

# Cloudera Data Engineering Prerequisites

Date published: 2020-07-30

Date modified: 2024-11-12



# Legal Notice

© Cloudera Inc. 2024. 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

<b>Amazon AWS prerequisites for Cloudera Data Engineering.....</b>	<b>4</b>
Enabling Customer Managed Keys (CMK) on Amazon Web Services (AWS).....	8
<b>Microsoft Azure prerequisites for Cloudera Data Engineering.....</b>	<b>9</b>
<b>Supporting Azure private storage.....</b>	<b>12</b>
<b>Using AWS IAM restricted roles and policies for compute and CDE.....</b>	<b>14</b>
Create IAM roles and instance profile pair.....	14
Create role and policy used to deploy CDP environments for CDE.....	21

# Amazon AWS prerequisites for Cloudera Data Engineering

Amazon Web Services (AWS) prerequisites for Cloudera Data Engineering (CDE).

## Review AWS account prerequisites for CDP

Refer to the [CDP AWS account requirements](#) and verify that the AWS account you are using for CDP has the required resources, and that you have the permissions required to manage these resources.

## Review CDE-specific AWS resource requirements

Provisioning a CDE service and virtual clusters require access to the following AWS resources.

AWS Services used by Cloudera Data Engineering (CDE)

- Network – Amazon VPC (see below for requirements)
- Compute – Amazon Elastic Kubernetes Service (EKS)
- Load Balancing – Amazon ELB Classic Load Balancer
- Key Management – AWS Key Management Service (KMS)
- DNS – Amazon Route 53 (CDE makes use of this but it is hosted in Cloudera's AWS infrastructure)
- Persistent Instance Storage – Amazon Elastic Block Store (EBS)
- Persistent Service and Virtual Cluster Storage – Amazon Elastic File System (EFS)
- Database – Amazon Relational Database Service (RDS)

VPC Requirements

You can use an existing VPC, or allow CDP to create one when you create an environment.

Option 1: use your own VPC

Minimum requirements:

- CDE requires at least two subnets, each in a different Availability Zone (AZ). If you require a public endpoint for CDE, provision at least one public subnet.
- Ensure that the CIDR block for the subnets is sized appropriately. For each CDE environment, in addition to ensuring enough IPs to accommodate the maximum number of autoscaling compute instances, allow for a fixed overhead of three instances for core CDE services and approximately one instance for every two virtual clusters.
- You must enable DNS for the VPC.

Recommended setup:

- Cloudera recommends that you provision at least three subnets, each in a different Availability Zone (AZ). If you do not require a public endpoint, use three private subnets. If you require a public endpoint, use at least two private subnets and one public subnet.
- Private subnets should have routable IPs over your internal VPN. If IPs are not routable, private CDE endpoints must be accessed via a SOCKS. This is not recommended.
- Tag the VPC and the subnets as shared so that Kubernetes can find them. For load balancers to be able to choose the subnets correctly, you are also required to tag private subnets with the `kubernetes.io/role/internal-elb:1` tag, and public subnets with the `kubernetes.io/role/elb:1` tag.

Note that only the load balancer needs to be on a public subnet for access to CDE. By default, if they are available, CDE will configure the EKS to run on private subnets.

Option 2: CDP creates a new VPC

If you choose to allow CDP to create a new VPC, three subnets will be automatically created. One subnet is created for each availability zone assuming three AZs per region; If a region has two AZs instead of three, three subnets are still created, with two in the same AZ.

You will be asked to specify a valid CIDR in IPv4 range that will be used to define the range of private IPs for EC2 instances provisioned into these subnets.

Related AWS documentation: [Amazon EKS - Cluster VPC Considerations](#), [Creating a VPC for your Amazon EKS Cluster](#)

Firewall requirements

HTTPS access to CDE endpoints is available over port 443 for the following cases:

- Internal only – Should be accessible from your organization's network, but not the public internet.
- Internet facing (public endpoint) – Should be accessible from the public internet as well as your organization's internal network.

If you are using a firewall or a security group setting to prevent egress traffic from the service, make sure that the `container.repository.cloudera.com` and `docker.repository.cloudera.com` URLs on port 443 are allowed at all times.

If egress traffic is blocked to these URLs, then autoscaling cannot pull images, which can result in broken pods. For more information on required outbound access, see [Outbound network access destinations for AWS](#) and [Security groups](#).

Do not remove firewall rules added during provisioning. The rules are also required for regular operation. You must also maintain the minimum firewall requirements set by the cloud provider. For more information, see [Amazon EKS security group considerations](#) in the Amazon AWS documentation.

If you're using Amazon Relational Database Service (RDS), you'll need to ensure that you are using `*.rds.amazonaws.com` and TCP 5432 / 3306 / 443 ports.

### Review the default AWS service limits and your current AWS account limits

By default, AWS imposes certain default limits for AWS services for each user account. Make sure you review your account's current usage status and resource limits before you start provisioning additional resources for CDP and CDE.

For example, depending on your AWS account, you may only be allowed to provision a certain number of EC2 instances. Be sure to review your AWS service limits before you proceed.

Related AWS documentation: [AWS Service Limits](#), [Amazon EC2 Resource Limits](#).

### Supported AWS regions

CDP supports the following AWS regions: [Supported AWS regions](#). However, the CDE service also requires AWS Elastic Kubernetes Service (EKS). Make sure you select a region that includes EKS.

Related AWS documentation: [Region Table](#).

### Set up an AWS Cloud Credential

Create a role-based AWS credential that allows CDP to authenticate with your AWS account and has authorization to provision AWS resources on your behalf. Role-based authentication uses an IAM role with an attached IAM policy that has the minimum permissions required to use CDP.

Once you have created this IAM policy, register it in CDP as a cloud credential. Reference this credential when you register an AWS environment in CDP environment as described in the next step.

Instructions: [Cross-account access IAM role](#)

## Register an AWS Environment in CDP

A CDP user must have the Environment Creator role in order to register an environment. An environment determines the specific cloud provider region and virtual network in which resources can be provisioned, and includes the credential that should be used to access the cloud provider account.

CDE supports deployment into environments with non-transparent proxies. To use this feature, you need to [register a proxy](#) and add it to the environment during environment registration. Registering a proxy requires Power User privileges.

Instructions: [Register an AWS environment](#)

## CDE Role Requirements

There are two CDP user roles associated with the CDE service: DEAdmin and DEUser. Any CDP user with the EnvironmentAdmin (or higher) access level must assign these roles to users who require access to the Cloudera Data Engineering console within their environment.

Furthermore, if you want to allow users to log in to provisioned workspaces and run workloads on them, this will need to be configured separately.

