

Service upgrade

Date published: 2021-04-06

Date modified: 2024-06-03

CLOUDERA

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

Pre-upgrade information.....	4
Update cdp-liftie-instance-profile.....	4
Upgrade a Cloudera DataFlow service.....	7
Troubleshooting Cloudera DataFlow upgrade errors.....	9

Pre-upgrade information

Learn about tasks you need to perform and information you need to consider before starting a Cloudera DataFlow service upgrade.



Important: If you are upgrading on AWS to Cloudera DataFlow 2.3.0 or higher from release 2.2.0 or lower and you use a pre-created IAM policy, you need to update the `cdp-liftie-instance-profile` before the upgrade. For more information, see [Update cdp-liftie-instance-profile](#).



Note: In Cloudera DataFlow versions 2.4.0 and lower, ZooKeeper may lose component state information under certain conditions due to its previous configuration to use ephemeral storage. If a ZooKeeper pod fails, it retains component state information as long as it is rescheduled to the same Kubernetes (K8s) node, as it can reattach to storage already present there. If the ZooKeeper pod gets rescheduled to a different K8s node, existing storage is not moved with it, therefore state information is lost. If your use case requires that component state information is retained even during an upgrade, contact Cloudera Support to provide you with a patch that you need to apply to every deployment before starting the upgrade.

- You trigger an upgrade for a selected Cloudera DataFlow environment. Upgrades move from an older version to the latest version available.
- Once you trigger the upgrade, the service enters UPGRADING state and is upgraded to the latest supported Cloudera DataFlow and Kubernetes versions. Kubernetes upgrade can take up to an hour to complete. During the upgrade service actions are restricted, that is, you are not able to create or manage deployments in the given Cloudera DataFlow service.
- As part of a Cloudera DataFlow Service (Environment) upgrade, all existing deployments are also upgraded.
- To keep existing Cloudera DataFlow deployments working, you may choose not to upgrade the NiFi version of existing deployments during the upgrade. This selection applies to all existing deployments.



Note: The option to skip NiFi version upgrade is not always available: NiFi version upgrade is mandatory if the upgrade wizard determines that the new workload version is not compatible with the current NiFi version running in any deployment. The same automatism prevents you from accidentally downgrading NiFi below the minimum supported version later.

You can manually upgrade NiFi for individual deployments later.

- If upgrading a deployment fails, all components are rolled back to their original state and it causes the overall Cloudera DataFlow service upgrade to be rolled back as well.

Update cdp-liftie-instance-profile

When upgrading to Cloudera DataFlow 2.3.0 or higher from release 2.2.0-h4 or lower on AWS and you use a pre-created IAM policy, you need to update the `cdp-liftie-instance-profile` before the upgrade. Without these policy changes, volumes provisioned by a pre-2.3.0 Cloudera DataFlow version will be left behind if the corresponding deployment is terminated or the data flow is disabled after the upgrade.

About this task

These updates are necessary because Cloudera DataFlow has migrated to use EKS k8s 1.23 and CSI EBS provisioner. This storage provisioner requires updated policies to allow the creation and deletion of volumes from within the EKS cluster.

Update the `cdp-liftie-instance-profile` using either the AWS UI or CLI.

Procedure

1. Add the ebs-csi policy to the cdp-liftie-instance-profile IAM profile.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateSnapshot",
        "ec2:AttachVolume",
        "ec2:DetachVolume",
        "ec2:ModifyVolume",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInstances",
        "ec2:DescribeSnapshots",
        "ec2:DescribeTags",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumesModifications"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": [
            "CreateVolume",
            "CreateSnapshot"
          ]
        }
      },
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:snapshot/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2>DeleteTags"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:snapshot/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringLike": {
          "aws:RequestTag/ebs.csi.aws.com/cluster": "true"
        }
      },
      "Action": [
        "ec2:CreateVolume"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

```

        "Condition": {
            "StringLike": {
                "aws:RequestTag/CSIVolumeName": "*"
            }
        },
        "Action": [
            "ec2:CreateVolume"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/ebs.csi.aws.com/cluster": "true"
            }
        },
        "Action": [
            "ec2:DeleteVolume"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/CSIVolumeName": "*"
            }
        },
        "Action": [
            "ec2:DeleteVolume"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/kubernetes.io/created-for/pvc/name":
**
            }
        },
        "Action": [
            "ec2:DeleteVolume"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/CSIVolumeSnapshotName": "*"
            }
        },
        "Action": [
            "ec2:DeleteSnapshot"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/ebs.csi.aws.com/cluster": "true"
            }
        },

