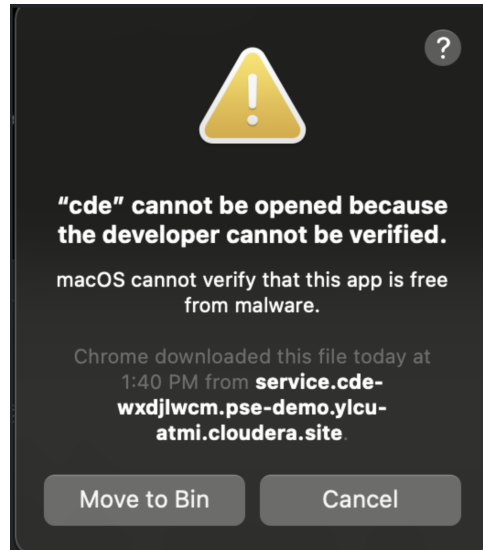


## For Mac users:

- Make sure that the `cde` file is executable by running the below command.

```
chmod +x /path/to/cde
```

- Open the CDE CLI. You might get the below error.



- Go to System Preferences → Security and Privacy and add this app in the trust center.
- Once done, add the path to the cli in the PATH variable. Run the below command in the terminal to update the path variable.

```
export PATH=$PATH:"<path-to-cde>"
```

**Example :** If the file is situated in the path  
`/home/user/applications/cde`

Then the export command will look something like this

```
export PATH=$PATH:/home/user/applications/
```

**Note :** Once the terminal is closed, the step to add the path to the PATH variable needs to be performed again.

- To validate the installation, run the below command from the terminal.  
`cde --help`

```
> cde --help
Cloudera Data Engineering

Usage:
  cde [command]

Available Commands:
  airflow      Airflow commands
  backup       Create and Restore CDE backups
  credential   Manage CDE credentials
  help         Help about any command
  job          Manage CDE jobs
  resource     Manage CDE resources
  run          Manage CDE runs
  spark        Spark commands

Flags:
  --access-key-id string      access key identifier
  --access-key-secret string  access key secret
  --auth-cache-file string    token file cache location (default "$USERCACHE/token-cache")
  --auth-no-cache             do not cache authentication tokens
  --auth-pass-file string     authentication password file location
  --credentials-file string    CDP credentials file location
  --credentials-profile string CDP credentials profile name (default "default")
  -h, --help                  help for cde
  --hide-progress-bars        hide progress bars for file uploads
  --insecure                  API does not require authentication
  --skip-credentials-file      skip CDP credentials file discovery
```

- If you get the output as shown above, then the installation is completed successfully. We now need to configure the CLI to connect to our virtual cluster.
- For configuring the CDE CLI, we create a new file and add the cluster details and use it as an environment variable for connecting to the CDE virtual cluster.
- Create a file as config.yaml and add the following details.
- Here, **user** is the username you have been assigned. *DO NOT INCLUDE "@workshop.com" as part of user (example: user: wuser01)*
- Use the **Jobs API URL** for the virtual cluster.

CLI TOOL : [API DOC](#) [JOBS API URL](#) [GRAFANA CHARTS](#) [AIRFLOW UI](#)

```
touch config.yaml
```

```
use cde [command] help for more information about a command
[mmehra@MacBook-Pro workshop % touch config.yaml]
[mmehra@MacBook-Pro workshop % ls]
cde          config.yaml
mmehra@MacBook-Pro workshop %
```

```
vi config.yaml
```

```
user: <CDP_user>
vcluster-endpoint: <CDE_virtual_cluster_endpoint>
```

```
user: wuser01
vcluster-endpoint: https://hg98mkgj.cde-7fsd4m65.pse-part.dp5i-5vkq.cloudera.site/dex/api/v1
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

- Open terminal and run the below command to create an environment variable.


```
export CDE_CONFIG=/path/to/config.yaml
```

- Run the below command to verify whether the above step is completed successfully. You should see the `path-to-config.yaml` as the output.

```
echo $CDE_CONFIG
```

- Run the below command to validate the configuration. Upon running it, you will be asked to provide the API password. Please enter the password as mentioned in the excel sheet.

```
cde job list
```

```
> cde job list
API User Password: 
```

- Once you enter the password, you should see all the jobs present in the virtual cluster.

```
mmehra@MacBook-Pro workshop % cde job list
API User Password:
{
  "name": "pkattil-Lab6A",
  "type": "spark",
  "created": "2021-11-03T07:10:23Z",
  "modified": "2021-11-03T07:13:00Z",
  "retentionPolicy": "keep_indefinitely",
  "mounts": [
    {
      "resourceName": "AutoResource-1635923577528"
    }
  ],
  "spark": {
    "file": "Lab681_PreSetupDW.py",
    "driverMemory": "1g",
    "driverCores": 1,
    "executorMemory": "1g",
    "executorCores": 1,
    "conf": {
      "dex.safariEnabled": "false",
      "spark.pyspark.python": "python3"
    },
    "logLevel": "INFO"
  },
  "schedule": {
    "enabled": false,
    "user": "pkattil"
  },
  "enabled": true,
  "user": "pkattil"
},
{
  "name": "pkattil-Lab6B",
  "type": "spark",
  "created": "2021-11-03T07:14:20Z",
  "modified": "2021-11-03T07:14:02Z",
  "retentionPolicy": "keep_indefinitely",
  "mounts": [
    {
      "resourceName": "AutoResource-1635926318679"
    }
  ],
  "spark": {
    "file": "Lab682_EnrichData_ETL.py",
    "driverMemory": "1g",
    "driverCores": 1,
    "executorMemory": "1g",
    "executorCores": 1,
    "conf": {
      "dex.safariEnabled": "false",
      "spark.pyspark.python": "python3"
    },
    "logLevel": "INFO"
  },
  "schedule": {
    "enabled": false,
    "user": "pkattil"
  },
  "enabled": true,
  "user": "pkattil"
},
{
  "name": "test_s3_write",
  "type": "spark",
  "created": "2021-10-30T08:24:21Z",
  "modified": "2021-10-31T07:53:33Z",
  "retentionPolicy": "keep_indefinitely",
  "mounts": [
    {
      "resourceName": "AutoResource-1635582769149"
    }
  ],
  "spark": {
    "file": "test_s3_write.py",
    "driverMemory": "1g",
    "driverCores": 1,
    "executorMemory": "1g",
    "executorCores": 1,
    "conf": {

```

- If you get any error related to the certificate, please add the flag to skip tls verification.

```
cde job list --tls-insecure
```

- This marks the end of installation and configuration of CDE CLI. Now, head over to the next lab to trigger the jobs from CLI.