## Set up the AWS account to run kubectl commands

1. In the AWS console, create an IAM user ( for example, kubectl-user) with Programmatic access (you don't need to grant any permissions).
2. Note the User ARN and copy the Access key ID and Secret access key and set up an AWS profile as follows:

```
[kubectl-user]

aws_access_key_id = <Access Key ID>
aws_secret_access_key = <Secret access key>
```

3. Navigate to IAM Roles and edit the cross-account IAM role (note the Role ARN) that was created as part of the CDP prerequisites.
4. Navigate to Trust relationships > Edit trust relationships.
5. Add the following to the policy document, then click Update trust policy.

```
"Effect": "Allow",
"Principal": {
  "AWS": "User ARN from step 2"
},
"Action": "sts:AssumeRole"
},
```

6. Download the kubeconfig file from the CDE UI and save it ( ~/.kube/cde-env1-kube-config, for example), then run the following shell commands:

```
$ export AWS_PROFILE=kubectl-user
$ unset AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY AWS_SESSION_TOKEN
$ cred=$(aws sts assume-role --role-arn <Role ARN from step 3> --role-session-name test | jq .Credentials)
$ export AWS_ACCESS_KEY_ID=$(echo $cred|jq .AccessKeyId|tr -d ' ')
$ export AWS_SECRET_ACCESS_KEY=$(echo $cred|jq .SecretAccessKey|tr -d ' ')
$ export AWS_SESSION_TOKEN=$(echo $cred|jq .SessionToken|tr -d ' ')
$ export KUBECONFIG=~/.kube/cde-env1-kube-config
$ export TILLER_NAMESPACE=tiller
```

7. You should now be able to run kubectl commands.

## Using AWS S3 buckets with encryption

You may need to incorporate a policy to use at-rest encryption on your Amazon Web Services (AWS) S3 buckets and telemetry log bucket. Starting with CDE 1.18 or higher, telemetry buckets with a customer-managed key is supported. When the policy is used, the data is encrypted before it is saved to your disk in S3 and is decrypted when read. This encryption and decryption takes place in the S3 infrastructure and is transparent to authenticated clients. See server-side encryption listed below under Encrypting Data on S3.

For CDE to write and read data to and from an encrypted S3 bucket, you must configure a KMS Key ARN under the Customer Managed Encryption Key for a CDP environment before you create a CDE service.

Once the KMS KEY ARN is configured, newly created CDE services will use that key to access the encrypted bucket. If the key is not configured or is invalid, then CDE can't access the encrypted telemetry bucket. This results in service/jobs logs not being stored on S3 and will be available on the Virtual Cluster user interface or for Diagnostics bundles. Additionally, the Spark user interface will not be available for completed applications.

There may be cases when you want the telemetry bucket to be encrypted with a key that is different from the one that is specified under the Customer Managed Encryption Key (see Adding a customer managed encryption key to a CDP environment running on AWS linked below) and use it to encrypt the EBS volumes and RDS instances running in the environment. In those cases, it's possible to override KMS KEY ARN via the "telemetry.encryption.key" property during service creation.

## Using Customer Managed Keys (CMK) encryption

You can use customer managed key (CMK) enabled environments for Cloudera Data Engineering (CDE) services deployed on AWS to use CMK-based data-at-rest encryption for Amazon Relational Database Service (RDS), Amazon Elastic File System (EFS) and Kubernetes secrets. For more information, see Enabling Customer Managed Keys (CMK) on Amazon Web Services (AWS) linked below.

While creating a CMK, if not provided, the AWS Key Management Service (KMS) creates a default key-policy for the newly created KMS key. The default key-policy allows CDP cross-account roles to use CMK. Instead of using default key-policy, if you want to change the key-policy, then the key-policy must contain the permission blocks below for CDE service to use CMK:

```
{
  "Sid": "AllowAutoscalingAndCDPCrossAccountRoleUseOfTheCMK",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE",
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowAutoscalingAndCDPCrossAccountRoleUseOfTheCMKGrants",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE",
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

