable Data Services cluster.

If the installation fails, and you see the following error message in the stderr output during the Install Longhorn UI step, retry the installation by clicking the Resume button.

```
++ openssl passwd -stdin -apr1 + echo 'cm-longhorn:$apr1$gp2nrbtq$1KYPGI0QNl
FJ2lo5sV62l0' + kubectl -n longhorn-system create secret generic basic-auth
--from-file=auth + rm -f auth + kubectl -n longhorn-system apply -f /opt/clo
udera/cm-agent/service/ecs/longhorn-ingress.yaml Error from server (Internal
Error): error when creating "/opt/cloudera/cm-agent/service/ecs/longhorn-ing
ress.yaml":
Internal error occurred: failed calling webhook "validate.nginx.ingress.kub
ernetes.io": Post "https://rke2-ingress-nginx-controller-admission.kube-syst
em.svc:443/networking/v1/ingresses?timeout=10s": x509: certificate signed by
 unknown authority
```

What to do next

- If you specified a custom certificate, select the ECS cluster in Cloudera Manager, then select Actions > Update Ingress Controller. This command copies the cert.pem and key.pem files from the Cloudera Manager server host to the ECS Management Console host.
- Click Open Private Cloud Data Services to launch your Cloudera Data Services on premises instance.
- Log in using the default username and password admin.
- On the Welcome to CDP Private Cloud page, click Reset Admin Password to change the Local Administrator Account password.
- Set up external authentication using the URL of the LDAP server and a CA certificate of your secure LDAP. Set up external authentication using the URL of the SAML and uplolad SAML Identity Provider Metadata file. Follow the instructions on the Welcome to CDP Private Cloud page to complete this step.
- Click Test Connection to ensure that you are able to connect to the configured LDAP server.
- Create your first Virtual Warehouse in the Cloudera Data Warehouse Data Service
- Provision an AI Workbench in the Cloudera AI Data Service
- Add a CDE service in the Cloudera Data Engineering Data Service

Related Information

No Wildcard DNS/TLS Setup

Setting up Certification Manager using Venafi TPP

Follow the steps in this topic to setup cluster issuer for certification Manager using Venafi TPP. For more information, refer to the steps given here: https://cert-manager.io/docs/configuration/venafi/#creating-a-venafi-trust-protection-platform-issuer

Before you begin

When you start a Cloudera Data Services on premises service installation, make sure that you have installed a cluster issuer to use third-party certificates. To validate if there is a valid cluster issuer, see the following rules:

- We can create a clusterissuer without annotation. It is not activated until we add the below annotation:

```
kubectl annotate clusterissuer <ISSUER_NAME>
issuer.cdp.cloudera.com/type=longlived/shortlived
```

- The cluster issuer must have the following annotation to be activated, along with the label set as follows:

```
kubectl label clusterissuer <ISSUER_NAME> issuer.cdp.cloudera.com/projec
t=<CDP_NAMESPACE>
```

In ECS the CDP_NAMESPACE is "cdp". Once this is setup, you can test this by creating a test certificate and checking in their Venafi TPP instance that the certificate is created. A sample certificate will look like:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
    name: test-cert
    namespace: default
spec:
    secretName: test-venafi-tls # This will store the certificate
issuerRef:
    name: tpp-issuer
    kind: ClusterIssuer
    commonName: test.cdp.svc.cluster.local
dnsNames:
    - test.cdp.svc.cluster.local
privateKey:
    algorithm: RSA
    size: 2048
```

Procedure

An example setup is as follows:

1. Make sure the cert-manager is installed and all pods are up and running in the cert-manager namespace.
2. Create a secret in cert-manager namespace with the credentials to communicate with Venafi TPP instance.

   Refer to the below example:

```
Put the following contents in a file called tpp-secret.yaml
apiVersion: v1
data:
  password: Q2xvdWRlcmFAMTIz
  username: bG9jYWw6YWRtaW4=
kind: Secret
metadata:
  name: tpp-secret
  namespace: cert-manager
type: Opaque

kubectl apply -f tpp-secret.yaml

or

kubectl create secret generic tpp-secret \
  --from-literal=username='local:admin' \
  --from-literal=password='Cloudera@123' \
  -n cert-manager
```

3. Create a secret in the cert-manager namespace with below command if Venafi is configured with custom CA (Microsoft ADCS).

   Refer to the below example:

```
kubectl create secret generic qe-tpp-ca --from-file=ca.crt -n cert-manager

ca.crt file contents
-----BEGIN CERTIFICATE-----
MIIDkjCCAnqgAwIBAgIQMMMF/SbNzEuvDt9MJUkn/DANBgkqhkiG9w0BAQsFADBM
MScwJQYDVQQLEx5WZW5hZmkgT3BlcmF0aW9uYWwgQ2VydGlmaWNhdGUxITAfBgNV
BAMTGGFkMi5xZS1hZC0xLmNsb3VkZXJhLmNvbTAeFw0yNTA0MDMxMTI5NTRaFw0y
NjA0MDMxMTI5NTRaMEwxJzAlBgNVBAsTHlZlbmFmaSBPcGVyYXRpb25hbCBDZXJ0
aWZpY2F0ZTEhMB8GA1UEAxMYYWQyLnFlLWFkLTEuY2xvdWRlcmEuY29tMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3xj5x0OH1061qOPYCb0pbWnivfUo
C8JoktjWlGPK3LOFamKj46qWY7dF4AQHqQAypwLQTLwT7RsSUgZTZbkUGOQKN4CC
kLsxYa5WaHrKuypoxb5WxC31JrJDGUGcUapPcvXLVYKpqBOZLkH5JnJQN1U5KYuX
SBsgEDpU6v0ab51aB+KSdWZ9yyPnMXeJNfYbmQAO8yz7pHtDYZNXDoDoWAjOrUDg
6MUBeOkGsas5zB3Z0pXohT3BU5LQVRlEDsdM3gx5KG/KeaPALHhr74BSkZ7ogHxp
w8qDFFookoFQhwxu3H2hGImh9LJgnV+pMZ9MLBdw/dJxwlNY/LNuVagW8QIDAQAB
o3AwbjAdBgNVHQ4EFgQUI9UXcQZvJLFmkkRJV048s6ZeTSQwCQYDVR0TBAIwADAj
BgNVHREEHDAaghhhZDIucWUtYWQtMS5jbG91ZGVyYS5jb20wHQYDVR0lBBYwFAYI
KwYBBQUHAwEGCCsGAQUFBwMCMA0GCSqGSIb3DQEBCwUAA4IBAQA5hBjSlwF++pnG
T90XIZfe47+MaWRFG9KoT5nDhv+VA0355s13gWhAGDeZYPdtc9RLcKeTZQLWxF5k
09Rv0NkKxF+FkznaIUyabUOacAaRLUjSnB7sIaOGATvP2MRNo7NTZ4seMuMevCx+
Dx1jKwEBdwGZTiCkZPkYI473JapkHvPM/Zs4DGro9Pcapnf4e1z2s1IPgFZPmCWr
Z0yeQQ99uU24Ge+n2Fqy4TtcOW/OI+VEWS7uMZj2rAc9xanVxt54hdW+6SJgamf+
Cniw8vU8QriaAallq89uVFYXpfwxnoCA5I3qKh0KFpfxHuB+KjFFfqoY9pIyt/7Q
WANrC2+c
-----END CERTIFICATE-----
```

4. Create clusterissuer resource to be used with cert manager using below commands.

   Refer to the below example:

   a) Longlived cluster issuer - 365 days validity

```
Put the following contents in a file called longlived-issuer.yaml

apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  annotations:
    issuer.cdp.cloudera.com/type: longlived
  labels:
    issuer.cdp.cloudera.com/project: cdp
  name: tpp-issuer
spec:
  venafi:
    tpp:
      url: https://ad2.qe-ad-1.cloudera.com:8443
      credentialsRef:
        name: tpp-secret
      caBundleSecretRef:
        name: qe-tpp-ca
        key: ca.crt
    zone: \VED\Policy\Cloudera\Longlived
Run the following command to create the ClusterIssuer resource
kubectl apply -f longlived-issuer.yaml
```

   The ClusterIssuer should be configured successfully, i.e. READY column should have the value True.

