Cloudera Data Engineering 1.20.3

Managing virtual clusters

Date published: 2020-07-30
Date modified: 2024-02-26

CLOUDERA

Legal Notice

Contents

Creating virtual clusters

In Cloudera Data Engineering (CDE), a virtual cluster is an individual auto-scaling cluster with defined CPU and memory ranges. Jobs are associated with virtual clusters, and virtual clusters are associated with an environment. You can create as many virtual clusters as you need. See Recommendations for scaling CDE ceployments linked below.

Before you begin

⚠️ **Important:** The user interface for CDE 1.17 and above has been updated. The left-hand menu was updated to provide easy access to commonly used pages. The steps below will vary slightly CDE version prior to 1.17, for example, the Overview page has been replaced with the Home page. The new Home page still displays Virtual Clusters, but now includes quick-access links located at the top for the following categories: Jobs, Resources, and Download & Docs.

📝 **Note:** Virtual Cluster installation fails when Ozone S3 gateway proxy is enabled. Ozone S3 gateway proxy gets enabled when more than one Ozone S3 Gateway is configured in the CDP Private Cloud Base cluster.

As a workaround, add the `127.0.0.1 s3proxy-<environment-name>.<private-cloud-control-plane-name>-services.svc.cluster.local` entry in the /etc/hosts file of all nodes in the CDP Private Cloud Base cluster where the Ozone S3 gateway is installed. For example, if the private cloud environment name is `cdp-env-1` and private cloud control plane name is `cdp`, then add the following entry in the `/etc/hosts` file:

```
127.0.0.1 s3proxy-cdp-env-1.cdp-services.svc.cluster.local
```

To create a virtual cluster, you must have an environment with Cloudera Data Engineering (CDE) enabled.

Procedure

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
2. Click Administration in the left navigation menu, select the environment you want to create a virtual cluster in.

3.

In the Virtual Clusters column, click ⊕ at the top right to create a new virtual cluster.

If the environment has no virtual clusters associated with it, the page displays a Create DE Cluster button that launches the same wizard.

Cluster Details

Cluster Name *

GenericVCName

Service*

g0-awsservice-name001   ▼

What do you intend to use this cluster for? ⓘ

⦿ Core (Tier 1)

Operational deployment & monitoring of batch based transformations

○ All Purpose (Tier 2)   NEW

Develop using interactive sessions and deploy both batch & streaming (coming soon) workloads.

Unsure which to select? Review our feature comparison or pricing breakdown ☒

Select Spark Version

Spark 2.4.8   ▼

Auto-Scale Max Capacity

CPU

200   800   200

Memory (GB)

80   3200   80

Additional Options

Enable Iceberg tables, Restrict user access

4. Enter a Cluster Name.

Cluster names must include the following:

- Begin with a letter
- Be between 3 and 30 characters (inclusive)
- Contain only alphanumeric characters and hyphens

5. Select the Service to create the virtual cluster in.

The environment you selected before launching the wizard is selected by default, but you can use the wizard to create a virtual cluster in a different CDE service.

6. Select one of the following CDE cluster types:

> **Note:** These tiers are available in CDE 1.19 and above only.

- Core (Tier 1): Batch-based transformation and engineering options include:
  - Autoscaling Cluster
  - Spot Instances
  - SDX/Lakehouse
  - Job Lifecycle
  - Monitoring
  - Workflow Orchestration
- All-Purpose (Tier 2) - Develop using interactive sessions and deploy both batch and streaming workloads. This option includes all options in Tier 1 with the following:
  - Shell Sessions - CLI and Web
  - JDBC/SparkSQL (Coming soon)
  - IDE (Coming Soon)

7. Select the Spark Version to use in the virtual cluster. You cannot use Spark 2 and Spark 3 in the same virtual cluster, but you can have separate Spark 2 and Spark 3 virtual clusters within the same CDE service. You can use multiple Spark 3.x in your virtual cluster.