```

    "AWS" :
    [
        "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE",
        "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/aws-service-role/autoscali
ng.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action":
  [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition":
  {
    "Bool":
    {
      "kms:GrantIsForAWSResource": "true"
    }
  }
},
{
  "Sid": "AllowCreateGrantToLiftieCluster",
  "Effect": "Allow",
  "Principal":
  {
    "AWS": "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition":
  {
    "StringEquals":
    {
      "aws:CalledViaFirst": "cloudformation.amazonaws.com"
    },
    "ForAllValues:StringEquals":
    {
      "kms:GrantOperations":
      [
        "Encrypt",
        "Decrypt"
      ]
    }
  }
}
}

```

### Related Information

[Accessing the Kubernetes dashboard](#)

[Enabling CMK on AWS](#)

[Encrypting Data on S3](#)

## Enabling Customer Managed Keys (CMK) on Amazon Web Services (AWS)

Learn how to use customer managed keys (CMK) enabled environments for Cloudera Data Engineering (CDE) services deployed on AWS using CMK-based data at rest encryption for Amazon Relational Database Service (RDS), Kubernetes secrets, and data at rest encryption.



### Before you begin

To perform these steps, you must be a CDE Admin and obtain your Customer-managed key from your Amazon Key Management Service (KMS). This CMK that you obtain will be associated with the environment for the steps below.

Add the CMK

Once you've obtained the CMK, complete the following steps:

1. Go to the Cloudera Management Console.
2. Click Environments.
3. Click the environment where your CDE Service is deployed.
4. In the Customer Managed Encryption Key section, click Edit.
5. Toggle Enable Customer-Managed Keys.
6. Select the Select Encryption Key field.
7. Click Save. Once saved, the key is associated with the environment and you are unable to change the CMK. Now, any CDE Service deployed using this CMK enabled environment uses the CMK-based data at rest encryption for EFS.

## Microsoft Azure prerequisites for Cloudera Data Engineering

Microsoft Azure prerequisites for Cloudera Data Engineering (CDE).

### Review Azure account prerequisites for CDP

Refer to the [Azure subscription requirements](#), and make sure that the Azure account you are using for CDP has the required resources, and that you have the permissions required to manage these resources.

### Review CDE-specific Azure resource requirements

The following Azure services are required to provision a CDE Service and virtual clusters:

Azure Services used by Cloudera Data Engineering (CDE)

- Network – VNet and Subnets(see below for requirements)
- Database – Azure Database for MySQL server
- Compute – Azure Kubernetes Service (AKS)
- Load Balancing - Azure Load Balancer
- Virtual machine scale set
- Storage account - CDE stores workload data and logs in Azure Data Lake Store Gen 2 (ADLS Gen2) environment storage account. The AKS service also generates a separate storage account for use with Azure Files.
- Azure Files - Contain job resources, application code, Apache Airflow DAG files and any other uploaded files. The AKS service generates an ADLS Gen2 storage account for these files.
- Log Analytics workspace

Refer to the [Azure resources used by CDP](#) to check the Azure resources used by CDP.

Vnet and Subnet Requirements

When registering an Azure environment in CDP, you will be asked to select a VNet and one or more subnets. Cloudera Data Engineering runs in the VNet registered in CDP as part of your Azure environment.

You have two options:

- Use your existing VNet and subnets for provisioning CDP resources
- Have CDP create a new VNet and subnets

New VNet and subnets

If you would like CDP to create a new VNet, you will need to specify a valid CIDR in IPv4 range that will be used to define the range of private IPs for VM instances provisioned into these subnets. This must be a /16 CIDR, but you can customize the IP Range. The default is 10.10.0.0/16.

CDP will divide this address range as follows:

- 32 x /24 private subnet for ML and CDE
- 3 x /19 private subnet for DW
- 3 x /19 private subnet for Data Lake and Data Hub
- 3 x /24 public subnet

Existing VNet and subnets

VNet Requirements

If you would like to use your own VNet, it needs to fulfill the following requirements:

- The VNet has at least one subnet
- VNet should be able to make an outbound connection with the internet or set of CIDRs and ports provided by Cloudera

Subnet Requirements

Each CDE service requires its own subnet, and must not share subnets or routing tables with any other CDP service or AKS cluster. CDE on AKS uses the Kubenet CNI plugin provided by Azure. In order to use Kubenet CNI, you must create multiple smaller subnets when creating an Azure environment. Cloudera recommends partitioning the VNet with subnets that accommodate the maximum number of expected cluster nodes.

If you are not sure what size of subnet to use, Cloudera recommends a 24-bit subnet mask (/24 CIDR prefix), which can accommodate 254 usable IP addresses per subnet. If you want to use a different subnet size, you can calculate the number of IP addresses required per CDE service as follows:

- Each CDE service can scale up to 100 compute nodes; each node consumes one IP address.
- Each CDE service requires 3 IP addresses for the infrastructure nodes
- Each virtual cluster within a CDE service requires 2 IP addresses

For more information, see [VNet and subnets](#), [VNet and subnet planning](#)

Firewall requirements

HTTPS access to CDE endpoints is available over port 443 for the following cases:

- Internal only – Should be accessible from your organization's network, but not the public internet.
- Internet facing (public endpoint) – Should be accessible from the public internet as well as your organization's internal network.

If you are using a firewall or a security group setting to prevent egress traffic from the service, make sure that the `container.repository.cloudera.com` and `docker.repository.cloudera.com` URLs on port 443 are allowed at all times.

If egress traffic is blocked to these URLs, then autoscaling cannot pull images, which can result in broken pods. For more information on required outbound access, see [Outbound network access destinations for Azure](#) and [Default security group settings on Azure](#).

Do not remove firewall rules added during provisioning. The rules are also required for regular operation. You must also maintain the minimum firewall requirements set by the cloud provider. For more information, see [Use Azure Firewall to protect Azure Kubernetes Service \(AKS\) Deployments](#) in the Microsoft Azure documentation.

### Review the default Azure service limits and your current Azure account limits

Azure portal imposes default limits to the resources available to each user subscription, which may vary for different regions. Make sure you review your Azure subscription's current usage status and resource limits before you start provisioning additional resources for CDP and CDE.

If you require more resources than the limit set by Azure, you can create a support request on your Azure Portal.

For example, To register an Azure environment in CDP, you may need to increase some of these limits for the region(s) that you are planning to use. CDP creates resources such as VMs in your Azure subscription. Depending on the number of clusters that CDP creates in your Azure subscription, you might need to raise the limits for certain resources such as VMs and vCPUs in your Azure subscription.

Related Azure quotas documentation: [Azure subscription and service limits, quotas, and constraints](#).

### Supported Azure regions

CDP supports the following Azure regions: [Supported Azure regions](#).

- A single Azure environment registered in CDP corresponds to a single VNet located in a specific region, and all the resources deployed by CDP on Azure are deployed into that VNet.
- Deploying clusters into the region containing the ADLS Gen2 containers that you want to access for input and output data, speeds up the data access. Therefore, when selecting the region to use, you should consider where your data is located.
- CDP requires that the ADLS Gen2 storage location provided during environment registration is in the same region as the region selected for the environment.

If you need to use multiple regions, you need to register multiple environments, one per region.

Related Azure documentation: [Azure geographies](#).

### Set up an Azure Cloud Credential

You must create the Azure provisioning credential for CDP prior to registering an environment. The credential allows CDP to access and provision a set of resources in your Azure account.

When working with an Azure environment, you can use the app-based credential to authenticate your Azure account and obtain authorization to create resources on your behalf. The app-based credential allows you to manually configure the service principal created within the Azure Active Directory.

Instructions: [Azure Credentials](#)

### Register an Azure Environment in CDP

Once you have met cloud provider requirements and have created the Azure provisioning credential for CDP, you may proceed to register an Azure Environment.



#### Important:

If you intend to use the Enable Private Network - create CDE with fully private Azure services (AKS, MySQL, etc.) option during environment creation, Cloudera recommends using one resource group per environment. You can accomplish this by selecting a pre-created resource group during CDP environment creation. This causes all CDP resources to be provisioned into the same resource group, allowing you to create multiple CDE services with private network enabled.

If you select creating new resource groups together with the environment, several separate resource groups are created. Since Azure restricts creating private link connections from private DNS zones in different resource groups to the same vnet, in this scenario you can only have a single active private CDE service for a given vnet.

As of CDE 1.16, private storage accounts are supported.

Instructions: [Register an Azure environment](#)

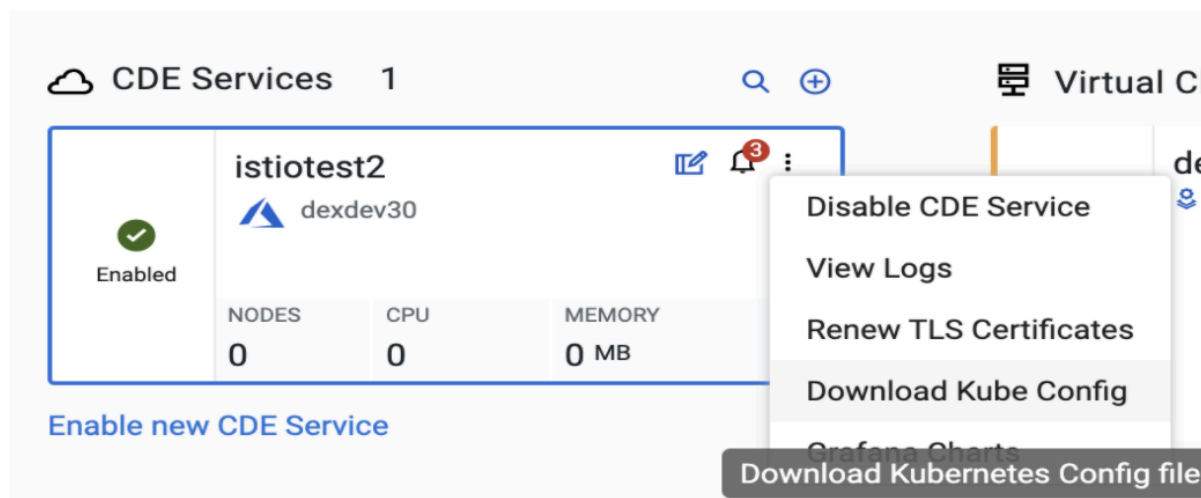
### CDE Role Requirements

There are two CDP user roles associated with the CDE service: DEAdmin and DEUser. Any CDP user with the EnvironmentAdmin (or higher) access level must assign these roles to users who require access to the Cloudera Data Engineering console within their environment.

Furthermore, if you want to allow users to log in to provisioned workspaces and run workloads on them, this will need to be configured separately.

### Set up to run kubectl

1. Go to the three dots on top right side of the CDE UI to see a dropdown menu.
2. Click on Download the Kube Config option and save it (For example, `~/.kube/cde-env1-kube-config`).



3. Run the following shell command.

