Cloudera DataFlow

# Troubleshooting

**Date published: 2021-04-06**
**Date modified: 2024-01-09**

**https://docs.cloudera.com/**

# Legal Notice

# Contents

# Overview

You may encounter some common errors while using Cloudera DataFlow (CDF), and it is useful to understand how to recognize and correct them.

Troubleshooting errors that occur when enabling CDF for an environment

Learn how to recognize and correct common errors that occur when you are enabling CDF for an environment.

Troubleshooting errors that occur when CDF is disabled for an environment

Learn how to recognize and correct common errors that occur when you are disabling CDF for an environment.

Troubleshooting errors that occur when CDF is enabled for an environment

Learn how to recognize and correct common errors with environments for which CDF has been enabled.

Troubleshooting flow deployment errors

Learn how to recognize and address common errors with your CDF flow deployments.

# Collect and send diagnostic bundle using CLI

Learn about collecting diagnostic logs and uploading them using the CLI to facilitate troubleshooting both environment and deployment issues.

```
cdp df start-get-diagnostics-collection \
--df-service-crn '[***DATAFLOW SERVICE CRN***]' \
--description 'some description' \
--destination 'CLOUD_STORAGE' \
--environment-components ‘[“CFM_OPERATOR”, “CERT_MANAGER”]’ \
--start-time [***YYYY-MM-DDTHH:MM***] \
--end-time [***YYYY-MM-DDTHH:MM***] \
--deployments ‘[“[***CRN1***]”, “[***CRN2***]”]’ \
--collection-scope ALL
```

**--df-service-crn**

**Description**

Provide the Cloud Resource Name (CRN) of the DataFlow service for which you want to collect diagnostic data.

**Possible values**

Valid CRN

**Importance**

Required

**--description**

**Description**

Provide a free text description of the case for which diagnostic data is being collected. The provided description is persisted in the database and returned in the listing for future reference.

**Possible values**

Free text

**Importance**

Required

**--destination**

**Description**

Specify the upload destination for the generated bundle.

**Possible values**

- CLOUD_STORAGE - uploads the bundle to cloud storage associated with the DataFlow Service environment. A bundle path is included in the listing output with the format <bucket/container>/<datalake-name>/cdf/….
- SUPPORT - uploads bundle to the UDX backend where the bundle is associated with the provided case number and is automatically attached to the case.

**Importance**

Required

**--environment-components**

**Description**

Specify a list of DataFlow Environment Components for which you want to collect logs. This list operates as an allowed list when present, only collecting logs for specified components. If this list is not present, it implies log collection from all components.

**Possible values**

| Environment component | Namespace |
|---|---|
| CFM_OPERATOR | cfm-operator-system |
| CADENCE | cadence |
| CERT_MANAGER | cert-manager |
| DFX_LOCAL | dfx-local |
| DFX_LOGGING | dfx-logging |
| FLUXCD | flux-system |
| NFS_PROVISIONER | nfs-provisioner-system |
| NGINX | nginx-ingress |
| REDIS | redis |
| VAULT | vault |
| VPA | vpa |
| ZOOKEEPER_OPERATOR | zookeeper-operator-system |

**Importance**

Optional

**--start-time**

**Description**

Specify the time from which you want logs to be collected. No logs generated prior to the start time are collected.

If no end or start time is specified, log collection defaults to logs from the past 24 hours. Time zone is assumed to be UTC. Collection throws an error and does not begin if the provided end time is earlier than the provided start time.

**Possible values**

Date and time in *YYYY-MM-DD*T*hh:mm* format. For example, 2023-02-03T11:00

**Importance**

Optional

**--end-time**

**Description**

Specify the time to which you want logs to be collected. No logs generated after the end time are collected.

If no end or start time is specified, log collection defaults to logs from the past 24 hours. Time zone is assumed to be UTC. Collection throws an error and does not begin if the provided end time is earlier than the provided start time.

**Possible values**

Date and time in *YYYY-MM-DD*T*hh:mm* format. For example, 2023-02-03T11:01

**Importance**

Optional

**--deployments**

**Description**

Specify a list of DataFlow Deployment CRNs for which logs should be collected. This list operates as an allowed list when present, only collecting logs for specified deployments. If this list is not present, it allows log collection from all Flow Deployments in the environment.

**Possible values**

List of valid CRNs

**Importance**

Optional

**--collection-scope**

**Description**

Specify the scope of data collection.

**Possible values**

| | |
|---|---|
| ALL | collects logs from environment components and deployments, honoring the provided allowed lists. |
| ENVIRONMENT | collects logs from environment components, honoring the environment-components allowed list, and no logs from deployments except for those specified in the deployments allowed list. |
| DEPLOYMENT | collects logs from deployments, honoring the deployments allowed list, and no logs from environment components except for those specified in the environment-components allowed list. |

**Importance**

Optional

Default collection:

```
cdp df start-get-diagnostics-collection \
--df-service-crn '[***DATAFLOW SERVICE CRN***]' \
--description 'some description' \
--destination 'CLOUD_STORAGE'
```

Support upload:

```
cdp df start-get-diagnostics-collection \
 --df-service-crn '[***DATAFLOW SERVICE CRN***]' \
 --description 'some description' \
 --destination 'SUPPORT' \
 --case-number '123456'
```

Note that case-number is required when destination is SUPPORT

List collection attempts:

```
cdp df list-diagnostics \
--df-service-crn '[***DATAFLOW SERVICE CRN***]'
```

### Related Tasks

Collect and send a diagnostic bundle using Unified Diagnostics

### Related Information

Send a diagnostic bundle using the CDP Management Console

# Collect and send a diagnostic bundle using Unified Diagnostics

You can trigger diagnostic bundle collection for your CDP environment in order to send Data Lake, FreeIPA, and DataFlow service and environment logs to Cloudera Support for support case resolution.

