

AWS Onboarding Quickstart

Date published: 2019-08-22

Date modified: 2025-08-18

CLOUDERA

Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

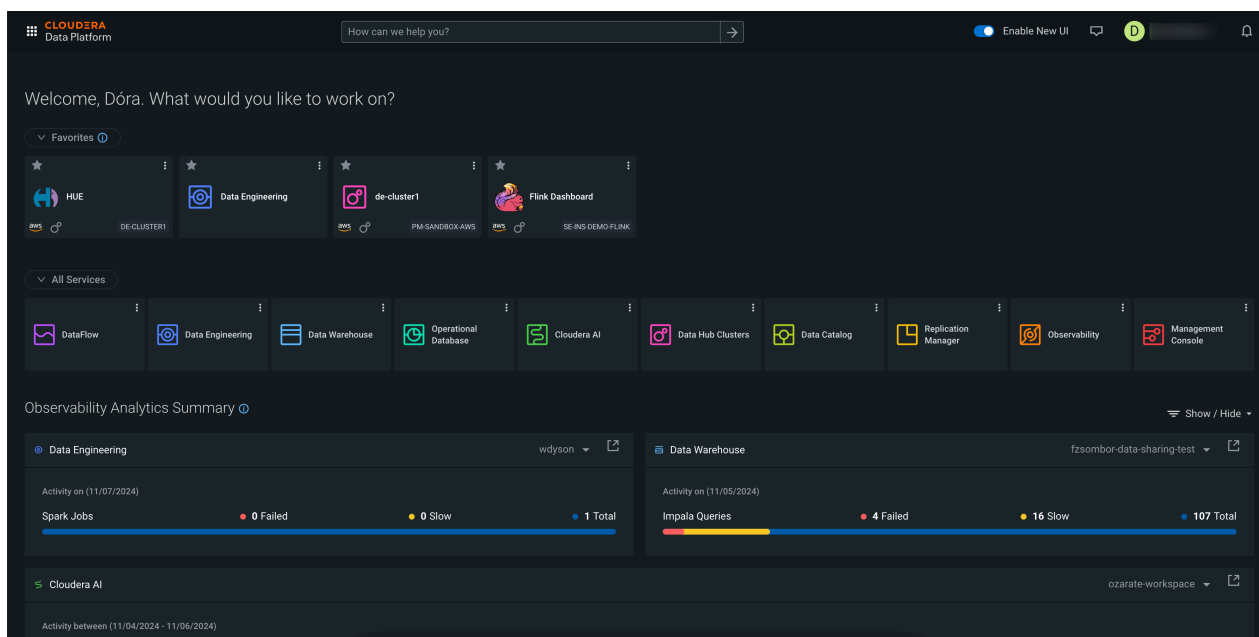
AWS quickstart (Deprecated).....	4
Verify AWS prerequisites.....	4
Create a Cloudera credential.....	5
Register a Cloudera environment.....	7

AWS 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 AWS account, 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 AWS, refer to [Deploy Cloudera using Terraform](#).



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

- The Cloudera console pictured above
- The AWS console



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

The steps that we will perform are:

Step 0: Verify the AWS prerequisites

Step 1: Create a provisioning credential

Step 2: Register an AWS environment in Cloudera

Verify AWS cloud platform prerequisites

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

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

- Users running the AWS onboarding quickstart should have:
 - AWS Administrator permissions on the AWS account that you would like to use for Cloudera.
 - Rights to create AWS resources required by Cloudera. See list of [AWS resources used by Cloudera](#).
 - Cloudera Admin role or Power User role in Cloudera subscription.
- This AWS onboarding quickstart uses a CloudFormation template that automatically creates the required resources such as buckets, IAM roles and policies, and so on.
- Cloudera on cloud relies on several AWS services that should be available and enabled in your region of choice. Verify if you have enough quota for each AWS service to set up Cloudera in your AWS account. See list of [AWS resources used by Cloudera](#).

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

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 Create Credential.
5. Click the Copy icon to the right of the **Create Cross-account Access Policy** text box.