```
$ export KUBECONFIG=~/.kube/cde-env1-kube-config
```

4. You should now be able to run Kubectl Commands

### Browser Requirements

Supported browsers:

- Chrome
- Safari

## Supporting Azure private storage

Cloudera Data Engineering (CDE) stores all the metadata and application files in Azure Files. By default, CDE relies on Azure Kubernetes Service (AKS) to create a public Azure Storage Account in the AKS Node Resource Group. The provisioned Storage Account will be used only for Azure Files in CDE and will be deleted once the CDE service is deleted.

### About this task

With CDE 1.16, the Azure Files used for storing CDE internal metadata can be overridden with custom private/public Azure Files.

### Creating Private Azure files

#### Before you begin

Create a Private Azure Account. For more information, see Private Endpoints for Azure Storage below. Be sure to create a Private Endpoint for Files as target sub-resource since CDE only uses Azure Files. This means that blobs, tables, queues, and so on are not used.

**Table 1: Sample creation commands**

Action	Command
Entitlement	Enable : DE_AZURE_PRIVATE_STORAGE for customer tenant Id.
Export Variables	<pre>RESOURCE_GROUP=TestCDEPrivateResourceGroup \ STORAGE_ACCOUNT_NAME=testcdeprivatestorageaccount \ PRIVATE_ENDPOINT_NAME=private-endpoint- \${STORAGE_ACCOUNT_NAME} \ VNET_NAME=cde-vnet \ VNET_RESOURCE_GROUP=TestCDEPrivateResourceGroup \ DNS_ZONE_NAME= \${STORAGE_ACCOUNT_NAME}.privatelink.file.core.windows.net \ VNET_LINK_NAME=vnet-link- \${STORAGE_ACCOUNT_NAME}</pre>
Create Private Storage Account	<pre>az storage account create --name \${STORAGE_ACCOUNT_NAME} --resource-group \${RESOURCE_GROUP} --allow-blob-public-access false --https true --public-network-access Disabled --sku Standard_LRS --location westus2</pre>
Get Storage Account ID	<pre>STORAGE_ACCOUNT_ID=\$(az storage account show --resource-group \${RESOURCE_GROUP} --name \${STORAGE_ACCOUNT_NAME}   jq -r .id)</pre>
Get Subnet ID	<pre>SUBNET_ID=\$(az network vnet subnet list -- resource-group \${VNET_RESOURCE_GROUP} --vnet- name \${VNET_NAME}   jq -r '.[0].id')</pre>
Create Private Endpoint	<pre>az network private-endpoint create -- connection-name myConnection --name \${PRIVATE_ENDPOINT_NAME} --private-connection- resource-id \${STORAGE_ACCOUNT_ID} --location westus2 --subnet \${SUBNET_ID} --resource-group \${RESOURCE_GROUP} --group-id file</pre>
Get Private Endpoint IP	<pre>PRIVATE_ENDPOINT_IP=\$(az network private- endpoint show --name \${PRIVATE_ENDPOINT_NAME} --resource-group \${RESOURCE_GROUP}   jq -r '.customDnsConfigs   .[0].ipAddresses   .[0]')</pre>
Create Private DNS Zone	<pre>az network private-dns zone create --resource- group \${RESOURCE_GROUP} --name \${DNS_ZONE_NAME}</pre>
Create Private DNS Zone record	<pre>az network private-dns record-set a add-record --resource-group \${RESOURCE_GROUP} --zone-name \${DNS_ZONE_NAME} --record-set-name @ --ipv4- address \${PRIVATE_ENDPOINT_IP}</pre>
Get VNet ID	<pre>VNET_ID=\$(az network vnet show --resource-group \${VNET_RESOURCE_GROUP} --name \${VNET_NAME}   jq -r .id)</pre>
Link DNS Zone to VNet	<pre>az network private-dns link vnet create --name \${VNET_LINK_NAME} --registration-enabled false --resource-group \${RESOURCE_GROUP} --virtual- network \${VNET_ID} --zone-name \${DNS_ZONE_NAME}</pre>

#### Enable CDE Service for Azure

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 CDE Environments column, click the plus icon at the top or the Enable new CDE link at the bottom to enable CDE for an environment.
3. Select an Azure environment.
4. Select the Override Azure Files Storage Server option.
5. Enter the Storage Account name and Resource group of the Storage Account.

6. For Azure Files FQDN, enter the Private Endpoint FQDN that you created in the prerequisites above.
7. Fill out the necessary details and click Create.

Using customer-managed key encryption

You can enable CMK encryption with CDE in Azure. Below are high-level instructions to enable CMK encryption. For detailed instructions, see [Customer-managed keys for Azure Storage encryption](#) linked below.

1. Create a key vault and grant corresponding permission to the service principle that is used to create CDP environment.
2. Associate the private storage account and key vault when enabling the CMK encryption.
3. In their job's code, set up ABFS authentication configurations and access the abfs:// files.

### Related Information

[Why are two resource groups created with AKS?](#)

[Azure Files](#)

[Private Endpoints for Azure Storage](#)

[Customer-managed keys for Azure storage encryption](#)

## Using AWS IAM restricted roles and policies for compute and CDE

AWS IAM write permissions are used by the Cloudera Data Engineering (CDE) compute infrastructure to create and delete roles and instance profiles.

Some customers may not be willing to provide IAM write permission in the role's policy. Instead, customers can set up static pre-created roles and instance profiles defined and used by the CDE compute infrastructure to provision clusters.



**Note:** The compute infrastructure is only able to use the pre-created roles and instance profile if the entitlement `LIFTIE_USE_PRECREATED_IAM_RESOURCES` is set for the tenant in use. The pre-created roles and instance profiles will continue to exist for the lifetime of the cluster.

The two main tasks for AWS IAM write permissions are the following:

1. Create roles and an instance profile.
2. Create restricted IAM policies for use by the compute infrastructure.

After the two tasks are completed, you may create a cross-account credential if needed.

See the following topics for the procedures for creating the roles and policies.

### Create IAM roles and instance profile pair

This step describes the role and instance profile pair that you will create and attach to EKS master and worker instances at runtime. This step is required in customer environments where write permissions are not provided to Cloudera Data Engineering (CDE). The roles created here are used exclusively within the customer's account. This step needs to be performed in the user's AWS console.

Use the following CloudFormation template to create the following:

- IAM role called `cdp-eks-master-role`
- IAM role and Instance Profile pair called `cdp-liftie-instance-profile`



**Important:** For CDE 1.19.4 and above, when you use a Restricted Policy, you must use an additional permission of `elasticfilesystem:TagResource` in the `efs-csi` policy for the `cdp-liftie-instance-profile` role.

- To apply the template, you need to provide values for the following parameters in the AWS console CloudFormation wizard:
  - Stack Name: Provide an appropriate name. (Example: compute-precreated-roles-and-instanceprofile)
  - TelemetryLoggingBucket: Name of the log bucket (just the name, not s3://) (Example : compute-logging-bucket)
  - TelemetryLoggingEnabled: Set it to true
  - TelemetryLoggingRootDir: Keep the default value (which is cluster-logs)
  - TelemetryKMSKeyARN: If the telemetry bucket is encrypted, give the KMS Key ARN. Default value is null.

CloudFormation > Stacks > Create stack

Step 1  
Specify template

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review

### Specify stack details

**Stack name**

Stack name

compute-precreated-roles-and-instanceprofile

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.

**CsiEnabled**  
If CSI is enabled

false

**TelemetryKmsKeyARN**  
KMS Key ARN For Telemetry logging bucket.

arn:aws:kms:us-west-2:112233445566:key/112233445566

**TelemetryLoggingBucket**  
Telemetry logging bucket where Liftie logs will be stored.

compute-logging-bucket

**TelemetryLoggingEnabled**  
Telemetry logging is enabled

true

**TelemetryLoggingRootDir**  
Telemetry logging root directory inside telemetry logging bucket used for storing logs.

cluster-logs

Cancel Previous Next

- On the last page of the wizard, select the checkbox to allow creation of IAM resources with special names. If not selected, CloudFormation prepends the provided name with random prefixes to ensure uniqueness.

**Capabilities**

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

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

The result of this procedure resembles the following:

compute-precreated-roles-and-instanceprofile						Delete	Update	Stack actions ▼	Create stack ▼
Stack info   Events   <b>Resources</b>   Outputs   Parameters   Template   Change sets									
<b>Resources (3)</b>									
<input type="text" value="Search resources"/>									
Logical ID ▲	Physical ID ▼	Type ▼	Status ▼	Status reason ▼					
AWSServiceRoleForAmazonEKS	<a href="#">cdp-eks-master-role</a>	AWS::IAM::Role	✓ CREATE_COMPLETE	-					
NodeInstanceProfile	cdp-liftie-instance-profile	AWS::IAM::InstanceProfile	✓ CREATE_COMPLETE	-					
NodeInstanceRole	<a href="#">cdp-liftie-instance-profile</a>	AWS::IAM::Role	✓ CREATE_COMPLETE	-					

Use the following CloudFormation template for this process.

CloudFormation Template (format: YAML)