### Before you begin

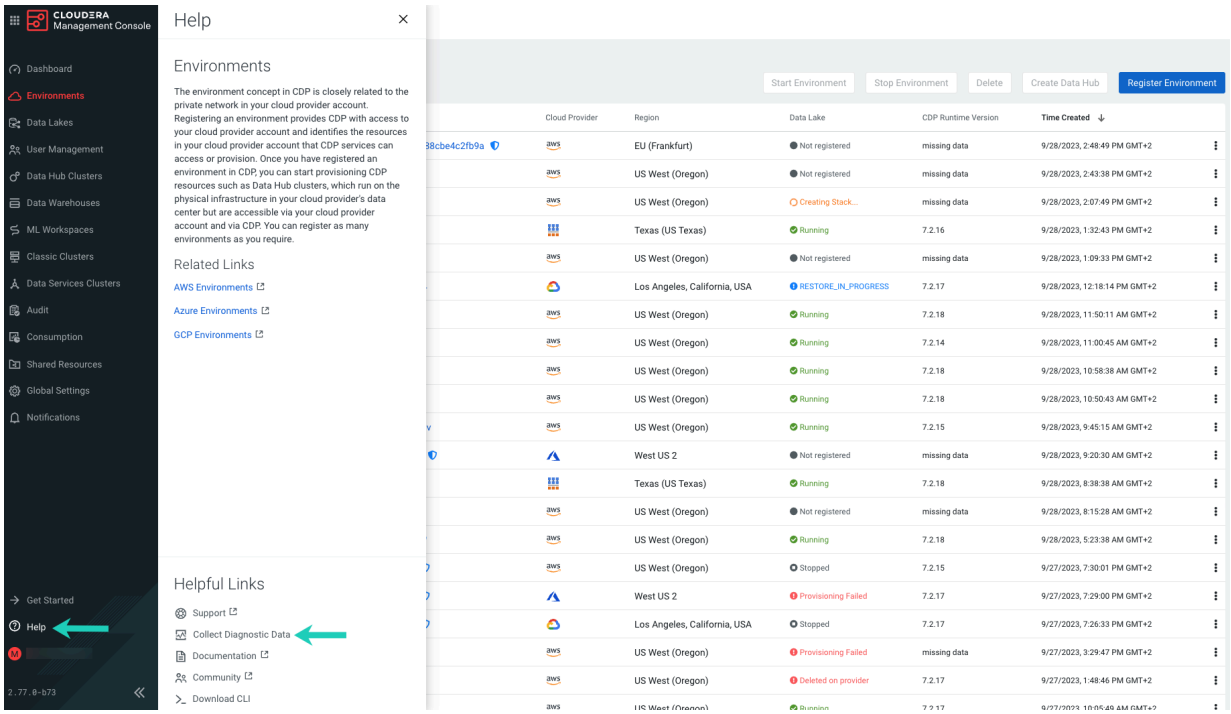
Required role: You need the EnvironmentAdmin or Owner role for the environment for which you would like to create a bundle.

- FreeIPA deployment time must be September 2020 or later.
- The VMs from which diagnostic data is to be collected must be running.
- Salt minions must be started properly on the Management Console nodes.

### Procedure

1. Log in to the CDP web interface.
2. Navigate to the Management Console.

3. Click Help in the bottom left corner and select Collect Diagnostic Data.




4. Click the Collect New Diagnostic Data.
5. On the **General Information** page, provide the following:

   **Envionment**

   Select the environment for which you would like to generate a diagnostic bundle. This is a required field.

   **Case Number**

   Enter the related support case number. You can obtain the case number from the MyCloudera portal or from your email by checking support case notification emails. This is a required field.

   **Description**

   Provide a free text description of the case for which you would like to generate diagnostic data. The provided description is persisted in the database and returned in the listing for future reference. This is a required field.

   **Time Range**

   You can specify specific points in time (date and time) from which to start and end the collection. This is an optional field.
6. Click Next.

7. On the **Selected Services** page under DataFlow, you can configure the diagnostic bundles you want to generate for CDF environment components and DataFlow deployments:

   **Environment Components**

   Select a list of DataFlow Environment Components from which you want to collect logs. This list operates as an allowed list when present, only collecting logs for specified components.

   If you leave this list empty, it implies you want to collect logs from all components.

   **Select Components**

   Specify a list of DataFlow Deployment CRNs for which you want to collect logs. This list operates as an allowed list when present, only collecting logs for specified deployments.

   If you leave this list empty, it implies log collection from all Flow Deployments in the environment.

   **Collection Scope**

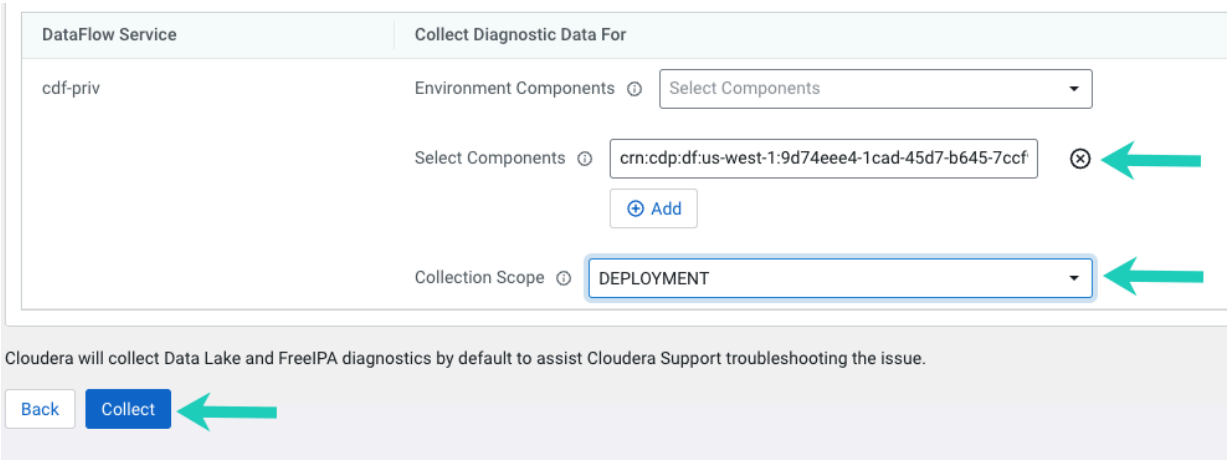
   Specify whether you want to collect environment logs, deployment logs, or both.