```
kubectl get clusterissuer tpp-issuer
NAME                READY   AGE
tpp-issuer          True    26h
```

   b) Shortlived cluster issuer - 24 hours validity. Refer to the below example:

```
Put the following contents in a file called shortlived-issuer.yaml

apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  annotations:
    issuer.cdp.cloudera.com/type: shortlived
  labels:
    issuer.cdp.cloudera.com/project: cdp
  name: tpp-issuer-short
spec:
  venafi:
    tpp:
      url: https://ad2.qe-ad-1.cloudera.com:8443
      credentialsRef:
        name: tpp-secret
      caBundleSecretRef:
        name: qe-tpp-ca
        key: ca.crt
    zone: \VED\Policy\Cloudera\Shortlived

Run the following command to create the ClusterIssuer resource
```

```
kubectl apply -f shortlived-issuer.yaml
```

The Cluster Issuer should be configured successfully. That is, READY column should have the value True.

```
kubectl get clusterissuer tpp-issuer-short
NAME                READY   AGE
tpp-issuer-short          True    26h
```

Once the test is successfully verified, it can be deleted by running the following command:

```
kubectl delete certificate test-cert
```

Manually revoking certificates from Venafi TPP

The certrevoke operator is responsible for revoking certificates in Venafi TPP when they are deleted from Kubernetes. When the operator is offline (either scaled down or offline due to some issue) and the certificates are deleted, the corresponding CertMeta custom resources may become orphaned without being properly marked for revocation. These orphaned CertMeta resources do not have the certificate.cdp.cloudera.com/deleted-during-down time annotation set, making them invisible to the normal recovery process when the operator starts up again.

Impact

Certificates that were deleted while the certrevoke operator was offline remain valid in Venafi TPP, creating potential security risks if those certificates are compromised. The operator cannot automatically detect or revoke these certificates without manual intervention.

Detection

Use the following command to identify orphaned CertMeta resources that do not have corresponding Certificate resources anywhere in the cluster:

```
# Get all certificate names from all namespaces
CERT_NAMES=$(kubectl get certificate --all-namespaces -o jsonpath='{range .
items[*]}{.metadata.name}{"\n"}{end}')

# Check each CertMeta against the list of all certificate names
kubectl get certmeta -n cert-manager -o custom-columns="NAME:.metadata.name,
ISSUER:.metadata.annotations.certificate\.cdp\.cloudera\.com/issuer-name,PIC
KUP_ID:.spec.pickupId" | grep -v '^NAME' | while read -r name issuer pickup_
id; do
  if ! echo "$CERT_NAMES" | grep -q "^$name$"; then
    echo "Orphaned CertMeta: $name, Issuer: $issuer, Pickup ID: $pickup_id"
  fi
done
```

Manual Revocation

For each orphaned CertMeta identified, manually revoke the certificates in Venafi TPP:

1. Use the information from the detection command to identify the certificates.
2. Revoke them manually through the Venafi TPP iterface.
3. Delete the CertMeta resources using the following command:

```
kubectl delete certmeta <certmeta-name> -n cert-manager
```

ECS Server High Availability

ECS Server High Availability (HA) is not enabled by default – you must enable it after installing Cloudera Embedded Container Service. If you do not wish to enable ECS HA, you can safely ignore this section. If you are enabling Cloudera Embedded Container Service 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 Cloudera Embedded Container Service 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 Cloudera Embedded Container Service HA, you can safely ignore this section.

As a prerequisite, during the installation, you must have installed Cloudera Embedded Container Service 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 Cloudera Embedded Container Service master nodes

You must install iptables on all of the additional Cloudera Embedded Container Service master nodes.

If your Cloudera Embedded Container Service hosts are running on the CentOS 8.4, OEL 8.4, RHEL 8, or RHEL 9 operating systems, you must install iptables on all the Cloudera Embedded Container Service hosts. Run the following command on each additional Cloudera Embedded Container Service master node:

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

For RHEL 9 only:

Note: RHEL 9 has deprecated iptables-nft and must use iptables-legacy. Install iptables if you do not have iptables installed on the hosts.

1. If iptables have been installed, then check their iptables version by using the following command:

```
iptables -V
```

2. This returns the iptables version running in the backend. For example:

```
iptables v1.8.4 (nf_tables)
```

3. If the version provides `nf_tables` instead of legacy, you must change the iptables binary to use `iptables-legacy` in the backend. Remove the symlink between iptables and iptables-nft and then symlink iptables to iptables-legacy by running the following command:

```
ln -s /usr/sbin/iptables-legacy /usr/sbin/iptables
ln -s /usr/sbin/iptables-legacy-save  /usr/sbin/iptables-save
ln -s /usr/sbin/iptables-legacy-restore /usr/sbin/iptables-restore
```

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.

Rolling Restart of an Cloudera Embedded Container Service

You must perform a rolling restart of Cloudera Embedded Container Service.

1. Log in to Cloudera Manager.
2. Navigate to the ECS service.
3. On the Home > Status tab, click the Actions Menu to the right of the cluster name and select Rolling Restart.
4. Click the Rolling Restart button that appears in the next screen to confirm. On this screen, you can select the services (Docker or /and ECS), Roles (Workers only, Non-workers only, All Roles).

   Note: Workers only refers to ECS agents, Non-workers only refers to all docker roles and ECS server.The Command Details window shows the progress of rolling restart of a batch of nodes. Here, batch size refers to the number of worker roles that can be restarted in parallel. The Batch size is 1 by default.
5. Click Actions > Unseal Vault .

   Note: There should always be one ECS server up at all times. Hence, a rolling restart or individually restarting the ECS server is required.

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 Cloudera Embedded Container Service Installation

If you want to enable ECS Server for High Availability after installing Cloudera Embedded Container Service, 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 Cloudera Embedded Container Service 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 Cloudera Embedded Container Service cluster installed and configured with a single ECS Server.

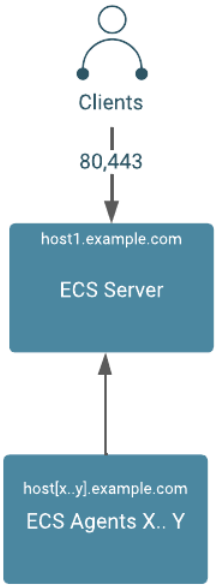
If you have a production quality Cloudera Embedded Container Service cluster, Cloudera recommends that you configure ECS Server High Availability. You can also consider having an ECS Server HA for any non-production Cloudera Embedded Container Service cluster that you expect to be available long-term.

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:

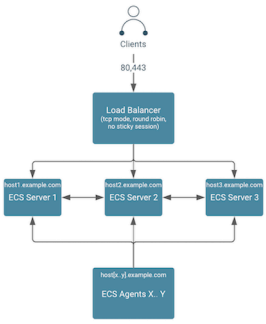
- An odd number of server nodes that will run etcd, the Kubernetes API, and other control plane services. Cloudera recommends a minimum of three ECS Server nodes.
- Two or more agent nodes that are designated to run Cloudera data services.
- A software or hardware Load balancer using TCP mode (non-terminating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



Architecture of Cloudera Data Services on premises 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 Cloudera Embedded Container Service hosts. However, Cloudera recommends six or more hosts.
2. DNS requirements for Cloudera Embedded Container Service High Availability must be fulfilled.

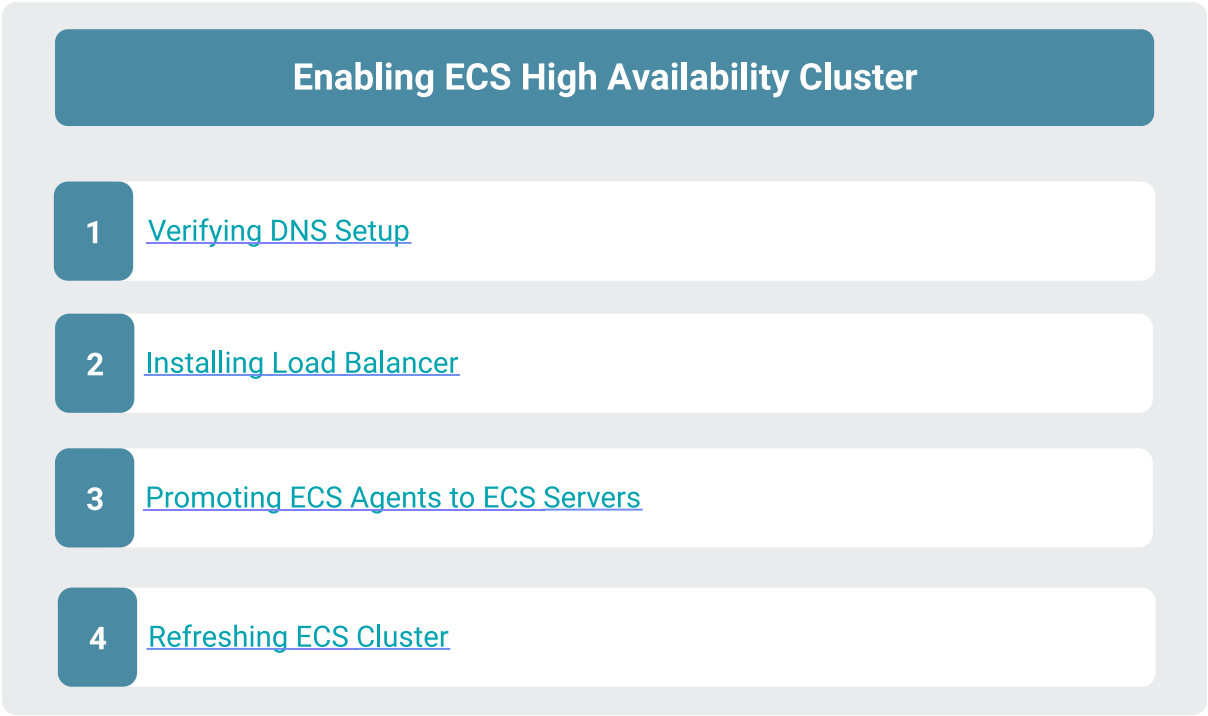
Hostname	Subdomain	Expected Roles	DNS ForwardZone	Reverse Zone PTR
“Wildcard” (hostname = *)	apps.ecs.example.com The string “apps” is required, “ecs” is up to user	Virtual app domain wildcard	“A Record” wildcard (hostname = *), may be a CNAME on certain DNS systems that use text-based config. Resolves 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 Cloudera Embedded Container Service 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 Cloudera Embedded Container Service High Availability cluster

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



Note:

1. You must have installed an Cloudera Embedded Container Service 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.

Procedure

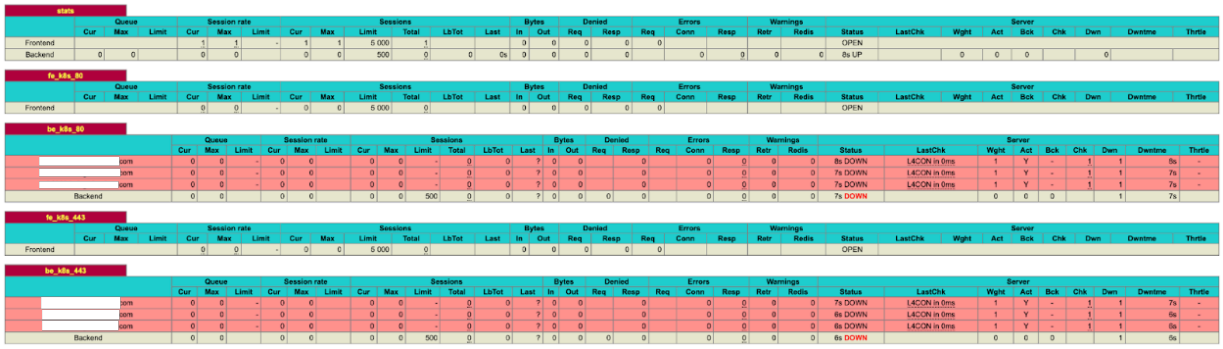
1. To install a HAProxy Load Balancer, you must ssh into the HAProxy host, install, and then configure HAProxy:

```
sudo su -
yum install haproxy -y
cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
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
```

2. 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

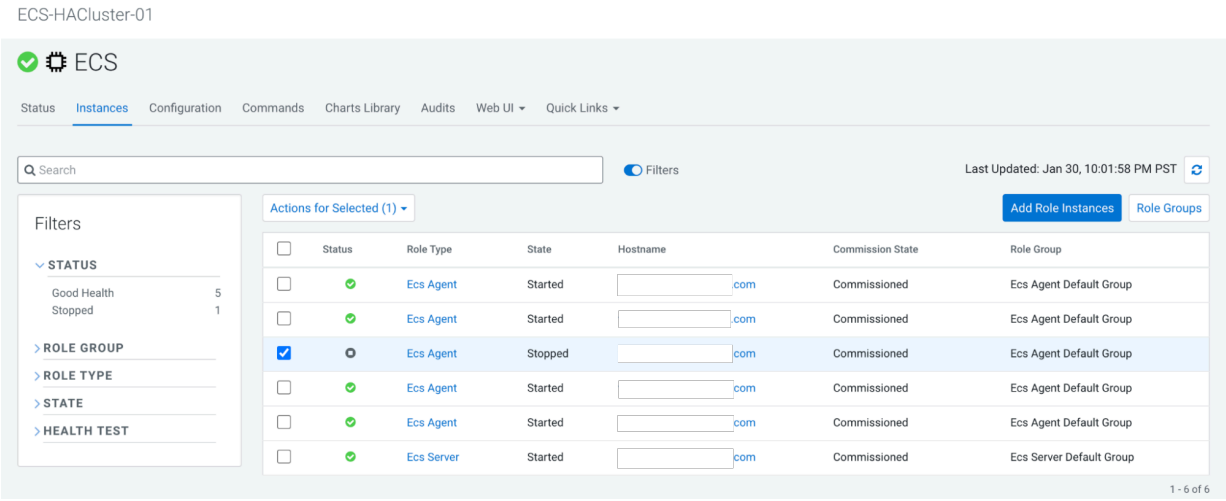
After installing the Load Balancer, you must reconfigure the existing Cloudera Embedded Container Service Agents to ECS Servers. This process is referred to as promoting the agents to servers. You must promote only one agent at a time.

About this task

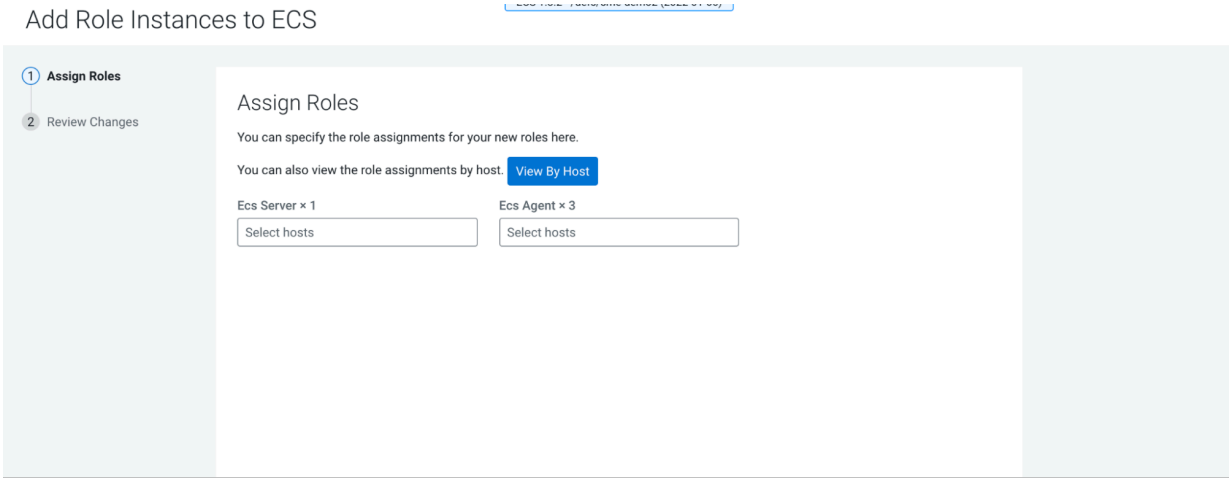
In this example we will promote the ECS agent on agent1.example.com and then promote the ECS agent on agent2.example.com.

Procedure

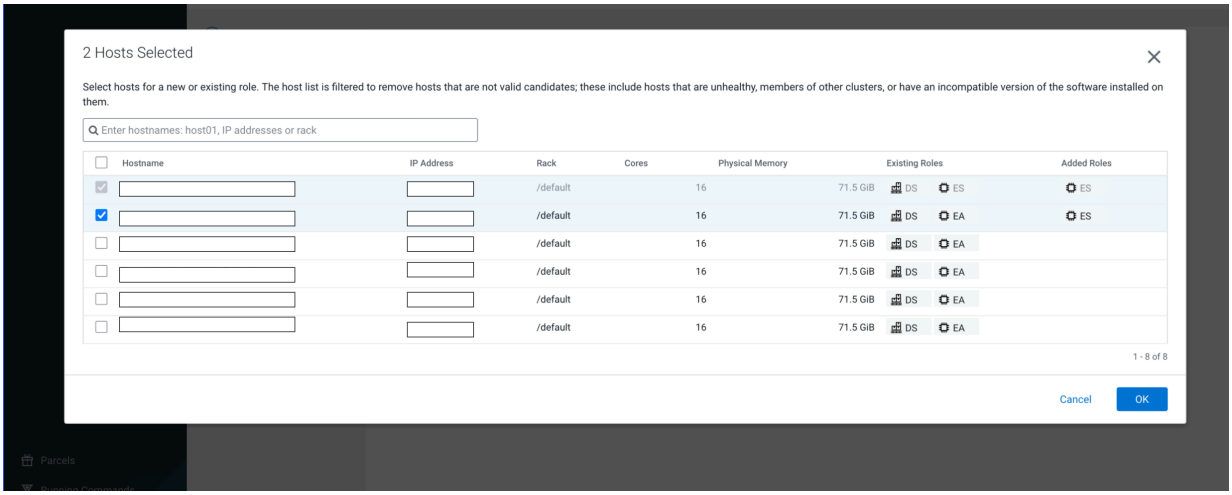
1. In Cloudera Manager, select the ECS cluster, then click ECS. Stop the ECS agent running on agent1 and then delete the agent.

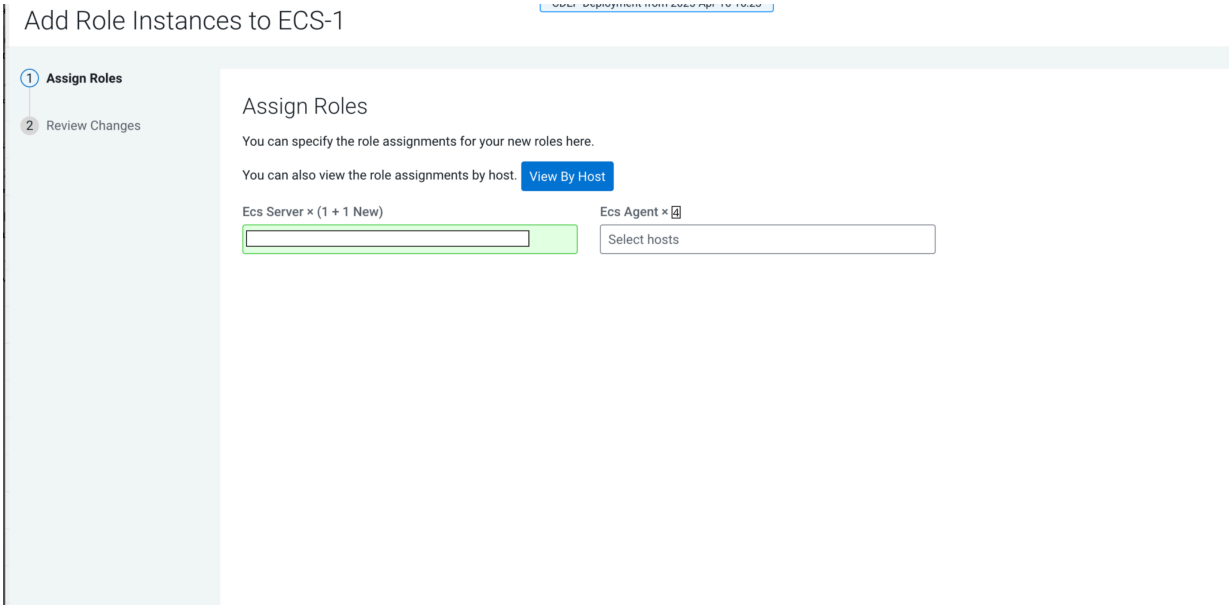


2. In ECS, click Add Role Instances.

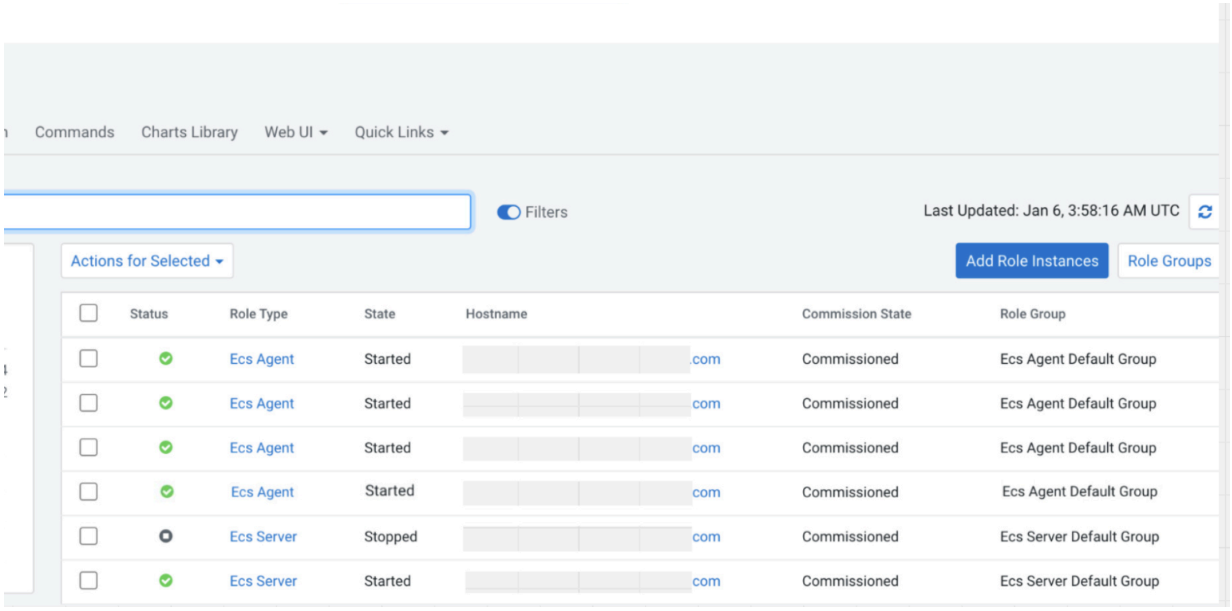