```
AWSTemplateFormatVersion: "2010-09-09"

Description: "Creates Liftie IAM resources"
Parameters:

  TelemetryLoggingEnabled:
    Description: Telemetry logging is enabled
    Type: String

  TelemetryLoggingBucket:
    Description: Telemetry logging bucket where Liftie logs will be stored.
    Type: String

  TelemetryKmsKeyARN:
    Description: KMS Key ARN For Telemetry logging bucket.
    Type: String
    Default: ""

  TelemetryLoggingRootDir:
    Description: Telemetry logging root directory inside telemetry logging bucket used for storing logs.
    Default: "cluster-logs"
    Type: String

Conditions:
  TelemetryLoggingEnabled:
    Fn::Equals:
      - {Ref: TelemetryLoggingEnabled}
      - true
  KMSKeyARNForTelemetryLoggingBucketIsEmpty: !Not [!Equals [!Ref TelemetryKmsKeyARN, ""]]

Resources:

  AWSServiceRoleForAmazonEKS:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
```



```

Statement:
  - Effect: Allow
    Principal:
      Service:
        - eks.amazonaws.com
    Action:
      - sts:AssumeRole
ManagedPolicyArns:
  - arn:aws:iam::aws:policy/AmazonEKSServicePolicy
  - arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
RoleName: cdp-eks-master-role
NodeInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: "/"
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
      - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
      - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
    RoleName: cdp-liftie-instance-profile
    Policies:
      - PolicyName: 'ssm-required'
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - ssm:GetParameters
              Resource:
                - "*"
      - PolicyName: 'cde-specific-permissions'
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - cloudwatch:GetMetricData
              Resource:
                - "*"
      - PolicyName: 'cluster-autoscaler'
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - autoscaling:DescribeAutoScalingGroups
                - autoscaling:DescribeAutoScalingInstances
                - autoscaling:DescribeLaunchConfigurations
                - autoscaling:DescribeScalingActivities
                - autoscaling:DescribeTags
                - ec2:DescribeImages
                - ec2:DescribeInstanceTypes
                - ec2:DescribeLaunchTemplateVersions
                - ec2:GetInstanceTypesFromInstanceRequirements
                - eks:DescribeNodegroup

```

```

        Resource:
        - "*"
    - Effect: Allow
      Action:
        - autoscaling:SetDesiredCapacity
        - autoscaling:TerminateInstanceInAutoScalingGroup
      Resource:
        - "*"
      Condition:
        StringEquals:
          "aws:ResourceTag/k8s.io/cluster-autoscaler/enabled": "
true"
- PolicyName: ebs-csi
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Action:
          - ec2:CreateSnapshot
          - ec2:AttachVolume
          - ec2:DetachVolume
          - ec2:ModifyVolume
          - ec2:DescribeAvailabilityZones
          - ec2:DescribeInstances
          - ec2:DescribeSnapshots
          - ec2:DescribeTags
          - ec2:DescribeVolumes
          - ec2:DescribeVolumesModifications
        Resource: "*"
      - Effect: Allow
        Action:
          - ec2:CreateTags
        Resource:
          - "arn:aws:ec2:*:*:volume/*"
          - "arn:aws:ec2:*:*:snapshot/*"
        Condition:
          StringEquals:
            "ec2:CreateAction":
              - CreateVolume
              - CreateSnapshot
      - Effect: Allow
        Action:
          - ec2>DeleteTags
        Resource:
          - "arn:aws:ec2:*:*:volume/*"
          - "arn:aws:ec2:*:*:snapshot/*"
      - Effect: Allow
        Action:
          - ec2:CreateVolume
        Resource: "*"
        Condition:
          StringLike:
            "aws:RequestTag/ebs.csi.aws.com/cluster": "true"
      - Effect: Allow
        Action:
          - ec2:CreateVolume
        Resource: "*"
        Condition:
          StringLike:
            "aws:RequestTag/CSIVolumeName": "*"
      - Effect: Allow
        Action:
          - ec2:CreateVolume
        Resource: "*"

```

```

        Condition:
          StringLike:
            "aws:RequestTag/kubernetes.io/cluster/*": "owned"
- Effect: Allow
  Action:
    - ec2:DeleteVolume
  Resource: "*"
  Condition:
    StringLike:
      "ec2:ResourceTag/ebs.csi.aws.com/cluster": "true"
- Effect: Allow
  Action:
    - ec2:DeleteVolume
  Resource: "*"
  Condition:
    StringLike:
      "ec2:ResourceTag/CSIVolumeName": "*"
- Effect: Allow
  Action:
    - ec2:DeleteVolume
  Resource: "*"
  Condition:
    StringLike:
      "ec2:ResourceTag/kubernetes.io/created-for/pvc/name": "*"
- Effect: Allow
  Action:
    - ec2:DeleteSnapshot
  Resource: "*"
  Condition:
    StringLike:
      "ec2:ResourceTag/CSIVolumeSnapshotName": "*"
- Effect: Allow
  Action:
    - ec2:DeleteSnapshot
  Resource: "*"
  Condition:
    StringLike:
      "ec2:ResourceTag/ebs.csi.aws.com/cluster": "true"
- PolicyName: efs-csi
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Action:
          - elasticfilesystem:DescribeAccessPoints
          - elasticfilesystem:DescribeFileSystems
          - elasticfilesystem:DescribeMountTargets
          - elasticfilesystem:TagResource
        Resource: "*"
      - Effect: Allow
        Action:
          - elasticfilesystem:CreateAccessPoint
        Resource: "*"
        Condition:
          StringLike:
            "aws:RequestTag/efs.csi.aws.com/cluster": "true"
      - Effect: Allow
        Action:
          - elasticfilesystem:DeleteAccessPoint
        Resource: "*"
        Condition:
          StringEquals:
            "aws:ResourceTag/efs.csi.aws.com/cluster": "true"
- !If

```

