Troubleshooting

This topic describes a recommended series of steps to help you start diagnosing issues with a Cloudera Machine Learning workspace.

- Issues with Provisioning ML Workspaces: If provisioning an ML workspace fails, first go to your cloud provider account and make sure that you have all the resources required to provision an ML workspace. If failures persist, start debugging by reviewing the error messages on the screen. Check the workspace logs to see what went wrong. For more details on the troubleshooting resources available to you, see Troubleshooting ML Workspaces on AWS on page 4.
- Issues with Accessing ML Workspaces: If your ML Admin has already provisioned a workspace for you but attempting to access the workspace fails, confirm with your ML Admin that they have completed all the steps required to grant you access: Configuring User Access to CML.
- Issues with Running Workloads: If you have access to a workspace but are having trouble running sessions/jobs/experiments, and so on, see if your error is already listed here: Troubleshooting Issues with Workloads on page 10.

Cloudera Support

If you need assistance, contact Cloudera Support. Cloudera customers can register for an account to create a support ticket at the support portal. For CDP issues in particular, make sure you include the Request ID associated with your error message in the support case you create.

Troubleshooting ML Workspaces on AWS

This topic describes the ML workspace provisioning workflow and tells you how to start debugging issues with ML workspaces on AWS.

ML Workspace Provisioning Workflow

When you provision a new ML Workspace on AWS, CML performs the following actions:

1. Communicates with the CDP Management Console to check your AWS credentials. It will also enable Single Sign-On so that authorized CDP users are automatically logged in to the workspace that will be created.
2. Provision an NFS filesystem for the workspace on your cloud service provider. On AWS, CML will provision storage on EFS.
3. Provision a Kubernetes cluster on your cloud service provider. This cluster runs the workspace infrastructure and compute resources. On AWS, CML provisions an EKS cluster.
4. Mounts the provisioned NFS filesystem to the Kubernetes cluster.
5. Provision TLS certificates for the workspace using LetsEncrypt.
6. Registers the workspace with the cloud provider’s DNS service. On AWS, this is Route53.
7. Installs Cloudera Machine Learning onto the EKS cluster.

Troubleshooting Resources

Any of the steps listed above can experience failures. To start debugging, you will require access to one or more of the following resources.

- Workspace > Details Page

Each workspace has an associated Details page that lists important information about the workspace. To access this page, sign in to CDP, go to ML Workspaces and click on the workspace name.

This page lists basic information about the workspace such as who created it and when. More importantly, it includes a link to the environment where the workspace was created, a link to the underlying EKS cluster on
AWS, a list of tags associated with the workspace, and the computing resources in use. The rest of this topic explains how to use these resources.

- **Workspace > Events Page**

  Each workspace also has an associated Events page that captures every action performed on the workspace. This includes creating, upgrading, and removing the workspace, among other actions. To access this page, sign in to CDP, go to ML Workspaces, click on the workspace name, and then click **Events**.

  Click the **View Logs** button associated with an action to see a high-level overview of all the steps performed by CML to complete the action.

  The Request ID associated with each action is especially useful in case of a failure as it allows Cloudera Support to efficiently track the series of operations that led to the failure.

- **Environment > Summary Page**

  CML workspaces depend quite heavily on the environment in which they are provisioned. Each environment's Summary page lists useful information that can help you debug issues with the CML service. You can access the environment directly from the workspace Details page.

  This page includes important information such as:

  - **Credential Setup** - Tells you how security has been configured for the environment. Your cloud credential gives CDP access to the region and virtual network that make up the environment thus allowing CDP to provision resources within that environment.
  - **Region** - The AWS region where the environment is provisioned. This is especially important because it tells you which region's AWS console you might need to access for further debugging.
  - **Network** - The VPC and subnets that were created for the environment. Each CML workspace requires a set of unique IP addresses to run all of its associated Kubernetes services. If you begin to run out of IP addresses, you will need these VPC and subnet IDs to debug further in the AWS console.
  - **Logs** - When you create a CDP environment, you are asked to specify an S3 bucket in that environment that will be used to store logs. All CML operational logs and Spark logs are also written to this bucket.

  You can use the AWS console to access these logs. Alternatively, Site administrators can download these logs directly from their workspace Site Admin panel (**Admin > Support**).

  **Note**: If you file a support case, Cloudera Support will not automatically have access to these logs because they live in your environment.

- **AWS Management Console**

  If you have all the relevant information about the environment and the workspace, you can go to the AWS console (for the region where your environment was created) to investigate further. The AWS Management Console has links to dashboards for all the services used by CML.