Create Credential

aws cloudera

Name *
Enter credential name

Description
Enter description

☐ Enable Permission Verification

Create Cross-account Access Policy
Copy the following JSON to create an AWS IAM policy

Default Minimal

The default role allows for the default set of operations including everything that the minimal role allows for.

```
{
  "Statement": [
    {
      "Sid": "CloudFormationFull",
      "Action": [
        "cloudformation:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

6. In a second browser tab, open the **AWS Console** and navigate to Identity and Access Management Policies. Click Create Policy.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Account settings

Policies (952)

A policy is an object in AWS that defines permissions.

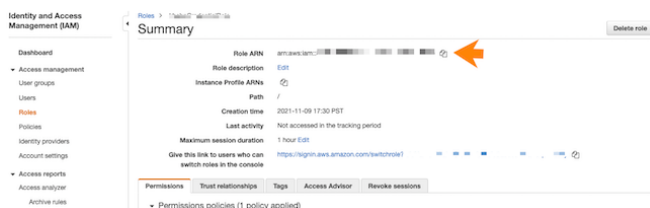
Filter policies by property or policy name and press enter

Policy name	Type	Used as	Description
AwsS3OutpostsAccessPolicy-20200218T124200	Customer managed	Permissions policy (1)	
aws-ec2-policy	Customer managed	Permissions policy (1)	AWS - EC2 Standardize
ec2_permissions	Customer managed	Permissions policy (1)	Incremental Permissions
demo-ec2-pm-bucket-policy-s3access	Customer managed	None	

7. Click on the JSON tab and paste the access policy in the text box.

You may get a warning related to using wildcards. You may ignore it and proceed to the next step.

8. Click Next: Tags.
9. Click Review Policy.
10. Give the policy a unique name and a description.
11. Click Create Policy.
Next, you create the required cross-account role.
12. In the AWS console, navigate back to Identity and Access Management.
13. Click Roles>Create Role.
14. Under **Select type of trusted entity**, select AWS Account > Another AWS account.
15. Return to the Cloudera Management Console and copy the contents of the Service Manager Account ID field on the **Credentials** page.
16. In the AWS console, paste the Service Manager Account ID into the Account ID field.
17. Return to the Cloudera Management Console and copy the contents of the External ID field on the **Credentials** page.
18. In the AWS console, check the "Require external ID" options box, and then paste the External ID copied from Cloudera into the External ID field.
19. Click Next: Permissions.
20. Under Permissions, select the checkbox next to the name of the policy that you created in Step 8.
21. Click Next: Tags.
22. Click Next: Review.
23. Give the role a unique name and description, then click Create Role.
24. Still in the role page of the AWS console, search for the role you just created, and click on it.
25. Copy the Role ARN at the top of the **Summary** page.



26. Return to the **Credentials** page in the Cloudera Management Console.
27. Give the Cloudera credential a name and description. The name can be any valid name.

28. Paste the Role ARN that you copied from the AWS console into the Cross-account Role ARN field, then click Create.

Create Cross-account Access Policy
Copy the following JSON to create an [AWS IAM policy](#)

Default Minimal

The default role allows for the default set of operations including everything that the minimal role allows for.

```
{
  "Statement": [
    {
      "Sid": "CloudFormationFull",
      "Action": [
        "cloudformation:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Create Cross-account Access Role
Use Service Manager Account ID and External ID to create an [AWS IAM role](#)

Service Manager Account ID *

387553343826

External ID *

bb90432f-29b0-4492-9e2a-8ff979164abe

Cross-account Role ARN *

Enter Cross-account Role ARN

Create

HOW CLI COMMAND

Now that you've created a cross-account role, proceed to creating a Cloudera environment.

Register a Cloudera environment

Before you register an environment, you'll want to create specific IAM roles and policies so that Cloudera can operate in a secure manner.

About this task

For background information, a description of what we're building and why can be found [here](#). For this quickstart, we'll use CloudFormation to set all of this up for you.

Procedure

1. Download the CloudFormation provided template [here](#).

2. In the AWS console, deploy the CloudFormation template:
 - a) In **AWS Services**, search for CloudFormation.
 - b) Click Create Stack and select With new resources (standard).
 - c) Select Template is ready and then Upload a template file.

The screenshot shows the 'Create stack' wizard in the AWS console, specifically the 'Specify template' step. The page is titled 'Create stack' and has a sub-header 'Prerequisite - Prepare template'. Below this, there's a section 'Prepare template' with three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The next section is 'Specify template', which explains that a template is a JSON or YAML file. It has two radio buttons: 'Amazon S3 URL' and 'Upload a template file' (selected). Under 'Upload a template file', there's a 'Choose file' button and a text input field containing 'setup.json'. Below this, it says 'JSON or YAML formatted file'. At the bottom, there's a text input field for the 'S3 URL' with the value 'https://s3.us-east-2.amazonaws.com/cf-templates-11mg5w1r6abx5-us-east-2/2020104Hxs-setup.json' and a 'View in Designer' button. At the very bottom, there are 'Cancel' and 'Next' buttons.