```

- TelemetryLoggingEnabled
- PolicyName: telemetry-s3-list-bucket
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Action:
          - 's3:ListBucket'
        Resource:
          - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}'
          - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}/${TelemetryLoggingRootDir}/*'
      - !Ref 'AWS::NoValue'
- !If
  - TelemetryLoggingEnabled
  - PolicyName: telemetry-s3-read-write
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Action:
            - 's3:*Object'
            - 's3:AbortMultipartUpload'
          Resource:
            - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}'
            - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}/${TelemetryLoggingRootDir}/*'
        - !Ref 'AWS::NoValue'
- !If
  - KMSKeyARNForTelemetryLoggingBucketIsEmpty
  - PolicyName: s3-kms-read-write-policy
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Action:
            - 'kms:Decrypt'
            - 'kms:GenerateDataKey'
          Resource:
            - !Sub ${TelemetryKmsKeyARN}
  - !Ref 'AWS::NoValue'
- PolicyName: 'calico-cni'
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Action:
          - ec2:ModifyInstanceAttribute
        Resource:
          - "*"
        Condition:
          StringEquals:
            "ec2:Attribute": "SourceDestCheck"
NodeInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: "/"
    InstanceProfileName: cdp-liftie-instance-profile
    Roles:
      - !Ref NodeInstanceRole

```

## Create role and policy used to deploy CDP environments for CDE

The Cloudera Data Engineering (CDE) control plane requires a role and policies to create CDP environments. In this step, you create a common policy for creating environments, as well as a policy that is specific to CDE environments.

The following two policies are created in this step:

- Compute infrastructure restricted IAM policy - A common policy for all data services deployed on CDP.
- CDE restricted IAM policy - A policy with additional permissions for CDE.

There are two options for the timing of attaching the role: during environment creation, or prior to enabling the CDE data service.

### Option #1: During environment creation

The Cloudbreak environment creation UI should be set up as shown here:

[cloudera.dps.mow-dev.cloudera.com/cloud/environments/register/general/\(credential:credential/amazon/role-based\)?provider=amazon](https://cloudera.dps.mow-dev.cloudera.com/cloud/environments/register/general/(credential:credential/amazon/role-based)?provider=amazon)

Environments / Environments

#### Create Cross-account Access Policy

Copy the following JSON to create an [AWS IAM policy](#)

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

#### Create Cross-account Access Role

Use Service Manager Account ID and External ID to create an [AWS IAM role](#)

Service Manager Account ID\*

External ID\*

Cross-account Role ARN\*



Create Credential

> SHOW CLI COMMAND

**Note:**

- For the AWS IAM policy mentioned in the “Create Cross-account Access Policy” section, use the Compute infrastructure Restricted IAM and CDE Restricted IAM policies below and create as new policies in AWS IAM. There may be one or more restricted policies already attached to the cross-account role, in addition to the Compute infrastructure and CDE restricted policies. For example, there may also be a Data Hub restricted policy.
- For the “Create Cross-account Access Role” section, create the cross-account role as instructed (or update the role if one already exists) and attach the newly created Compute infrastructure Restricted IAM policy and CDE Restricted IAM policy. Finally, update the cross-account role to use it.

**Option #2: Prior to enabling CDE data service**

If the Cloudbreak environment has already been created, you can create and attach the Compute infrastructure Restricted IAM policy and CDE restricted IAM policy to the existing cross-account role associated with the environment.

To view the existing cross-account role, in the Environments section of the CDP management console, on the Summary tab, see Credentials.



**Note:** There may be one or more restricted policies already attached to the cross-account role, in addition to the Compute infrastructure and CDE restricted policies. For example, there might be a Data Hub restricted policy. These should be left in place.

**Compute (Liftie) Restricted IAM policy**

Replace the following placeholders in the JSON file:

- [YOUR-ACCOUNT-ID] with your account ID in use.
- [YOUR-IAM-ROLE-NAME] with the IAM restricted role associated with this policy.
- [YOUR-SUBNET-ARN-\*) supplied during the Cloudbreak Environment(s) creation. Note: Please provide all the subnets present in all the Cloudbreak Environment(s) that you intend to use it for the experience. If at any point a new Cloudbreak Environment is created or an existing one is updated for subnets, the same should be updated here.
- [YOUR-IDBROKER-ROLE-NAME] with the ID Broker Role name in use.
- [YOUR-LOG-ROLE-NAME] with the Log Role name in use.
- [YOUR-KMS-CUSTOMER-MANAGED-KEY-ARN] with KMS key ARN.
- [YOUR-ACCOUNT-REGION] with the AWS region.

```
{
  "Version": "2012-10-17",
  "Id": "ComputePolicy_v12",
  "Statement": [
    {
      "Sid": "SimulatePrincipalPolicy",
      "Effect": "Allow",
      "Action": [
        "iam:SimulatePrincipalPolicy"
      ],
      "Resource": [
        "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/[YOUR-IAM-ROLE-NAME]"
      ]
    },
    {
      "Sid": "RestrictedPermissionsViaClouderaRequestTag",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:CreateChangeSet",

```

```

        "ec2:createTags",
        "eks:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:RequestTag/Cloudera-Resource-Name": [
                "crn:{{ .CRNPartition }}:*"
            ]
        }
    }
},
{
    "Sid": "RestrictedPermissionsViaClouderaResourceTag",
    "Effect": "Allow",
    "Action": [
        "autoscaling:DeleteTags",
        "autoscaling:DetachInstances",
        "autoscaling:ResumeProcesses",
        "autoscaling:SetDesiredCapacity",
        "autoscaling:SuspendProcesses",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "cloudformation:DeleteChangeSet",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStacks",
        "cloudformation:CancelUpdateStack",
        "cloudformation:ContinueUpdateRollback",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:ListStacks",
        "cloudwatch:deleteAlarms",
        "cloudwatch:putMetricAlarm",
        "ec2:AttachVolume",
        "ec2:CreateNetworkInterface",
        "ec2:CreateVolume",
        "ec2:DeleteVolume",
        "ec2:RunInstances",
        "eks:DescribeUpdate",
        "eks:ListUpdates",
        "eks:UpdateClusterConfig",
        "eks:UpdateClusterVersion",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfiles",
        "iam:ListRoleTags",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:TagRole",
        "iam:UntagRole",
        "logs:DescribeLogStreams",
        "logs:FilterLogEvents"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/Cloudera-Resource-Name": [
                "crn:{{ .CRNPartition }}:*"
            ]
        }
    }
},
{

```

