Cloudera AI

on premises requirements

Date published: 2020-07-16
Date modified: 2025-10-31

https://docs.cloudera.com/

Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Introduction to Cloudera on premises

With the Cloudera AI service, data scientists and partners can build and run machine learning experiments and workloads in a secure environment. Cloudera AI on premises provides an identical experience to Cloudera AI on cloud, but running in your own on-premises data center.

Cloudera AI enables you to:

- Easily onboard a new tenant and provision an Cloudera AI Workbench in a shared OpenShift or Cloudera Embedded Container Service environment.
- Enable data scientists to access shared data on Cloudera Base on premises and Cloudera Data Warehouse.
- Leverage Spark-on-K8s to spin up and down Spark clusters on demand.

Requirements for Cloudera AI on Openshift Container Platform

To launch the Cloudera AI service, the OpenShift Container Platform (OCP) host must meet several requirements. Review the following Cloudera AI-specific software, NFS server, and storage requirements.

Requirements

Note:

Only the usage of SSD disks is supported with Cloudera on premises on OpenShift Container Platform.

If needed, reach out to your Administrator to ensure the following requirements are met.

Compatibility requirements

- If you use OpenShift, check that the version of the installed OpenShift Container Platform is exactly as listed in Software Support Matrix for OpenShift.

Required accesses

- If Cloudera AI needs access to a database on the Cloudera Base on premises cluster, then the user must be authenticated using Kerberos and must have Ranger policies set up to allow read/write operations to the default (or other specified) database.
- Ensure that Kerberos is enabled for all services in the cluster. Custom Kerberos principals are not supported currently. For more information, see Authenticating Hue users with Kerberos.
- Cloudera AI assumes it has cluster-admin privileges on the cluster.
- If external NFS is used, the NFS directory and assumed permissions must be those of the cdsw user. For details see Using an External NFS Server.
- If you intend to access a workbench over https, see Deploying a Cloudera AI Workbench with support for TLS.

Requirements for functioning

- On OpenShift Container Platform, CephFS is used as the underlying storage provisioner for any new internal workbench on Cloudera on premises 1.5.x. A storage class named ocs-storagecluster-cephfs with csi driver set to openshift-storage.cephfs.csi.ceph.com must exist in the cluster for new internal workbenches to get provisioned.
- A block storage class must be marked as default in the cluster. This may be rook-ceph-block, Portworx, or another storage system. Confirm the storage class by listing the storage classes (run `oc get sc`) in the cluster, and check that one of them is marked default.

DNS-related requirements

- Forward and reverse DNS must be working.

- DNS lookups to sub-domains and the Cloudera AI Workbench itself shall work properly.
- In DNS, wildcard subdomains (such as *.cml.yourcompany.com) must be set to resolve to the master domain (such as cml.yourcompany.com). The TLS certificate (if TLS is used) must also include the wildcard subdomains. When a session or job is started, an engine is created for it, and the engine is assigned to a random, unique subdomain.

Configuration requirements

- The external load balancer server timeout needs to be set to 5 min. Without this, creating a project in a Cloudera AI Workbench with `git clone` or with the API may result in API timeout errors. For workarounds, see *Known Issue DSE-11837*.
- For non-TLS Cloudera AI Workbench, websockets need to be allowed for port 80 on the external load balancer.
- Only a 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 on premises or on cloud trusted Certificate Authority (CA).
- On OpenShift, due to a Red Hat issue with OpenShift Container Platform 4.17, the image registry cluster operator configuration must be set to Managed.
- Check if storage is set up in the cluster image registry operator. See *Known Issues DSE-12778* for further information.

For more information on requirements, see Cloudera Base on premises Installation Guide.

Hardware requirements

Storage

The cluster must have persistent storage classes defined for both block and filesystem volume Modes of storage. Ensure that a block storage class is set up. The exact amount of storage classified as block or filesystem storage depends on the specific workload used:

Table 1: Storage requirements for Cloudera AI on OCP

	Local Storage (for example, ext4)	**Block PV (for example, Ceph or Portworx)**	**NFS (for Cloudera AI user project files)**
Cloudera Control Plane	N/A	250 GB	N/A
Cloudera AI	N/A	The total (not per node) storage needed only for Cloudera AI in OCP without disaster recovery (drs) is 1800 Gi per workbench with the external NFS. If the Cloudera AI Workbench is using the internal NFS, the minimum storage needed per workbench is 3800 Gi, considering the replication factor of 2. If you have a different replication configured, this will change accordingly. Considering the DRS and a single backup of the workbench, the total storage needed is 1800 Gi * 2 = 3600 Gi for the workbench with external NFS. If the workbench uses internal NFS, the total storage needed is 7600Gi.	1 TB per workbench (dependent on the size of the Cloudera AI user files)

Note:

NFS storage must be routable from all pods running in the cluster.

