Cloudera Data Science Workbench

Managing Cloudera Data Science Workbench

Date published: 2020-02-28
Date modified:

CLOUDERA

Legal Notice

Contents

Cluster Management

This section describes some common tasks and guidelines related to cluster management for Cloudera Data Science Workbench.

For a basic overview of how Cloudera Data Science Workbench fits onto a Cloudera Manager and CDH cluster, refer to Architecture Overview.

Cluster Monitoring with Grafana

Cloudera Data Science Workbench leverages Prometheus and Grafana to provide a dashboard that allows you to monitor how CPU, memory, storage, and other resources are being consumed by your CDSW deployment. Prometheus is an internal data source that is auto-populated with resource consumption data for each deployment. Grafana is the monitoring dashboard that allows you to create visualizations for resource consumption data from Prometheus.

By default, CDSW provides you with three Grafana dashboards: K8 Cluster, K8s Containers, and K8s Node. You can extend these dashboards or create more panels for other metrics. For more information, see the Grafana documentation.

Accessing the Grafana Dashboard

To access the Grafana dashboard for your deployment.

Procedure

1. Log into Cloudera Data Science Workbench with site administrator privileges.
2. Click Admin Overview .
3. Click the Grafana dashboard link. This will take you to the built-in Grafana server.
4. To see all the available dashboards, click Home K8 Cluster (or K8s Containers or K8s Node).
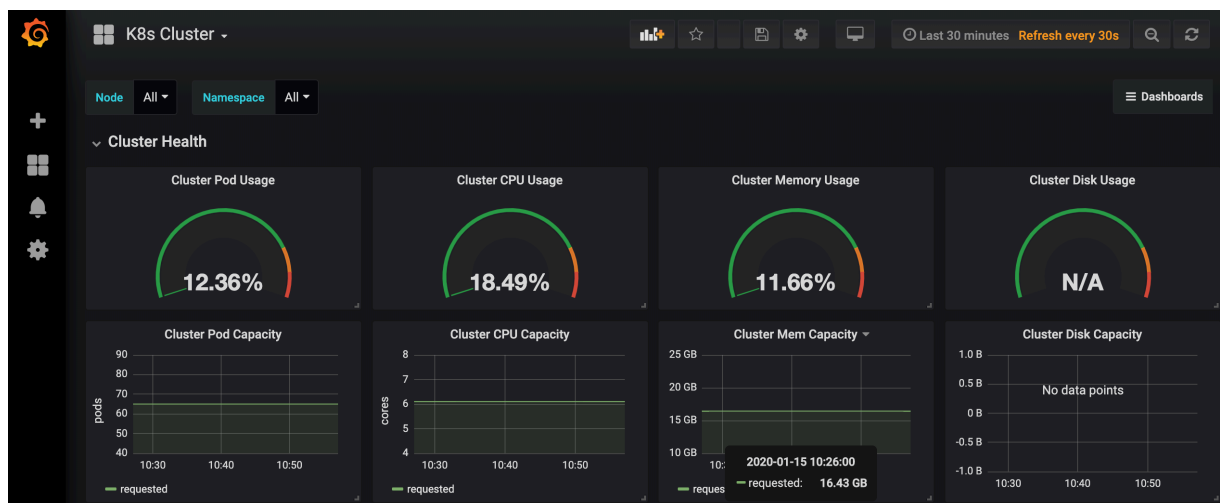
Available Grafana Dashboards

This topic lists all of the available Grafana dashboards.
K8s Cluster

Provides metrics for the following:

- Overview of nodes, pods, and containers
- CPU capacity usage
- Memory capacity usage
- Pod capacity usage
- Disk capacity usage

K8s Containers

Provides metrics for the following:

- Memory usage per pod
- CPU Usage per pod
- Read/Write IOPS per pod

K8s Node

Provides metrics for the following:

- CPU usage per node
- Memory Usage per node
- Read/Write IOPS per node
- Available memory per node
- Network traffic per node

Backup and Disaster Recovery for Cloudera Data Science Workbench

All application data for Cloudera Data Science Workbench, including project files and database state, is stored on the master host at /var/lib/cdsw.

Given typical access patterns, it is strongly recommended that /var/lib/cdsw be stored on a dedicated SSD block device or SSD RAID configuration. Because application data is not replicated to HDFS or backed up by default, site administrators must enable a backup strategy to meet any disaster recovery scenarios.