```

    "Sid": "RestrictedPermissionsViaCloudFormation",
    "Effect": "Allow",
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateLaunchConfiguration",
        "autoscaling:CreateOrUpdateTags",
        "autoscaling:DeleteAutoScalingGroup",
        "autoscaling:DeleteLaunchConfiguration",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeLaunchConfigurations",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeTags",
        "dynamodb:DescribeTable",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateSecurityGroup",
        "ec2:DeleteLaunchTemplate",
        "ec2:DeletePlacementGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeImages",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribePlacementGroups",
        "ec2:DescribeRegions",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVolumes",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "eks:CreateCluster",
        "eks>DeleteCluster"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "cloudformation.amazonaws.com"
            ]
        }
    }
},
{
    "Sid": "RestrictedEC2PermissionsViaClouderaResourceTag",
    "Effect": "Allow",
    "Action": [
        "ec2:RebootInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "ForAnyValue:StringLike": {
            "ec2:ResourceTag/Cloudera-Resource-Name": [
                "crn:{{ .CRNPartition }}:*"
            ]
        }
    }
}

```



```

    }
  },
  {
    "Sid": "RestrictedIamPermissionsToClouderaResources",
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/[YOUR-IDBROKER-ROLE-NAME]",
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/[YOUR-LOG-ROLE-NAME]",
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/liftie-*--eks-service-role",
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/liftie-*--eks-worker-nodes",
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/cdp-eks-master-role",
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/cdp-liftie-instance-profile"
    ]
  },
  {
    "Sid": "RestrictedKMSPermissionsUsingCustomerProvidedKey",
    "Effect": "Allow",
    "Action": [
      "kms:CreateGrant",
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*"
    ],
    "Resource": [
      "[YOUR-KMS-CUSTOMER-MANAGED-KEY-ARN]"
    ]
  },
  {
    "Sid": "AllowCreateDeleteTagsForSubnets",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags",
      "ec2:DeleteTags"
    ],
    "Resource": [
      "arn:{{ .ARNPartition }}:ec2:[YOUR-SUBNET-REGION]:[YOUR-ACCOUNT-ID]:subnet/*"
    ]
  },
  {
    "Sid": "ModifyInstanceAttribute",
    "Effect": "Allow",
    "Action": [
      "ec2:ModifyInstanceAttribute"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:Attribute": "SourceDestCheck"
      }
    }
  }
}

```

```

    },
    {
      "Sid": "OtherPermissions",
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreatePlacementGroup",
        "ec2:DeleteKeyPair",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs",
        "ec2:ImportKeyPair",
        "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
        "ec2:GetInstanceTypesFromInstanceRequirements",
        "eks:DescribeCluster",
        "elasticloadbalancing:DescribeLoadBalancers",
        "iam:GetRole",
        "iam:ListRoles",
        "iam:GetInstanceProfile"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowSsmParams",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeParameters",
        "ssm:GetParameter",
        "ssm:GetParameters",
        "ssm:GetParameterHistory",
        "ssm:GetParametersByPath"
      ],
      "Resource": [
        "arn:aws:ssm:*:*:parameter/aws/service/eks/optimized-ami/*"
      ]
    },
    {
      "Sid": "CfDeny",
      "Effect": "Deny",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "ForAnyValue:StringLike": {
          "cloudformation:ImportResourceTypes": [
            "*"
          ]
        }
      }
    },
    {
      "Sid": "ForAutoscalingLinkedRole",
      "Effect": "Allow",

```

```

    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": [
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/aws-service-ro
le/autoscaling-plans.amazonaws.com/AWSServiceRoleForAutoScalingPlans_EC2Auto
Scaling"
    ],
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "autoscaling-plans.amazonaws.com"
      }
    }
  },
  {
    "Sid": "ForEksLinkedRole",
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": [
      "arn:{{ .ARNPartition }}:iam::[YOUR-ACCOUNT-ID]:role/aws-service-
role/eks.amazonaws.com/AWSServiceRoleForEKS"
    ],
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "eks.amazonaws.com"
      }
    }
  }
]
}

```

### Supporting Customer Managed CMKs

Along with providing the KMS Customer Managed Customer Master Key (CMK) for volume encryption in the policy section with Sid: `RestrictedKMSPermissionsUsingCustomerProvidedKey`, you need to verify that the policy for the Customer Managed Customer Master Key (CMK) at KMS (this is not an IAM policy) has the following permission blocks defined for `AWSServiceRoleForAutoScaling`:

```

{
  "Sid": "AllowAutoscalingAndCDPCrossAccountRoleUseOfTheCMK",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE",
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/aws-service-role/autoscali
ng.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},

```

```

{
  "Sid": "AllowAutoscalingAndCDPCrossAccountRoleToCreateGrantsOfTheCMK",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE",
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/aws-service-role/autoscali
ng.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": "true"
    }
  }
},
{
  "Sid": "AllowCreateGrantToLiftieCluster",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/CDP-CROSSACCOUNT-ROLE"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:CalledViaFirst": "cloudformation.amazonaws.com"
    },
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "Decrypt"
      ]
    }
  }
},
{
  "Sid": "AllowEKSAccessToEBS",
  "Effect": "Allow",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "kms:CreateGrant",
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",

```

```

    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "[YOUR-ACCOUNT-ID]",
      "kms:viaService": "ec2.[YOUR-ACCOUNT-REGION].amazonaws.com"
    }
  }
}

```

After the policy is attached, the KMS service page will show the CMK as having the policy attached, similar to this screenshot:

The screenshot shows the AWS KMS console interface. On the left, there's a sidebar with 'Key Management Service (KMS)' and options for 'AWS managed keys', 'Customer managed keys' (selected), and 'Custom key stores'. The main panel shows the 'General configuration' for a specific CMK, including its Alias, ARN, Status (Enabled), Creation date, and Regionality. Below this, the 'Key policy' tab is selected, displaying a JSON policy document. The policy has two statements: one for 'Allow Autoscailing service-linked role for attachment of persistent resources' and another for 'Allow Autoscailing service-linked role use of the CMK', both granting actions like 'kms:CreateGrant', 'kms:Encrypt', 'kms:Decrypt', 'kms:ReEncrypt\*', 'kms:GenerateDataKey\*', and 'kms:DescribeKey' to the 'arn:aws:iam::[account-id]:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling'.