  - **EC2**

    You can use the EC2 service dashboards to check the instance-type (CPU, GPU), VPC, subnet, and security group limits imposed on your AWS account. For example, there is typically a limit of 5 VPCs per region.

    If you need more resources, submit a request to Amazon to raise the limit of a resource.

  - **EKS**

    EKS will give you more information such as the version of Kubernetes CML is using, network information, and the status of the cluster. The workspace Details page gives you a direct link to the provisioned EKS cluster on the AWS console.

    **Note**: By default, users do not have Kubernetes-level access to the EKS cluster. If a user wants to use **kubectl** to debug issues with the EKS cluster directly, an MLAdmin must explicitly grant access using the instructions provided here: **Granting Remote Access to ML Workspaces on EKS**.

  - **VPC**

    Use the VPC ID obtained from the CDP environment Summary page to search for the relevant VPC where you have provisioned or are trying to provision an ML workspace. Each CML workspace requires a set of
unique IP addresses to run all of its associated Kubernetes services. You can use this service to see how many IP addresses are available for each subnet.

- S3
  Use the S3 bucket configured for the environment to check/download logs for more debugging.

- Tags
  When provisioning an ML workspace, you will have the option to assign one or more tags to the workspace. These tags are then applied to all the underlying AWS resources used by the workspace. If failures occur during provisioning or de-provisioning, it can be very useful to simply query the tags associated with the workspace to see if any resources need to be cleaned up manually. Tags associated with a workspace are available on the workspace Details page.

  You can search by tags in the EC2 and VPC services. You can also use the AWS CLI to search for specific tags: resourcegroupstaggingapi

- Trusted Advisor (available with AWS Support)
  Use the Trusted Advisor dashboard for a high-level view of how you are doing with your AWS account. The dashboard displays security risks, service limits, and possible areas to optimize resource usage. If you have access to AWS Support, it's a good idea to review your current account status with Trusted Advisor before you start provisioning ML workspaces.

**Common CML Errors and Solutions**

The following sections describe recommended steps to start debugging common error messages you might see in the workspace logs (found under [Events > View Logs](#)).

**Before you begin**

Make sure you have reviewed the list of resources available to you for debugging on CML and AWS:

Troubleshooting ML Workspaces on AWS on page 4

**AWS Account Resource Limits Exceeded (Compute, VPC, etc.)**

ML workspace provisioning fails because CDP could not get access to all the AWS resources needed to deploy a CML workspace. This is likely because your AWS account either does not have access to those resources or is hitting the resource limits imposed on it.

Sample errors include (from [Events > View Logs](#)):

```
Failed to provision cluster. Reason: Failed to wait for provisioner: Wait for status failed with status CREATE_FAILED: error creating eks cluster (cause: InvalidParameterException: Provided subnets subnet-0a648a0cc5976b7a9 Free IPs: 0 , need at least 3 IPs in each subnet to be free for this operation
```

```
Failed to mount storage. Reason: Failed to create mount target: NoFreeAddressesInSubnet: The specified subnet does not have enough free addresses to satisfy the request.
```

AWS accounts have certain hard and soft resource limits imposed on them by default. For example, certain CPU/GPU instances that CML allows you to provision might even have an initial default limit of 0 (set by AWS). This means if you attempt to provision a cluster with those instance types, your request will fail.

Aside from the CPU and GPU compute resource limits, there are other types of limits you can run into. For example, the second error shows that the subnets in your VPC don't have any more free IP addresses for the workspace (and each of the underlying Kubernetes pods). This occurs if the CIDR range mentioned while registering the environment was not large enough for your current needs.
You can use the AWS console to request an increase in limits as needed. Go to the AWS console for the region where the environment was provisioned and then navigate to **EC2 > Limits**.

For networking failures, navigate to **EC2 > VPC**. Search for the environment's VPC ID (available on environment Summary page) to see the list of available IP addresses for each subnet. Request more resources as needed.

Related AWS documentation: AWS Service Limits, Amazon EC2 Resource Limits, EKS Cluster VPC Considerations, AWS CNI Custom Networking.

### Access denied to AWS credential used for authentication

The cloud credential used to give CDP access to your AWS account failed authentication. Therefore, CDP could not provision the resources required to deploy a CML workspace.

Sample error (from **Events > View Logs**):

```
Failed to provision storage. Reason: Failed to create new file system: AccessDenied: User: arn:aws:iam::1234567890:user/cross-account-trust-user is not authorized to perform:
```

Your cloud credential gives CDP access to the region and virtual network that make up the environment thus allowing CDP to provision resources within that environment. If authentication fails, go to your environment to see how the cloud credentials were set up and confirm whether your account has the permission to perform these actions.