3. Select the available host as an ECS server and then select the Add Role Instances to ECS in the pop-up. Click OK.

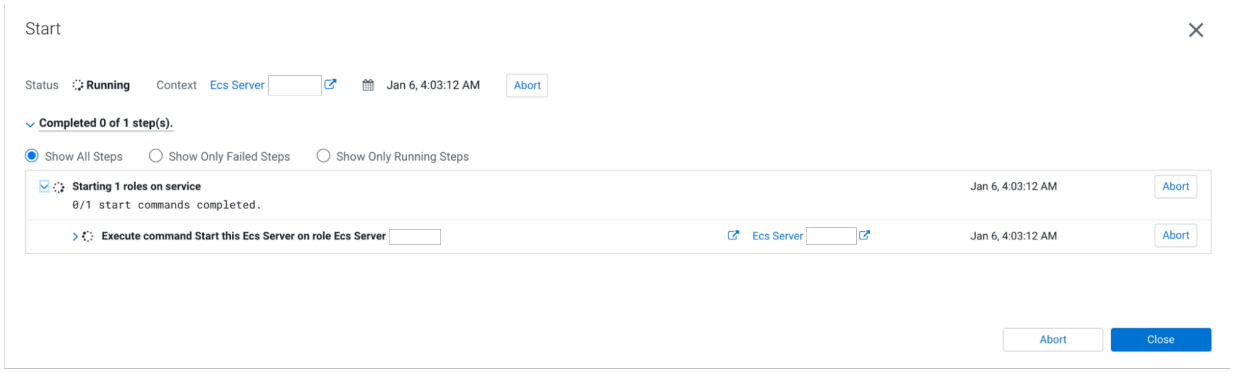




4. Click Continue.



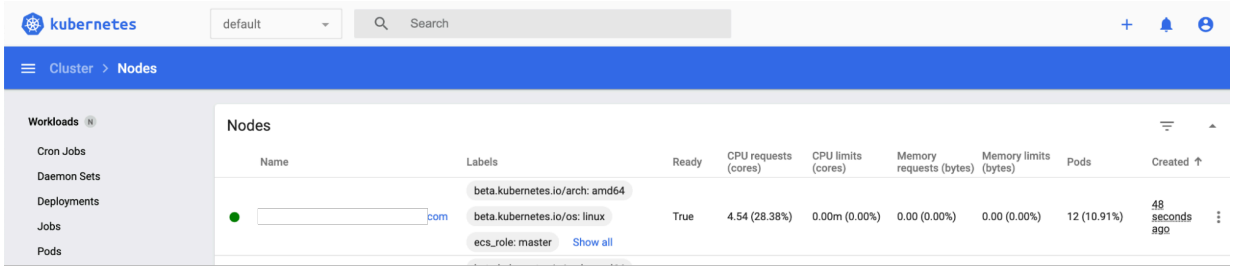
5. Start the new ECS server from the ECS Instances view. For example, start the ECS server on agent1.



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

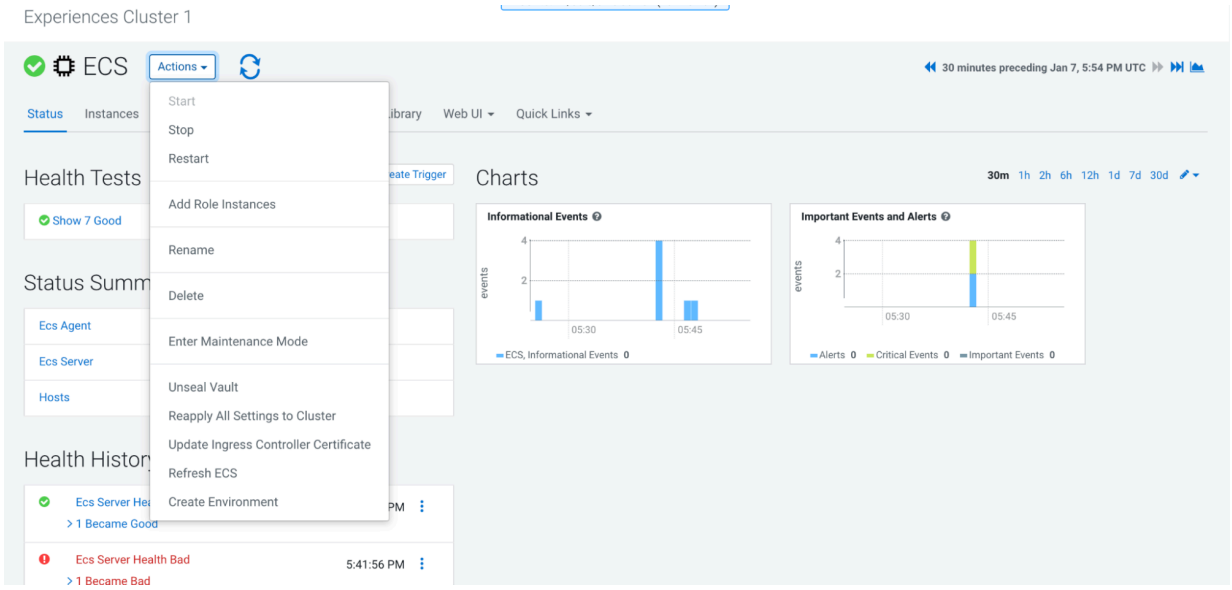
When agent1 is ready, you can promote agent2. To promote agent2, perform steps 1-8 again using agent2.example.com.

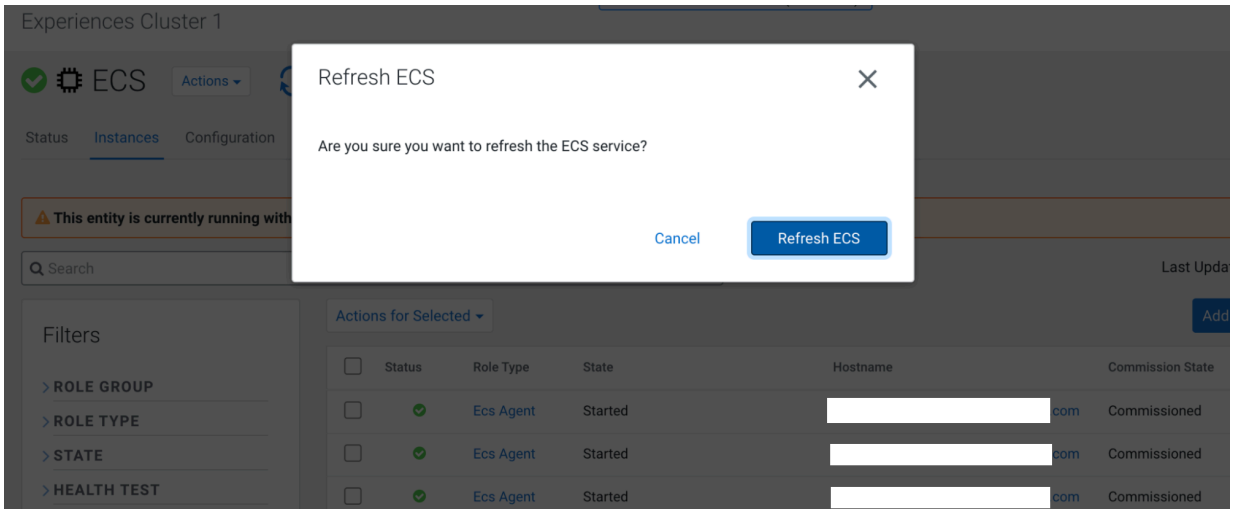
Refreshing Cloudera Embedded Container Service

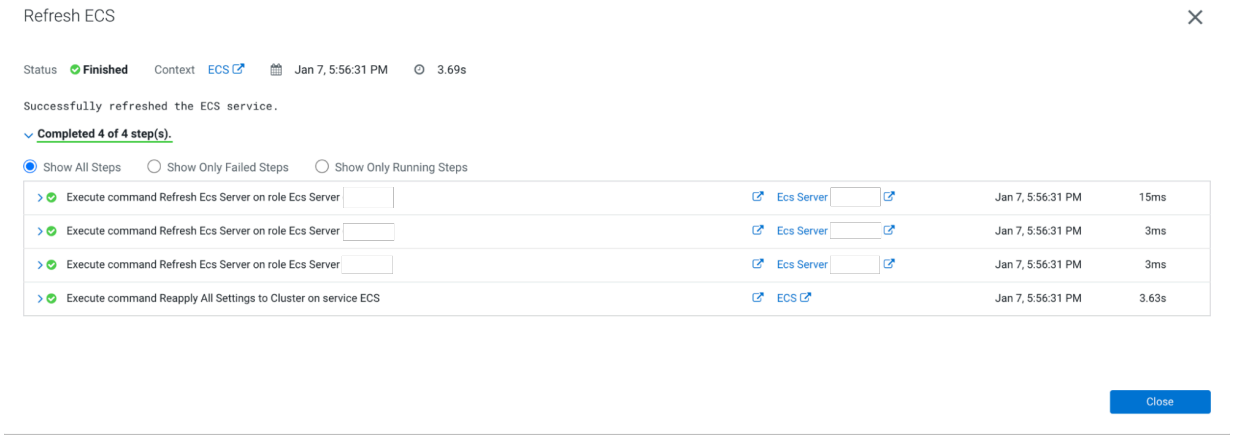
After all the ECS Agents are promoted to ECS Servers, you must log in to Cloudera Manager and refresh the ECS cluster.

Procedure

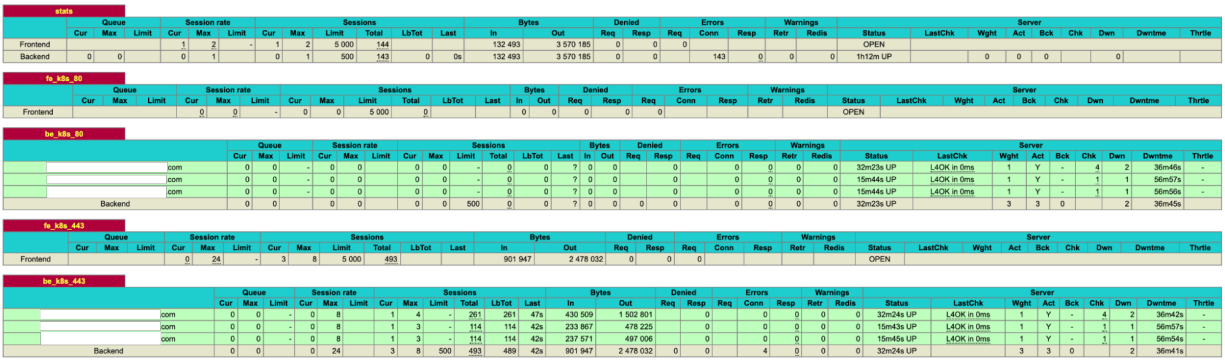
1. Navigate to ECS Cluster >> ECS view >> Actions >> Refresh ECS. This sets the ingress proxy so that all three servers are eligible to process incoming commands.







