Cloudera Public Cloud

Azure Onboarding Quickstart

Date published: 2019-08-22
Date modified:

https://docs.cloudera.com/

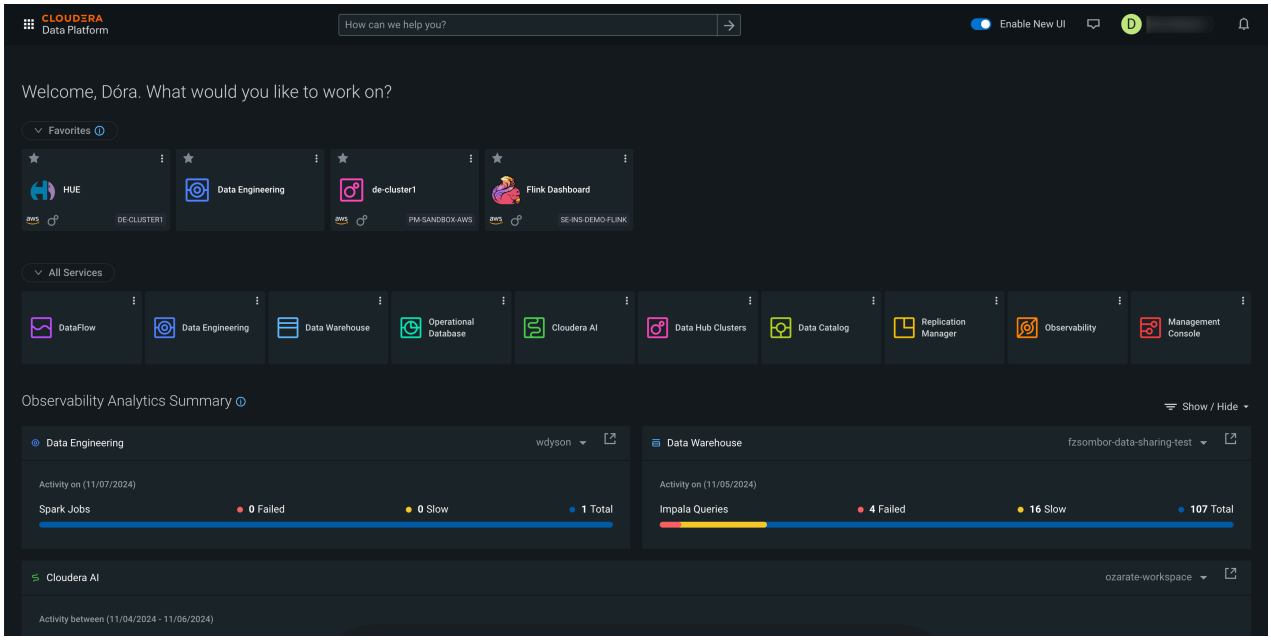
Legal Notice

Contents

Azure quickstart (Deprecated)

If you've reached the Cloudera landing page for the first time, you've come to the right place! In this quickstart, we'll show you step-by-step how to connect Cloudera to your Azure subscription, so that you can begin to provision clusters and workloads.

Warning: This quickstart has been deprecated and is no longer being maintained. For quickly setting up Cloudera on Azure, refer to Deploy Cloudera using Terraform.