- d) Click Choose file and select the CloudFormation template that you downloaded.
- e) Click Next.
- f) Under Stack name, enter a stack name. The name can be any valid name.
- g) Under **Parameters**, complete the following fields:
 - BackupLocationBase: Choose an unused bucket name and path for the FreeIPA backups. Cloudera will be creating the bucket for you. The same bucket can be used for BackupLocationBase, LogsLocationBase, and StorageLocationBase. By default this is set to my-bucket/my-backups.
 - CrossAccountARN: Do not change the default value. This parameter is only required when encryption is enabled, but since in this quickstart we do not enable encryption, you should leave this value as is.
 - LogsLocationBase: Choose an unused bucket name and path for the logs. Cloudera will be creating the bucket for you. The same bucket can be used for BackupLocationBase, LogsLocationBase, and StorageLocationBase. By default this is set to my-bucket/my-logs.
 - StorageLocationBase: Choose an unused bucket name and path for the data. Cloudera will be creating the bucket for you. The same bucket can be used for BackupLocationBase, LogsLocationBase, and StorageLocationBase. By default this is set to my-bucket/my-data.
 - Prefix: A short prefix of your choosing, which will be added to the names of the IAM resources Cloudera will be creating. We chose "cloudera" as an example.
 - s3KmsEncryption: Encryption will be disabled for the created bucket. You don't need to change this value.

For example:

Specify stack details

Stack name

Stack name

mc-cdp-stack

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

BackupLocationBase

The storage base path to create an S3 bucket with default encryption for CDP. By default CDP will create the optional subdirectory in the bucket. It is possible to either use the same bucket or different buckets for StorageLocationBase and LogsLocationBase.

my-bucket/my-backups

CrossAccountARN

Required if s3 KMS Encryption is selected

arn:aws:iam::<ACCT_ID>:role/<ROLE_NAME>

LogsLocationBase

The storage base path to create an S3 bucket with default encryption for CDP. By default CDP will create the optional subdirectory in the bucket. It is possible to either use the same bucket or different buckets for StorageLocationBase and LogsLocationBase.

my-bucket/my-logs

StorageLocationBase

The logging base path to create an S3 bucket with default encryption for CDP. By default CDP will create the optional subdirectory in the bucket. It is possible to either use the same bucket or different buckets for StorageLocationBase and LogsLocationBase.

my-bucket/my-data

prefix

prefix for IAM objects, separated by a dash.

cloudera

s3KmsEncryption

If set to True S3 will be configured with AWS managed KMS server side encryption

false

Make a note of the BackupLocationBase, LogsLocationBase, StorageLocationBase, and Prefix that you define. You will need them later.

- h) Click Next.
- i) At the **Configure Stack Options** page, click Next.
- j) At the bottom of the **Review** page, under Capabilities, click the checkbox next to I acknowledge that AWS CloudFormation might create IAM resources with custom names, as that is exactly what we will be doing.

Capabilities

The following resource(s) require capabilities: [AWS::IAM::ManagedPolicy]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

Cancel Previous Create change set **Create stack**

- k) Click Submit.
3. Still in the AWS console, create an SSH key in the region of your choice. If there is already an SSH key in your preferred region that you'd like to use, you can skip these steps.
 - a) In **AWS Services**, search for EC2.
 - b) In the top right corner, verify that you are in your preferred region.
 - c) On the left hand navigation bar, choose Key Pairs.
 - d) On the top right of the screen, select Create Key Pair.
 - e) Provide a name. The name can be any valid name.
 - f) Choose RSA type, and then choose the pem format.
 - g) Click Create key pair.
 4. Return to the Cloudera Management Console and navigate to **Environments Register Environments**.
 5. Provide an environment name and description. The name can be any valid name.
 6. Choose Amazon as the cloud provider.

7. Under **Amazon Web Services Credential**, choose the credential that you created earlier.
8. Click Next.
9. Under **Data Lake Settings**, give your new data lake a name. The name can be any valid name. Choose the latest data lake version.
10. Under **Data Access and Audit**:
 - Choose prefix-data-access-instance-profile>
 - For Storage Location Base, choose the StorageLocationBase from the cloud formation template.
 - For Data Access Role, choose prefix-datalake-admin-role.
 - For Ranger Audit Role, choose prefix-ranger-audit-role, where "prefix" is the prefix you defined in the **Parameters** section of the stack details in AWS.

For example:



Data Access and Audit

Provide an existing location where workload data will be stored.