8. Use the Auto-Scale Max Capacity sliders to set the maximum number of CPU cores and the maximum memory in gigabytes. The cluster will scale up and down as needed to run the submitted Spark applications.

9. Optional for spot instances enabled at the CDE service level: From the Driver and Executors will run on drop-down menu, select whether you want to run drivers and executors on spot instances or on-demand instances. By default, the driver runs on on-demand instances, and the executors run on spot instances. For SLA-bound workloads, select On-demand. For non-SLA workloads, Cloudera recommends leaving the default configuration to take advantage of the cost savings afforded by spot instances. For more information, see Cloudera Data Engineering Spot Instances.

10. Click Enable Remote Shuffle Service (Technical Preview) if you want to store Spark shuffle data on remote servers. Usage of Remote Shuffle Service (RSS) improves resilience in the case of executor loss and it allows to run jobs with regular Dynamic Allocation (without shuffle tracking). It may also decrease the job execution time depending on the number of RSS instances.

> **Note:** Remote Shuffle Service feature is supported only on Amazon Web Services.

a) Override Instances: Click to override the recommended RSS instances for this virtual cluster.

The recommended number of RSS instances: the recommendation on a number of RSS instances for a virtual cluster is according to the defined virtual cluster CPU quota. This number is obtained according to the RSS performance testing results. Increasing the number of RSS servers to some point could help decrease job execution time, but will also have cost implications.

b) Instances: Drag the slider button to specify the number of RSS instances.

Each RSS instance runs on i3.xlarge AWS EC2 instance, so each instance will contribute i3.xlarge cost to the total CDE cost . Overall shuffle data volume that can be stored by a virtual cluster at the same time is number of RSS instances * 880 GiB (gibibytes).

11. Optional: Select Enable Iceberg analytic tables to enable Spark jobs running within the virtual cluster to create and access Apache Iceberg tables.

12. Select Restrict Access to add access control for the virtual cluster. You can search for users to add by name or email address. You can manage users using the Cloudera Data Platform Management Console. For more information, see Managing user access and authorization.

13. Click Configure Email Alerting (Technical Preview) if you want to receive notification mails.The email configuration options appear.

  a) You must provide at least Sender Email Address and SMTP Host information.

  b) Test SMTP Configs: Click Test SMTP Configs to test the configurations set for SMTP. This helps you to test the SMTP configuration before creating the cluster.

> **Note:** You will have the option to enable email alerts when you create or edit a job after the Virtual Cluster is created, but step 10 must be completed first.

14. Click Create.

Results

On the CDE Home page, select the environment to view the virtual cluster initialization status. You can also click the three-dot menu for the virtual cluster to view the logs.

Related Information

Recommendations for scaling CDE deployments

Viewing and managing virtual cluster details

You can view details such as configuration, metrics, and logs for existing virtual clusters. You can also view jobs associated with a cluster, clone a cluster, and delete a cluster

> **Important:** The user interface for CDE 1.17 and above has been updated. The left-hand menu was updated to provide easy access to commonly used pages. The steps below will vary slightly, for example, the Overview page has been replaced with the Home page. The new home page still displays Virtual Clusters, but now includes quick-access links located at the top for the following categories: Jobs, Resources, and Download & Docs.

To view cluster details:

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.

2. Click Administration in the left navigation menu. The Administration page displays.

3. In the Services column, select the environment containing the virtual cluster you want to manage.

4. In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to manage.

5. On the Configuration tab, you can view details about the cluster, including the cluster name, CDP environment, and the CPU and memory auto-scale ranges.

The virtual cluster Overview page provides details about the virtual cluster, and includes links to several resources, such as the API documentation, and the Jobs API URL. You can switch between the following three tabs to view additional information:

Configuration

The Configuration tab lists details about the cluster, including the cluster name, CDP environment, and the CPU and memory auto-scale ranges. You can modify the virtual cluster quotas (CPU/ memory) dynamically. CDE provides an Edit option to make the configuration changes dynamically which may take a few mintutes to update. For more information, see Editing virtual clusters linked below.