### CML Installation Failures

While the steps to provision resources on AWS were completed successfully, the CML workspace installation on EKS failed.

Sample error (from **Events > View Logs**):

```
Failed to install ML workspace. Reason: Error: release mlx-mlx failed: timed out waiting for the condition
```

If you are an advanced user, you can log in to the underlying EKS cluster and use **kubectl** to investigate further into which pods are failing.

**Note:** This error might be an indication that DNS has been turned off for the VPC. Go to the AWS console for the region where the environment was provisioned and then navigate to **EC2 > Load Balancers** to confirm that DNS is configured properly for the environment's VPC.

Related AWS documentation: EKS and kubectl

### Failures due to API Throttling

These errors can be harder to prepare for due to their seemingly random nature. Occasionally, AWS will block API calls if it receives too many requests at the same time. For example, this can occur when multiple users are attempting to provision/delete/upgrade clusters at the same time.

Sample error (from **Events > View Logs**):

```
Failed to delete cluster. Reason: Failed to wait for deletion: Wait for status DELETE_FAILED: Throttling: Rate exceeded
```

Currently, if you see a 'Throttling: Rate exceeded' error, our recommendation is that you simply try again later.

Related AWS documentation: AWS API Request Throttling

### De-provisioning Failures

De-provisioning operations can sometimes fail if AWS resources are not terminated in the right order. This is usually due to timing issues where certain resources might take too long to terminate. This can result in a cascading set
of failures where AWS cannot delete the next set of resources because they still have active dependencies on the previous set.

Sample error (from Events > View Logs):

```
Failed to delete cluster. Reason: Failed to wait for deletion: DELETE_FAILED
  msg: failed to delete aws stack
Cloudformation says resource xyz has a dependent object (Service: AmazonEC2;
  Status Code: 400; Error Code: DependencyViolation; Request ID: 815928e2-277e-4b8b-9fed-4b89716a205b) EKS - cluster still existed, was blocking CF delete
```

CML includes a Force Delete option now that will remove the workspace from the CML service. However, this not mean all the underlying resources have been cleaned up. This is where tags are very useful.

If you assigned tags to the workspace at the time of provisioning, you can use the AWS console or the CLI to query the tags associated with the workspace to see if any resources need to be cleaned up manually. Tags associated with a workspace are available on the workspace Details page.

You can search by tags in the EC2 and VPC services. You can also use the AWS CLI to search for specific tags:

```
resourcegroupstaggingapi
```

**Users unable to access provisioned ML workspaces**

If you have provisioned a workspace but your colleagues cannot automatically access the workspace using CDP Single-Sign on, make sure that you have completed all the steps required to grant users access to workspaces: Configuring User Access to CML. All CML users must have CDP accounts.

## Troubleshooting ML Workspaces on Azure

You can collect logs to troubleshoot issues that occur in ML Workspaces with Azure.

**How to access Azure logs**

Logs from the AKS control plane can be found in the "Logs" blade of the liftie-xxxxxxx resource group (not to be confused with the "Logs" blade of the AKS cluster itself or the Log Analytics Workspace in that resource group). The logs can be looked up using a query language developed by Microsoft.

**Cluster fails to scale down**

If a worker node is idle but is not being scaled down, check the cluster autoscaler logs.

Use this example to look up the logs:

```
AzureDiagnostics | where Category == "cluster-autoscaler"
```

The logs list the pods that are scheduled on a given node that are preventing it from being scaled down, or other reasons for its scaling decisions. Services running in the kube-system namespace (such as tunnelfront, or metrics-server) have been known to delay scale-down when scheduled on an otherwise idle node.

**Delete ML Workspace fails**

If you delete a workspace, and the delete operation fails, you can use Force delete to remove the workspace.

In this case, CML attempts to delete associated cloud resources for the workspace including metadata files. However, users should check that all such resources have been deleted, and delete manually if necessary.
Logs for ML Workspaces

You can access logs to troubleshoot issues with the CML service and your workloads on ML workspaces.

Access to logs

When you create a CDP environment, you specify an S3 bucket (on AWS) or an Azure Storage container (on Azure) in that environment for storing logs. If you have access to the log storage, you can use the AWS or Azure console to access certain CML and Spark logs directly. You can get the details of the specific bucket or container from the environment’s Summary page.

Note: If you file a support case, Cloudera Support will not automatically have access to these logs because they live in your environment.

ML Workspace access to logs

CML workspace users also have access to these logs depending on their authorization level:

- Site Administrators
  Site administrators can download the same logs directly from their workspace Site Admin panel (Admin > Support). For more details, see Downloading Diagnostic Bundles for a Workspace on page 9.