Cloudera strongly recommends both regular backups and backups before upgrades and is not responsible for any data loss.

Creating a Backup for Disaster Recovery for Cloudera Data Science Workbench

Cloudera strongly recommends both regular backups and backups before upgrades and is not responsible for any data loss.

Procedure

1. Stop Cloudera Data Science Workbench.

    Use one of the following steps depending on your Cloudera Data Science workbench version.

    Cloudera Data Science Workbench 1.4.2 or lower

    > Do not stop or restart Cloudera Data Science Workbench without using the cdsw_protect_stop_restart.sh script. This is to help avoid the data loss issue detailed in TSB-346 .

    > Run the script on your master host and stop Cloudera Data Science Workbench (instructions below) only when instructed to do so by the script. Then proceed with step 2 of this process.

    Cloudera Data Science Workbench 1.4.3 or higher

    > Depending on your deployment, use one of the following sets of instructions to stop the application.

    - CSD - Log in to Cloudera Manager. On the Home Status tab, click [dropdown] to the right of the CDSW service and select Stop from the dropdown. Wait for the action to complete.

      OR
    - RPM - Run the following command on the master host:

      ```
      cdsw stop
      ```

2. After stopping CDSW, and before running the following tar command, wait 2-5 minutes (depending on your disk speed) to ensure that all data from CDSW is successfully written to the disks. Otherwise the tar command may not capture all recent changes.

3. To create the backup, run the following command on the master host:

    ```
    tar cvzf cdsw.tar.gz /var/lib/cdsw/*
    ```

    > **Note:** The /var/lib/cdsw directory contains all the persistent information as well as project-related information, such as database information, configurations, image details, etc. Using tar to create the backup preserves important file metadata such as file ownership, and is the recommended method to migrate the information to the new host. Other methods of copying/saving files might not preserve this information. This metadata is required for tasks such as migrating CDSW to another cluster.

4. (Optional) If needed, the following command can be used to unpack the tar bundle.

    ```
    tar xvzf cdsw.tar.gz -C /var/lib/cdsw
    ```

Cloudera Data Science Workbench Scaling Guidelines

New hosts can be added and removed from a Cloudera Data Science Workbench deployment without interrupting any jobs already scheduled on existing hosts. Therefore, it is rather straightforward to increase capacity based on observed usage.

At a minimum, Cloudera recommends you allocate at least 1 CPU core and 2 GB of RAM per concurrent session or job. CPU can burst above a 1 CPU core share when spare resources are available. Therefore, a 1 CPU core allocation is often adequate for light workloads. Allocating less than 2 GB of RAM can lead to out-of-memory errors for many applications.

As a general guideline, Cloudera recommends hosts with RAM between 60GB and 256GB, and between 16 and 48 cores. This provides a useful range of options for end users. SSDs are strongly recommended for application data storage.

Note: At present, Kubernetes does not automatically pick the CPU/RAM that is dynamically added to the virtual machines (VM) while the VM is on. Therefore, you need to restart the CDSW service/role on the node on which you have updated the capacity.

For some data science and machine learning applications, users can collect a significant amount of data in memory within a single R or Python process, or use a significant amount of CPU resources that cannot be easily distributed into the CDH cluster. If individual users frequently run larger workloads or run workloads in parallel over long durations, increase the total resources accordingly. Understanding your users' concurrent workload requirements or observing actual usage is the best approach to scaling Cloudera Data Science Workbench.

Ports Used By Cloudera Data Science Workbench

Cloudera Data Science Workbench runs on gateway hosts in a CDH/HDP cluster. As such, Cloudera Data Science Workbench acts as a gateway and requires full connectivity to cluster services such as Impala, Spark 2, etc. Additionally, in the case of Spark 2, cluster hosts will require access to the Spark driver running on a set of random ports (20050-32767) on Cloudera Data Science Workbench hosts.

Firewall restrictions must be disabled across Cloudera Data Science Workbench and CDH/HDP cluster hosts. Internally, the Cloudera Data Science Workbench master and worker hosts require full connectivity with no firewalls. Externally, end users connect to Cloudera Data Science Workbench exclusively through a web server running on the master host, and therefore do not need direct access to any other internal Cloudera Data Science Workbench or CDH services.

This information has been summarized in the following table.