Note:

For monitoring, the recommended volume size is 60 GB.

Note:

The storage calculation is based on the replication factor of two. If a different replication factor is used, the calculation will adjust accordingly.

External NFS considerations

Cloudera AI supports NFS versions 3.0, 4.0, 4.1 and 4.2 versions for storing project files and folders. NFS storage is to be used only for storing project files and folders, and not for any other Cloudera AI data, such as PostgreSQL database and LiveLog.

OpenShift requirements for NFS storage

An internal user-space NFS server can be deployed into the cluster which serves a block storage device (persistent volume) managed by the cluster's software defined storage (SDS) system, such as Ceph or Portworx. This is the recommended option for Cloudera AI on OpenShift. Alternatively, the NFS server can be external to the cluster, such as a NetApp filer that is accessible from the on premises cluster nodes. NFS storage is to be used only for storing project files and folders, and not for any other Cloudera AI data, such as PostgreSQL database and LiveLog.

Cloudera AI does not support shared volumes, such as Portworx shared volumes, for storing project files. A read-write-once (RWO) persistent volume must be allocated to the internal NFS server (for example, NFS server provisioner) as the persistence layer. The NFS server uses the volume to dynamically provision read-write-many (RWX) NFS volumes for the Cloudera AI clients.

Related Information

Cloudera Private Cloud Base Installation Guide

Cloudera Data Services on premises Software Requirements

Cloudera Data Services on premises Hardware Requirements

Known Issues and Limitations

Deploy a Cloudera AI Workbench with Support for TLS

Using an External NFS Server

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

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*.

Related Information

Installation using the Embedded Container Service (ECS)

Certification Manager service for increased security

Cloudera AI requires a wildcard certificate to support Cloudera AI workloads.

When workloads such as sessions, jobs, applications, and models are created, Cloudera AI generates random, unique subdomains. As these subdomains are not deterministic, a wildcard certificate is necessary to manage them effectively.

To address concerns about using wildcard certificates, Cloudera AI leverages the open-source service Certificate Manager. This approach enables you to use an automatic certificate signing service, known as 'issuer.' Cloudera AI then relies on the Certificate Manager service to request certificates from your managed automatic signing service, ensuring a more secure and streamlined process.

A custom revocation service is in place to automatically revoke certificates for terminated workloads.

Configuring cluster issuer for Certificate Manager

Certificate Manager is installed by default as part of the Cloudera Embedded Container Service installation.

About this task

To enable the usage of cert-manager in Cloudera AI, cluster issuers must be configured with the appropriate annotations.

Note: Cloudera currently supports only Venafi Trust Protection Platform (TPP) as the certificate issuer.

Before you begin

Procedure

1. Configure at least one cluster issuer with the type longlived.

   ```
   kubectl annotate clusterissuer <ISSUER_NAME> issuer.cdp.cloudera.com/typ
   e=longlived
   ```

2. Ensure that the cluster issuers are labelled for the CDP namespace for Cloudera Embedded Container Service.

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

3. Configure a different cluster issuer with the type shortlived as below:

   ```
   kubectl annotate clusterissuer <ISSUER_NAME> issuer.cdp.cloudera.com/typ
   e=shortlived
   ```

   Note:

   There can be only one cluster issuer configured with longlived or shortlived type for the CDP namespace.

4. Option: To create a certificate with a specific duration instead of using the default value, configure the duration annotation in the cluster issuer.

   ```
   kubectl annotate clusterissuer <ISSUER_NAME> issuer.cdp.cloudera.com/dur
   ation=”24h”
   ```

   When the duration is configured, the cert-manager.io/duration annotation is also configured in the ingress definition. This annotation is utilized by the cert-manager for certificate signing.

Results

Cloudera AI will prioritize using the shortlived issuer, if available, to sign certificates for temporary workloads such as jobs, sessions, and experiments. For Cloudera AI infrastructure endpoints and application workloads, the longlived issuer will be used. In cases where a shortlived issuer is not configured, the longlived issuer will handle certificate signing for all workloads and infrastructure endpoints.

Certified scale limitations for Cloudera AI Workbenches on Cloudera Embedded Container Service

The following scale limitations have been certified with Cloudera AI Workbenches on premises on Cloudera Embedded Container Service.