- Data Scientists
  While data scientists don’t have access to the full set of workspace logs, they do have access to engine logs for their own workloads (sessions/jobs/experiments). While in an interactive session or on a job/experiment’s Overview page, click Download Logs at any time to review the full set of logs for that workload’s engine. In the case of Spark workloads, Spark executor and event logs are also downloaded as part of this bundle.

Related Information

Configure lifecycle management for logs on AWS
Configure lifecycle management for logs on Azure

Downloading Diagnostic Bundles for a Workspace

This topic describes how to download diagnostic bundles for an ML workspace.

Before you begin

Note: This topic applies only to public cloud releases.

Required Role: MLAdmin

Make sure you are assigned the MLAdmin role in CDP. Only users with the MLAdmin role will be logged into ML workspaces with Site Administrator privileges.

Procedure

1. Log in to the CDP web interface.
2. Click ML Workspaces.
3. Click Admin > Support.
4. Select Include Engine Logs if needed.
5. Select the time period from the dropdown and click Download Logs.
Results
This will download a .zip file to your machine. The data in these bundles may be incomplete. If it does not contain logs for time period you are looking for, there are a number of possible reasons:

- There is a delay between the time the logs are initially generated by a workload and the time they are visible in cloud storage. This may be approximately 1 minute due to buffering during streaming, but can be significantly longer due to eventual consistency in the cloud storage.
- Another user or process may have deleted data from your bucket; this is beyond the control of Cloudera Machine Learning.
- There may be a misconfiguration or an invalid parameter in your request. Retrieving logs requires a valid cloud storage location to be configured for logging, as well as authentication for Cloudera Machine Learning to be set up properly for it. Requests must pertain to a valid engine in a valid project.

Troubleshooting Issues with Workloads

This section describes some potential issues data scientists might encounter once the ML workspace is running workloads.

Engines cannot be scheduled due to lack of CPU or memory
A symptom of this is the following error message in the Workbench: "Unschedulable: No node in the cluster currently has enough CPU or memory to run the engine."

Either shut down some running sessions or jobs or provision more hosts for Cloudera Machine Learning.

Workbench prompt flashes red and does not take input
The Workbench prompt flashing red indicates that the session is not currently ready to take input.

Cloudera Machine Learning does not currently support non-REPL interaction. One workaround is to skip the prompt using appropriate command-line arguments. Otherwise, consider using the terminal to answer interactive prompts.

PySpark jobs fail due to Python version mismatch

Exception: Python in worker has different version 2.6 than that in driver 2.7, PySpark cannot run with different minor versions

One solution is to install the matching Python 2.7 version on all the cluster hosts. A better solution is to install the Anaconda parcel on all CDH cluster hosts. Cloudera Machine Learning Python engines will use the version of Python included in the Anaconda parcel which ensures Python versions between driver and workers will always match. Any library paths in workloads sent from drivers to workers will also match because Anaconda is present in the same location across all hosts. Once the parcel has been installed, set the PYSPARK_PYTHON environment variable in the Cloudera Machine Learning Admin dashboard.

Troubleshooting Kerberos Errors

This topic describes some common Kerberos issues and their recommended solutions.

HDFS commands fail with Kerberos errors even though Kerberos authentication is successful in the web application

If Kerberos authentication is successful in the web application, and the output of klist in the engine reveals a valid-looking TGT, but commands such as hdfs dfs -ls / still fail with a Kerberos error, it is possible that your cluster is missing the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy File. The JCE policy
file is required when Red Hat uses AES-256 encryption. This library should be installed on each cluster host and will live under $JAVA_HOME. For more information, see Using AES-256 Encryption.

**Cannot find renewable Kerberos TGT**

Cloudera Machine Learning runs its own Kerberos TGT renewer which produces non-renewable TGT. However, this confuses Hadoop's renewer which looks for renewable TGTs. If the Spark 2 logging level is set to WARN or lower, you may see exceptions such as:

```
16/12/24 16:38:40 WARN security.UserGroupInformation: Exception encountered while running the renewal command. Aborting renew thread. ExitCodeException
exitCode=1: kinit: Resource temporarily unavailable while renewing credentials
16/12/24 16:41:23 WARN security.UserGroupInformation: PrivilegedActionException as:user@CLOUDERA.LOCAL (auth:KERBEROS) cause:javax.security.sasl.SaslException
GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]
```

This is not a bug. Spark 2 workloads will not be affected by this. Access to Kerberized resources should also work as expected.