Components	Details
Communication with the CDH / HDP cluster	CDH / HDP -> Cloudera Data Science Workbench The CDH/HDP cluster must have access to the Spark driver that runs on Cloudera Data Science Workbench hosts, on a set of randomized ports in the range, 20050-32767. You can customize this range to restrict the number of reserved ports. See Configuring spark-port-forwarder ports.
	Cloudera Data Science Workbench -> CDH / HDP As a gateway service, Cloudera Data Science Workbench must have access to all the ports used by CDH and Cloudera Manager.
Communication with the Web Browser	The Cloudera Data Science Workbench web application is available at port 80. HTTPS access is available over port 443.

Note: Prometheus captures data for the previous 30 days.

Configuring spark-port-forwarder ports

The CDH/HDP cluster must have access to the Spark driver that runs on Cloudera Data Science Workbench hosts, on a set of randomized ports in the range, 20050-32767. You can customize this range to restrict the number of reserved ports.

Procedure

1. Log in to the Cloudera Manager Admin Console.

2. On the Home > Status tab, click CDSW.

3. Click the Configuration tab.

4. Scroll to the Minimum port configured for the spark-port-forwarder field and enter the minimum port number you want to reserve.

Minimum port configured for the spark-port-forwarder	CDSW-1 (Service-Wide)
SPARK_PORT_FORWARDER_MINIMUM_PORT ⚙ cdsw.sparkportforwarder.minimum.port	20050

Maximum port configured for the spark-port-forwarder	CDSW-1 (Service-Wide)
SPARK_PORT_FORWARDER_MAXIMUM_PORT ⚙ cdsw.sparkportforwarder.maximum.port	32767

The default minimum port numer is 20050. You cannot specify a port lower than this number.

5. In the Maximum port configured for the spark-port-forwarder field, enter the maximum port number you want to reserve.

The maximum port number cannot be lower than the minimum port number. You must reserve a minimum of 100 ports for use. The maximum port number allowed is 32767.

Monitoring/Reducing CDSW disk space

If the /var/lib/cdsw disk on the CDSW master node fills up to 80% or more, then the CDSW system starts to act very poorly, with symptoms such as unable to start sessions, Kerberos tickets not working, unable to create projects, and pods just crashing. There are a number of maintenance chores that Admin teams should perform monthly on CDSW to clean up the disk storage space

About this task

Possible symptoms that CDSW storage is filling up include but are not limited to:

- unable to start sessions
- sessions randomly get killed with "Engine exited with status 1"
- unable to build models
- unable to "kinit" through Hadoop Authentication
- the UI is slow

Any of these symptoms "Go away" temporarily when CDSW is restarted. This is a major point that the problem is with the storage space, since restarting CDSW will free up a lot of temporary space.

Remove Deleted Orphan Projects

When projects are deleted from the CDSW UI, the files are left on the disk. These projects cannot be recovered in CDSW UI, since they have been removed from the database. Therefore, since nobody can view these files, they are completely safe to delete. The following command will show these files that can be safely deleted:

```
{{
for project_id in `ls /var/lib/cdsw/current/projects/projects/*/`
do
 echo "Processing $project_id"
 rows=`kubectl exec $(kubectl get pods -l role=db --no-headers=true -o cu
stom-columns=NAME:.metadata.name) -ti -- psql -U sense -P pager=off -c "SELE
CT * FROM projects WHERE ID = $project_id" | grep '0 row' | wc -l`
 if [ $rows -gt 0 ]
 then
```

```
  echo -n "Project with ID $project_id has been deleted, you can archive its
  directory: " `ls -d /var/lib/cdsw/current/projects/projects/*/$project_id`;
echo
  fi
done
}}
```

This will print out all of the projects that can safely be removed from the disk. These projects have already been deleted from the UI, so there is no risk that collaborators cannot access the files, etc.

Note: The project folders are organized like /projects/projects/0/1, etc. The 0 here actually stands for "the first thousand." This means if a system has more than a thousand projects, this command might not be complete. If you have more than one thousand projects in the file system, you may have to update the above script.

After you run this command and clean up these orphaned projects, check the disk usage again to see if you are back below 70%.

Identify Huge Projects on Disk

If you still need to clean up some disk space, the next best solution is to look through the file system and find huge projects on disk. Projects can grow huge for a number of reasons. It is important to understand why so that the Admin team can educate the end users to prevent this from happening in the future. The biggest reasons for the project growing to crazy sizes are:

*   Users are doing ML training on huge training data sets, and instead of using the data in HDFS, they pull massive training sets into their projects.