The Cloudera AI Workbench is certified to scale up to:

- 350 concurrent interactive users
- 1000 concurrent user workloads which include interactive sessions, scheduled jobs, long running applications and model deployments
- 200 cron jobs scheduled to start at the same time
- 500 concurrent applications
- 300 concurrently served model deployments

Cloudera recommends maintaining only one production Cloudera AI Workbench per Cloudera Embedded Container Service cluster. While deploying multiple Cloudera AI Workbenches on the same Cloudera Embedded Container Service cluster is supported, a single Workbench per cluster ensures optimal scalability and allows workloads to reach the specified scale limits effectively.

Getting Started with Cloudera AI on premises

To get started as a user with Cloudera AI on premises, follow the steps described below. The instructions guide you on how to set up a Project and work on data.

Before you begin

Make sure the Administrator creates a new workbench for you. If you are an Administrator, see: Provision a Cloudera AI Workbench.

Note: Make sure that an Administrator user logs into the workbench first.

Procedure

1. Log in to your workbench. On the workbenches tab, click Launch workbench.
2. Next, create a Project, see: Creating a Project.

3. Once you have a Project, run a session to start your work, see: Launch a Session.
4. Test your access to the base cluster (Data Lake), see: Cloudera-Cloudera Data Catalog cluster connectivity test.
5. You can then run a Model. Learn about Models here: Creating and Deploying a Model.
6. When you are finished with your workbench, your Administrator can remove it if necessary, as described here: Removing Cloudera AI Workbench.

Testing your connectivity to the Cloudera-Data Center Cluster

Test that you can create a Project in your Cloudera AI Workbench and access data that is stored in the data center cluster.

Procedure

1. Create a new Project, using the PySpark template.
2. Create a new file called testdata.txt (use this exact filename).
3. Add 2-3 lines of any text in the file to serve as sample data.
4. Run the following Spark commands to test the connection.

```
from pyspark.sql import SparkSession

# Instantiate Spark-on-K8s Cluster
spark = SparkSession\
.builder\
.appName("Simple Spark Test")\
.config("spark.executor.memory","8g")\
.config("spark.executor.cores","2")\
.config("spark.driver.memory","2g")\
.config("spark.executor.instances","2")\
.getOrCreate()

# Validate Spark Connectivity
spark.sql("SHOW databases").show()
spark.sql('create table testcml (abc integer)').show()
spark.sql('insert into table testcml  select t.* from (select 1) t').show(
)
spark.sql('select * from testcml').show()
# Stop Spark Session
spark.stop()
```

5. Run the following direct HDFS commands to test the connection.

```
# Run sample HDFS commands
# Requires an additional testdata.txt file to be created with sample data
 in project home dir
!hdfs dfs -mkdir /tmp/testcml/
!hdfs dfs -copyFromLocal /home/cdsw/testdata.txt /tmp/testcml/
!hdfs dfs -cat /tmp/testcml/testdata.txt
```

What to do next

If you get errors, then check with your Administrator to make sure that your user ID is set up in the Hadoop Authentication settings to access the Cloudera-DC cluster, and that the correct Ranger permissions have been applied.

Differences Between Cloudera on cloud and Cloudera on premises

There are some differences in Cloudera AI functionality between Cloudera on cloud and Cloudera on premises.

Feature	Cloudera on cloud	Cloudera on premises 1.5.x
Cloudera AI application control plane (infrastructure containers and workload containers)	Control plane is hosted on Cloudera on cloud servers.	Control plane is hosted on customer's cluster.
Storage - Cloudera AI internal state data (database, images, logs)	EBS on AWS, Azure Disks on Azure.	Software Defined Storage System, such as Ceph or Portworx.
Storage - User project files	EFS on AWS, external NFS on Azure.	Internal NFS storage is recommended.
Autoscaling	CPU/GPU nodes scale up and down as needed.	Autoscaling concept is different; Cloudera on premises shares a pooled set of resources among workloads.
Logging	Per-workbench diagnostic bundles can be downloaded from the workbench.	Diagnostic bundles are not supported at workbench level, but can be downloaded from the control plane at the cluster level.
Monitoring dashboards	Provides four dashboards.	Provides two dashboards, for K8s Container and K8s Cluster.
NFS support	AWS uses EFS; Azure requires external NFS.	Internal NFS is recommended, external NFS is supported.
TLS support	TLS access to workbenches is supported.	TLS access is supported, but requires manual setup of certificate and other steps.
Hadoop Authentication	Uses FreeIPA	User needs to provide credentials to communicate with the Cloudera Base on premises cluster.
Remote Access	Available from each workbench.	Not available in the workbench. Instead, the environment's kubeconfig file may be downloaded from Environments using the Download Kubernetes configuration action for the specified environment.
Roles	MLAdmin, MLUser	The corresponding roles are: EnvironmentAdmin, EnvironmentUser

Limitations on Cloudera on premises

There are some limitations to keep in mind when you are working with Cloudera AI on premises.

