Cloudera Data Flow

AWS Resource Panning

Date published: 2021-04-06
Date modified: 2025-09-30

https://docs.cloudera.com/

Legal Notice

Contents

AWS requirements for Cloudera Data Flow

As the administrator for your AWS environment, ensure that the environment meets the requirements for Cloudera on cloud and Cloudera Data Flow. Then set up your AWS cloud credential and register the environment.

Follow the steps to ensure that your AWS environment meets the Cloudera and Cloudera Data Flow requirements:

Understand your AWS account requirements for Cloudera

- Review the *Cloudera AWS account requirements*. The link is in the *Related information* section below.
- Verify that your AWS account for Cloudera has the required resources.
- Verify that you have the permissions to manage these resources.

Understand the Cloudera Data Flow requirements

- Verify that the following services are available in your environment for Cloudera Data Flow to use:
  - Network – Amazon VPC
  - Compute – Amazon Elastic Kubernetes Service (EKS)
  - Load Balancing – Amazon ELB Classic Load Balancer
  - Persistent Instance Storage – Amazon Elastic Block Store (EBS)
  - Database – Amazon Relational Database Service (RDS)
- Determine your networking option:
  - Use your own VPC
  - Allow Cloudera to create a VPC

  To understand each option, see: *Cloudera Data Flow Networking*. The link is in the *Related information* section below.
- Regions:
  - Select a Cloudera on cloud-supported region that also includes the AWS Elastic Kubernetes Service (EKS).

    For more information, see: *Cloudera Supported AWS regions* and the Region Table in *AWS Regional Services*. The links are in the *Related information* section below.
- Ports and outbound network access:
  - Review the port requirements for the Cloudera default security group. See: *Cloudera Management Console - Security groups*. The link is in the *Related information* section below.
  - Configure ports for NiFi to access your source and destination systems in the data flow.
  - If you are using a firewall or a security group setting to prevent egress from the workspace, you must ensure that the outbound destinations required by Cloudera Data Flow are reachable. For more information, see *Outbound network access destinations for AWS*. The link is in the *Related information* section below.
  - If the egress is blocked to these URLs, then autoscaling fails to pull new images and the instances will have broken pods.

    Follow the recommended and minimum required security group settings by AWS. For more information, see *Amazon EKS security group considerations*. The link is in the *Related information* section below.

Set up an AWS Cloud credential

Create a role-based AWS credential that allows Cloudera on cloud 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 Cloudera.

To set up an AWS Cloud credential, see *Creating a role based provisioning credential for AWS*. The link is in the *Related information* section below.

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

Register an AWS environment in Cloudera on cloud

A Cloudera user must have the PowerUser 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.

To register an AWS environment in Cloudera on cloud, see *Cloudera AWS Environments*.

Related Concepts

Cloudera Data Flow networking in AWS

Related Information

Cloudera on cloud
Cloudera on cloud supported AWS regions
AWS Regional Services
Cloudera Management Console
Amazon EKS security group considerations
Creating a role based provisioning credential for AWS
Cloudera on cloud
Outbound network access destinations for AWS

Cloudera Data Flow networking in AWS

Cloudera Data Flow supports different networking options depending on how you have set up your VPC and subnets. If you want Cloudera Data Flow to use specific subnets, make sure that you specify them when registering a Cloudera environment.

If you specified a mix of public and private subnets during environment registration, Cloudera Data Flow by default will provision the Kubernetes nodes in the private subnets. For Cloudera Data Flow to work, the private subnets require outbound internet access. This can be achieved by configuring NAT gateways in separate public subnets and making sure that outbound internet traffic is routed via the NAT gateway. The VPC you are using must have an Internet Gateway set up which ultimately provides internet access to the public subnets. Following this approach allows the Cloudera Data Flow services running on Kubernetes nodes in your private network to connect to the internet while also preventing inbound connections from the internet.

You can configure Cloudera Data Flow to either use a private or public load balancer to allow users to connect to flow deployments. Using a private load balancer is possible when your Cloudera environment contains at least two private subnets. When you are using a private load balancer, you need to ensure connectivity between the client network from where your users are initiating connections and the private subnets in your VPC. This is typically done by setting up VPN access between the private subnets in AWS and the corporate network.

