Cloudera Data Engineering 1.15.2

Accessing the Cloudera Data Engineering service using the API

Date published: 2020-07-30
Date modified: 2022-09-14

https://docs.cloudera.com/

Legal Notice

Contents

Using the Cloudera Data Engineering API

Cloudera Data Engineering (CDE) provides a robust API for integration with your existing continuous integration/continuous delivery platforms.

The Cloudera Data Engineering service API is documented in Swagger. You can view the API documentation and try out individual API calls by accessing the API DOC link in any virtual cluster:

1. In the Data Engineering web console, select an environment.
2. Click the Cluster Details icon in any of the listed virtual clusters.
3. Click the link under API DOC.

Related Information

Using CLI-API to Automate Access to Cloudera Data Engineering

Using Cloudera Data Engineering CLI

Getting a Cloudera Data Engineering API access token

Cloudera Data Engineering uses JSON Web Tokens (JWT) for API authentication. To interact with a virtual cluster using the API, you must obtain an access token for that cluster.

Before you begin

Determine the authentication endpoint for your virtual cluster:

1. Navigate to the Cloudera Data Engineering Overview page by clicking the Data Engineering tile in the Cloudera Data Platform (CDP) management console.
2. In the Services column, select the environment containing the virtual cluster you want to interact with.
3. In the Virtual Clusters column on the right, click the Cluster Details icon on the virtual cluster you want to interact with.
4. Click the link under GRAFANA CHARTS. The hostname of the URL in your browser is the base URL, and /gateway/<****AUTHTKN-OR-CDPTKN*****>/knoxtoken/api/v1/token is the endpoint.
   - Example: LDAP Authentication URL

     ```
     https://service.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/gateway/authtkn/knoxtoken/api/v1/token
     ```

   - Example: Access Key Authentication URL

     ```
     https://service.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/gateway/cdptkn/knoxtoken/api/v1/token
     ```

For Using LDAP authentication

1. From the client you want to use to access the API, run curl -u <workload_user> <auth_endpoint>. Enter your workload password when prompted.

   For example:

   ```
   curl -u csso_psherman https://service.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/gateway/authtkn/knoxtoken/api/v1/token
   ```

   The user account is your CDP workload user .

2. In the output, the access_token value is the JWT. For convenience, copy it and set it as an environment variable:

```
export CDE_TOKEN=<ACCESS_TOKEN>
```

Alternatively, you can set the token in a single step using jq to extract the token:

```
export CDE_TOKEN=$(curl -s -u <WORKLOAD_USER> <AUTH_ENDPOINT> | jq -r
 '.access_token')
```

For Using CDP Access keys

- Create a requirements.txt file specifying the Python package and version dependencies required by your CDE job.

Steps

1. Generate CDP Access Keys in User Management Console.
2. Generate DE workload auth token using CDP IAM API.

   The IAM API endpoint <CDP_ENDPOINT>/api/v1/iam/generateWorkloadAuthToken is called to generate a CDP Access Token. A CDP API call requires a request signature to be passed in the "x-altus-auth" header, along with a corresponding timestamp in the "x-altus-date" header. The cdpcurl library constructs the headers automatically. However, if you would rather use a different HTTP client, such as ordinary curl, you can use the cdpv1sign script within cdpcurl to generate these required headers.

   The request body contains workload-name as DE and is authenticated using the CDP Access Key. This request must also be signed as per the specification is available here either manually or use the cdpv1sign library to generate these necessary headers through automation script.

```
curl -X POST '<CDP_ENDPOINT>/api/v1/iam/generateWorkloadAuthToken' \
-H "Content-Type: application/json" \
-H "x-altus-date: Tue, 15 Mar 2022 07:22:57 GMT" \
-H "x-altus-auth: <signature-string-as-per-the-specification>" \
-i --insecure \
-d '{
   "workloadName": "DE"
}'
```

   Note: CDP_ENDPOINT corresponds to CDP Control Plane console url. For example, https://console-test-cp.apps.shared-os-dev-01.kcloud.cloudera.com

3. The response will include the CDP Access Token in the token field and expiry time in the expireAt.

```
 {
    "token": "<token-string>",
    "expireAt": "2021-05-03T15:34:03.727000+00:00"
}
```