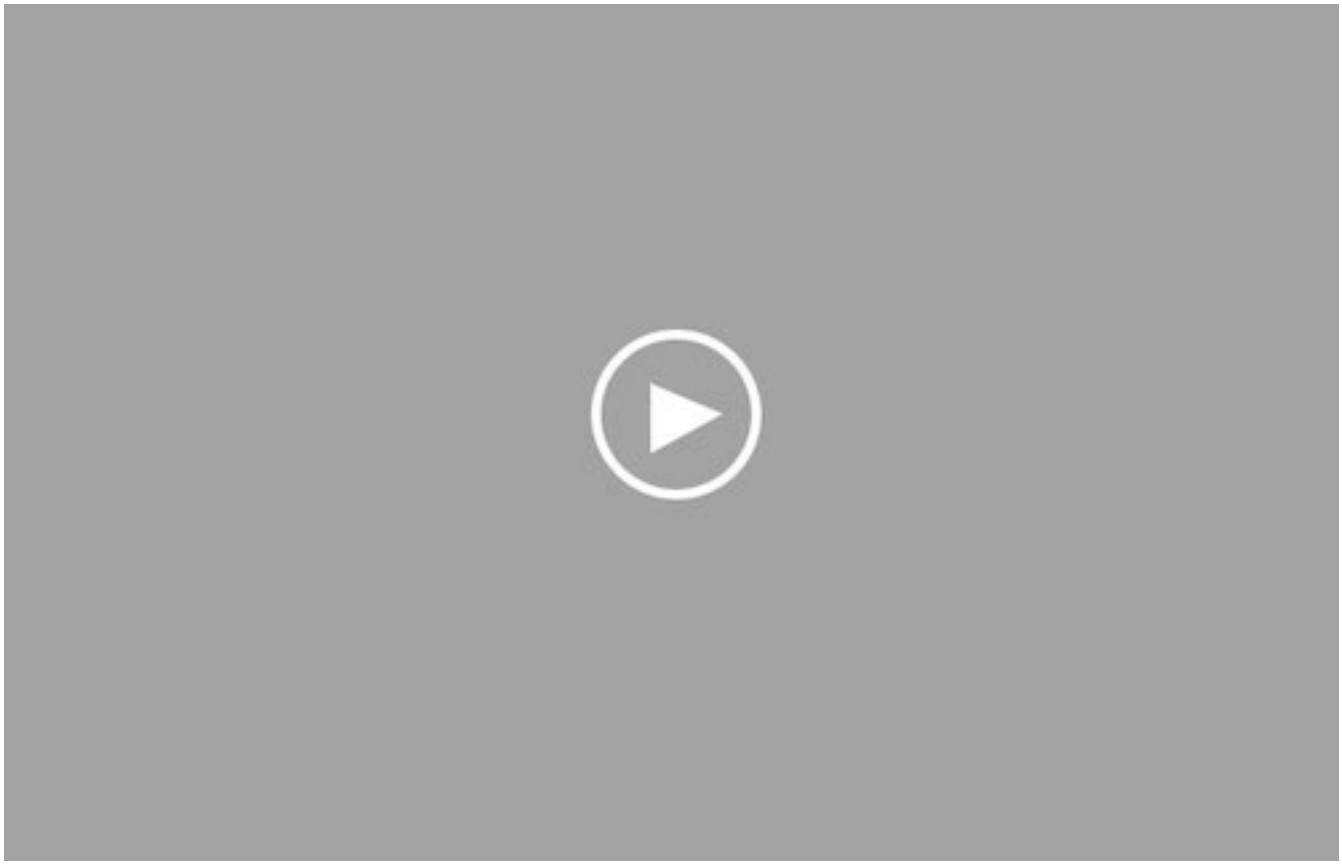


To complete this quickstart, you'll need access to three things:

- The Cloudera console pictured above
- The Azure console
- Azure Cloud shell

Note: This Azure onboarding quickstart is intended for simple Cloudera evaluation deployments only. It may not work for scenarios where Azure resources such as VNet, security group, storage accounts, and so on, are pre-created or Azure accounts have restrictions in place.

In addition to this documentation, you can refer to the following video:



The steps that we will perform are:

Step 0: Verify the Azure prerequisites

Step 1: Create an Azure AD app

Step 2: Deploy the Azure quickstart template

Step 3: Assign roles

Step 4: Create or locate an SSH key

Step 5: Create a Cloudera credential

Step 6: Register a Cloudera environment

Verify Azure cloud platform prerequisites

Before getting started with the Azure onboarding quickstart, review and acknowledge the following:

- This Azure onboarding quickstart is intended for simple Cloudera evaluation deployments only. It may not work for scenarios where Azure resources such as VNet, security group, storage accounts, and so on, are pre-created or Azure accounts have restrictions in place.
- User running the Azure onboarding quickstart should have:
  - Owner permissions on the Azure subscription that you would like to use for Cloudera.
  - Rights to create Azure resources required by Cloudera. See list of Azure resources used by Cloudera.
  - Rights to create an Azure AD application (service principal) and assign Contributor role at subscription level.
  - Cloudera Admin role or Power User role in Cloudera subscription.
- This Azure onboarding quickstart uses an Azure ARM template that automatically creates the required resources such as storage accounts, containers, managed identities, resource groups, and so on.

- Cloudera on cloud relies on several Azure services that should be available and enabled in your region of choice. Verify if you have enough quota for each Azure service to set up Cloudera in your Azure account. See list of Azure resources used by Cloudera.