If you want to allow users to connect to flow deployments from the internet you can use the public load balancer option. This option will provision public load balancers in public subnets allowing your users to connect to flow deployments without the need to set up VPN connectivity between the private subnets and your corporate network.

Important: Cloudera recommends that you either use a fully private deployment in private subnets with private load balancers or a mix of private subnets with a public load balancer. Cloudera does not recommend provisioning Cloudera Data Flow in public subnets.

The image below represents a fully private deployment where Kubernetes nodes and load balancers are deployed in the private subnets.

Use your own VPC

If you choose to use your own VPC, verify that it meets the minimum requirements and review Cloudera's recommended setup.

VPCs can be created and managed from the *VPC console on AWS*. For instructions on how to create a new VPC on AWS, refer to *Create and configure your VPC* in the AWS documentation.

Verify that your VPC meets the following requirements and recommendations:

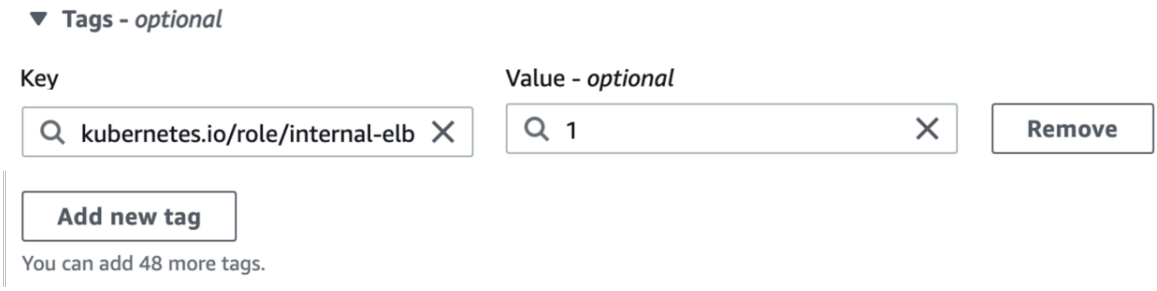
Minimum requirements

- Cloudera Data Flow requires at least two subnets, each in a different Availability Zone (AZ). If you require a public endpoint for Cloudera Data Flow, provision at least one public subnet.
- Ensure that the CIDR block for the subnets is sized appropriately for each Cloudera Data Flow environment. You must have enough IPs to accommodate:
  - The maximum number of autoscaling compute instances.
  - A fixed overhead of 48 IP addresses for three instances for core Cloudera Data Flow services.
- You must enable DNS for the VPC.

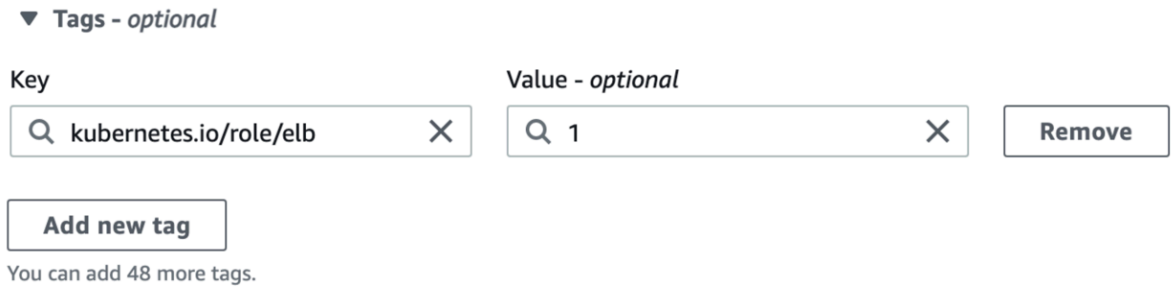
Cloudera's recommended setup

- Provision two subnets, each in a different Availability Zone (AZ).
  - If you do not require a public endpoint, use two private subnets.
  - If you require a public endpoint, use one private subnet and one public subnet.
- Private subnets should have routable IPs over your internal VPN. If IPs are not routable, private Cloudera Data Flow 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. Also, for load balancers to be able to choose the subnets correctly, you must tag either the private or public subnets.