The following features are not yet supported in Cloudera AI on premises:

- Logging is limited, and diagnostic bundles for each workbench cannot be downloaded from the workbench UI. Instead, diagnostic bundles for the entire cluster can be downloaded from the control plane.
- Monitoring on Cloudera on premises does not support node-level resource metrics, hence only K8s Cluster and K8s Container dashboards are available.
- Cloudera AI does not support the NVIDIA Multi-Instance GPU (MIG) feature.

Network File System (NFS)

A Network File System (NFS) is a protocol to access storage on a network that emulates accessing storage in a local file system. Cloudera AI requires an NFS server for storing project files and folders, and the NFS export must be configured before you provision the first Cloudera AI Workbench in the cluster.

There are many different products or packages that can create an NFS in your private network. A Kubernetes cluster can host an internal NFS server, or an external NFS server can be installed on another cluster that is accessible by the on premises cluster nodes. NFS storage is used only for storing project files and folders, and not for any other Cloudera AI data, such as PostgreSQL database and livelog files.

Cloudera AI does not support shared volumes, such as Portworx shared volumes, for storing project files. A read-write-once (RWO) persistent volume must be allocated to the internal NFS server (for example, NFS server provisioner) as the persistence layer. The NFS server uses the volume to dynamically provision read-write-many (RWX) NFS volumes for the Cloudera AI clients.

An external NFS server option is currently the recommended option for Cloudera on premises production workloads. Not specifying an external NFS Server for your Cloudera AI Workbench will use/require a deprecated internal NFS provisioner, which should only be used for small, proof-of-concept deployments. There are several options for setting up an internal NFS provisioner, described in the appendix. The Cloudera on premises Administrator is responsible for setting up an NFS for use by your cluster.

Note: See *Cloudera on premises Installation Software Requirements* for some information about installing NFS.

Related Information

Cloudera Private Cloud Experiences Installation Software Requirements

NFS options for on premises

Cloudera AI on premises requires a Network File System (NFS) server for storing project files and folders.

On Cloudera Embedded Container Service, NFS is part of the overall installation, and no additional setup steps are required.

You can use an NFS server that is external to the cluster, such as a NetApp Filer appliance. In this case, you must manually create a directory for each workbench. Additionally, the NFS server must be configured before deploying the first Cloudera AI Workbench in the cluster. One important limitation is that Cloudera AI does not support using shared volumes for storing project files.

Note:

For Production environments, it is strongly recommended to setup an External NFS environment.

Storage provisioner change

On Openshift Container Platform, CephFS is used as the underlying storage provisioner for any new internal workbench on Cloudera on premises. A storage class named ocs-storagecluster-cephfs with CSI driver set to openshift-storage.cephfs.csi.ceph.com must exist in the cluster for new internal workbenches to get provisioned. Each workbench will have separate 1 TB internal storage.

On Cloudera Embedded Container Service, any new internal workbench will use Longhorn as the underlying storage provisioner. A storage class named longhorn with CSI driver set to driver.longhorn.io must exist in the cluster for new internal workbenches to get provisioned. Each workbench will have separate 1 TB internal storage.

Portworx storage provisioner

Cloudera AI supports Portworx as storage backend for workbenches.

About this task

New Cloudera AI Workbenches can be configured to use Portworx volumes for storing data.

Procedure

1. Create a Portworx storage class in your OpenShift Container Platform cluster manually. This storage class must have the provisioner field set to either `pxd.portworx.com` or `kubernetes.io/portworx-volume`.

   Note:

   Although `kubernetes.io/portworx-volume` is operational, this provisioner is deprecated in upstream Kubernetes and might be removed in the future (See: portworxVolumes).

   `pxd.portworx.com` is the preferred provisioner to use. Also the storage class created must have the `allow_all_ips: "true"` parameter (or have the `allow_ips` parameter set to the Kubernetes node in CIDR notation, such as `allow_ips: 10.17.0.0/16`).

   Sample storage classes are:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: sample-portworx-storage-class-k8s
parameters:
repl: "3"
allow_all_ips: "true"
provisioner: kubernetes.io/portworx-volume
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate

---


apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: sample-portworx-storage-class-pxd
parameters:
repl: "2"
sharedv4: "true"
sharedv4_svc_type: ClusterIP
allow_all_ips: "true"
provisioner: pxd.portworx.com
reclaimPolicy: Retain
allowVolumeExpansion: true
volumeBindingMode: Immediate

---

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: sample-portworx-storage-class-pxd
parameters:
repl: "2"
sharedv4: "true"
```

```
sharedv4_svc_type: ClusterIP
allow_ips: 10.17.0.0/16
provisioner: pxd.portworx.com
reclaimPolicy: Retain
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

2. In the Cloudera AI Workbench provisioning page, enable the Set Custom NFS Storage Class Name option. Enter the name of storage class you manually created earlier in the Storage Class Name text box.
3. Enter the rest of the workbench details and submit. This sets up Cloudera AI Workbench backed by Portworx.