Charts

The Charts tab displays charts of virtual cluster metrics, such as network and filesystem usage.

> **Note:** If your browser blocks third-party cookies, Charts may not display correctly. If this happens, add *.cloudera.com as an exception in your browser settings.

Security

The Security tab displays the Access Management that you can set for the virtual cluster. See Applying user and group access for virtual clustersfor more details.

Logs

The Logs tab displays the latest log entries for the virtual cluster.

Related Information

Editing virtual clusters

Applying user and group access for virtual clusters

Learn how to restrict or grant access to a virtual cluster for specific users or groups that you specify in Cloudera Data Engineering (CDE).

Procedure

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
2. Click Administration in the left navigation menu. The Administration page displays.
3. In the Services column, select the environment containing the virtual cluster you want to manage.
4. In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to update security for.

5. Click the Security Access Add User or Group .

The Add User(s) or Group(s) screen is displayed.

6. In the Search for User(s) or Group(s) field, enter the name or email of the user(s) or group name(s).

7. Under Set Role(s) drop-down, select Read-only Admin or Admin.

The Admin option can access endpoints including all jobs, resources, and sessions. The Read-only Admin option can only view all cluster activities but cannot modifiy the cluster or activities.

8. Click Add.

The Access Management screen is displayed with the users and groups that you added. Only one role can be set each time, but selection can be modified by editing once the user or groups are added.

9. To add more users or groups, click Add.

The Add User(s) or Group(s) screen is displayed.

a) Repeat steps 6 through 8.

Editing user and group access for virtual clusters

Learn how to edit or delete a user or group access that you have added for a virtual cluster in Cloudera Data Engineering (CDE).

Procedure

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
2. Click Administration in the left navigation menu. The Administration page displays.
3. In the Services column, select the environment containing the virtual cluster you want to manage.
4. In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to update security for.
5. Click the Security Access .
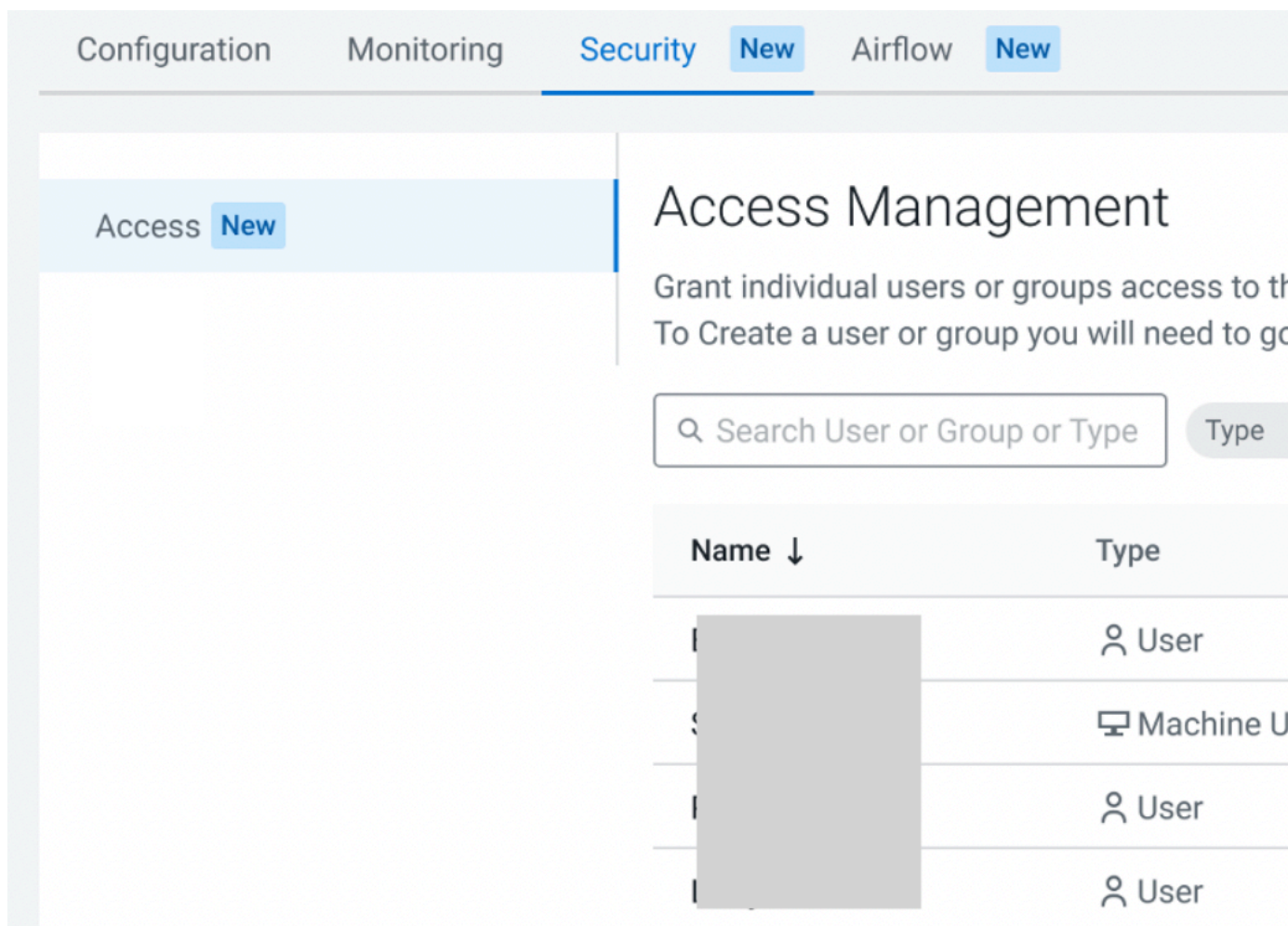
6. Locate the name of the user or group that you want to edit.
7. Click the Pencil icon to edit the current user or group.