  A tag in AWS consists of a key and a value.
  - To tag private subnets, enter kubernetes.io/role/internal-elb for the key and 1 for the value.

  - To tag public subnets, enter kubernetes.io/role/elb for the key and 1 for the value.

Note: The load balancer must be on a public subnet for access to Cloudera Data Flow. By default, if they are available, Cloudera Data Flow will configure the EKS to run on private subnets.

Related Information

VPC Console on AWS

Create and configure your VPC

Allow Cloudera to create a VPC

You can choose to create a VPC through Cloudera.

If you choose to create a VPC through Cloudera, three subnets will be automatically created.

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.

For more information, see the AWS documentation *Amazon EKS - Cluster VPC Considerations* and *Creating a VPC for your Amazon EKS Cluster*. The links are in the *Related information* section below.

Related Information

Amazon EKS - Cluster VPC Considerations

Creating a VPC for your Amazon EKS Cluster

Limitations on AWS

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 Cloudera and Cloudera Data Flow.

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 your proceed.

For more information, see the AWS documentation: *AWS Service Limits* and *Amazon EC2 Resource Limits*.

Cloudera Data Flow environments have the following resource limits on AWS:

- Certificate creation (for TLS) uses LetsEncrypt which is limited to 2000 certs/week. As such, a single tenant in Cloudera can create a maximum of 2000 flows per week.

Related Information

AWS Service Limits

Amazon EC2 Resource Limits

ENI Max Pods

AWS restricted policies

Customers with strict security policies beyond what the default Cloudera cross-account policy permits can enable Cloudera Data Flow for a Cloudera environment with more restricted IAM policies. To do so, an administrator must attach the Compute Restricted IAM policy with the cross-account role associated with the Cloudera environment.

Cloudera Data Flow uses AWS IAM write permissions to create/delete Roles and Instance Profiles. If due to security requirements you cannot provide IAM write permission in the role’s policy, you can set up static pre-created roles and an instance profile. Cloudera Data Flow makes use of these static pre-created roles and instance-profile while provisioning the cluster.

Note: Cloudera Data Flow will only be able to use the pre-created Roles and Instance Profile if the entitlement *LIFTIE_USE_PRECREATED_IAM_RESOURCES* for the tenant in use is set.

To enable Cloudera Data Flow with restricted IAM policies, perform the following tasks:

1. Create the IAM Roles and Instance Profile pair.
2. Create the restricted policies and attach them to the Cloudera cross-account role

Create IAM roles and instance profile

Enable the *LIFTIE_USE_PRECREATED_IAM_RESOURCES* entitlement and then create the IAM roles and instance profile.

Before you begin

Confirm that the *LIFTIE_USE_PRECREATED_IAM_RESOURCES* entitlement is enabled for the tenant in consideration.