Related Information

Using an External NFS Server

Internal Network File System on Cloudera Embedded Container Service

On Cloudera Embedded Container Service, NFS is part of the overall installation, and no additional setup steps are required.

The internal NFS does not have a backup feature.

Storage provisioner change

On Cloudera Embedded Container Service, Longhorn is used as the underlying storage provisioner for any new internal workbench on Cloudera 1.5.0. A storage class named longhorn with csi driver set to driver.longhorn.io must exist in the cluster for new internal workbenches to get provisioned. Each workbench will have separate 1 TB internal storage.

Internal workbenches running on Cloudera 1.4.0 and 1.4.1 use the NFS server provisioner as the storage provisioner. These workbenches when upgraded to 1.5.0 will continue to run with NFS Provisioner. However, NFS server provisioner is deprecated now and will not be supported in the 1.5.1 release. So, customers are expected to migrate their 1.5.0 upgraded workbench from NFS server provisioner to Longhorn if they want to continue using the same workbench in 1.5.1 as well. If not, then customers should create a new 1.5.0 workbench and migrate their existing workloads to that before the 1.5.1 release.

- **Note:** There is no change in the underlying storage of external NFS backed workbenches and these can be simply upgraded to 1.5.0.

Using an External NFS Server

You can install an NFS server that is external to the cluster.

About this task

Cloudera AI currently supports NFS versions 3.0, 4.0, 4.1, and 4.2. The NFS client within Cloudera AI must be able to mount the NFS storage with default options, and also assumes these export options:

```
rw,sync,no_root_squash,no_all_squash,no_subtree_check
```

Before you begin

Before creating a Cloudera AI Workbench, the storage administrator must create a directory that will be exported to the cluster for storing ML project files for that workbench. Either a dedicated NFS export path, or a subdirectory in an existing export must be specified for each workbench.

Each Cloudera AI Workbench needs a unique directory that does not have files in it from a different or previous workbench. For example, if 10 Cloudera AI Workbenches are expected, the storage administrator will need to create

10 unique directories. Either one NFS export and 10 subdirectories within it need to be created, or 10 unique exports need to be created.

For example, to use a dedicated NFS share for a workbench named workspace1 from NFS server nfs_server, do the following:

Procedure

1. Create NFS export directory /workspace1.
2. Change ownership for the exported directory
   a) Cloudera AI accesses this directory as a user with a UID and GID of 8536. Therefore, run chown 8536:8536 /workspace1
   b) Make the export directory group-writeable and set the GID:
   chmod g+srwx /workspace1
3. Provide the NFS export path nfs_server:/workspace1 when prompted by the Cloudera AI Control Plane App while creating the workbench.
4. To use a subdirectory in an existing NFS share, say nfs_server:/export, do the following:
   a) Create a subdirectory /export/workspace1
   b) Change ownership: chown 8536:8536 /export/workspace1
   c) Set GID and make directory group writeable: chmod g+srwx /export/workspace1
   d) Provide the export path nfs_server:/export/workspace1 when prompted by the Cloudera AI Control Plane App.

NFS share sizing

Cloudera AI workloads are sensitive to latency and IO/s instead of throughput.

The minimum recommended file share size is 100 GB. The file share must support online volume capacity expansion. It must provide at least the following performance characteristics:

IO / s	3100
Throughput rate	110.0 MiBytes/s

Deploying a Cloudera AI Workbench with support for TLS

You can provision a Cloudera AI Workbench with TLS enabled both on Cloudera Embedded Container Service and on OpenShift Container Platform (OCP), so that it can be accessed through https.

About this task

You must obtain a certificate from the Certificate Authority used by your organization. This might be an internal certificate authority. Additionally, you must have a computer with CLI access to the cluster, and with kubectl installed.

Note: Cloudera AI static subdomain is a custom name for the workbench endpoint, and it is also used for the URLs of models, applications, and experiments. Only one workbench with the specific subdomain endpoint name can be running at a time. You can create a wildcard certificate for this endpoint in advance.

The workbench subdomain is either the static subdomain the user elects or it can also be a workbench endpoint name that the deployment autogenerates. Also note that app_domain is defined at the Data Services deployment.