    Resolution: Pull data from Hadoop and run your training, then delete this data. Or simply train the algorithms on the CDH cluster via spark or some other method.
*   Users have created ML jobs that run on a scheduled basis and these generate output artifacts. After enough time, this can grow to substantial sizes.

    Resolution: Add another job to delete this old data.

You can use the following script to list the projects in the system, along with the username, project name, email, and size. This will spit things out in a CSV format, and can be thrown into Excel and used, for example, to email the top ten users and ask them if it is possible to clean up their project sizes.

```
echo "Project Path, Size, Project Name, User name, email"
for project_id in `du -hx --max-depth=2 /var/lib/cdsw/current/projects/pro
jects/ | sort -hr | awk 'NR>2 {print $2}'`
do
 echo -n $project_id ","
 fileNameWithFullPath="${project_id%/}";
 pid="${fileNameWithFullPath##*/}"
 rows=`kubectl exec $(kubectl get pods -l role=db --no-headers=true -o custo
m-columns=NAME:.metadata.name) -ti -- psql -P pager=off -U sense -c "SELECT
* FROM projects WHERE ID = $pid" | grep '0 row' | wc -l`
 if [ $rows -gt 0 ]
 then
 echo -n "Project with ID $pid has been deleted, you can archive its dire
ctory: " `ls -d /var/lib/cdsw/current/projects/projects/*/$pid`;echo
 else
 size=`du -sh /var/lib/cdsw/current/projects/projects/0/$pid | awk '{print
$1}'`
 echo -n "$size,"
 user_data=`kubectl exec $(kubectl get pods -l role=db --no-headers=true -o
 custom-columns=NAME:.metadata.name) -ti -- psql -U sense -t -A -c "select
p.name,u.name, u.email from projects p, users u where p.user_id=u.id and p.i
d=$pid;" | tr '|' ','`;
 echo "$user_data"
 fi
```

```
done
```

You should put this inside a file, chmod to 777, and run it like get-large-projects.sh > large-projects.csv

This can make a significant difference. In practice, the top ten projects can take up 30-40% of the free space.

Increasing Docker storage

Cloudera requires a dedicated disk for docker (the "docker block device") on every single CDSW node. This this disk should be of size 1TB. You can increase the Docker storage space for CDSW.

About this task
If the Docker storage on any of the CDSW nodes reaches 75% capacity or more, you may see issues with pods being evicted from those nodes with error messages including but not limited to "The node had condition: [DiskPressure]." You can view the free space with the docker info command.

Note: This process will remove all existing docker images. This means that custom ML Runtimes and custom Legacy Engines will need to be loaded back onto the CDSW hosts. It also means that any Models which are in Stopped status may need to be rebuilt.

To increase the Docker storage space, complete the following steps:

About this task

Procedure

1. Remove existing docker devices
   a) On all nodes that will get new disks, run the file located at /opt/cloudera/parcels/CDSW/scripts/teardown-docker.sh This will remove all of the exiting docker devices on the host that it is ran on.

2. Option 1 - Add the same disk on every worker node

a) Add a new, empty disk to all CDSW worker nodes.

This disk does not need to be 1TB. Each node should have 1TB of docker space but that can be split across multiple disks.

b) Ensure the new disks are added to the same location on all CDSW worker nodes, such as: /dev/sdc.

c) Navigate to Cloudera Manager > CDSW > Configuration > then search for Docker block device and add the new location:

This location must be the same on every worker node and must have a disk. If this does not happen, one or more of the worker nodes will fail to start.

d) Save the new setting, then stop CDSW.

e) Run Prepare Node on all nodes, then restart CDSW from Cloudera Manager. Copy any existing Legacy Engine docker images to all nodes.

3. Option 2 - Add different disks to different worker nodes

Using Role Groups (see Role Groups) in Cloudera Manager, you can add disks to worker nodes in different locations.

a) Navigate to Cloudera Manager > CDSW and add a new Role Group.