Procedure

1. Apply 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

Figure 1: CloudFormation Template (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 store
d.
    Type: String
  TelemetryKmsKeyARN:
    Description: KMS Key ARN For Telemetry logging bucket.
    Type: String
    Default: ""
  TelemetryLoggingRootDir:
    Description: Telemetry logging root directory inside telemetry logg
ing bucket used for storing logs.
    Default: "cluster-logs"
    Type: String
Conditions:
  TelemetryLoggingEnabled:
    Fn::Equals:
      - {Ref: TelemetryLoggingEnabled}
      - true
  KMSKeyARNForTelemetryLoggingBucketIsEmpty: !Not [!Equals [!Ref Telemetry
KmsKeyARN, ""]]
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: cluster-autoscaler
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - autoscaling:DescribeAutoScalingGroups
                  - autoscaling:DescribeAutoScalingInstances
                  - autoscaling:DescribeTags
                  - autoscaling:DescribeLaunchConfigurations
                  - autoscaling:SetDesiredCapacity
                  - autoscaling:TerminateInstanceInAutoScalingGroup
                  - ec2:DescribeLaunchTemplateVersions
                Resource:
                  - "*"
        - 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
```

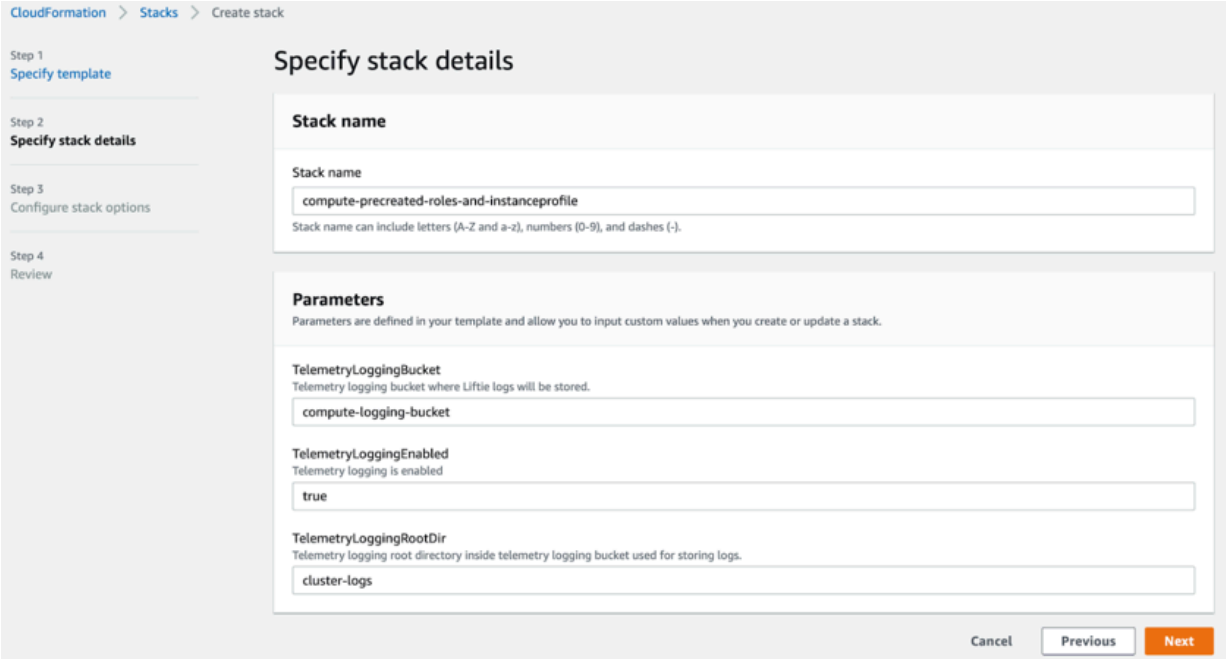
```
                  Resource:
                    - "arn:aws:ec2:*:*:volume/*"
                    - "arn:aws:ec2:*:*:snapshot/*"
                  Condition:
                    StringEquals:
                      "ec2:CreateAction":
                        - CreateVolume
                        - 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
                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}/${Te
lemetryLoggingRootDir}/*'