4. Export the CDP token to CDP_TOKEN from above output.

```
export CDP_TOKEN=<token-string>
```

5. Once you have a CDP access token CDP_TOKEN from the previous step, you can manually exchange it for a CDE access token.

```
curl https://service.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/gateway/
authtkn/knoxtoken/api/v1/token \
-XPOST \
-H "Accept: application/json" \
-H "Authorization: Bearer ${CDP_TOKEN}" \
```

```
-i --insecure
```

6. In the output, the access_token value is the JWT. For convenience, copy it and set it as an environment variable:

```
export CDE_TOKEN=<access_token>
```

Alternatively, you can set the token in a single step using jq to extract the token:.

```
export CDE_TOKEN=$(curl <auth_endpoint> -XPOST \
-H "Accept: application/json" \
-H "Authorization: Bearer ${CDP_TOKEN}" \
-i --insecure | jq -r '.access_token')
```

See Using an access token in Cloudera Data Engineering API calls for instructions on using the token in API calls.

Using an access token in Cloudera Data Engineering API calls

Cloudera Data Engineering (CDE) uses JSON Web Tokens (JWT) for API authentication.

Before you begin

Get an access token and save it as an environment variable as described in Getting a Cloudera Data Engineering API access token.

Procedure

1. Determine the API URL for the virtual cluster you want to access using the API. The API URL for managing jobs is different from the URL used to obtain the access token.
   a) Navigate to the Cloudera Data Engineering Overview page.
   b) In the Environments column, select the environment containing the virtual cluster you want to interact with using the API.
   c) In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to interact with.
   d) Click JOBS API URL to copy the link to your clipboard.
   For example: https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1
2. When you make an API call, include the JWT as a bearer token. For example, to list all jobs associated with the virtual cluster, assuming you have saved the token as an environment variable named CDE_TOKEN:

```
curl -H "Authorization: Bearer ${CDE_TOKEN}" -H "Content-Type: applicati
on/json" -X GET "https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.
site/dex/api/v1/jobs"
```

Managing Cloudera Data Engineering job resources using the API

A *resource* in Cloudera Data Engineering (CDE) is a named collection of files referenced by a job. The files can include application code, configuration files, and even Python virtual environment specifications (requirements.txt). These resources can be managed using the CDE API.

Creating a Cloudera Data Engineering resource using the API

A *resource* in Cloudera Data Engineering (CDE) is a named collection of files referenced by a job. The files can include application code, configuration files, and even Python virtual environment specifications (requirements.txt). You can create a resource using the CDE API.

Before you begin

As with all API calls, make sure you have a valid access token. For instructions, see Getting a Cloudera Data Engineering API access token.

Procedure

1. Create a JSON file describing the resource, using the structure for the type of resource you want to create.

   The JSON payload for a resource is structured as follows:
   files resource type

   ```
   {
      "name":"<RESOURCENAME>",
      "type":"files",
      "retentionPolicy":"keep_indefinitely"
   }
   ```

   python-env resource type

   ```
   {
      "name":"<RESOURCENAME>",
      "type":"python-env",
      "retentionPolicy":"keep_indefinitely",
      "pythonEnv":{
         "pythonVersion":"python3"
      }
   }
   ```

   custom-runtime-image resource type

   ```
   {
     "customRuntimeImage": {
       "credential": "string",
       "engine": "string",
       "image": "string",
       "modified": "string"
     },
     "name": "<RESOURCENAME>",
     "retentionPolicy": "keep_indefinitely",
     "type": "custom-runtime-image"
   }
   ```

2. Create the resource by submitting a POST request to the resources endpoint. The JSON filename is referenced using the @/path/to/filename.json convention. In this example, the JSON filename is fileResource.json, and describes a files type resource named example-job-files.

   ```
   curl <jobs_api_url>/resources
   -H "Authorization: Bearer ${CDE_TOKEN}" \
   -H "Content-Type: application/json" \
   -X POST -d @${HOME}/fileResource.json
   ```

3. Verify that the resource was creating by requesting the resource details from the /resources/<resourceName> endpoint:

```
curl <jobs_api_url>/example-job-files \
-H "Authorization: Bearer ${CDE_TOKEN}" \
-H "Content-Type: application/json" \
-X GET
```

Deleting a Cloudera Data Engineering resource using the API

A *resource* in Cloudera Data Engineering (CDE) is a named collection of files referenced by a job. The files can include application code, configuration files, and even Python virtual environment specifications (requirements.txt). You can delete a resource using the CDE API.