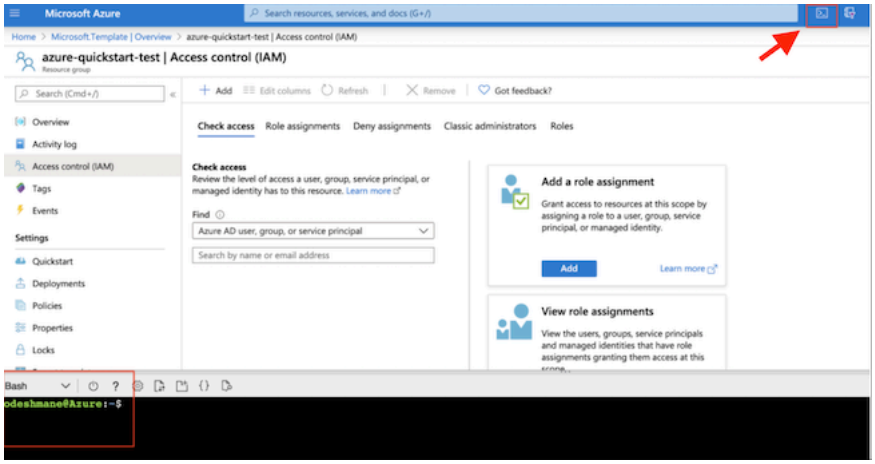
If you have more complex requirements than those listed here, contact Cloudera Sales Team to help you with Cloudera onboarding.

Create an Azure AD app

In the Azure portal, create an application in your Azure Active Directory tenant. This steps allows you to use the native Cloud Shell terminal and not have to set up Azure CLI.

Procedure

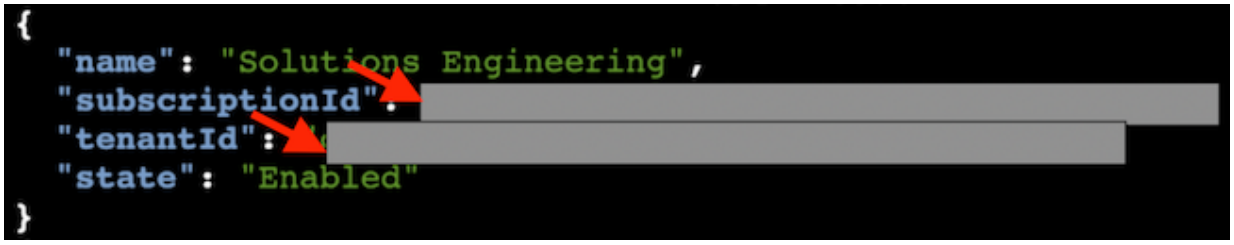
1. Log in to the Azure portal and launch Cloud Shell.



2. When prompted, select Bash to use Bash shell.
3. Run the following command to return the subscription ID and tenant ID:

```
az account list|jq '.[]|{"name": .name, "subscriptionId": .id, "tenantId
": .tenantId, "state": .state}'
```

The output of this command is shown below:



Make a note of the subscriptionId and tenantId values. You will need them later.

Note: In case you have more than one subscription, make sure to only make a note of the subscription that you would like to use for Cloudera.

4. Run the command below to create an app in Azure AD and assign the "Contributor" role at the subscription.

   Note: Replace {subscriptionId} in the command with the subscription ID value from the previous step.

   ```
   az ad sp create-for-rbac --name http://cloudbreak-app --role Contributor
    --scopes /subscriptions/{subscriptionId}
   ```

   The output of this command is shown below:

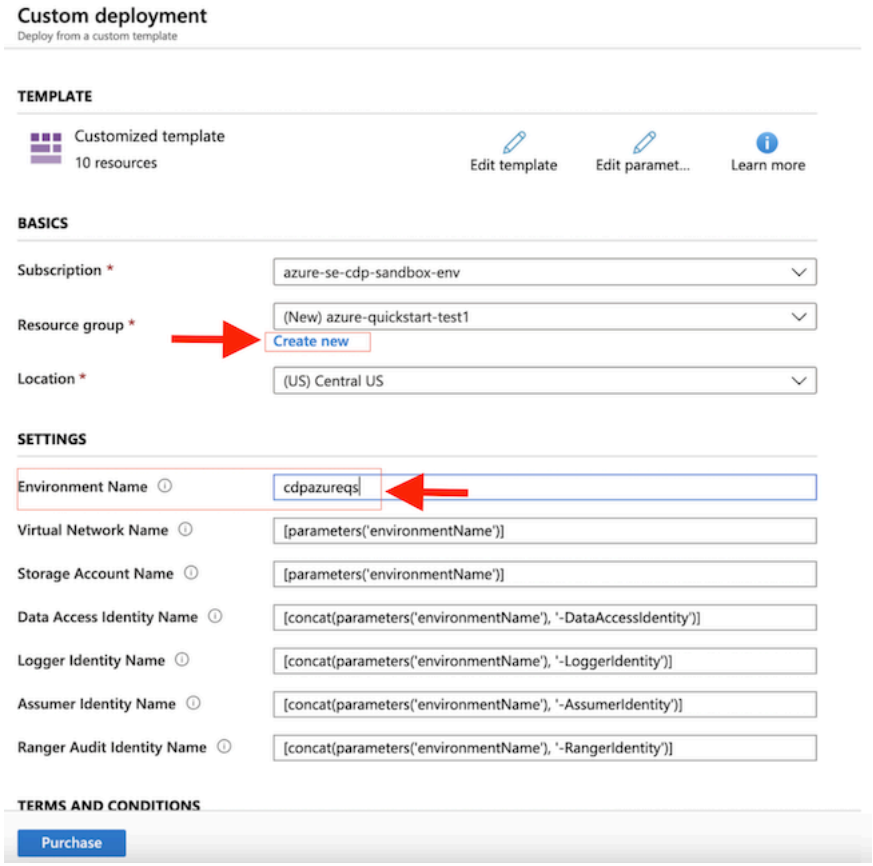
Deploy the Azure quickstart template

The Azure quickstart template is a customized ARM template that deploys essential Azure resources for the Cloudera environment.

Procedure

1. Click Deploy to Azure to begin ARM template deployment of Cloudera prerequisites in your Azure subscription.
2. Log in to Azure to create the resources in your subscription that are required for Cloudera deployment. These resources include VNet, ADLS Gen2, and 4 user managed identities.
3. On the **Custom deployment** screen, click Create new under the Resource group field and then give the resource group a name (it should only contain letters, numbers, and hyphens).

4. Under **Settings**, provide an Environment Name in the corresponding field. The Environment Name should be short (between five and twelve characters) and should include only lowercase characters and hyphens (no underscores).



5. Accept the terms and conditions, and click Purchase.

   An ARM script begins to run in the background, creating the resources required for a Cloudera environment. This may take around 10 minutes.
6. When your resource group is up, navigate to the **Overview** page of the resource group.
7. Copy and paste the following values into a note, as you will need them in the next task:
   - Subscription ID: Your subscription ID is found at the top of the resource group **Overview** page.
   - Resource group: The name of the resource group that you created.

Assign roles

Azure Resource Manager templates do not support role assignments at a scope other than resource groups. Perform the following role assignments through UI or CLI.

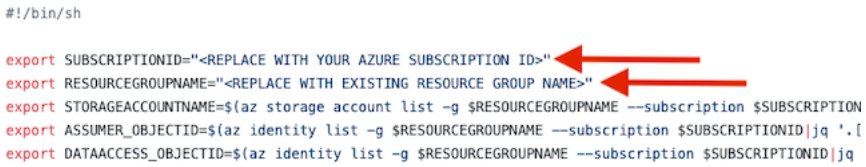
Before you begin

Make sure that you have your note from the previous step, where you copied values for the Subscription ID and resource group name.

Procedure

1. Once you have values for the subscription ID, resource group name, storage account, environment name, and all four managed identities, click here to download a script.
2. Create a new file in Cloud Shell with the same name, and copy the content of the script there.