          - !Ref 'AWS::NoValue'
        - !If
          - TelemetryLoggingEnabled
          - PolicyName: telemetry-s3-read-write
            PolicyDocument:
              Version: 2012-10-17
              Statement:
                - Effect: Allow
                  Action:
                    - s3:*Object
                    - s3:AbortMultipartUpload
                    - s3:GetBucketAcl
                  Resource:
                    - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}'
                    - !Sub 'arn:aws:s3:::${TelemetryLoggingBucket}/${Telem
etryLoggingRootDir}/*'
          - !Ref 'AWS::NoValue'
        - !If
          - KMSKeyARNForTelemetryLoggingBucketIsEmpty
          - PolicyName: s3-kms-read-write-policy
```

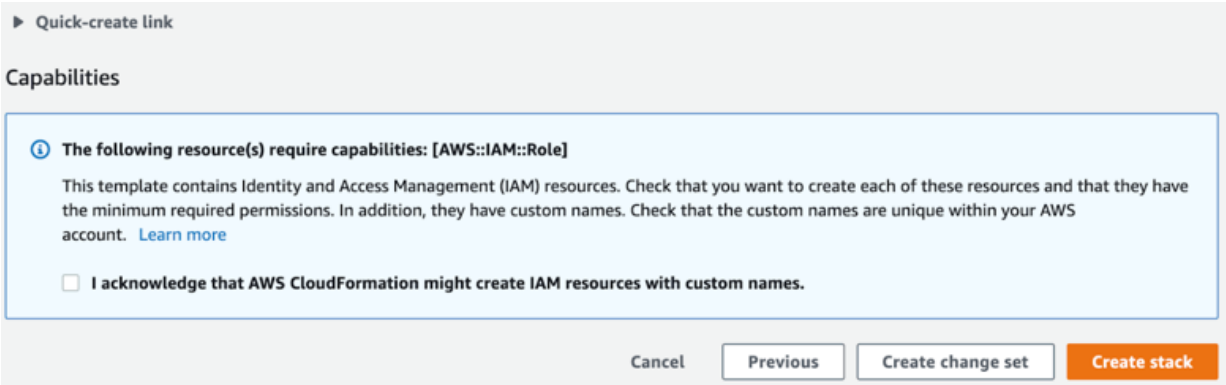
```
            PolicyDocument:
              Version: 2012-10-17
              Statement:
                - Effect: Allow
                  Action:
                    - kms:Decrypt
                    - kms:GenerateDataKey
                  Resource:
                    - !Sub ${TelemetryKmsKeyARN}
        - !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
```

2. In the AWS console Cloudformation wizard, provide values for the following properties:

   - Stack Name: Provide an appropriate name. Example: compute-precreated-roles-and-instanceprofile)
   - TelemetryLoggingBucket: Name of the log bucket. Example: compute-logging-bucket
   - TelemetryLoggingEnabled: Set it to true.
   - TelemetryLoggingRootDir: Verify that it is set to the default value cluster-logs.
   - TelemetryKMSKeyARN: If the telemetry bucket is encrypted, specify the KMS Key ARN. The default value is null.



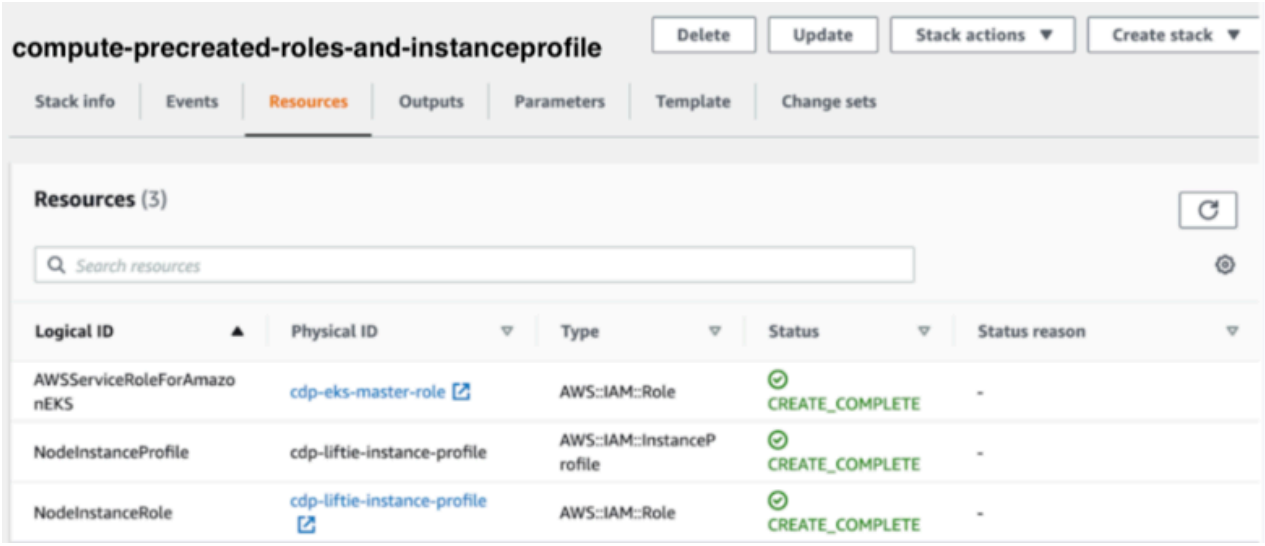