   The Add User(s) or Group(s) screen is displayed.
8. If you want to update the user or group, in the Search for User(s) or Group(s) field, enter the name or email of the user(s) or group name(s).
9. If you want to edit the role, under Set Role(s) drop-down, select Read-only Admin or Admin.

   The Admin option can access endpoints including all jobs, resources, and sessions. The Read-only Admin option can only view all cluster activities but cannot modifiy the cluster or activities.

10. If you want to delete the user or group, locate the name or group of the user that you want to delete, and click the Trash icon.

Editing virtual clusters

You can edit virtual cluster configurations in the UI or use a script to apply configurations to multiple jobs at once.

⚠️ **Important:** The user interface for CDE 1.17 and above has been updated. The left-hand menu was updated to provide easy access to commonly used pages. The steps below will vary slightly, for example, the Overview page has been replaced with the Home page. The new home page still displays Virtual Clusters, but now includes quick-access links located at the top for the following categories: Jobs, Resources, and Download & Docs.

For Edit in UI

To edit virtual cluster details:

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
2. Click Administration in the left navigation menu. The Administration page displays.
3. In the Services column, select the environment containing the virtual cluster you want to manage.
4. In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to manage.
5. On the Configuration tab, you can modify the virtual cluster quotas (CPU/memory) dynamically. CDE provides an Edit option to make the configuration changes dynamically which may take a few minutes to update.

For Edit with Script

As an Administrator, you can further customize the virtual cluster by applying additional configurations all at once using the cde-utils script. Click cdp-cde-utils.sh to download the script. Customizations include applying spark configurations to all jobs, enabling and disabling gang schedueling, and initiating faster autoscaling times for nodes in an autoscaler cluster for Amazon Web Services (AWS). See the following cde-utils script and options to apply these changes:

⚠️ **Important:** Before using the script, the user must export the kubeconfig file to ensure that the virtual cluster is accessible via kubectl.