3. Replace the following values in the script with the values you have collected thus far:



For example, your script should look similar to this:

```
#!/bin/sh

export SUBSCRIPTIONID="jfs85ls8-sik8-8329-fq0m-jqo7v06dk6sy"
export RESOURCEGROUPNAME="myCDPresourcegroup"
export STORAGEACCOUNTNAME=$(az storage account list -g $RESOURCEGROUPNAME
--subscription $SUBSCRIPTIONID|jq '.[]|.name'| tr -d '"')
export ASSUMER_OBJECTID=$(az identity list -g $RESOURCEGROUPNAME --subscri
ption $SUBSCRIPTIONID|jq '.[]|{"name":.name,"principalId":.principalId}|
select(.name | test("AssumerIdentity"))|.principalId'| tr -d '"')
export DATAACCESS_OBJECTID=$(az identity list -g $RESOURCEGROUPNAME --subs
cription $SUBSCRIPTIONID|jq '.[]|{"name":.name,"principalId":.principalI
d}|select(.name | test("DataAccessIdentity"))|.principalId'| tr -d '"')
export LOGGER_OBJECTID=$(az identity list -g $RESOURCEGROUPNAME --subscr
iption $SUBSCRIPTIONID|jq '.[]|{"name":.name,"principalId":.principalId}|
select(.name | test("LoggerIdentity"))|.principalId'| tr -d '"')
export RANGER_OBJECTID=$(az identity list -g $RESOURCEGROUPNAME --subscrip
tion $SUBSCRIPTIONID|jq '.[]|{"name":.name,"principalId":.principalId}|s
elect(.name | test("RangerIdentity"))|.principalId'| tr -d '"')
# Assign Managed Identity Operator role to the assumerIdentity principal
 at subscription scope
az role assignment create --assignee $ASSUMER_OBJECTID --role 'f1a07417-
d97a-45cb-824c-7a7467783830' --scope "/subscriptions/$SUBSCRIPTIONID"
# Assign Virtual Machine Contributor role to the assumerIdentity principal
 at subscription scope
az role assignment create --assignee $ASSUMER_OBJECTID --role '9980e02c-c
2be-4d73-94e8-173b1dc7cf3c' --scope "/subscriptions/$SUBSCRIPTIONID"
# Assign Storage Blob Data Contributor role to the assumerIdentity prin
cipal at logs filesystem scope
az role assignment create --assignee $ASSUMER_OBJECTID --role 'ba92f5b4-2
d11-453d-a403-e96b0029c9fe' --scope "/subscriptions/$SUBSCRIPTIONID/reso
urceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAccounts/
$STORAGEACCOUNTNAME/blobServices/default/containers/logs"
# Assign Storage Blob Data Contributor role to the loggerIdentity princ
ipal at logs/backup filesystem scope
az role assignment create --assignee $LOGGER_OBJECTID --role 'ba92f5b4-
2d11-453d-a403-e96b0029c9fe' --scope "/subscriptions/$SUBSCRIPTIONID/res
ourceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAccoun
ts/$STORAGEACCOUNTNAME/blobServices/default/containers/logs"
az role assignment create --assignee $LOGGER_OBJECTID --role 'ba92f5b4-
2d11-453d-a403-e96b0029c9fe' --scope "/subscriptions/$SUBSCRIPTIONID/res
ourceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAccoun
ts/$STORAGEACCOUNTNAME/blobServices/default/containers/backups"
# Assign Storage Blob Data Owner role to the dataAccessIdentity principal
 at logs/data/backup filesystem scope
az role assignment create --assignee $DATAACCESS_OBJECTID --role 'b7e6dc6
d-f1e8-4753-8033-0f276bb0955b' --scope "/subscriptions/$SUBSCRIPTIONID/r
esourceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAcco
unts/$STORAGEACCOUNTNAME/blobServices/default/containers/data"
az role assignment create --assignee $DATAACCESS_OBJECTID --role 'b7e6dc6
d-f1e8-4753-8033-0f276bb0955b' --scope "/subscriptions/$SUBSCRIPTIONID/r
esourceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAcco
unts/$STORAGEACCOUNTNAME/blobServices/default/containers/logs"
```

```
az role assignment create --assignee $DATAACCESS_OBJECTID --role 'b7e6dc6
d-f1e8-4753-8033-0f276bb0955b' --scope "/subscriptions/$SUBSCRIPTIONID/r
esourceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAcco
unts/$STORAGEACCOUNTNAME/blobServices/default/containers/backups"
# Assign Storage Blob Data Contributor role to the rangerIdentity principa
l at data/backup filesystem scope
az role assignment create --assignee $RANGER_OBJECTID --role 'ba92f5b4-2d1
1-453d-a403-e96b0029c9fe' --scope "/subscriptions/$SUBSCRIPTIONID/resour
ceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAccounts/
$STORAGEACCOUNTNAME/blobServices/default/containers/data"
az role assignment create --assignee $RANGER_OBJECTID --role 'ba92f5b4-2d1
1-453d-a403-e96b0029c9fe' --scope "/subscriptions/$SUBSCRIPTIONID/resour
ceGroups/$RESOURCEGROUPNAME/providers/Microsoft.Storage/storageAccounts/
$STORAGEACCOUNTNAME/blobServices/default/containers/backups"
```

4. Run the Cloud Shell script: sh azure_msi_role_assign.sh

Create or locate an SSH Key

Cloudera requires that you provide a public SSH key for admin access to VM instances.

You can find more information on SSH key requirement in the topic SSH key. If you need to create one, you can do so by running ssh-keygen -t rsa.

When you complete this step, you have created all of the Azure resources required for this quickstart.

Create a Cloudera credential

In the Cloudera Console, the first step is to create a Cloudera credential. The Cloudera credential is the mechanism that allows Cloudera to create resources inside of your cloud account.

Procedure

1. Log in to the Cloudera web interface.
2. From the Cloudera home screen, click the Cloudera Management Console icon.
3. In the Cloudera Management Console, select Shared Resources > Credentials from the navigation pane.
4. Click in the Create Credential button to create a new credential.
5. Select the Azure tab, name your credential, under "Credential Type", select "App based", and enter the values you previously collected for subscription ID, app ID, and password.

