# Overview

This document explains how to use CDP CLI to configure and manage the environment.

# Pre-requisites

**ssh Key**: you will need to paste the public key into your parameters file

## CDP Installation:

* CDP CLI ([Instructions](https://docs.cloudera.com/management-console/cloud/cli/topics/mc-cli-client-setup.html))
* CDP Credential ([Instructions](https://docs.cloudera.com/management-console/cloud/credentials-azure/topics/mc-create-app-based-credential.html))
  + You must set your Workload Password in your CDP Profile ([Shortcut](https://console.cdp.cloudera.com/iam/index.html#/my-account))
  + You must generate a CLI Access Key in your CDP Profile, and configure it to your local CDP CLI ([Shortcut](https://console.cdp.cloudera.com/iam/index.html#/my-account))

## To set up the CDP client, complete the following tasks: ([Instructions](https://docs.cloudera.com/management-console/cloud/cli/topics/mc-cli-generating-an-api-access-key.html))

* Install the CDP client.
* Generate an API access key.
* Configure the CDP client with the API access key.

Optional : Install the latest version of the [Azure CLI](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli). To start, run az login to create a connection with Azure.

# CLI Profile:

By default, profile name should be default:

cdp configure –-profile ${profileName}

Enter the following information at the prompt:

1. CDP Access key. Copy and paste the access key ID that you generated in the Management Console.
2. CDP Private key. Copy and paste the private key that you generated in the Management Console.

The configuration utility creates the following file to store your user credentials: ~/.cdp/credentials

# CDP PARAMETER:

## Create Environment parameter file like below

{

"environmentName": "**ENVIRONMENT\_NAME**",

"credentialName": "[CREDENTIAL\_NAME](https://docs.cloudera.com/management-console/cloud/credentials-azure/topics/mc-create-app-based-credential.html)",

"region": "**REGION\_TO\_USE**",

"datalakeName": "**DATALAKE\_NAME**",

"scale": **"LIGHT\_DUTY",**

"runtime": "**7.2.1**",

"groupName": "**GROUP\_NAME**",

"cdp\_profile": "**default**",

"workload\_pwd": "**PASSWORD**",

"publicKey": "**SSH\_KEYS",**

"proxyConfigName": "**PROXY\_CONFIG\_NAME**",

"resourceGroupName": "**RESOURCE\_GROUP\_NAME**",

"securityAccess": {

"cidr": "",

"securityGroupIdForKnox": "",

"defaultSecurityGroupId": ""

},

"cloudProviderConfiguration": {

"assumerManagedIdentity": "",

"dataStorageLocation": "**abfs://DATA\_CONTAINER@STORAGE\_NAME.dfs.core.windows.net**",

"dataManagedIdentity": "",

"rangerManagedIdentity": ""

},

"logStorage": {

"logStorageLocationBase": "**abfs://LOG\_CONTAINER@STORAGE\_NAME.dfs.core.windows.net**",

"logManagedIdentity": ""

},

"existingNetworkParams": {

"networkId": "**NETWORK\_NAME**",

"subnetIds": [

**"SUBNET\_1","SUBNET\_2","SUBNET\_3"**

]

},

"newNetworkParams": {

"networkCidr": ""

},

"description": "**DESCRIPTION**",

"freeIpa": {

"instanceCountByGroup": 1

},

"mappings": [

{

"accessorCrn": "",

"role": ""

}

],

"tags": [

{

"key": "",

"value": ""

}

],

"datahub\_list": [

{

"definition": "kafka.json",

"custom\_script": ""

},

{

"definition": "flink.json",

"custom\_script": ""

}

]

}

# Set workload password:

cdp --profile ${profileName}

cdp iam set-workload-password --password ${workload\_pwd}

Environment Registration is three step process

# 1) CDP ENVIRONMENT:

## If Network is created already then use below CDP command to register in CDP

cdp environments create-azure-environment \

--environment-name ${environmentName} \

--credential-name ${credential} \

--region "${region}" \

--public-key "${publicKey}" \

--log-storage storageLocationBase="${logStorageLocationBase}”,managedIdentity="${logManagedIdentity}" \

--no-use-public-ip \

--workload-analytics \

--enable-tunnel \

--existing-network-params networkId="${networkID}”,resourceGroupName="${resourcegroup}",subnetIds="${subnetIds}" \

--security-access

cidr="${cidr}”, securityGroupIdForKnox="${knox\_id}”,defaultSecurityGroupId="${default\_id}", \