```

```

    },
    "Action": [
      "ec2:DeleteSnapshot"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }
]
}

```

2. Add the efs-csi policy to the cdp-liftie-instance-profile IAM profile.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticfilesystem:DescribeAccessPoints",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringLike": {
          "aws:RequestTag/efs.csi.aws.com/cluster": "true"
        }
      },
      "Action": [
        "elasticfilesystem:CreateAccessPoint"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/efs.csi.aws.com/cluster": "true"
        }
      },
      "Action": [
        "elasticfilesystem:DeleteAccessPoint"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

Upgrade a Cloudera DataFlow service

You can upgrade your Cloudera DataFlow service together with all DataFlow deployments in that environment with the push of a button.

Procedure

1. In Cloudera DataFlow, select Environments from the left navigation pane.


2. Select the environment you want to upgrade and click Manage DataFlow.
3. Click Actions Upgrade DataFlow .




Note: If the upgrade wizard determines that the NiFi version of your Cloudera DataFlow deployments is compatible with the Cloudera DataFlow service after the upgrade, you have the option to skip upgrading NiFi in current deployments. In this case you can upgrade NiFi manually later.

Upgrade DataFlow ✕

i DataFlow will be upgraded to the latest supported DataFlow and Kubernetes versions. The upgrade requires performing a Kubernetes upgrade. Kubernetes upgrade can take up to an hour to complete. During the upgrade you won't be able to create or manage deployments in this DataFlow service.

INSTANCE TYPE	NAME	REGION
 Standard_D16s_v3		West US 2

Current Version in Use	Version After Upgrade
CDF 2.0.0-b294	→ CDF 2.1.0-b117
Kubernetes 1.21	→ Kubernetes 1.22

☐ Preserve current deployment NiFi versions 

☐ Skip Validation **i**

Cancel Upgrade

4. Click Upgrade.

The environment status changes to Upgrading and you see a similar message under Event History:

i DataFlow Upgrade Initiated

2022-06-22 11:32 CEST

⤴

ALERT DETAILS:

Initiated dataflow upgrade on cluster with ID liftie-l2623bk9.























DURATION:

1 second

5. You can monitor the upgrade procedure in Event History. Click Manage DataFlow then select Alerts.

Event History [?](#)

SHOW ONLY: ☒ Info ☐ Warning ☐ Error

 DataFlow Successfully Upgraded	2022-06-27 19:43 CEST	
 DataFlow Post Upgrade Cleanup Initiated	2022-06-27 19:41 CEST	
 NiFi Deployments Upgraded	2022-06-27 19:41 CEST	
 Upgrading NiFi Deployments	2022-06-27 19:30 CEST	
 NiFi Dependencies Upgraded	2022-06-27 19:30 CEST	
 Upgrading NiFi Dependencies	2022-06-27 19:25 CEST	
 FluxCD Helm Operators Are Upgraded	2022-06-27 19:25 CEST	
 Upgrading FluxCD Helm Operators	2022-06-27 19:25 CEST	
 Kubernetes Version Upgraded	2022-06-27 19:25 CEST	
 Upgrading Kubernetes Version	2022-06-27 18:58 CEST	
 DataFlow Upgrade Initiated	2022-06-27 18:57 CEST	

Results

The message DataFlow Successfully Upgraded in **Event History** and the Status of the service changing from Upgrading to Good Health on the **Environments** page signal that the upgrade has ended. Depending on the number of running nodes in the cluster, the upgrade can take more than an hour to complete.

What to do next

If you had restricted policies in place before the upgrade, run the

```
aws ec2 describe-volumes --profile <aws-profile> --query
    'Volumes[?Tags[?contains(Key,`kubernetes.io/cluster/liftie-`)] &&
    State==`available`].[VolumeId,State,Tags[?contains(Key,`k
    ubernetes.io/cluster/liftie-`)].Key][0]'
```

command to confirm that no unused volumes were left behind.

Troubleshooting Cloudera DataFlow upgrade errors

Learn how to recognize and address common errors with your Cloudera DataFlow service and flow deployment upgrades.

Setting up `kubectl` to connect to the Kubernetes cluster

It is helpful to have access to the Kubernetes cluster using command line tools such as `kubectl` when you are troubleshooting deployment or upgrade failures. To set up `kubectl` access, follow these steps:

1. In , from the Environments page, select the service for which you want to add or remove user access.
2. Click Manage DataFlow.
3. From the Actions menu, click Manage Kubernetes API Server User Access.
4. Add the AWS IAM role that you will authenticate as to the list of authorized users for DataFlow by entering the ARN in the Add User dialog.