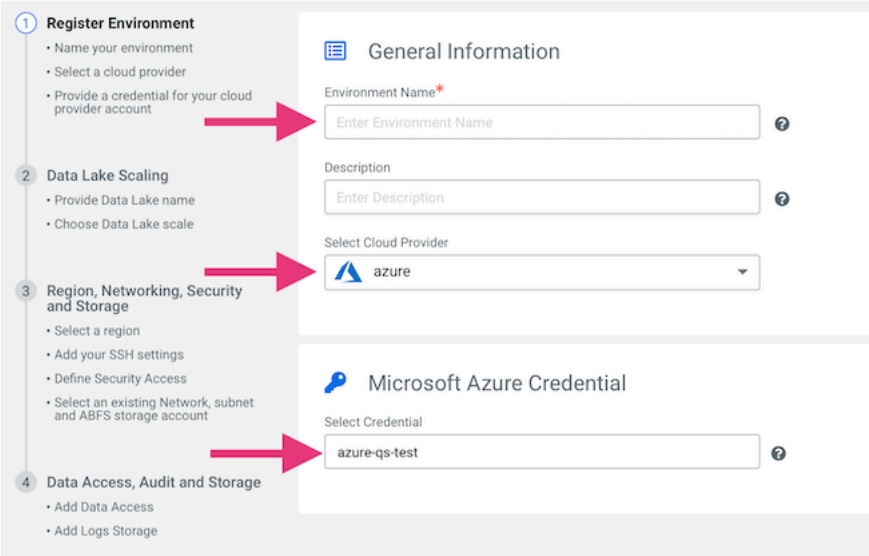
Register a Cloudera environment

When you register an environment, you set properties related to data lake scaling, networking, security, and storage. You will need your Azure environment name, resource group name, storage account name, and virtual network name from your resource group.

Procedure

1. In the Cloudera Management Console, navigate to Environments and click Register Environment.
2. Provide an Environment Name and description. The name can be any valid name.
3. Choose Azure as the cloud provider.