Apply Spark configurations to all new jobs

This script allows users to add and update default configurations for all Spark jobs all at once. The options required are as follows:

- -h <host_name>: You must provide this option, and paste the domain of the URL into a text editor to identify the endpoint host. For example, if the URL is https://xyz.cde-vjgd8ksl.dsp-azur.xcu2-8y8x.dev.cldr.work/dex/ui/jobs, then paste the following :

```
xyz.cde-vjgd8ksl.dsp-azur.xcu2-8y8x.dev.cldr.work
```

- -c <spark_default_configurations>: You must provide option -c <spark_configuration> to add all the spark-configurations inside the single quotes (' ') with a delimiter comma (,) at the virtual cluster level. There must be at least one configuration in the command. For example:

```
./cde-utils.sh add-spark-config-in-virtual-cluster -h <host_
name> -c 'spark.driver.supervise="false",spark.executor.inst
ances="2"'
```

The ,user can add multiple configurations as shown above in the example: '   "key1"="val1", "key2"="val2" , ...' . The , <host_name> is is determined by the provided information in the host name option.

Enabling and disabling gang scheduling

This script enables and disables gang scheduling at the virtual cluster level for all jobs that are inside the virtual cluster. The options required are as follows:

- -h <host_name>: You must provide this option, and paste the domain of the URL into a text editor to identify the endpoint host. For example, if the URL is https://xyz.cde-vjgd8ksl.dsp-azur.xcu2-8y8x.dev.cldr.work/dex/ui/jobs, then paste the following:

```
xyz.cde-vjgd8ksl.dsp-azur.xcu2-8y8x.dev.cldr.work
```

- --gang-scheduling <gang_scheduling_status>: You must provide this option to toggle the status of the gang scheduling at the virtual cluster level. You can only use enable or disable options. To disable gang scheduling, see the following example:

```
./cde-utils.sh add-spark-config-in-virtual-cluster -h <host_
name> --gang-scheduling disable
```

The , <host_name> is determined by the provided information in the host name option.

> ⚠️ **Important:** You must provide either configurations option (-c) or gang-scheduling option(--gang-scheduling) for this sub-command. You may also provide both options.

Initiating faster autoscaling times

You can increase the speed of autoscaling times for nodes in an autoscaler cluster for Amazon Web Services (AWS) by using a script. This script allows you to speed up the scale down progress for situations where you scale up to hundreds of nodes. This is a cost-saving measure. The options required for scaling down nodes are as follows:

- -i <interactive_menu>: You can use this option to use an interactive menu that uses the arrow keys to navigate through the menu options. For example:

```
./cde-utils.sh edit-cluster-autoscaler -i
```

> ⚠️ **Important:** In the edit-cluster-autoscaler sub command, you can provide all the other options except -i <interactive_menu>, or only provide -i <interactive_menu> to change the value of other options using the interactive menu.

- --scale-down-delay-after-add: Reduce the value of this option which specifies the interval after scaling up when scale down evaluation resumes. This allows you to speed up the process. The default value is 10 minutes. Permitted values are 10 seconds, 5 minutes, and 1 hour. The format is as follows: <time>[s / m / h]. For Example:

```
./cde-utils.sh edit-cluster-autoscaler --scale-down-delay-af
ter-add 10m
```

- --scale-down-delay-after-delete: You can decrease the value for this option which is the duration of time after node deletion when the scale down evaluation resumes. The default is 10 seconds.

Permitted values are 10 seconds, 5 minutes, and 1 hour. The format is as follows: <time>[s / m / h]. For example:

```
./cde-utils.sh edit-cluster-autoscaler --scale-down-delay-af
ter-delete 10s
```

- --scale-down-delay-after-failure: You can decrease the value for this option which is a duration of time after a scale down failure when the scale down evaluation resumes. The default value is 3 minutes. Permitted values are 10 seconds, 5 minutes, and 1 hour. The format is as follows: <time>[s / m / h]. For example:

```
./cde-utils.sh edit-cluster-autoscaler --scale-down-delay-af
ter-failure 3m
```