   Regardless of the selection you make here, logs for environment components and deployments you specified in their respective allowed lists will be collected.

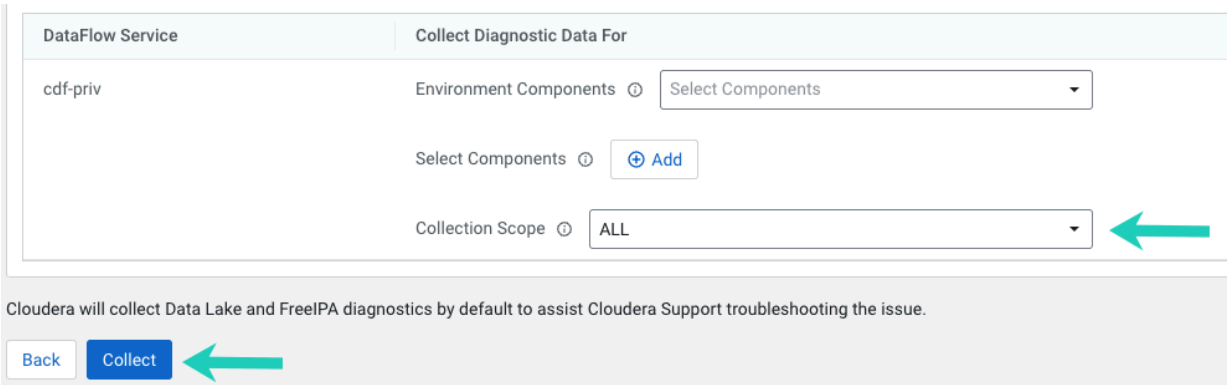
   To generate diagnostic data for a specific DataFlow Deployment, enter the CRN of the DataFlow Deployment for which you want to collect logs in the Select Components field and select DEPLOYMENT as Collection Scope.

   **Note:** To obtain the CRN of a DataFlow Deployment:

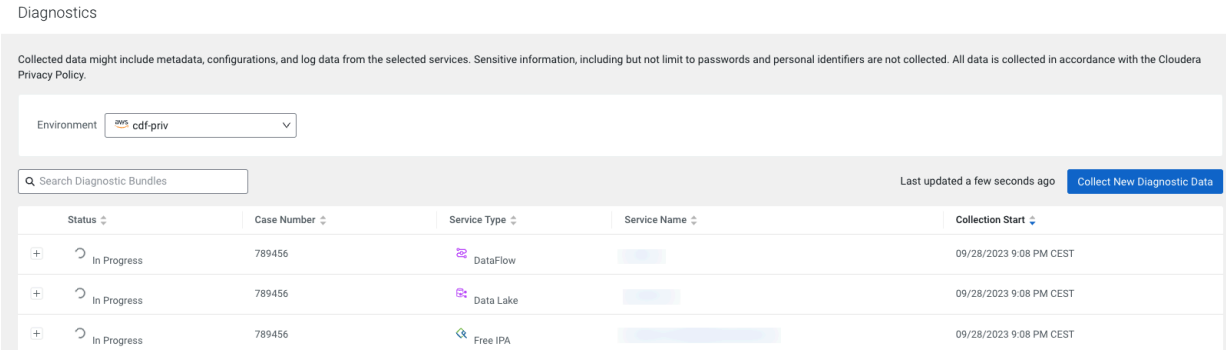
   a. Navigate to DataFlow on the CDP web interface.
   b. On the **Dashboard** select the deployment you want to collect diagnostic data for.
   c. Click Copy under CRN #.




   To obtain all diagnostic data for a CDF Service, set Collection Scope to ALL.

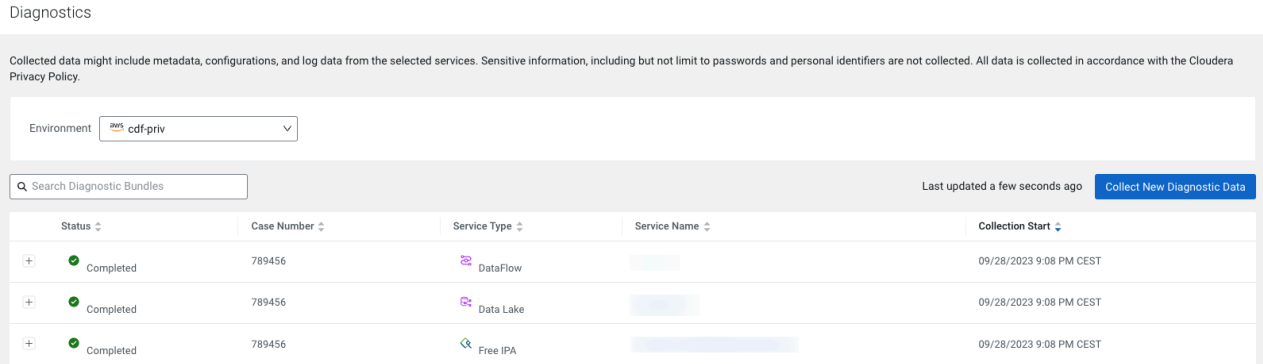


8. Click Collect.
   New entries appear on the page, allowing you to track the status of the log collection. Since a separate bundle is created for each service, there is one entry for each of the following: Data Lake, FreeIPA, and one entry for each DataFlow service. While the collection is in progress, the collection status is In progress:




### Results

Once the collection is complete, the status changes to Completed. This means that Cloudera Support can now access the diagnostic bundles:




### Related reference

Collect and send diagnostic bundle using CLI

# Downloading NiFi application log

You can download the NiFi application log from the CDF Deployment Manager to use it for troubleshooting.