A workbench name has the following format: https://[***WORKBENCH-SUBDOMAIN***].apps.[***APP_DOMAIN***].com.

Workloads created in a Cloudera AI Workbench are containers provisioned in Kubernetes and must be addressable to the user. To do this, Cloudera AI creates a unique subdomain.

The URL for the workload is structured in the following format: https://[***WORKLOAD-ENDPOINTS***].[***WORKBENCH-SUBDOMAIN***].apps.[***APP_DOMAIN***].com.

As the workload endpoints are randomly generated, for TLS to work, a Cloudera AI Workbench must have a wildcard SAN entry in the TLS certificate and additionally the workbench subdomain SAN must also be provided.

The wildcard SAN entry has the following format: SAN:*.[***WORKBENCH-SUBDOMAIN***].apps.[***APP_DOMAIN***].com.

The workbench subdomain SAN has the following format: [***WORKBENCH-SUBDOMAIN***].apps.[***APP_DOMAIN***].com.

The following elements can be used for creating a Cloudera AI Workbench with static subdomain in Cloudera Embedded Container Service environment:

Table 2: Creating a Cloudera AI Workbench with static subdomain in Cloudera Embedded Container Service

Element	Example value	Origin
User domain	mycompany.com	User-provided
Non-HA deployment master hostname	ecsmst01	Inherited hostname
User control plane deployment	cdp-dev	User-provided
User load-balanced endpoint for the control plane deployment	cdp-lb	User-provided
Application subdomain	apps	Hardcoded
Cloudera AI Workbench ID	ml-1234abc-123	Auto-generated
Cloudera AI static subdomain	cmlstatic	User-provided

With the provided details, the following examples can be created:

Table 3: Cloudera AI Workbench environment examples

Network topology	Domain set	Example
Control Plane	High Availability (HA)	HA: app_domain = cdp-lb.mycompany.com
		HA applications: *.apps.cdp-lb.mycompany.com
	Non-HA, with custom deployment domain set	Non-HA with the custom Cloudera Embedded Container Service domain: app_domain = cdp-dev.mycompany.com
		Non-HA with the custom Cloudera Embedded Container Service applications: *.apps.cdp-dev.mycompany.com
	Non-HA, with no custom deployment domain set	Non-HA without custom Cloudera Embedded Container Service domain: app_domain = ecsmst01.mycompany.com
		Non-HA without custom Cloudera Embedded Container Service domain applications: *.apps.ecsmst01.mycompany.com*.

Network topology	Domain set	Example
Cloudera AI Workbench without static subdomain	High Availability (HA)	Cloudera AI Workbench on HA Cloudera Embedded Container Service without static subdomain: [*.]ml-1234abc-123.apps.cdp-lb.mycompany.com
	Non-HA with the user domain	Cloudera AI Workbench on non-HA Cloudera Embedded Container Service without custom Cloudera Embedded Container Service domain without Cloudera AI Workbench static subdomain: [*.]ml-1234abc-123.apps.cdp-dev.mycompany.com
	Non-HA without the user domain	Cloudera AI Workbench on non-HA Cloudera Embedded Container Service without custom Cloudera Embedded Container Service domain without Cloudera AI Workbench static subdomain: [*.]ml-1234abc-123.apps.ecsmst01.mycompany.com
Cloudera AI Workbench with static subdomain	High Availability (HA)	Cloudera AI Workbench on HA Cloudera Embedded Container Service with Cloudera AI Workbench static subdomain: [*.]cmlstatic.apps.cdp-lb.mycompany.com
	Non-HA with the user domain	Cloudera AI Workbench on non-HA Cloudera Embedded Container Service with custom Cloudera Embedded Container Service domain with Cloudera AI Workbench static domain: [*.]cmlstatic.apps.cdp-dev.mycompany.com
	Non-HA without the user domain	Cloudera AI Workbench on non-HA Cloudera Embedded Container Service without custom Cloudera Embedded Container Service domain with Cloudera AI Workbench static domain: [*.]cmlstatic.apps.ecsmst01.mycompany.com

By using unique subdomains, the Cloudera AI Workbench is able to securely serve each interactive workload with proper isolation and protect each workload from code injection attacks such as Cross Site Scripting.

Procedure

1. Provision the Cloudera AI Workbench.

   Follow the procedure in *Provisioning a Cloudera AI Workbench*.

   Note: Ensure you select the Enable TLS option.

2. Obtain the .crt and .key files for the certificate from your Certificate Authority.

   The certificate URL has the following format: [***WORKBENCH-SUBDOMAIN***].apps.[***APP_DOMAIN***].com

   Example

   This example shows an URL for the certificate: cml.apps.cdp.mycompany.com.