5. Use the Download Kubeconfig action to retrieve the kubeconfig file for connecting to your cluster.
6. Set up your `kubectl` to use the downloaded kubeconfig file:

```
export KUBECONFIG=[ ***PATH/TO/DOWNLOADED/KUBECONFIG/FILE*** ]
```

7. Run `kubectl get ns` and validate that your output looks similar to:

NAME	STATUS	AGE
cadence	Active	37h
cert-manager	Active	37h
cfm-operator-system	Active	37h
default	Active	38h
dfx-dev-environment-ns	Active	36h
dfx-idbrokers3-ns	Active	36h
dfx-kafkatos3-ns	Active	22h
dfx-local	Active	37h
dfx-ops	Active	37h
kube-node-lease	Active	38h
kube-public	Active	38h
kube-system	Active	38h
liftie	Active	37h
logging	Active	37h
monitoring	Active	37h
nfs-provisioner-system	Active	37h
nginx-ingress	Active	37h
prometheus-operator-system	Active	37h

With `kubectl` being set up correctly, you are able to access NiFi and other logs directly through the CLI.

Understanding upgrade failures

There are various reasons that may prevent an upgrade from happening. This document covers the following scenarios:

- The upgrade fails upon initiation.
- The upgrade starts but fails, and rollback is possible.
- The upgrade starts but fails and rollback is not possible.

Upgrade fails on initiation

Symptom: When the upgrade fails to start, the status of the environment does not change to Upgrading and you receive an error message stating the cause of the failure.

Possible causes: The service is in one of the following states:

- The service is in one of the following states:
 - Failed to Update
 - Failed to Update NiFi version
 - Any transitional / in-progress state
- Any of the deployments is in one of the following states:
 - New
 - Failed to Update
 - Failed to Update NiFi version
 - Any transitional / in progress state

How to fix it: Fix the error indicated in the error message, then retry the upgrade. If the upgrade still fails, contact Cloudera Support.

Upgrade fails and rollback is possible

Symptom: Upgrade rollback process starts. If the rollback is successful, the **Status** of the service returns to **Good Health**, Cloudera DataFlow and NiFi versions return to their pre-upgrade values. If the rollback fails, the **Status** of both the Cloudera DataFlow service and running deployments becomes **Failed to Upgrade**. The order in which this transition happens depends on where the rollback actually failed.

How to fix it: In the event of an upgrade failure, Cloudera DataFlow determines if rollback is possible and automatically initiates it. No user intervention is required.

If the rollback is successful:

1. After the rollback ends, from the Environments page, select the Cloudera DataFlow service you were trying to upgrade and click the Alerts tab. Check Event History to see if there is anything obvious in there or something that narrows down the possible cause of the failure.
2. From the Deployments page select running deployments that failed to upgrade and check Alerts>Event History to see if there is anything obvious in there or something that narrows down the possible cause of the failure.
3. Check the K8s cluster using CLI and try to fix errors based on the information you obtained from Event History in the previous steps.
4. Retry the upgrade.
5. If the upgrade still fails contact Cloudera Support.

If the rollback fails:

The status of both the service and the running deployments changes to Failed to Update. The order in which they transition depends on where the rollback actually failed.



Note: A Cloudera DataFlow upgrade can only be rolled back if Kubernetes (K8s) was not upgraded.

Upgrade fails and rollback is not possible

Symptom: The status of the service, and shortly after the deployments, changes to Failed to Upgrade. The status icon on the service changes to Bad Health.

How to fix it:

1. If K8s upgrade failed, check the Event History if there is anything obvious in there.
2. Check on the cloud provider side whether there is anything obvious as a reason for the failure. For example, check activity logs and node pools on Azure, or EKS cluster on AWS.
3. If the K8s upgrade succeeded, but the upgrade failed somewhere later, then the previous point in the "Upgrade fails and rollback is possible" can follow here (check the events and then the K8s cluster).
4. If you managed to pinpoint and fix the possible cause of the failure, retry the upgrade.
5. If you did not find the cause of the failure, or the upgrade still fails upon retry, contact Cloudera Support.

Retrying an upgrade

1. On the Cloudera DataFlow Overview select Environments from the left navigation pane.
2. Select the service where you want to retry the upgrade and click Manage DataFlow.
3. Click Actions Retry Upgrade DataFlow .