```

14  {
15    "Sid": "Allow Autoscailing service-linked role for attachment of persistent resources",
16    "Effect": "Allow",
17    "Principal": {
18      "AWS": "arn:aws:iam::[account-id]:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
19    },
20    "Action": "kms:CreateGrant",
21    "Resource": "*",
22    "Condition": {
23      "Bool": {
24        "kms:GrantIsForAWSResource": "true"
25      }
26    }
27  },
28  {
29    "Sid": "Allow Autoscailing service-linked role use of the CMK",
30    "Effect": "Allow",
31    "Principal": {
32      "AWS": "arn:aws:iam::[account-id]:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
33    },
34    "Action": [
35      "kms:Encrypt",
36      "kms:Decrypt",
37      "kms:ReEncrypt*",
38      "kms:GenerateDataKey*",
39      "kms:DescribeKey"
40    ],
41    "Resource": "*"
42  }
43 ]

```

### CDE restricted IAM policy

```

{
  "Id": "CDEPolicy_v2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticFileSystem",
      "Action": [
        "elasticfilesystem:CreateMountTarget",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:TagResource",
        "elasticfilesystem:ClientMount"
      ]
    }
  ],
}

```

```

    "Effect": "Allow",
    "Resource": [
        "*"
    ]
},
{
    "Sid": "ElasticFileSystemRequest",
    "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateTags"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "aws:RequestTag/Cloudera-Resource-Name": [
                "crn:cdp:de:*"
            ]
        }
    }
},
{
    "Sid": "ElasticFileSystemResource",
    "Action": [
        "elasticfilesystem:DescribeFileSystemPolicy",
        "elasticfilesystem:DeleteFileSystem",
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:DeleteMountTarget",
        "elasticfilesystem:PutFileSystemPolicy"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/Cloudera-Resource-Name": [
                "crn:cdp:de:*"
            ]
        }
    }
},
{
    "Sid": "CloudWatch",
    "Action": [
        "cloudwatch:GetMetricData"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ]
},
{
    "Sid": "ElasticLoadBalancing",
    "Action": [
        "elasticloadbalancing:DescribeTags"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ]
},

```

```

{
  "Sid": "RelationalDatabaseServiceRequest",
  "Action": [
    "rds:CreateDBCluster",
    "rds:CreateDBSubnetGroup",
    "rds:AddTagsToResource",
    "rds:CreateDBInstance"
  ],
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringLike": {
      "aws:RequestTag/Cloudera-Resource-Name": [
        "crn:cdp:de:*"
      ]
    }
  }
},
{
  "Sid": "RelationalDatabaseServiceResource",
  "Action": [
    "rds>DeleteDBSubnetGroup",
    "rds:DescribeDBInstances",
    "rds:ModifyDBInstance",
    "rds>DeleteDBInstance"
  ],
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/Cloudera-Resource-Name": [
        "crn:cdp:de:*"
      ]
    }
  }
},
{
  "Sid": "RelationalDatabaseService",
  "Action": [
    "rds:DescribeDBEngineVersions"
  ],
  "Effect": "Allow",
  "Resource": [
    "*"
  ]
}
]
}

```

### Data Hub restricted policy

To add the Data Hub restricted policy, copy the following data in the Create Cross-account Access Policy field:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [

```

```

        "ec2:DeleteTags",
        "ec2:AssociateAddress",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:AttachVolume",
        "ec2:DescribeAddresses",
        "ec2:TerminateInstances",
        "ec2:DeleteSecurityGroup"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/Cloudera-Resource-Name": [
                "crn:cdp:*"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "cloudformation:DeleteStack",
        "autoscaling:SuspendProcesses",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:ResumeProcesses",
        "autoscaling:DetachInstances",
        "autoscaling:DeleteAutoScalingGroup",
        "rds:StopDBInstance",
        "rds:StartDBInstance"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/Cloudera-Resource-Name": [
                "crn:cdp:*"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateStack",
        "cloudformation:GetTemplate",
        "ec2:CreateTags"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "aws:RequestTag/Cloudera-Resource-Name": [
                "crn:cdp:*"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DeleteVolume",
        "ec2:CreateSecurityGroup",
        "ec2:DeleteKeyPair",
        "ec2:DescribeKeyPairs",

```



```
"ec2:DescribeAvailabilityZones",
"ec2:DescribeImages",
"ec2:DeleteLaunchTemplate",
"ec2:DescribeVolumes",
"ec2:CreateVolume",
"ec2:DescribeInstances",
"ec2:DescribeRegions",
"ec2:DescribeInstanceTypeOfferings",
"ec2:DescribeInstanceTypes",
"ec2:DescribeRouteTables",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVpcAttribute",
"ec2:DescribeVpcs",
"ec2:DescribeInternetGateways",
"ec2:DescribeVpcEndpoints",
"ec2:describeAddresses",
"ec2:DescribeNatGateways",
"ec2:DescribeVpcEndpointServices",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVpcAttribute",
"ec2:CreatePlacementGroup",
"ec2:DescribePlacementGroups",
"ec2:ImportKeyPair",
"ec2:DescribeLaunchTemplates",
"ec2:CreateLaunchTemplate",
"ec2:RunInstances",
"ec2:DescribeAccountAttributes",
"sts:DecodeAuthorizationMessage",
"cloudformation:DescribeStacks",
"dynamodb:DeleteTable",
"dynamodb:DescribeTable",
"iam:ListInstanceProfiles",
"iam:ListRoles",
"dynamodb:ListTables",
"autoscaling:DescribeAutoScalingGroups",
"autoscaling:DescribeScalingActivities",
"autoscaling:CreateAutoScalingGroup",
"autoscaling:TerminateInstanceInAutoScalingGroup",
"cloudwatch:DeleteAlarms",
"cloudwatch:PutMetricAlarm",
"cloudwatch:DescribeAlarms",
"elasticloadbalancing:CreateLoadBalancer",
"elasticloadbalancing:CreateTargetGroup",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:AddTags",
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:DescribeListeners",
"elasticloadbalancing:CreateListener",
"elasticloadbalancing:DeleteListener",
"elasticloadbalancing:DeleteTargetGroup",
"elasticloadbalancing:DeleteLoadBalancer",
"elasticloadbalancing:DeregisterTargets",
"s3:GetBucketLocation",
"cloudformation:DescribeStackEvents",
"cloudformation:DescribeStackResources",
"cloudformation:DescribeStackResource",
"cloudformation:ListStackResources",
"cloudformation:UpdateStack",
"cloudformation:GetTemplate",
"iam:GetInstanceProfile",
"iam:SimulatePrincipalPolicy",
```

```

        "iam:GetRole",
        "rds:AddTagsToResource",
        "rds:CreateDBInstance",
        "rds:CreateDBSubnetGroup",
        "rds:DeleteDBInstance",
        "rds:DeleteDBSubnetGroup",
        "rds:ListTagsForResource",
        "rds:RemoveTagsFromResource",
        "rds:CreateDBParameterGroup",
        "rds:DeleteDBParameterGroup",
        "rds:DescribeEngineDefaultParameters",
        "rds:ModifyDBParameterGroup",
        "rds:DescribeDBParameters",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeDBInstances",
        "rds:ModifyDBInstance",
        "rds:DescribeCertificates",
        "kms:ListKeys",
        "kms:ListAliases",
        "ec2:ModifyInstanceAttribute",
        "ec2:CreateLaunchTemplateVersion"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::[YOUR-ACCOUNT-ID]:role/[YOUR-IDBROKER-ROLE-NAME]"
    ]
  },
  {
    "Sid": "IdentityAccessManagementLimited",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:iam::*:role/aws-service-role/*"
    ]
  }
]
}

```