4. Under Microsoft Azure Credential, choose the credential you created in the previous task.

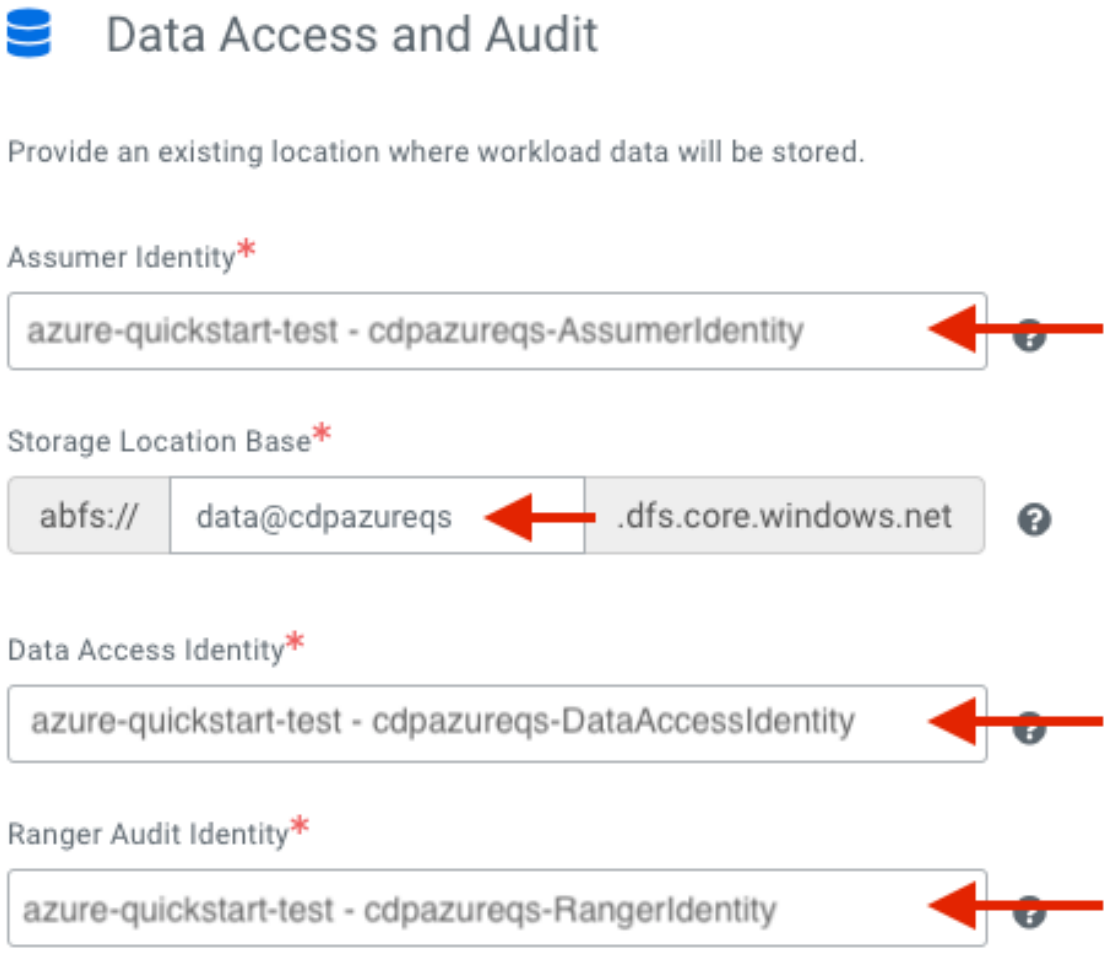


5. Click Next.
6. Under **Data Lake Settings**, give your new data lake a name. The name can be any valid name. Choose the latest data lake version.

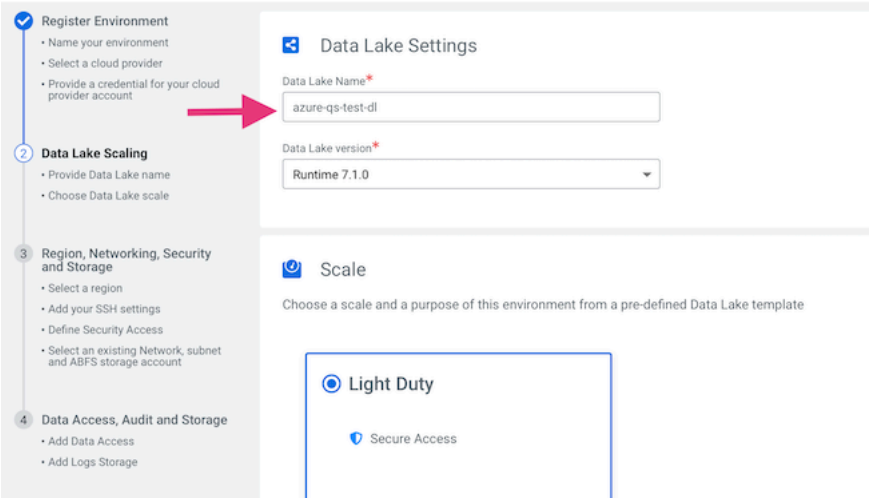
7. Under **Data Access and Audit**, choose the following:
   - Assumer Identity: <resourcegroup-name>-<envName>-AssumerIdentity
   - Storage Location Base: data@<storageaccount-name>
   - Data Access Identity: <resourcegroup-name>-<envName>-DataAccessIdentity
   - Ranger Audit Role: <resourcegroup-name>-<envName>-RangerIdentity

   Warning: Ensure that you have entered the correct location base. If the name entered does not match the actual location base created by the quickstart script, environment registration will fail.

   For example:

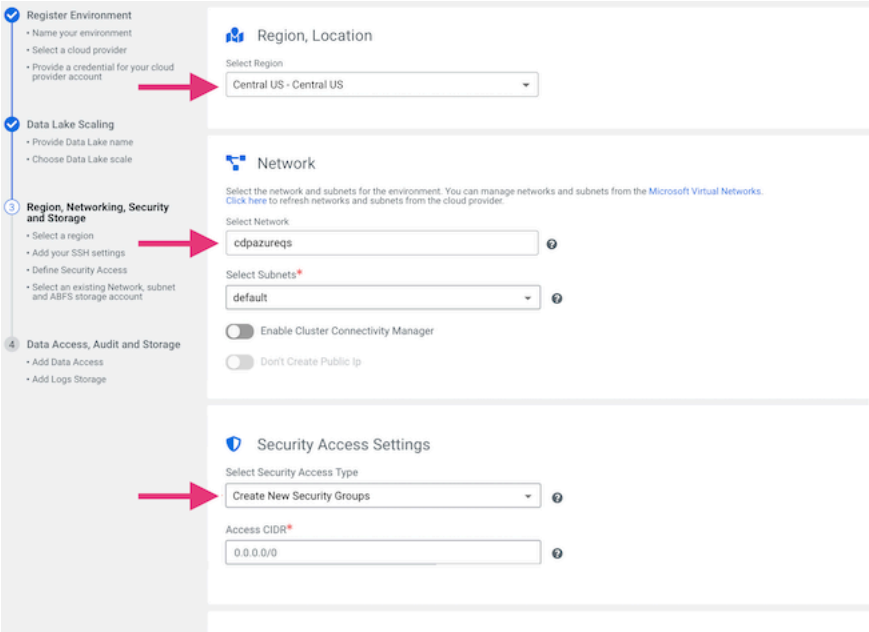


8. For Data Lake **Scale**, choose Light Duty.



9. Click Next.
10. Under Select Region, choose your desired region. This should be the same region you created an SSH key in previously.
11. Under Select Resource Group, choose your resource group <resourcegroup-name>.
12. For the Select Network field, select the name of the "Virtual Network" resource that was created when you deployed the ARM template to create the resource group. The name of the Virtual Network should be the same as your environment name, but you can verify this in the Azure portal on the Overview page of your resource group. In the following Select Subnets field, ensure all three subnets are selected.
13. Slide the Enable Public Endpoint Access Gateway toggle to the Enabled position. Select any of your three subnets.
14. The Create Public IPs toggle can remain in the default Enabled position.
15. In the next drop-down input box, keep Flexible Server selected.
16. Under the Encryption section, leave both the Enable encryption at host and Enable Customer-Managed Keys toggles in their default deselected state.
17. Under the Proxies section, leave the Select Proxy Configuration field in its default "Do not use Proxy Configuration" state.

18. Under **Security Access Settings**, select Create New Security Groups for the Security Access Type.

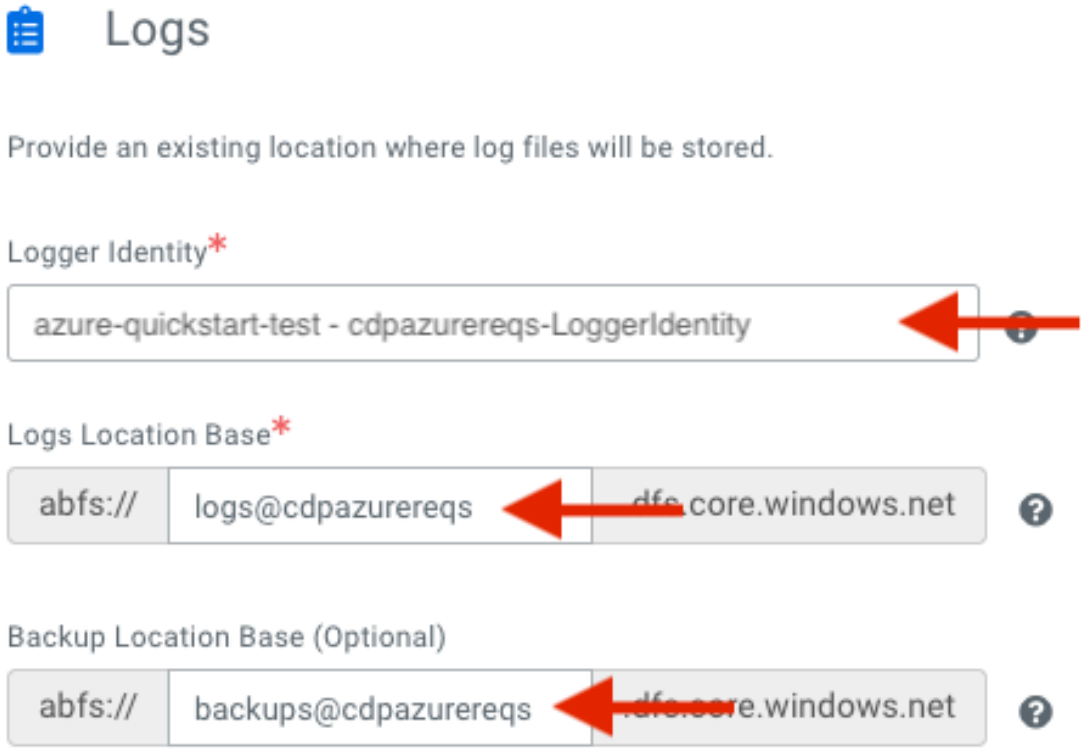


19. Under **SSH Settings**, paste the public SSH key that you created earlier.
20. Optionally, under **Add Tags**, provide any tags that you'd like the resources to be tagged with in your Azure account.
21. Click Next.

22. Under **Logs**, choose the following:
    - Logger Identity: <resourcegroup-name>-<envName>-LoggerIdentity
    - Logs Location Base: logs@<storageaccount-name>
    - Backup Location Base: backups@<storageaccount-name>

    Warning: Ensure that you have entered the correct location base. If the name entered does not match the actual location base created by the quickstart script, environment registration will fail.

    For example:



23. Under the Telemetry section, leave the inputs as default (Observability Enabled, Deployment Cluster Logs Collection Disabled).
24. Before finishing, click the Show CLI Command button and then Copy the full command for future reference. This can be useful for e.g. analyzing any errors in the deployment process.
25. Click Register Environment.