3. On the last page in the wizard process, click the I acknowledge... checkbox to allow creation of IAM resources with special names.



4. Click Create stack.

Results

On the Cloudformation **Resources** tab, you find the precreated role and instance profile.



What to do next

Update the environment role to use the restricted role and policy.

Create the restricted policies and attach them to the Cloudera cross-account role

Update the environment role to use the Cloudera Data Hub and Compute restricted policies. You can do this during the environment-creation process or before you enable the environment.

About this task

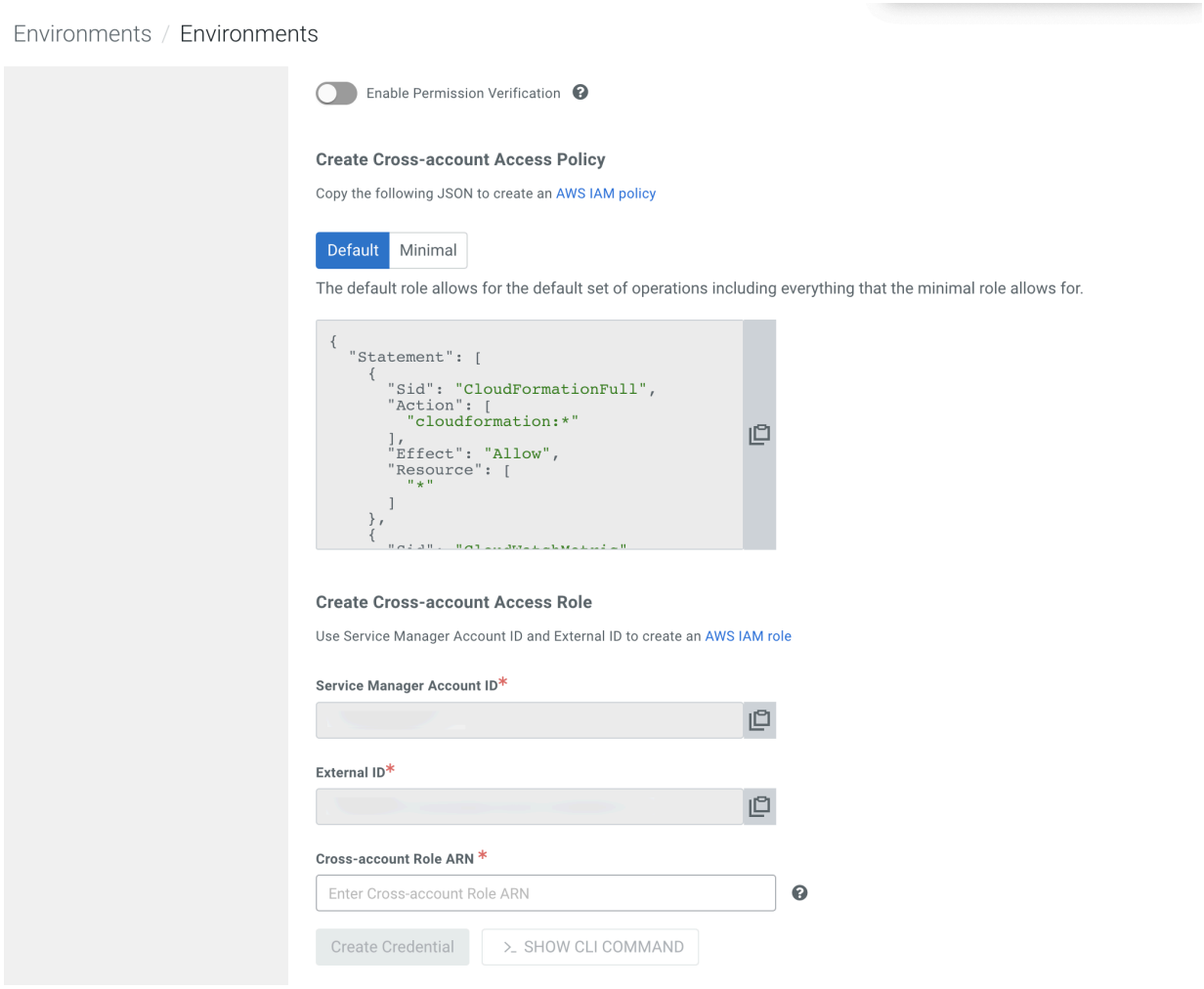
To enable the Cloudera Data Flow experience after the environment has been created, an Administrator needs to attach the Compute Restricted IAM policy and the Cloudera Data Hub restricted policy with the Cloudera cross-account role associated with the environment.

To use Cloudera Data Flow with the most restrictive options available, you need to create a custom Customer Managed Key (CMK) defining it on Environment level. The Compute Restricted IAM policy provided here requires

the existence of such a key. If you want to use Cloudera Data Flow with less restrictive options, you can modify the policy in a way that allows Cloudera to create and manage the key for you.

Procedure

1. Go to the **Environments** page.



2. In the Create Cross-account Access Policy field, attach the Compute 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 Cloudera Environment(s) creation.

     Note: Provide all the subnets present in all the Cloudera Environment(s) that you intend to use it for the experience. If at any point a new Cloudera Environment is created or an existing one is updated for subnets, provide it 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.