- --scale-down-unneeded-time: To speed up the process, the user can decrease the value of this option which specifies the amount of time before a node qualifies for scale down. The default value is 10 minutes. Permitted values are 10 seconds, 5 minutes, and 1 hour. The format is as follows: <time>[s / m / h]. For example:

```
./cde-utils.sh edit-cluster-autoscaler --scale-down-unneeded-
time 10m
```

- --unremovable-node-recheck-timeout: You can reduce the value for this option which is the duration of the timeout before the unremovable node is checked again. The default value is 5 minutes. Permitted values are 10 seconds, 5 minutes, and 1 hour. The format is as follows: <time>[s / m / h]. For example:

```
./cde-utils.sh edit-cluster-autoscaler --unremovable-node-re
check-timeout 5m
```

Example use cases with commands

These example use cases and commands show how you can use the cde-utils script.

- Initiate faster autoscaling times with the command below. The order of options is not significant in this command:

```
./cde-utils.sh edit-cluster-autoscaler --scale-down-delay-af
ter-add 1m --scale-down-delay-after-failure 1m --scale-down-
delay-after-delete 5s
--scale-down-unneeded-time 1m --unremovable-node-recheck-timeo
ut 1m
```

- Add a single Spark configuration at the virtual cluster level:

```
./cde-utils.sh add-spark-config-in-virtual-cluster -h <host_
name> -c spark.executor.instances="2"
```

- Add multiple spark configurations at the virtual cluster level:

```
./cde-utils.sh add-spark-config-in-virtual-cluster -h <host_
name> -c 'spark.driver.supervise="false",spark.executor.inst
ances="2"'
```

- Enable gang scheduling:

```
./cde-utils.sh add-spark-config-in-virtual-cluster -h <host_
name> --gang-scheduling enable
```

Related Information

Accessing the Kubernetes dashboard

Support Video: Enabling kubectl for CDE

Locking and unlocking a CDE Service

Locking a Cloudera Data Engineering (CDE) Service freezes the configuration of a Service and its corresponding Virtual Clusters. When a Service is locked, you are unable to edit, add, or delete a Service and its Virtual Clusters.

About this task

You may want to lock a Service in scenarios where you don't want configuration changes to occur during a planned upgrade of CDE. The lock and unlock options will display interchangeably, depending on the state that the Service is in. For example, if the Service is unlocked, you will only see the Lock Service option when you complete the steps below. Once your Service is locked, you will see the option to Unlock it as you complete the steps below.

Before you begin
Note that locking a Service is a preview feature starting in CDE 1.19. Ensure that you are on CDE 1.19 and above.

Procedure

1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
2. Click Administration.
3. Locate the Service that you want to lock or unlock.
4. Click ⋮ on the Service, and select the Lock Service or Unlock Service option.
5. Enter the name of the Service that you want to lock or unlock as this confirms your choice.
6. Click the appropriate Lock Service or Unlock button.

Deleting virtual clusters

If you no longer need a particular virtual cluster, you can delete it. Deleting a virtual cluster instantly deletes all metadata associated with the cluster, including job configurations and log files. Do not delete a virtual cluster unless you are certain you no longer need access to any of the applications, jobs, or log files. Job run history is maintained separately, and is not deleted even if you delete a virtual cluster.

Before you begin

Important: The user interface for CDE 1.17 and above has been updated. The left-hand menu was updated to provide easy access to commonly used pages. The steps below will vary slightly, for example, the Overview page has been replaced with the Home page. The new home page still displays Virtual Clusters, but now includes quick-access links located at the top for the following categories: Jobs, Resources, and Download & Docs.

Procedure

1. On the Cloudera Data Engineering (CDE) Overview page, select the environment containing the virtual cluster that you want to delete.
2. In the Virtual Clusters column for the selected environment, click the menu icon at the top right of the virtual cluster that you want to delete, and then click Delete.