--free-ipa instanceCountByGroup=1 \

--tags key=string,value=string

## If Network is not created then use below CDP command to register in CDP

cdp environments create-azure-environment

--environment-name ${environmentName} \

--credential-name ${credential} \

--region "${region}" \

--public-key "${publicKey}" \

--log-storage storageLocationBase="${logStorageLocationBase}”,managedIdentity="${logManagedIdentity}" \

--no-use-public-ip

--workload-analytics \

--enable-tunnel \

--resource-group-name "${resourcegroup}" \

--new-network-params networkCidr="{networkCidr}"

--security-access cidr="${cidr}" \

--free-ipa instanceCountByGroup=1 \

--tags key=string,value=string

## To check the status of the Environment creation: (Lookout for Available status)

cdp environments describe-environment --environment-name ${environmentName}| jq -r .environment.status

# 2) CDP SET ID BROKER MAPPING:

A CDP environment and Data Lake with no end user access to cloud storage. Adding users and groups to a CDP environment involves ensuring they are properly mapped to IAM roles to access cloud storage.

## Retrieve User CRN

user\_crn=$(cdp iam get-user | jq -r .user.crn)

group\_crn=$(cdp iam list-groups --group-names $group\_name | jq -r .groups[0].crn)

cdp environments set-id-broker-mappings --environment-name ${environmentName} \

--data-access-role ${dataManagedIdentity} --ranger-audit-role "${rangerManagedIdentity}" \

--ranger-cloud-access-authorizer-role "${rangerManagedIdentity}" --set-empty-mappings

## Adding CDP user/group to IAM role mappings

Under the IDBroker Mappings, you can change the mappings of users or groups to IAM roles. The user or group dropdown is prepopulated with CDP users and groups. On the right-hand side, specify the role ARN (copied from the IAM role page) for that user or group that you are configuring.

For example, in the example setup we created the following roles:

DATAENG\_ROLE - We created this role while onboarding users, and we assume that there is a DataEngineers group that was created in CDP.

DATASCI\_ROLE - We created this role while onboarding users, and we assume that there is a DataScientists group that was created in CDP.

cdp environments set-id-broker-mappings --environment-name ${environmentName} \

--data-access-role ${dataManagedIdentity} --ranger-audit-role "${rangerManagedIdentity}" \

--ranger-cloud-access-authorizer-role "${rangerManagedIdentity}" --no-set-empty-mappings \

**--mappings accessorCrn=${user\_crn or group\_crn},role=${anyManagedIdentity}**

# 3) CDP DATALAKE:

[CDP supported Data Lakes Scale.](https://docs.cloudera.com/management-console/cloud/data-lakes/topics/mc-data-lake-scale.html)

## Use below command to create SDX or Datalake

cdp datalake create-azure-datalake \

--datalake-name ${datalakeName} \

--environment-name ${environmentName} \

--cloud-provider-configuration

storageLocation="${dataStorageLocation}",managedIdentity="${assumerManagedIdentity}" \

--scale "${scale}" \

--runtime "${runtime}" \

--tags key=string,value=string

## To check the status of the Datalake creation: (Lookout for Available status)

cdp environments describe-environment --environment-name ${environmentName} | jq -r .datalake.datalakeName

# Sync ID Broker with datalake after datalake creation

cdp environments sync-id-broker-mappings --environment-name ${environmentName}

# Get Keytab:

cdp environments get-keytab --environment-name ${environmentName} | jq -r '.contents' | base64 –decode

# Synchronizes environments with user

## A user to the FreeIPA servers.

cdp environments sync-user

output:

{

  "operationId": "c52f9e7d-0a1e-43db-ae33-e23297b3c3f6",

  "operationType": "USER\_SYNC",

  "status": "RUNNING",

  "success": [],

  "failure": [],

  "startTime": "1602178446261"

}

## All users to the FreeIPA servers.

cdp environments sync-all-users  --environment-name ${environmentName}

## To check the user sync status, use below command.

cdp environments sync-status --operation-id c52f9e7d-0a1e-43db-ae33-e23297b3c3f6

check status:

cdp environments sync-status --operation-id c52f9e7d-0a1e-43db-ae33-e23297b3c3f6 | jq -r .status

# CDP DATAHUB:

cdp datahub create-azure-cluster --cli-input-json “$(cat <kafka\_template.json>)”

# Get Subnet ID from the environment:

cdp environments describe-environment --environment-name ${prefix}-cdp-env | jq -r .environment.network.subnetIds[0])