For instance, if you have a set of worker nodes that will use /dev/sdc for the new disk, and another set of worker nodes that will use /dev/vdb, you can create two role groups named "sdc" and "vdb." Ensure that these role groups are assigned to the right set of specific nodes.

b) Edit the Docker Block device per the above procedure but you can have different disks for different nodes. Example:

Docker Block Device

DOCKER_BLOCK_DEVICES

Edit Identical Values

⚙ cdsw.docker.devices.config

Docker Daemon Default Gro

/dev/vdb

worker3 ↺ Undo

/dev/vdb

/dev/sde

workers1-2 ↺ Undo

/dev/vdb

/dev/sdc

In this example, every worker nodes has a disk in /dev/vdb. The worker nodes that are using the "worker3" role group have an additional disk added at /dev/sde, while the nodes using "workers1-2" role group have the disk at /dev/sdc. Any nodes using the default role group ("Docker daemon default group") have a disk only in /dev/vdb.

c) Save the new setting, then stop CDSW.

d) Run Prepare Node on all nodes, then restart CDSW from Cloudera Manager. Copy any existing Legacy Engine docker images to all nodes.

Rollback Cloudera Data Science Workbench to an Older Version

All stateful data for Cloudera Data Science Workbench is stored in the /var/lib/cdsw directory on the Master host. The contents of this directory are forward compatible, which is what allows for upgrades. However, they are not backward compatible. Therefore, to rollback Cloudera Data Science Workbench to a previous version, you must have a backup of the /var/lib/cdsw directory, taken prior to the last upgrade.

About this task
In general, the steps required to restore a previous version of Cloudera Data Science Workbench are:

Procedure
1. Depending on your deployment, either uninstall the RPM or deactivate the current CDSW parcel in Cloudera Manager.
2. On the master host, restore the backup copy you have of /var/lib/cdsw. Note that any changes made after this backup will be lost.
3. Install a version of Cloudera Data Science Workbench that is equal to or greater than the version of the /var/lib/cdsw backup.

Uninstalling CPM Deployments

This topic describes how to uninstall Cloudera Data Science Workbench from a cluster.

About this task

Procedure
1. Log in to the Cloudera Manager Admin Console.
2.
   On the Home > Status tab, click [▼] to the right of the CDSW service and select Stop from the dropdown and confirm that you want to stop the service.

   > **Note:** On Cloudera Data Science Workbench 1.4.0 (and lower), do not stop or restart Cloudera Data Science Workbench without using the cdsw_protect_stop_restart.sh script. This is to help avoid the data loss issue detailed in TSB-346.

3. (Strongly Recommended) On the master host, backup the contents of the /var/lib/cdsw directory. This is the directory that stores all your application data.

   To create the backup, run the following command on the master host:

   ```
   tar cvzf cdsw.tar.gz /var/lib/cdsw/*
   ```

4. Go back to Cloudera Manager and click Hosts Parcels .
5. Go to the CDSW parcel and click Deactivate.
6. Select Deactivate Only from the list.
7.
   Click the [▼] to the right of the Activate button and select Remove From Hosts.
8. Click OK to confirm.

9.
Go back to the Home Status page and click ▼ to the right of the CDSW service. Select Delete from the dropdown and confirm that you want to delete the service.

10. Remove all your user data that is stored in /var/lib/cdsw on the master host from the deployment. This step will permanently remove all user data.

```
sudo rm -Rf /var/lib/cdsw
```

For more details on how to uninstall Cloudera Manager and it's components, see Uninstalling Cloudera Manager and Managed Software.

Uninstalling RPM Deployments

This topic describes how to uninstall Cloudera Data Science Workbench from a cluster.

RPM Deployments

Procedure

1. Run the following command on the master host to stop Cloudera Data Science Workbench.

```
cdsw stop
```

> **Note:** On Cloudera Data Science Workbench 1.4.0 (and lower), do not stop or restart Cloudera Data Science Workbench without using the cdsw_protect_stop_restart.sh script. This is to help avoid the data loss issue detailed in TSB-346.

2. (Strongly Recommended) On the master host, backup the contents of the /var/lib/cdsw directory. This is the directory that stores all your application data.

To create the backup, run the following command on the master host:

```
tar cvzf cdsw.tar.gz /var/lib/cdsw/*
```

3. To uninstall Cloudera Data Science Workbench, run the following commands on the master host and all the worker hosts.

```
cdsw stop
yum remove cloudera-data-science-workbench
```

4. Remove all your user data that is stored in /var/lib/cdsw on the master host from the deployment. This step will permanently remove all user data.

```
sudo rm -Rf /var/lib/cdsw
```