   Check that the certificate shows the corresponding Common Name (CN) and Subject Alternative Names (SAN) correctly. In this example, the following CN and SANs can be used:

   - CN: cml.apps.cdp.mycompany.com
   - SAN: *.cml.apps.cdp.mycompany.com
   - SAN: cml.apps.cdp.mycompany.com

   Note: If you want to install a new signed certificate, you must regenerate a new certificate from your Certificate Authority. Multi-level subdomains cannot be secured with a single wildcard certificate. If a wildcard certificate is issued for *.company.com, it can secure only the first-level subdomains of *.company.com. If you want to secure another subdomain, for example *.sub1.company.com, you must have another wildcard certificate for the*.sub1.company.com subdomain.

3. Create a Kubernetes secret inside the previously provisioned Cloudera AI Workbench namespace.

For Embedded Container Service

The certificate is automatically uploaded. Login to the Cloudera Embedded Container Service to run the following commands:

a. cd /opt/cloudera/parcels/ECS/bin/
b. ./cml_utils.sh -h

   Optional: A helper prompt appears, with explanation for the next command.
c. ./cml_utils.sh upload-cert -n [***NAMESPACE***] -c <path_to_cert> -k <path_to_key>

   For example: ./cml_utils.sh upload-cert -n bb-tls-1 -c /tmp/ws-cert.crt -k /tmp/ws-key.key

Note: To find the [***NAMESPACE***] of the workbench, go to the Cloudera AI Workbenches UI, and in the Actions menu for the workbench, select the View workbench Details option. Namespace is shown on the Details tab.

For OpenShift Container Platform

a. Name the secret cml-tls-secret.
b. Run this command on a machine with access to the .crt and .key files, and access to the cluster: kubectl create secret tls cml-tls-secret --cert=<pathtocrt.crt> --key=<pathtokey.key> -o yaml --dry-run | kubectl -n [***CML WORKBENCH-NAMESPACE***] create -f -

   You can replace or update certificates in the secret at any time.

4. Upload the root CA certificate to Site Administration Security Root CA configuration , unless it was already uploaded to Management Console Settings CA Certificates Miscellaneous before provisioning a Cloudera AI Workbench.

What to do next

The procedure creates routes to reflect the new state of ingress and secret, and enables TLS.

Related Information

Provisioning a Cloudera AI Workbench

CA certificate management in Cloudera AI Workbench

In enterprise clusters, internal endpoints, such as private Git repositories, Python package indexes, and internal services, are frequently secured with self-signed certificates or certificates issued by organization-specific Certificate Authorities (CAs).

CA Certificate Management in Cloudera AI Workbench enables secure, certificate-based trust for protected endpoints at the workbench level.

You can upload CA certificates directly through the Cloudera AI Workbench UI. You can also refresh certificates to apply and propagate trust across all relevant Cloudera AI Workbench services.

Managing CA certificates for Cloudera AI Workbench

Learn how to ensure that workbenches securely connect to internal resources protected by certificates.

Before you begin

Only Administrators are authorized to manage CA certificates in the Cloudera AI Workbench, including uploading or refreshing them.

Note: Cloudera AI Workbench has a certificate size limit of 1 MB, which applies to the combined size of the Cloudera Control Plane certificate and any certificates uploaded within the Cloudera AI Workbench.

Cloudera recommends generating a private root CA with the Basic Constraints extension set to CA:TRUE and creating endpoint certificates that include a valid and accurate Subject Alternative Name (SAN).

Procedure

1. Upload the certificates to the Cloudera AI Workbench.
   a) In the **Cloudera** console, click the Cloudera AI tile.

      The Cloudera AI Workbenches page displays.
   b) Select the required workbench.

      The Home page of the workbench is displayed.
   c) Select the View Workbench Details action from the Actions drop-down list.
   d) Scroll to the **CA Certificates** section on the Details page.
   e) Upload the certificate to the CA Certificate section and click the Upload button.
2. Refresh certificates.

   Use this action to activate the newly uploaded certificates and to refresh Workbench services with the updated certificate content. If you have already navigated to the required workbench, the Step 1 and Step 2 are optional.

   Note: The Refresh Certificate action results in temporary downtime for the workbench while the services are restarted.

   a) In the **Cloudera** console, click the Cloudera AI tile.

      The Cloudera AI Workbenches page displays.
   b) Select the required workbench.

      The Home page of the workbench is displayed.
   c) Select the Actions menu from the top-right navigation menu and select the Refresh Certificate action.