### About this task

This feature allows you to download the NiFi application log that is currently being written. As the log file is rotated and the old file is archived once the file size reaches 10 MB, this is the theoretical maximum you can download using this method. For information on downloading archived log files, see *Diagnostic bundle collection*.

### Before you begin

You need DFFlowAdmin permission to perform this action.

### Procedure

1. On the Dashboard select the deployment for which you want to download the NiFi application log.
2. Select  Actions Manage Deployment .
   You are redirected to the **Deployment Manager** page.
3. Select  Actions Download NiFi Log .

### Results

The current NiFi application log is downloaded to your computer in tar.gz format.

# Troubleshooting errors that occur when enabling DataFlow for an environment fails

Learn how to recognize and correct common errors that occur when you are enabling Cloudera DataFlow (CDF) for an environment.

Your ability to deploy flow definitions depends on DataFlow being enabled and in good health in your target environment. If a DataFlow deployment is unhealthy in an environment, it is typically because enabling or disabling DataFlow for an environment has failed. Review the environment troubleshooting information to understand common errors and their solutions.

When you enable DataFlow for an existing CDP environment, CDP creates the required infrastructure in your cloud environment and installs core CDP services as well as the DataFlow software in the Kubernetes cluster. The enablement process can be divided into three parts:

1. Provisioning Cloud Infrastructure
2. Installing core CDP software and services
3. Installing CDF software and services

To understand why the enablement process is failing, the first step is to identify at which stage the enablement process failed.

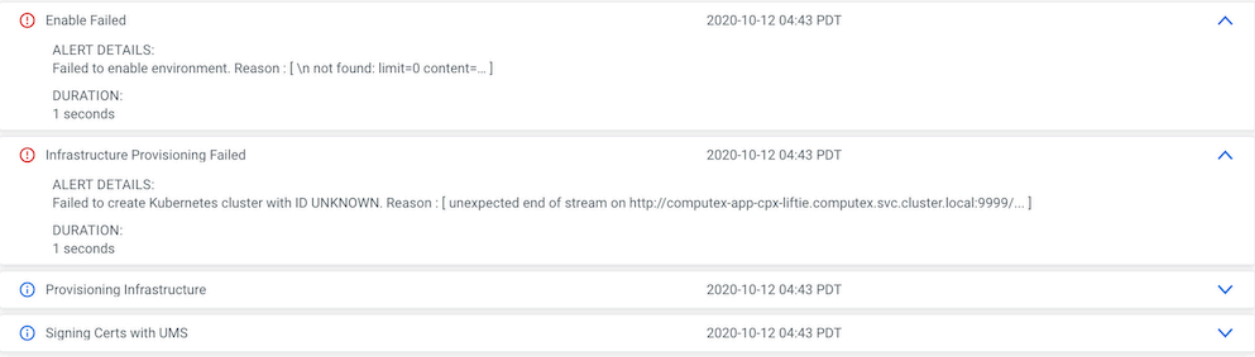
Identifying where the enablement process failed

Use the information provided in the hover state of an environment and the status messages that have been logged in the environment's Event History to identify where the enablement process has failed.

1. In DataFlow, navigate to the Environments page and find the environment where the enablement process failed.
2. Hover over the status icon and take note of the error message.
3. Click the environment, select the Alerts tab, and review the error and info events that have been logged during enablement.

## Enablement fails during infrastructure provisioning or core CDP software installation

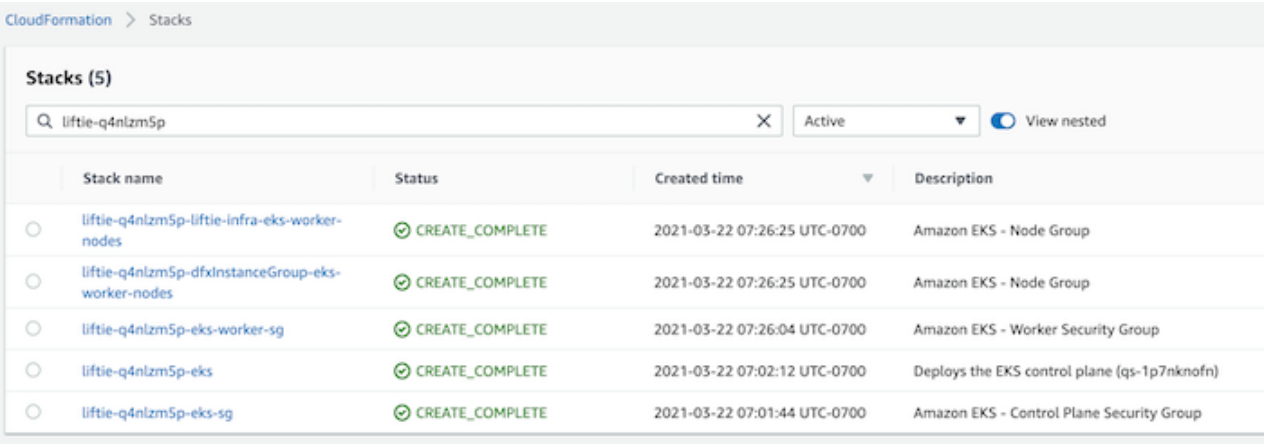
If you only see a status message in the Event History indicating that the infrastructure provisioning has started but you cannot see a corresponding status message confirming that the infrastructure has been provisioned successfully, CDP was either not able to create the required infrastructure or install the core CDP software and services afterwards.




The infrastructure provisioning failed error message indicates that there was either an issue with creating AWS infrastructure or setting up core CDP services.