Assumer Instance Profile*

[Click here](#) to refresh instance profiles from the cloud provider.



Storage Location Base*



Data Access Role*



Ranger Audit Role*



ID Broker Mappings

You may optionally provide mappings for users or groups.

11. For Data Lake **Scale**, choose Light Duty.
12. Click Next.
13. Under Select Region, choose your desired region. This should be the same region you created an SSH key in previously.

14. Under **Select Network**, choose Create New Network.

15. Create private subnets should be enabled by default. If it isn't, enable it.



Note:

By enabling private subnets you will not have SSH access to cluster nodes unless you have access to the VPC.

16. Click the toggle button to enable Enable Public Endpoint Access Gateway.

For example:

Region, Location

Select Region

US West (Oregon) - us-west-2

Network

Select the network and subnets for the environment. You can manage networks and subnets from the [VPC Console](#).
[Click here](#) to refresh networks and subnets from the cloud provider.

Select Network

Create new network

Network CIDR*

10.10.0.0/16

☒ Create private subnets

☐ Create Private Endpoints

⚠ Typical NAT gateway charges will be applied on your account, see [AWS pricing](#) for more details

☒ Enable Public Endpoint Access Gateway

17. Under **Security Access Settings**, choose Create New Security Groups.

18. Under **SSH Settings**, choose the SSH key you created earlier.

For example:

The image shows two sections of a web form. The first section, titled "Security Access Settings", has a "Select Security Access Type" dropdown menu with "Create New Security Groups" selected. Below it is an "Access CIDR*" text field with "0.0.0.0/0" entered. The second section, titled "SSH Settings", contains instructions to paste an SSH public key or select an existing one from the EC2 console. Two radio buttons are present: "New SSH public key" (unselected) and "Existing SSH public key" (selected). Below the "Existing SSH public key" option is a text field with the label "Name of an existing SSH key pair to use for accessing cluster node instances." and the value "docs-test" entered.

Security Access Settings

Select Security Access Type

Create New Security Groups

Access CIDR*

0.0.0.0/0

SSH Settings

Paste your SSH public key or select an SSH key that already exists in the [EC2 console](#) > Key Pairs in the specified region. [Click here](#) to refresh SSH keys from the cloud provider.

☐ New SSH public key ☒ Existing SSH public key

Name of an existing SSH key pair to use for accessing cluster node instances.

docs-test

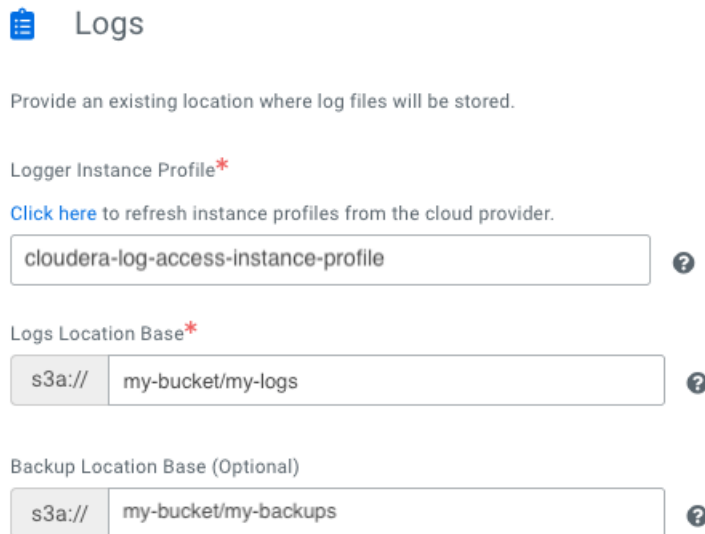
19. Optionally, under **Add Tags**, provide any tags that you'd like the resources to be tagged with in your AWS account.


20. Click Next.

21. Under Logs:

- a) Choose the Instance Profile titled prefix-log-access-instance-profile, where "prefix" is the prefix you defined in the **Parameters** section of the stack details in AWS.
- b) For Logs Location Base, choose the LogsLocationBase from the CloudFormation template.
- c) For Backup Location Base, choose the BackupLocationBase from the CloudFormation template.

For example, using the parameters we defined earlier:



 **Logs**

Provide an existing location where log files will be stored.

Logger Instance Profile*

[Click here](#) to refresh instance profiles from the cloud provider.

cloudera-log-access-instance-profile ?

Logs Location Base*

s3a:// my-bucket/my-logs ?

Backup Location Base (Optional)

s3a:// my-bucket/my-backups ?

22. Click Register Environment.