```
{
  "Version": "2012-10-17",
  "Id": "ComputePolicy_v12",
  "Statement": [
    {
```

```
      "Sid": "SimulatePrincipalPolicy",
      "Effect": "Allow",
      "Action": [
        "iam:SimulatePrincipalPolicy"
      ],
      "Resource": [
        "arn:{{ .ARNPartition }}:iam::[***YOUR-ACCOUNT-ID***]:rol
e/[***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***]:rol
e/[***YOUR-IDBROKER-ROLE-NAME***]",
        "arn:{{ .ARNPartition }}:iam::[***YOUR-ACCOUNT-ID***]:rol
e/[***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-ek
s-master-role",
        "arn:{{ .ARNPartition }}:iam::[***YOUR-ACCOUNT-ID***]:role/cdp-li
ftie-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",
        "eks:CreateAccessEntry",
        "eks:DeleteAccessEntry",
        "eks:ListAccessEntries",
        "eks:DescribeAccessEntry",
        "eks:AssociateAccessPolicy",
        "eks:DisassociateAccessPolicy",
        "eks:ListAssociatedAccessPolicies",
        "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-se
rvice-role/autoscaling-plans.amazonaws.com/AWSServiceRoleForAutoScalingP
lans_EC2AutoScaling"
      ],
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "autoscaling-plans.amazonaws.com"
        }
      }
    },
    {
      "Sid": "ForEksLinkedRole",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:{{ .ARNPartition }}:iam::[***YOUR-ACCOUNT-ID***]:role/aws-se
rvice-role/eks.amazonaws.com/AWSServiceRoleForEKS"
      ],
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "eks.amazonaws.com"
        }
      }
    }
  ]
}
```

3. Depending on whether you created a custom Customer Managed Key (CMK) to be used for EBS encryption or you want Cloudera to generate and manage the CMK for you, select one of the following options:
   - Provide own CMK - if you want to use your own custom CMK
   - Let Cloudera generate CMK - if you want to allow Cloudera to generate and manage your CMK

   For Provide own CMK

   a. If you have not already created it, create a custom CMK. Verify that the policy (this is different from the IAM policy) for CMK at KMS has the required additional permissions blocks defined.

      For more information, see Using CMKs with Cloudera Data Flow.

   b. Provide the KMS CMK for volume encryption in the policy section with Sid: RestrictedKMSPermissions UsingCustomerProvidedKey.

   For Let CDP generate CMK

   Replace the RestrictedKMSPermissionsUsingCustomerProvidedKey policy section with the following:

```
{
 "Sid": "AllCreateAndManageKMS",
 "Effect": "Allow",
 "Action": [
   "kms:Create*",
   "kms:Describe*",
   "kms:Enable*",
   "kms:List*",
   "kms:Put*",
   "kms:Update*",
   "kms:Revoke*",
   "kms:Disable*",
   "kms:Get*",
   "kms:Delete*",
   "kms:ScheduleKeyDeletion",
   "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

4. In the Create Cross-account Access Role section, associate the cross-account access role with the Compute Restricted IAM policy.
5. Click Create Credential.
6. Repeat the steps to add the Data Hub restricted policy.