Validating infrastructure creation – for AWS users

CDP uses CloudFormation scripts to create the required infrastructure in your AWS account. To validate whether the requested resources have been created successfully, log in to your AWS account, navigate to CloudFormation and search for the Kubernetes cluster ID that you have extracted from the environment events and looks similar to liftie-q4nlzm5p. Verify that the CloudFormation scripts completed successfully.

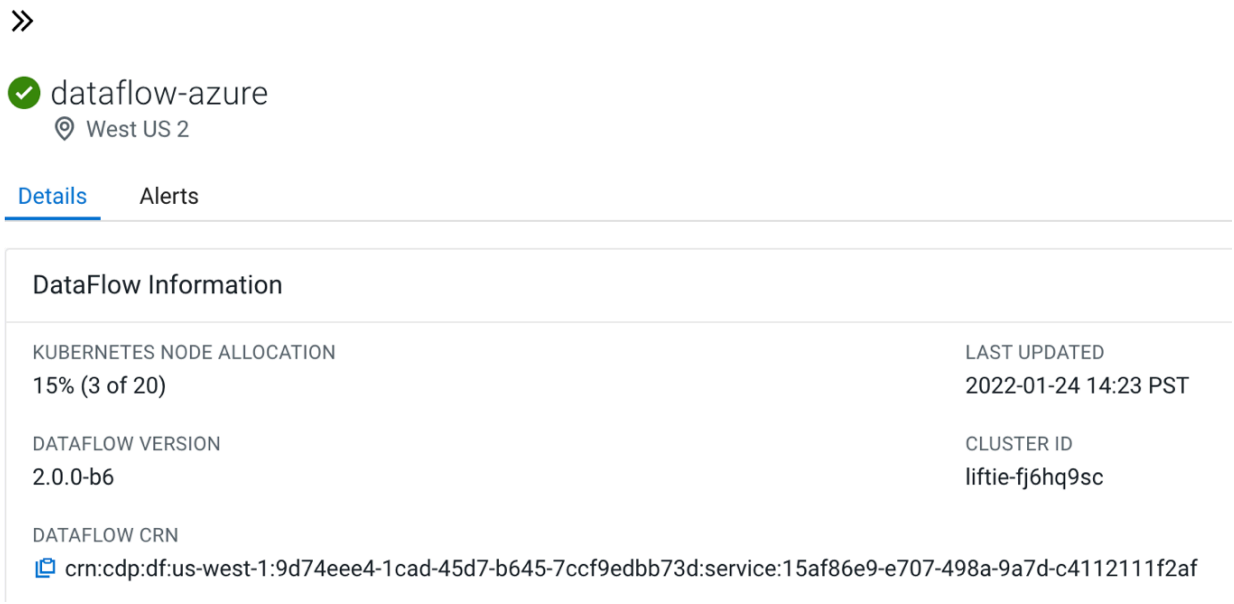



If the CloudFormation script did not complete successfully, make sure that the cross account role for your CDP environment has been assigned appropriate permissions.

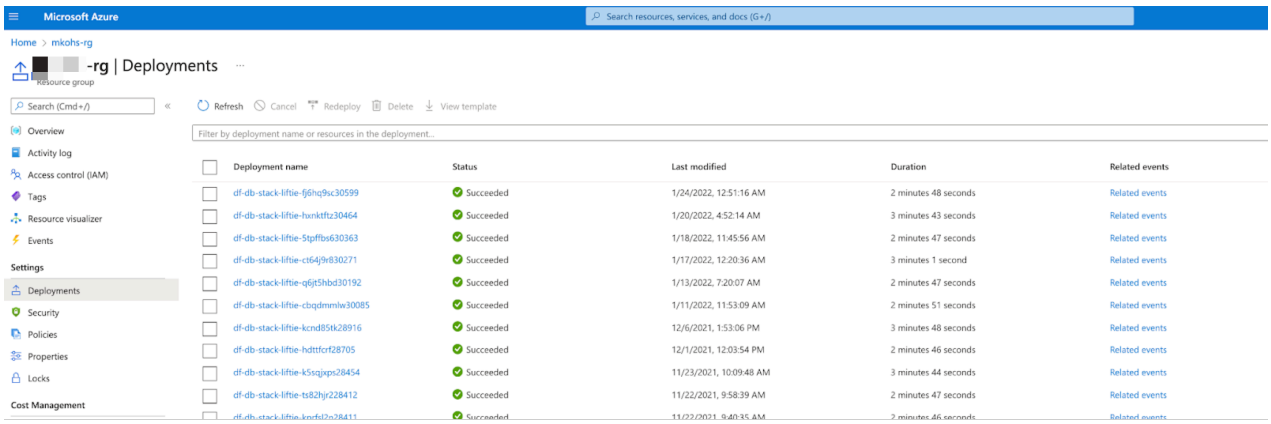
If the CloudFormation script completed successfully but enabling Dataflow failed before completing the infrastructure setup, this might be an indication that the Kubernetes cluster cannot communicate with the CDP control plane or other public endpoints like container image repositories. Make sure that the VPC and subnets you are using for DataFlow meet the CDP and DataFlow prerequisites.