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	32m23s UP		3	3	0		2	36m45s	

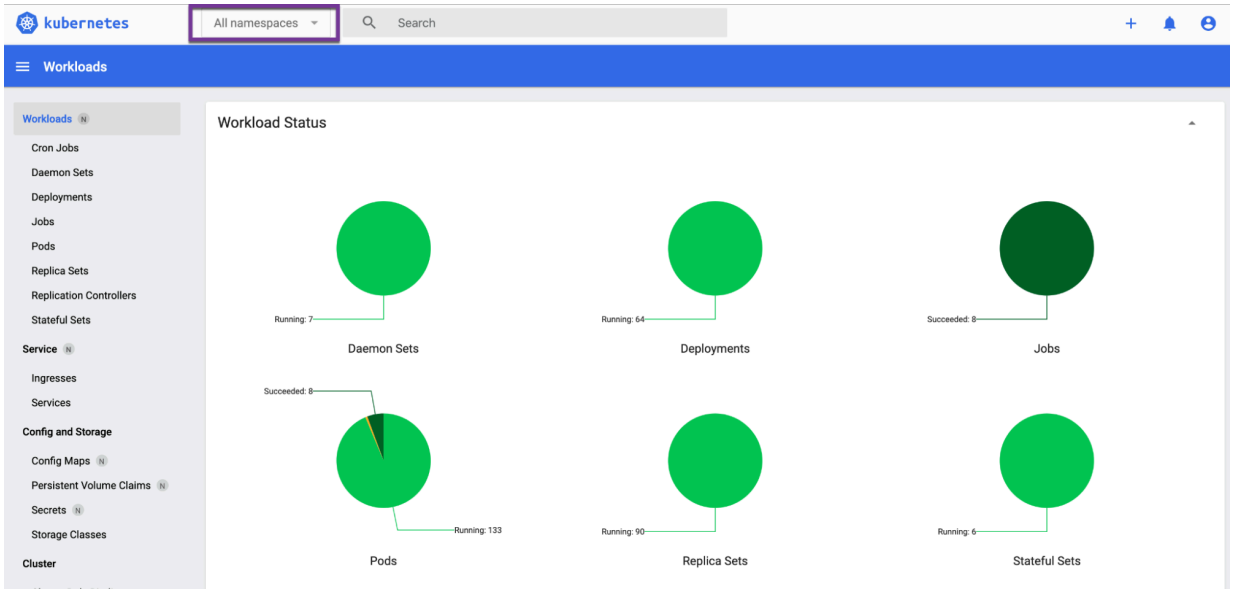
fe_k8s_443

	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								

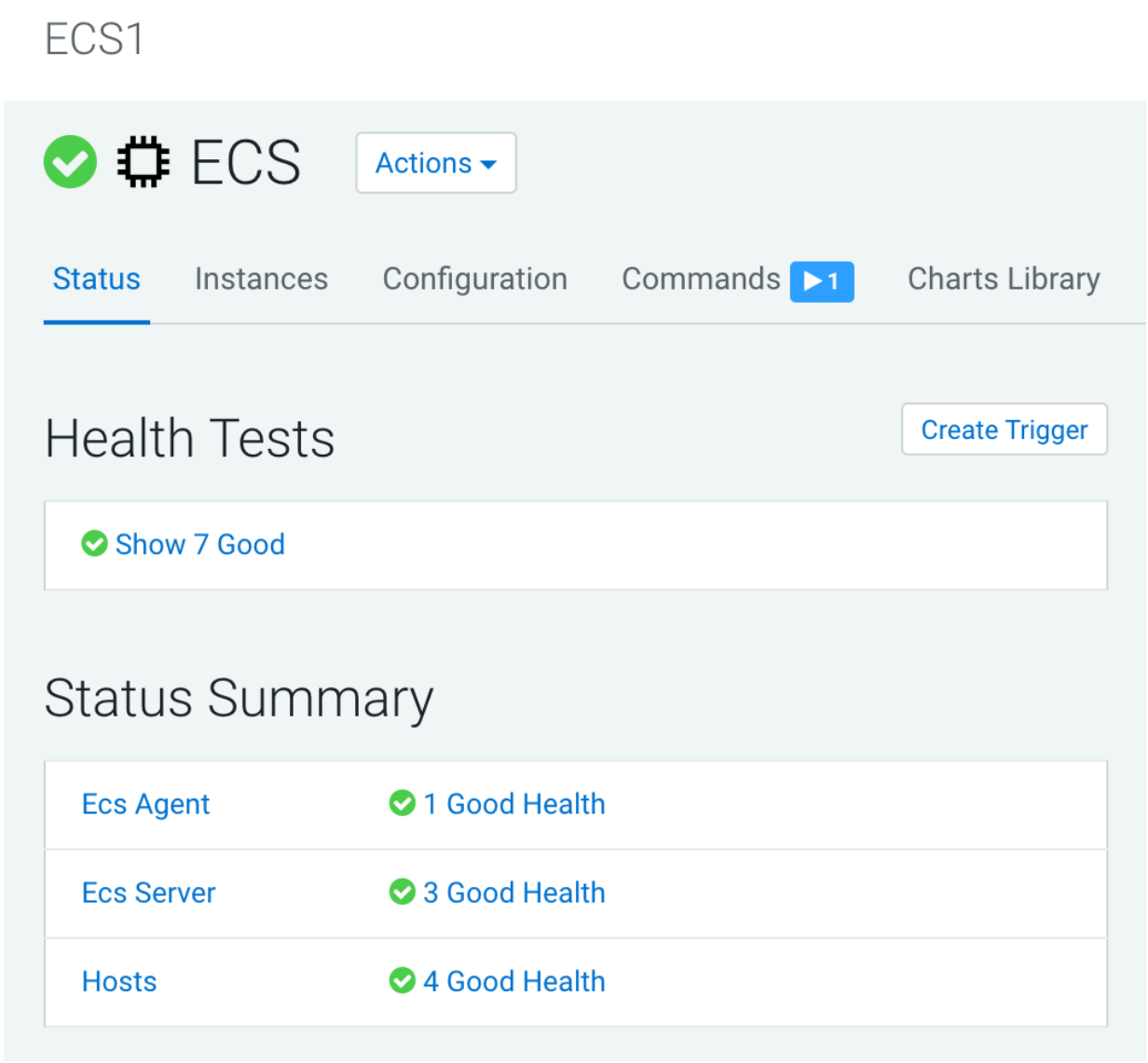
be_k8s_443

		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.

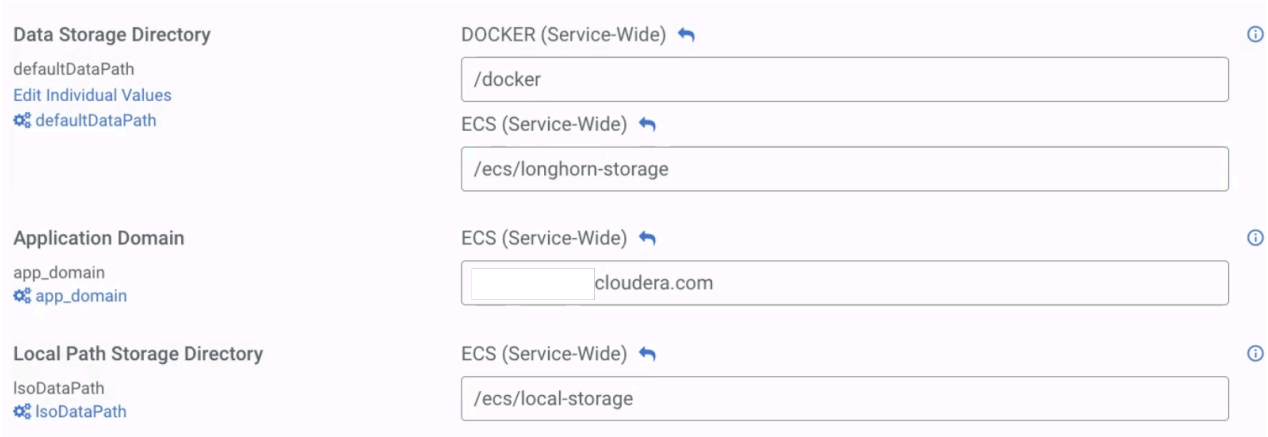
Manually uninstalling ECS from a cluster

You can manually uninstall ECS from your cluster.

Before you begin

Before performing this procedure, ensure that you have activated the ECS parcel on the cluster hosts.

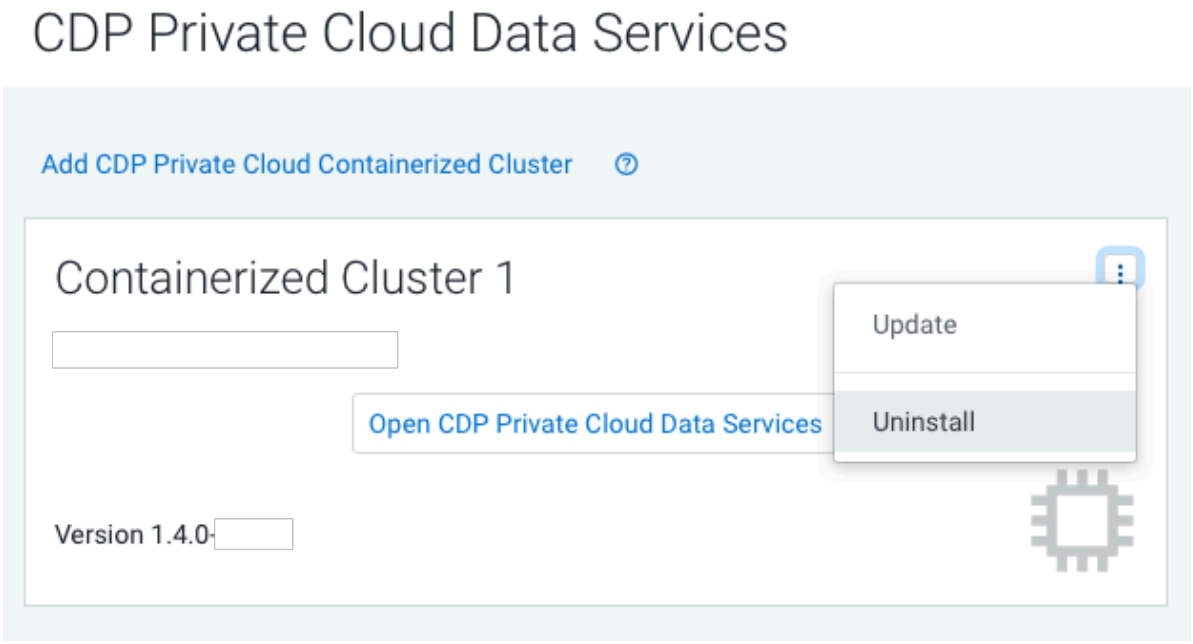
During the installation time of ECS, the directory for Longhorn and the LSO are decided by Cloudera Manager and defaults to /ecs.



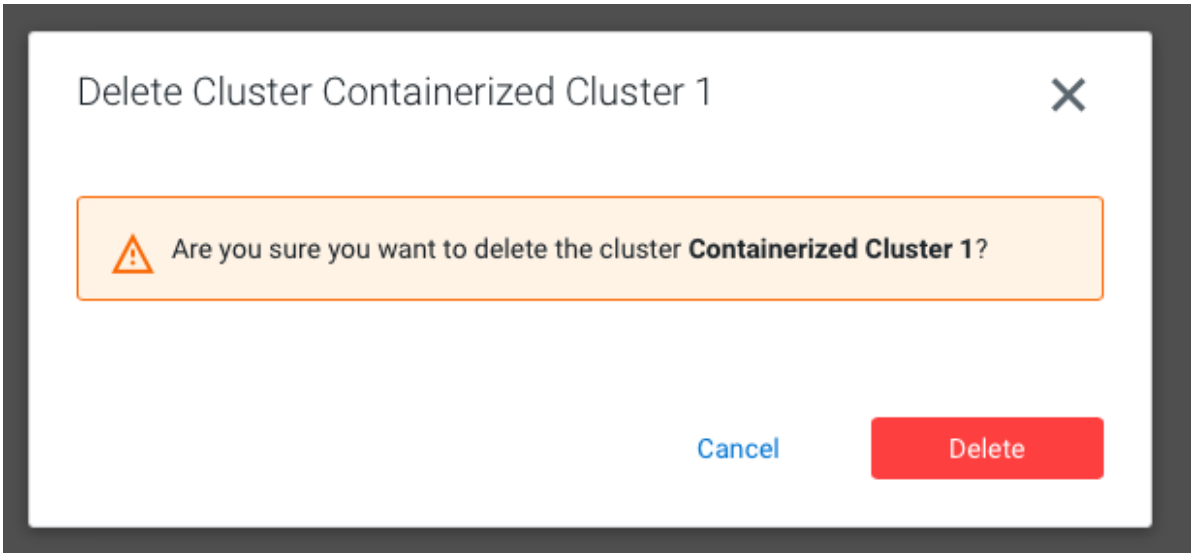
Procedure

1. On each host in the cluster:
   a) /opt/cloudera/parcels/ECS/docker/docker container stop registry