   Copy the following Cloudera Data Hub restricted policy in the Create Cross-account Access Policy field:

   Replace the following placeholders in the JSON file:

   - *[YOUR-ACCOUNT-ID]* with your account ID in use.
   - *[YOUR-IDBROKER-ROLE-NAME]* with your IDBroker role name.

```
{
    "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/*"
            ]
        }
    ]
}
```

Using Customer Managed Keys with Cloudera Data Flow

By default, Cloudera Data Flow uses your account level KMS key for EBS storage encryption. You can optionally secure your data with a custom KMS key.

You have two options to implement Customer Managed Keys (CMKs):

- define a new default KMS key for EBS encryption on AWS account level
- define a key on Cloudera Data Flow environment level

Define a new default KMS key for AWS account level EBS encryption

When you define a new account level default key in AWS, you need to add policies to your key definition that allow for storage provisioning and fulfilling scaling requests.

About this task

Important: Defining a new default key affects all EBS storage encryption within your account.

Procedure

1. Create a custom encryption key on the AWS Management Console.

   The key policy section of the new key must contain additional permissions. Add the three required permission blocks in the example below.

   Replace *[***YOUR ACCOUNT ID***]* and *[***YOUR ACCOUNT REGION***]* with your AWS account ID and with the AWS region where you want to deploy Cloudera Data Flow, respectively.

```
       {
            "Sid": "AllowAutoscalingServiceLinkedRoleForAttachmentOfPer
sistentResources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[***YOUR ACCOUNT ID***]:role/aws-se
rvice-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Sid": "AllowAutoscalingServiceLinkedRoleUseOfTheCMK",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[***YOUR ACCOUNT ID***]:role/aws-se
rvice-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow EKS access to EBS.",
            "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***].ama
zonaws.com"
                }
            }
        }
```

Important: If you fail to add these permissions, you will encounter failures when enabling Cloudera Data Flow since it will be unable to provision the necessary encrypted storage using the custom key.

2. Set the newly created key as the default KMS key for EBS encryption.

   For more information, see Default KMS key for EBS encryption.

3. If you are also use restricted IAM policies with Cloudera, make sure you provide the KMS CMK for volume encryption when you Create the restricted policies and attach them to the cross-account role .

Define a new default KMS key for Cloudera environment level EBS encryption

When you define a custom KMS key at the Cloudera environment level, you need to add policies to your key definition that allow for storage provisioning and fulfilling scaling requests.

Procedure

1. Create a custom encryption key on the AWS Management Console.

   The key policy section of the new key must contain additional permissions. Add the three required permission blocks in the example below.

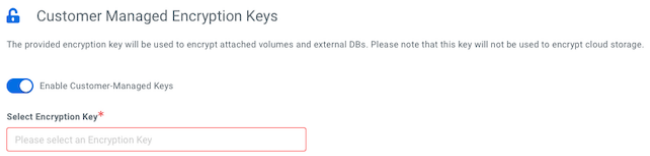
   Replace *[***YOUR ACCOUNT ID***]* and *[***YOUR ACCOUNT REGION***]* with your AWS account ID and with the AWS region where you want to deploy Cloudera Data Flow, respectively.

```
        {
            "Sid": "AllowAutoscalingServiceLinkedRoleForAttachmentOfPer
sistentResources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[***YOUR ACCOUNT ID***]:role/aws-se
rvice-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Sid": "AllowAutoscalingServiceLinkedRoleUseOfTheCMK",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[***YOUR ACCOUNT ID***]:role/aws-se
rvice-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
```

```
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow EKS access to EBS.",
            "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***].ama
zonaws.com"
                }
            }
        }
```

Important: If you fail to add these permissions, you will encounter failures when enabling Cloudera Data Flow since it will be unable to provision the necessary encrypted storage using the custom key.

2. When registering your Cloudera environment, follow these steps on the Region, Networking and Security page to assign the custom key:



   a. Under Customer-Managed Keys, click Enable Customer-Managed Keys.
   b. Select the CMK you want to enable for this environment from the Select Encryption Key drop-down list.

   For more information on registering a Cloudera environment, see Register an AWS environment from Cloudera UI.

3. If you are also using restricted IAM policies with Cloudera, make sure you provide the KMS CMK for volume encryption when you Create the restricted policies and attach them to the cross-account role .