Validating infrastructure creation – for Azure users

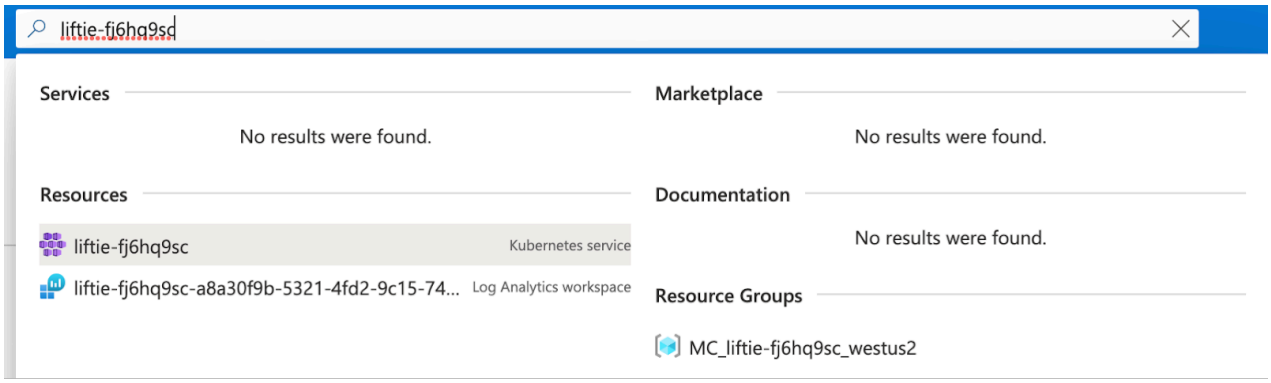
CDP uses Azure Resource Manager to orchestrate infrastructure creation in Azure. When you enable DataFlow for a CDP environment, a Cluster ID is generated that can be used to track all associated resources in Azure. Navigate to Environments in CDF, select your Azure environment and copy the Cluster ID, which will look similar to liftie-fq6hq9sc.



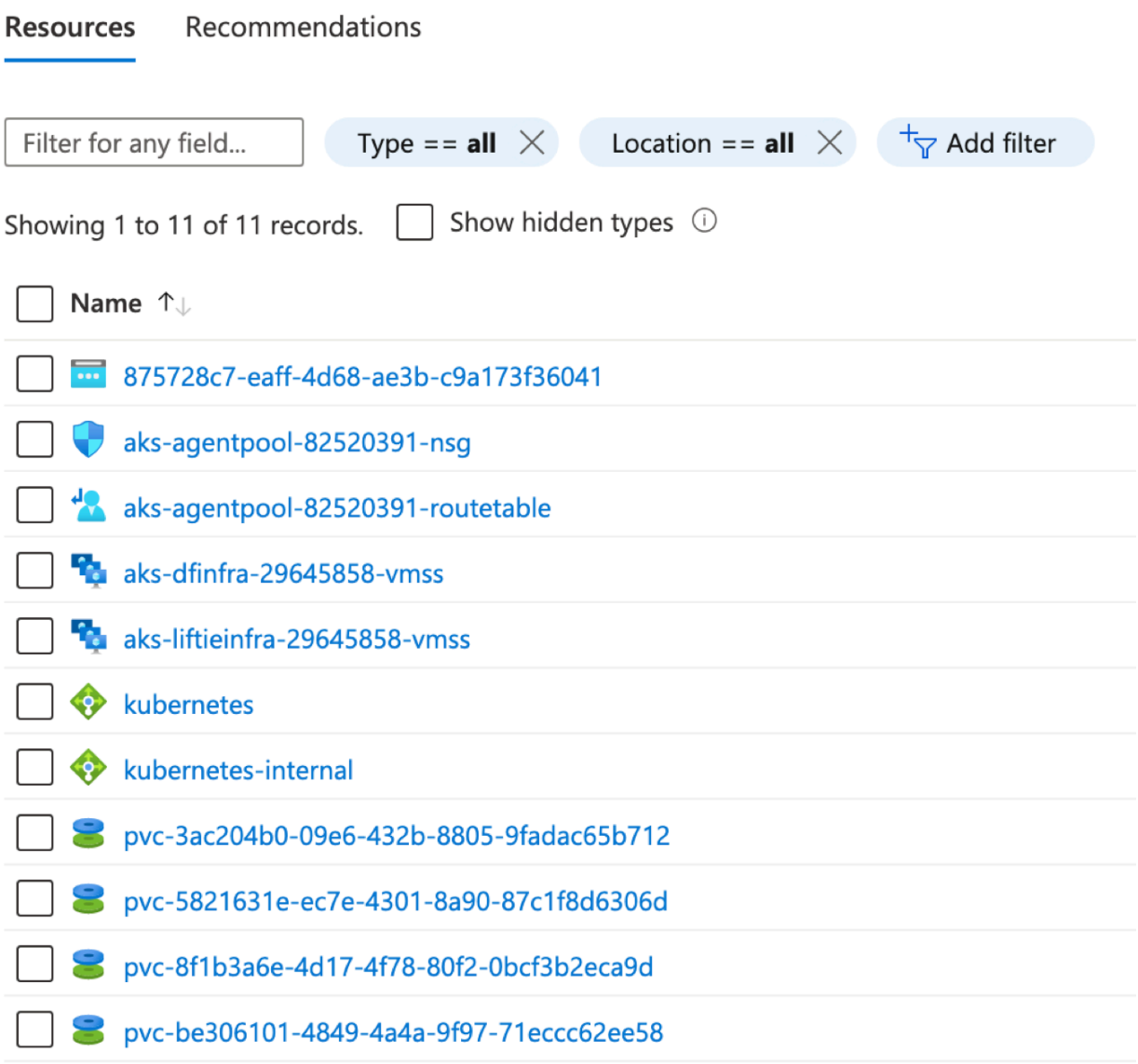

First, make sure that CDF's Azure PostgreSQL database has been created successfully. In the Azure Portal, navigate to your CDP resource group, explore Deployments under Settings, and find the database deployment associated with the previously obtained Cluster ID.




Next, validate that the AKS Kubernetes cluster and associated infrastructure has been created successfully. In the Azure Portal, use the Cluster ID to search for associated resources in your Azure subscription.




You should find a Kubernetes service as well as a new resource group called MC_<Cluster ID>_<AZURE Region>, which contains the virtual machine scale sets, load balancers, IP addresses, and storage disks that have been created for the Kubernetes cluster.




If the resources were not created successfully, make sure that the app-based credential for your CDP environment has been assigned the appropriate permissions. For more information, see *Prerequisites for the provisioning credential*.

