

Azure Requirements

Date published: 2020-07-16

Date modified: 2024-04-01



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

Azure Account Requirements for ML Workspaces.....	4
Limitations on Azure.....	4
Network Planning for Cloudera Machine Learning on Azure.....	5
Azure Environment Setup for ML Workspaces.....	5
Create subnets.....	6
Create a subnet with the CLI.....	6
Create Azure Files Storage Account and File Share.....	7
Create Azure NetApp Files Account, Capacity Pool and Volume.....	7
Create a Volume.....	8
Other NFS Options.....	8
Set up minimum permissions.....	9
Migrating from generic NFS to Azure Files NFS in CML.....	11
Backing up workspace.....	14
Restoring old data to a new workspace.....	15

Azure Account Requirements for ML Workspaces

The requirements for using ML Workspaces in Azure are described in *Working with Azure environments*, which is linked in the Related information section, below.

In addition, Cloudera Machine Learning on Azure has a few additional requirements:

- Cloudera Machine Learning requires one Azure virtual network.
- Each ML workspace requires its own subnet.
- Each ML workspace requires its own NFS file share.
- Azure Files NFS 4.1, or Azure NetApp Files, are the recommended services, in order of preference. For Azure Files NFS 4.1, only NFS version 4.1 is supported. For Azure NetApp Files, only NFS version 3 is supported. We require the “No Root Squash” (or Allow Root Access) export option, and the volume must be shared with read and write access. The NFS volume must be created with at least 100 GB of storage capacity to ensure adequate I/O performance. It is also possible, although not recommended, to use an external NFS. For more information, see [Other NFS Options](#) in Related information.

Related Information

[Working with Azure environments](#)

[Other NFS Options](#)

[Use Azure Firewall to protect Azure Kubernetes Service \(AKS\) Deployments](#)

Limitations on Azure

This section lists some resource limits that CML and Azure impose on workloads running in ML workspaces.

- There is no ability to grant or revoke remote access (via Kubeconfig) to specific users. Users with the MLAdmin role in the environment can download a Kubeconfig file. The Kubeconfig file will continue to allow access even if the MLAdmin role is later revoked.
- Support is limited to regions that provide AKS. Also, customers should check availability of Azure Files NFS or Azure NetApp Files and GPU instance types in their intended region. See [Supported Azure regions](#) for more information.
- Data is not encrypted in transit to Azure Files NFS or Azure NetApp Files or other NFS systems, so make sure to implement policies to ensure security at the network level.
- Each ML workspace requires a separate subnet. For more information on this issue, see [Use kubenet networking with your own IP address ranges in Azure Kubernetes Service \(AKS\)](#).
- Heavy AKS activity can cause default API rate limits to trigger, causing throttling and eventually failures for AKS clusters. For some examples, see [AKS issue 1187](#) and [AKS issue 1413](#).
- When you provision an Azure Kubernetes (AKS) cluster, a Standard load balancer is provisioned by default. The Standard load balancer always provisions a public IP for egress traffic, communication with the Kubernetes control plane, and backwards compatibility. Cloudera software does not use this public IP directly, or expose anything on it. For more information, see: [Use a public Standard Load Balancer in Azure Kubernetes Service \(AKS\)](#)
- The following Azure Policy add-on for AKS has a default policy that causes workspace upgrades to fail. To avoid this problem, the Kubernetes clusters should not allow container privilege escalation policy should not be enabled. For more information on AKS policies, see [Azure Policy built-in definitions for Azure Kubernetes Service](#).
- The Workspace Backup and Restore feature, which on Azure is only available through the CDP CLI, does not perform a backup of NFS.

Related Information

[Supported Azure regions](#)

Network Planning for Cloudera Machine Learning on Azure

Before attempting to deploy your Azure virtual network and set up your Azure Environment and ML Workspaces, you should plan the network.

As an example, a minimum architecture to support two ML Workspaces would comprise the following:

- An Azure virtual network. Cloudera Machine Learning can use an existing virtual network if available.
- One subnet dedicated to the Azure NetApp Files service, if this service is utilized. Azure Files NFS does not need a dedicated subnet.
- One subnet for each ML Workspace.

Keep the following considerations in mind when planning your network:

- Each ML workspace requires one subnet in the virtual network.
- Plan the CIDR addresses for each subnet so that the ranges do not overlap.
- Each subnet should use a /26 CIDR. This should accommodate a maximum of 30 worker nodes as well as 4 infrastructure nodes for Cloudera Machine Learning.
- To use GPUs, create a virtual network with /25 CIDR subnets to accommodate a maximum of 30 GPU nodes. If you created a /26 CIDR network originally, and then subsequently need to add GPU support, you must create a new network with /25 CIDR subnets. Subnets cannot be resized.
- Azure Files NFS v4.1 (first preference) and Azure NetApp Files v3 (second preference) are the recommended NFS services for use with Cloudera Machine Learning on Azure.
- Subnets may not use the following reserved CIDR blocks: 10.0.0.0/16 or 10.244.0.0/16
- Even if private IP addresses are used for the CML service, each AKS cluster provisions a public IP for egress traffic, communication with the Kubernetes control plane, and backwards compatibility. For more information, see [Use a public Standard Load Balancer in Azure Kubernetes Service \(AKS\)](#).
- In a Single Resource Group setup in Azure, the entire CDP stack is supposed to use the resource group provided by the customer, and not create any new resource groups. However, note that Azure will automatically create new resource groups for Azure managed resources, such as a new resource group for AKS compute worker nodes.
- Each CML workspace cluster must not share any subnets or routing tables with any other CDP experience or AKS cluster.
- After provisioning a workspace, do not remove firewall rules to allow inbound and outbound access.
- Installations must comply with firewall requirements set by cloud providers at all times. Ensure that ports required by the provider are not closed. For example, Kubernetes services have requirements documented in [Use Azure Firewall to protect Azure Kubernetes Service \(AKS\) Deployments](#). Also, for information on repositories that must be accessible to set up workspaces, see *Outbound network access destinations for Azure*.