      This operation merges the latest certificates from the Platform truststore with the CA certificates specific for the Cloudera AI Workbench. It then restarts the Cloudera AI services to recognize the combined certificates and establish a secure connection with the CA certificate endpoints.

      Note: The certificates uploaded through the UI are securely stored within the Cloudera AI Workbench. However, access to the endpoints does not become active until the certificate refresh action is initiated.

      Note:

      Certificates added through the UI in Cloudera Administration CA Certificates are shared across all workbenches. However, to apply these certificates to each workbench you must use the Refresh Certificate action for each workbench.

Trusting self-signed CA certificates for private docker registries

To enable Cloudera AI workloads to pull images from a private docker registry, the cluster container runtime must be configured to trust the registry certificate.

For Embedded Container Service

Manually pulling a Docker image from a private registry is particularly useful when the registry requires authentication and uses a self-signed certificate. To manually pull a Docker image from a private registry using the ctr command, a client for containerd, perform the following steps:

1. Obtain the certificate and the credentials.
   a. Obtain a copy of the registry self-signed certificate and place it in a designated path on the Cloudera Embedded Container Service nodes. Ensure the certificate file is in .pem or .crt format.
   b. Retrieve the username and password for a user with pull access to the private registry.
2. Manually pull the image on each node within the Cloudera Embedded Container Service cluster.
   - Run the ctr command with the necessary flags to provide the certificate, credentials, and image details. Ensure this command is executed on each node where the image needs to be pulled.

```
/opt/cloudera/parcels/ECS/docker/ctr --namespace=k8s.io -a /run/k3s/
containerd/containerd.sock image pull --tlscacert <path/to/cert.pem>
 -u <username>:<password> <registry_url>/<image_name>:<tag>
```

For Opensift Container Service

OpenShift handles trusted certificates for private registries through the cluster-wide image configuration settings.

1. Create a ConfigMap in the openshift-config namespace to store the trusted certificates.
2. Update the cluster image.config.openshift.io resource to reference this ConfigMap.

For more details, see the official OpenShift CA documentation.

Replacing a Certificate

You can replace a certificate in a deployed namespace.

About this task

Procedure

1. Obtain the new certificate .crt and .key files.
2. Run this command (example): kubectl create secret tls cml-tls-secret --cert=<pathtocrt.crt> --key=<pathtokey .key> -o yaml --dry-run | kubectl -n <cml-workspace-namespace> replace -f -

   Note: Before replacing a certificate, it is recommended to save the existing certificate in case you need to revert to it later. Run this command to create a backup of the existing certificate: kubectl get secret c ml-tls-secret -n <workspace namespace> -o yaml > oldsecret.txt

   Note: Make sure that any intermediate certificates are concatenated to the server certificate file.

What to do next

The certificate of an existing session does not get renewed. The new certificate only applies to newly created sessions.

Related Information

How To Renew and Redistribute Certificates

Setting up the GPU node

In Kubernetes, you can taint nodes to affect how the node is scheduled. You can ensure that nodes that have a GPU are reserved exclusively for Cloudera AI workloads that require a GPU.

To reserve a GPU node, assign a taint to the node.

Assigning taint to the node on OpenShift

On OpenShift, specify the node taint nvidia.com/gpu: true:NoSchedule for any nodes that host GPUs and are required to be used only for GPU workloads.

Assigning taint to the node on Cloudera Embedded Container Service

On Cloudera Embedded Container Service, set the node taint nvidia.com/gpu: true:NoSchedule in one of the following three ways:

1. During Cloudera Embedded Container Service installation

   After adding the GPU host(s) to Cloudera Manager but prior to creation of the Cloudera Embedded Container Service cluster:

   a. Visit the Host Configuration page.
   b. Select the Dedicated GPU Node for Data Services checkbox and Save the configuration.
   c. Repeat for all hosts on which the taint is desired.
   d. Proceed with installation via the Add Cluster wizard.
2. During Cloudera Embedded Container Service upgrade

   After upgrading Cloudera Manager (if applicable):

   a. Set the host configuration as described in the first step above on one or more hosts in the Cloudera Embedded Container Service cluster.
   b. Proceed with upgrade via the Upgrade Cluster wizard.
3. Independently of Cloudera Embedded Container Service install or upgrade

   a. Set the host configuration as described above on one or more hosts in the Cloudera Embedded Container Service cluster.
   b. Redeploy the client configuration on the Cloudera Embedded Container Service cluster.
   c. Complete a Rolling restart of the Cloudera Embedded Container Service cluster. Restart the Cloudera Embedded Container Service by selecting Cloudera Manager Clusters [***CLUSTER NAME***] Actions .