   b) /opt/cloudera/parcels/ECS/docker/docker container rm -v registry
   c) /opt/cloudera/parcels/ECS/docker/docker image rm registry:2
2. Stop the ECS cluster in Cloudera Manager
3. On each host:
   a) cd /opt/cloudera/parcels/ECS/bin
   b) ./rke2-killall.sh # usually 2 times is sufficient
   c) Use umount to unmount all NFS disks.
   d) ./rke2-uninstall.sh
   e) rm -rf /ecs/* # assumes the default defaultDataPath and lsoDataPath
   f) rm -rf /var/lib/docker_server/* # deletes the auth and certs
   g) rm -rf /etc/docker/certs.d/* # delete the ca.crt
   h) rm -rf /docker # assumes the default defaultDataPath for docker
   i) rm -rf /var/lib/rancher/*

4. Delete the ECS cluster in Cloudera Manager.
   a) In Cloudera Manager, navigate to CDP Private Cloud Data Services and click . Click Uninstall.



   b) The Delete Cluster wizard appears. Click Delete.



5. Clean IPtables on each host:

```
echo "Reset iptables to ACCEPT all, then flush and delete all other chai
ns";
                        declare -A chains=( [filter]=INPUT:FORWARD:OUTPUT
 [raw]=PREROUTING:OUTPUT [mangle]=PREROUTING:INPUT:FORWARD:OUTPUT:POSTROUT
ING [security]=INPUT:FORWARD:OUTPUT [nat]=PREROUTING:INPUT:OUTPUT:POSTRO
UTING );
                        for table in "${!chains[@]}"; do
                        echo "${chains[$table]}" | tr : $"\n" | while IFS=
 read -r;
                        do sudo iptables -t "$table" -P "$REPLY" ACCEPT
                        done
```

```
                        sudo iptables -t "$table" -F
                        sudo iptables -t "$table" -X
                        done
```

Note: Alternatively, an experimental script is available. This script combines steps three through five. The script is available here: https://github.com/cloudera-labs/snippets/blob/main/private-cloud/kill-2-rke.sh script

6. Reboot the host(s).
7. Before you install ECS again, ensure that the IP tables list is empty by executing the following command: #iptables -L