Before you begin

As with all API calls, make sure you have a valid access token. For instructions, see Getting a Cloudera Data Engineering API access token.

Procedure

1. Make sure that you no longer need the resource before deleting it.
2. Delete the resource by sending a DELETE request to the /resources/<*RESOURCENAME*> endpoint.
   For example, to delete a resource named example-job-files:

```
curl <jobs_api_url>/example-job-files \
-H "Authorization: Bearer ${CDE_TOKEN}" \
-H "Content-Type: application/json" \
-X DELETE
```

Creating a Cloudera Data Engineering job using the API

You can create a job in Cloudera Data Engineering (CDE) using the CDE jobs API endpoint.

Before you begin

Request an access token and save it as an environment variable to use in API calls. For instructions, see Getting a Cloudera Data Engineering API access token.

Procedure

1. Determine the API URL for the virtual cluster you want to access using the API:
   a) Navigate to the Cloudera Data Engineering Overview page.
   b) In the Environments column, select the environment containing the virtual cluster you want to interact with using the API.
   c) In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to interact with.
   d) Copy the URL under JOBS API URL.
   For example: https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1
2. Submit the job creation request using the API to the /jobs endpoint.

   The JSON payload to create a job is structured as follows:

```
{
```

```
  "name": "demoJob",
  "spark": {
    "className": "com.example.demoJobMainClass",
    "file": "local:/path/to/demoJobJar"
  },
  "type": "spark"
}
```

```
curl -H "Authorization: Bearer ${CDE_TOKEN}" <JOBS_API_URL>/jobs \
-H "Content-Type: application/json" \
-X POST -d "{\"name\":\"demoJob\",\"spark\":{\"className\":\"com.example.d
emoJobMainClass\",\"file\":\"local:/path/to/demoJobJar\"},\"type\":\"spark
\"}"
```

3. Verify the job was created. You can view job details using the /jobs/<*JOBNAME*> endpoint:

```
curl -H "Authorization: Bearer ${CDE_TOKEN}" -H "Content-Type: applicati
on/json" -X GET "https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.
site/dex/api/v1/jobs/demoJob"
```

Listing Cloudera Data Engineering jobs using the API

You can list Cloudera Data Engineering (CDE) jobs using the API by issuing a GET request to the /jobs endpoint.

Before you begin

Get an access token and save it as an environment variable as described in Getting a Cloudera Data Engineering API access token.

Procedure

1. Determine the API URL for the virtual cluster you want to access using the API:
   a) Navigate to the Cloudera Data Engineering Overview page.
   b) In the Environments column, select the environment containing the virtual cluster you want to interact with using the API.
   c) In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to interact with.
   d) Copy the URL under JOBS API URL.
   For example: https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1
2. Issue a GET request to the /jobs endpoint:

```
curl -H "Authorization: Bearer ${CDE_TOKEN}" -X GET "https://pmjkrgn5.cde-
czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1/jobs"
```

Getting Cloudera Data Engineering job information using the API

You can get Cloudera Data Engineering (CDE) job information using the API by issuing a GET request to the /jobs/<*JOB_NAME*> endpoint.

Before you begin

Get an access token and save it as an environment variable as described in Getting a Cloudera Data Engineering API access token.

Procedure

1. Determine the API URL for the virtual cluster you want to access using the API:
   a) Navigate to the Cloudera Data Engineering Overview page.
   b) In the Environments column, select the environment containing the virtual cluster you want to interact with using the API.
   c) In the Virtual Clusters column on the right, click the Cluster Details icon for the virtual cluster you want to interact with.
   d) Copy the URL under JOBS API URL.
   For example: https://pmjkrgn5.cde-czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1
2. Issue a GET request to the /jobs/<*JOBNAME*> endpoint:

```
curl -H "Authorization: Bearer ${CDE_TOKEN}" -X GET "https://pmjkrgn5.cde-
czlmkz4y.na-01.xvp2-7p8o.cloudera.site/dex/api/v1/jobs/SparkPi"
```