If the infrastructure resources were created successfully, but enabling DataFlow failed before completing the infrastructure setup, it might be an indication that the Kubernetes cluster cannot communicate with the CDP control plane or other public endpoints like container image repositories. Make sure that the virtual network and subnets you are using for DataFlow meet the CDP and DataFlow prerequisites. For more information, see *VNet and subnets*.

### Enablement fails during CDF software and service installation

If you see a status message that indicates that the required Infrastructure has been provisioned successfully but the enablement process still failed, this is an indication that installing and setting up the CDF software and services has failed.


The Infrastructure Provisioned status event indicates that the Kubernetes cluster has been created and core CDP services have been setup successfully.

To ensure that this is not a transient issue, use the Retry Enablement action to start the enablement process again. Retry Enablement terminates all existing resources and provisions new infrastructure.

If retrying does not help and enabling DataFlow still fails after successfully provisioning the infrastructure, copy the error message from the Event History and open a support case with Cloudera.

### Related Concepts

Overview

### Related Information

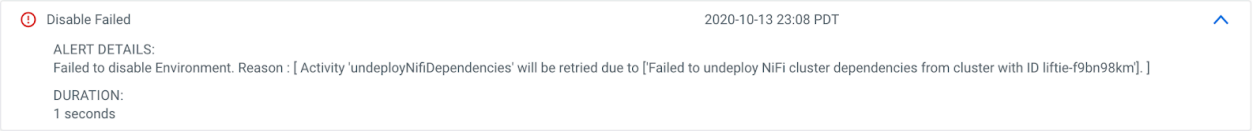
Prerequisites for the provisioning credential

VNet and subnets

# Troubleshooting errors that occur when disabling DataFlow for an environment fails

Learn how to recognize and correct common errors that occur when you are disabling Cloudera DataFlow (CDF) for an environment.

DataFlow can be in Bad Health due to a failed disablement process. If you see error messages in the Event History indicating that disabling DataFlow has failed, use the Retry Disable Process to try again to ensure that the issue is not transient.




If the issue persists, select  Environments Action Reset Environment  to clear DataFlow's state for the environment. Resetting the Environment allows you to enable DataFlow for the same environment again at a later stage.

**Note:**

Resetting DataFlow does not terminate the associated cloud infrastructure. After resetting DataFlow for an environment, you must manually delete the cloud resources that were created during DataFlow enablement.

### Related Concepts

Overview

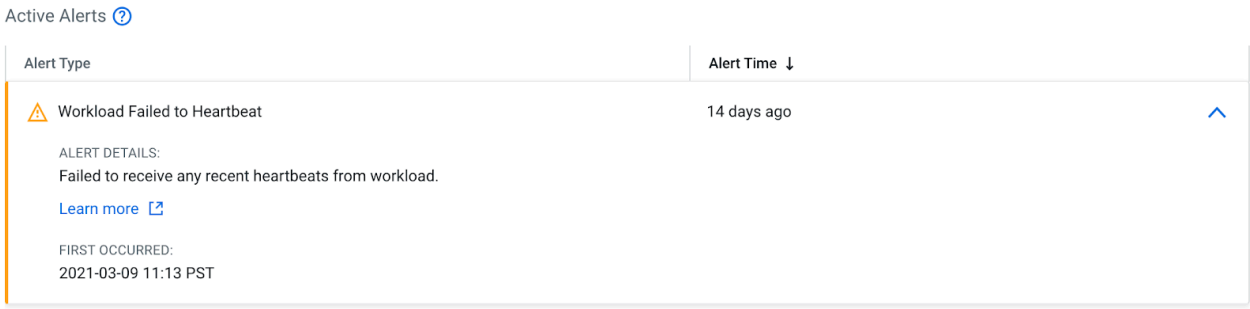
# Troubleshooting errors that occur after successful DataFlow enablement

Learn how to recognize and correct common errors with environments for which Cloudera DataFlow (CDF) has been enabled.

After you have successfully enabled DataFlow for an environment there are several reasons why DataFlow's health can become Concerning or Bad.

## Concerning health due to "Workload Failed to Heartbeat"

If DataFlow health is concerning for one of your environments, hover over the status icon and check the Alerts tab in the environment details to see details about the issue. Workload Failed to Heartbeat  means that the Cloudera Control Plane has not received a recent heartbeat from the DataFlow workload application running in your cloud account.




DataFlow fails to receive heartbeats from a particular environment

Heartbeat failures can have several reasons. Make sure that:

- The associated CDP environment has been started and is running.
- There was no networking related change in your VPC/subnet configuration and that your networking setup still meets the requirements outlined in the *DataFlow Networking*.

If the issue persists, open a support case with Cloudera.

## Concerning Health due to Nearing Maximum Kubernetes Limit

When your Kubernetes cluster is close to its maximum node count, DataFlow will show Concerning Health for the particular environment and display an Active Alert with more details about the boundaries. To return DataFlow to Good Health you can adjust the maximum node number through the Edit Configuration option in the Environment actions menu.

## Bad Health due to issues with the associated CDP environment

Certain issues with the associated CDP environment will result in DataFlow reporting Bad Health for an environment. Once DataFlow is in Bad Health you can no longer create or termnate flow deployments in the environment.

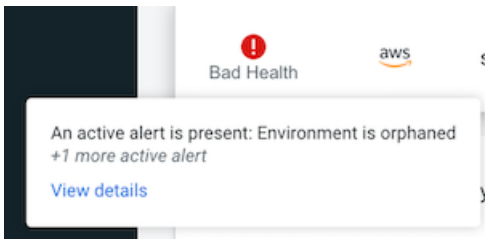
Bad Health due to CDP environment state

If the associated CDP environment is either unhealthy or in a starting/stopping state, DataFlow will report Bad Health. To return DataFlow to Good Health make sure that:

- The associated CDP environment has been started and is running.
- FreeIPA and DataLake are both running.

Bad Health due to CDP environment having been deleted

If the associated CDP environment has been deleted without disabling DataFlow first, DataFlow will report Bad Health indicating that it has been orphaned and required CDP services such as FreeIPA are no longer available in the environment.