   Warning: Deleting a virtual cluster deletes all metadata associated with the cluster, including job configurations and log files. Do not do this unless you are certain that you no longer need any of these.
3. Confirm that you want to delete the cluster by entering the cluster name and then clicking Delete.

Accessing non-public CDE virtual clusters via SOCKS proxy

A SOCKS proxy server, or jump host, allows your web browser to connect securely to a non-public Cloudera Data Engineering (CDE) endpoint. When you connect to the CDE endpoint using its internal IP, the browser connects to the proxy server, which performs the required SSH tunneling, and forwards the request to the internal CDE endpoint.

About this task

This topic describes how to configure a SOCKS proxy to access CDE Virtual Clusters on non-publicly routable VPCs.

Note: Using the below approach all the web traffic in the launched browser instance will be routed via the jump host.

Procedure

1. In a VPC with a public subnet, which can also route to the target VPC where the CDE Load Balancer is running, create an EC2 instance for your SOCKS server (for example, my-ec2-jump-host) with a public IP and an SSH key-pair (for example, my-key-file.pem).

   The instance does not need to be large, for example on AWS a micro instance is usually sufficient, but may need to be larger depending on how many users are using it concurrently.

   Depending on whether you want multiple users to share the jump host or have everyone create their own server, pick the SSH key pair for the instance accordingly.

   Note: Make sure to apply a locked down security group to your jump host instance to ensure only users from trusted IP ranges can SSH into the server.

   Related AWS Documentation: Getting Started with Amazon EC2 Instances, Amazon EC2 Key Pairs

   a) Set up a SOCKS proxy server via SSH to the EC2 instance using below command:

   ```
   nohup ssh -i "SSH-key-pair" -CND <SOCKS-Proxy-Port> ec2-user@<jump-host-
   public-IP> &
   Example: nohup ssh -i "my-key-file.pem" -CND 8157 ec2-user@<public ip for
    my-ec2-jump-host> &
   ```

   where

   - nohup(optional) is a POSIX command to ignore the HUP (hangup) signal so that the proxy process is not terminated automatically if the terminal process is later terminated.
   - my-key-file.pem is the private key you used to create the EC2 instance where the SOCKS server is running.
   - C sets up compression.
   - N suppresses any command execution once established.
   - D 8157 sets up the SOCKS 5 proxy on the port. The port number 8157 in this example is arbitrary, but must match the port number you specify in your browser configuration in step 2.
   - ec2-user is the AMI username for the EC2 instance. The AMI username can be found in the details for the instance displayed in the AWS Management Console on the Instances page under the Usage Instructions tab.
   - <public ip for my-ec2-jump-host>is the public IP address of the EC2 instance running the SOCKS server you created above.
   - & (optional) causes the SSH connection to run as an operating system background process, independent of the command shell. Without the &, you leave your terminal open while the proxy server is running and use another terminal window to issue other commands.

2. Configure Your Browser to Use the Proxy.
Using Google Chrome

By default, Google Chrome uses system-wide proxy settings on a per-profile basis. To get around that you can start Chrome using the Command Line and specify the following:

- The SOCKS proxy port to use (must be the same value used in step 1, 8157 in example above)
- The profile to use (new profile created in the example below)

a) Run the following commands to create a new profile and launch a new instance of Chrome that does not interfere with any currently running instance.

For Linux:

```
/usr/bin/google-chrome \
--user-data-dir="$HOME/chrome-with-proxy" \
--proxy-server="socks5://localhost:<SOCKS-Proxy-Port>"
```

For MacOS

```
"/Applications/Google Chrome.app/Contents/MacOS/Google Chrome" \
--user-data-dir="$HOME/chrome-with-proxy" \
--proxy-server="socks5://localhost:<SOCKS-Proxy-Port>"
```

For Windows

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" ^
--user-data-dir="%USERPROFILE%\chrome-with-proxy" ^
--proxy-server="socks5://localhost:<SOCKS-Proxy-Port>"
```

Results

These steps enable you to navigate to any CDE virtual cluster in the browser launched using SOCKS proxy.