You are not able to create new Flow Deployments for DataFlow. You cannot recover DataFlow health in this situation. Terminate your Flow Deployments and use the Disable Environment action to terminate DataFlow and associated cloud infrastructure and enable DataFlow again for a different CDP environment.

### Related Concepts

Overview

# Troubleshooting Flow Deployment errors

Learn how to recognize and address common errors with your Cloudera DataFlow (CDF) flow deployments.

### Setting up `kubectl` to connect to the DataFlow Kubernetes cluster

It is helpful to have access to the DataFlow 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 DataFlow, from the Environments page, select the DataFlow 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:

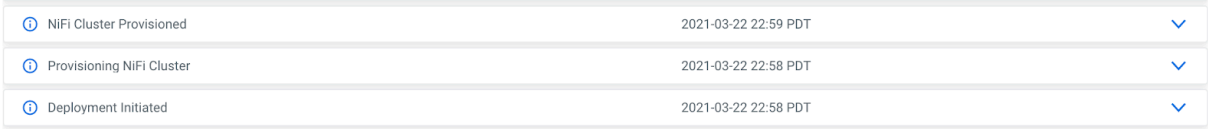
```
NAME                      STATUS   AGE
cadence                   Active   37h
cert-manager              Active   37h
cfm-operator-system       Active   37h
default                   Active   38h
dfx-dev-environment-ns    Active   36h
dfx-idbrokers3-ns         Active   36h
dfx-kafkatos3-ns          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 DataFlow logs directly through the CLI.

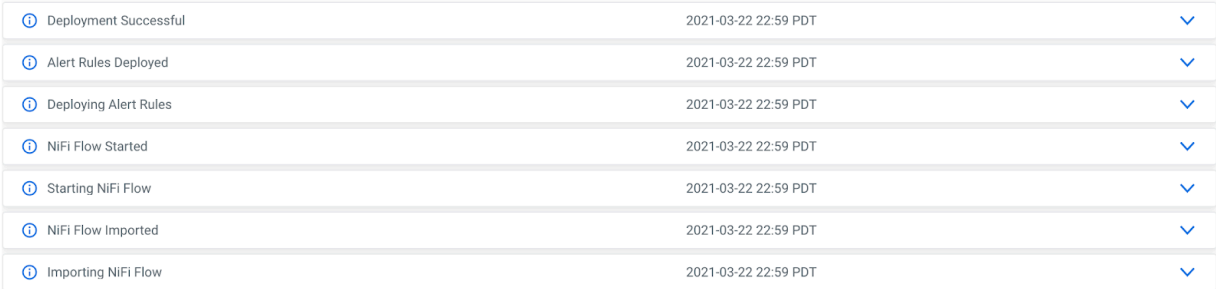
## Understanding flow deployment failures

The Flow Deployment process consists of two phases:

1. Scheduling resources on Kubernetes and creating a new NiFi cluster:




2. Importing, configuring, and starting the NiFi flow definition:




If your flow deployment fails, check the Event History to see where exactly the issue occurred.

## Deployment fails during Phase 1

If the issue occurs during Phase 1 while scheduling resources on Kubernetes and creating the NiFi cluster, you can get more details on why the deployment failed by looking at the DataFlow application logs.

Identify the DataFlow application pod by running:

```
kubectl get pods
--namespace dfx-local
```

The result should look similar to

```
NAME                                    READY   STATUS      RESTARTS   AGE
create-db-zxb2q                         0/1     Completed   0          38h
dfx-local-deployment-7f8b466c68-xwrbf   3/3     Running     0          38h
```

Copy the `dfx-local-deployment` pod ID and run to view the DataFlow application logs.

```
kubectl logs \
  -f dfx-local-deployment-7f8b466c68-xwrbf \
  -c dfx-local \
  --namespace dfx-local
```

**Note:**

`-f` tails the log file as DataFlow is writing to it.

A common reason for flow deployment issues is that the Kubernetes cluster does not have enough resources available for the deployment pods to be scheduled. Pods that are stuck in pending are an indicator for not enough resources being available.

You can explore flow deployments and their resources by running:

```
get pods --namespace dfx-deployment_name-ns
```

A healthy deployment should look similar to this with one or more NiFi pods (depending on the Sizing & Scaling settings), a ZooKeeper and a Prometheus pod.

```
NAME                      READY   STATUS    RESTARTS   AGE
dfx-nifi-0                5/5     Running   0          22h
dfx-zookeeper-0           1/1     Running   0          22h
prometheus-prometheus-0   4/4     Running   0          22h
```

If one of the pods is stuck in Pending you can explore the pod further and identify any potential issues by looking at its events.

If in the above screenshot dfx-nifi-0 was Pending and you wanted to find out why, you would run the follow command to get detailed information about the containers in the pod.

```
kubectl describe pod dfx-nifi-0 -n dfx-deployment_name-ns
```

Find the Events section and check if there are any messages about why a container could not be scheduled.

If the flow deployment failed because of insufficient resources in the Kubernetes cluster, you can increase the Kubernetes cluster size by using the Edit Configuration action of the affected environment.

### Deployment fails during Phase 2

If the issue occurs during Phase 2, check the NiFi canvas of the deployment for any error messages. To get there, open the deployment details, click Manage Deployment and in the following Deployment Manager page select the view in NiFi action.

If a processor or controller service failed to start, make sure that you have provided the correct values for the deployment parameters. You can adjust parameter values in the NiFi canvas and restart processors or controller services as needed. Once you have identified the issue, note down the correct parameter values and start a new deployment.

To view the NiFi log for a particular deployment run the following `kubectl` command.

```
kubectl logs -f dfx-nifi-0 -c app-log --namespace dfx-deployment_name-ns
```

**Note:**

`-f` tails the log file as NiFi is writing to it.

### Related Concepts

Overview