Related Information

[Use a public Standard Load Balancer in Azure Kubernetes Service \(AKS\)](#)

[Use Azure Firewall to protect Azure Kubernetes Service \(AKS\) clusters](#)

[Outbound network access destinations for Azure](#)

Azure Environment Setup for ML Workspaces

After an Azure environment has been created, there are several setup steps to prepare the environment for ML Workspaces. These steps are required when using Azure Files NFS or Azure NetApp Files, which are the recommended NFS services.

First, ensure that the Azure environment has been created. Instructions to do this can be found [here](#):

- [Working with Azure environments](#)

- [Azure subscription requirements](#)

The following steps must be performed by an Azure administrator to prepare the environment for use with an ML workspace.

Create subnets

Within a virtual network, you need to create one or more subnets for workspaces. CML workspaces cannot share subnets, so if you are planning to create multiple CML workspaces you will need to create one subnet for each workspace. In addition, if you plan to use Azure NetApp Files for NFS, one dedicated subnet must be created and delegated to Azure NetApp Files.

Procedure

1. Open the Virtual Network for your environment.
2. Click the Subnets in the left-hand navigation and then +Subnet.
3. Enter the name of the subnet, for example 'Workspace-1'.
4. Enter the IP address range as a CIDR address. A /25 network should be sufficient.
5. Create a subnet for each workspace you plan to provision.
6. Create a subnet named 'netapp'. Only one subnet needs to be delegated to 'netapp'. The 'netapp' subnet is used for Cloudera Machine Learning infrastructure purposes only, and cannot be used for a workspace. Make sure that the CIDR ranges specified do not overlap.



Note: This step is only needed if Azure NetApp Files is used.

7. In the 'netapp' subnet Info page, under Subnet delegation, select Microsoft.Netapp/volumes.



Note: This step is only needed if Azure NetApp Files is used.

8. Click Save.

Create a subnet with the CLI

You can use the CLI to create subnets, as shown in this example.

Example

```
az network vnet subnet create \  
--resource-group my-cdp-resource-group \  
--vnet-name my-cdp-vnet \  
--name my-cml-subnet-1 \  
--address-prefix 10.0.10.0/25 \  
--service-endpoints Microsoft.Sql
```

This example creates a subnet for the NetApp Files volume:



Note: This is only required if Azure NetApp Files is used.

```
az network vnet subnet create \  
--resource-group my-cdp-resource-group \  
--vnet-name my-cdp-vnet \  
--name my-anf-subnet \  
--address-prefix 10.0.10.0/25 \  

```

```
--delegations Microsoft.NetApp/volumes
```

The `--delegations` and `--service-endpoints` flags can also be applied to an existing subnet by using the `az network vnet subnet update` command.

Create Azure Files Storage Account and File Share

Azure Files NFS v4.1 is a managed, POSIX compliant NFS service on Azure. The file share is used to store files for the CML infrastructure and ML workspaces. This is the recommended NFS service for use with CML.

About this task

Consult the *Azure Files for NFS documentation*, below, for detailed information on how to create an NFS share. We outline the sequence of steps here.

Procedure

1. Create a FileStorage storage account.
2. Disable secure transfer.
3. See *Create a private endpoint*, below, to create a private endpoint between the FileStorage storage account for each of the subnets that you plan to use for creating the CML workspace. Without this step, CML workspace nodes will not have connectivity to the NFS server.
4. Create NFS file shares, one for each ML Workspace. Create each file share with an initial capacity of at least 100 GB. Ensure that the “No Root Squash” export policy is set. This is the default setting.
5. Create a Linux VM in the subnet in which you are going to create the CML workspace, and ensure that you can successfully mount the NFS share in the VM. This step will ensure that there are no connectivity issues to the NFS server from hosts in this subnet.
6. Create a directory within the NFS share to be used as the CML workspace’s projects folder. For example, if the NFS export path is ‘azurenfsv4.file.core.windows.net:/myshare’, and is mounted as ‘/mnt/myshare’, run ‘`sudo mkdir /mnt/myshare/ml-workspace-1`’; `sudo chown 8536:8536 /mnt/myshare/ml-workspace-1` to set ownership to the ‘cdsw’ user ID and group ID.
7. Go to the mount instructions page to copy the server name and export path details. Enter this into the CML control plane when prompted for an existing server. Due to a known issue in the control plane UI, the “nfs://” prefix must be specified if you are using the domain name of the file share server, e.g., “nfs://azurenfsv4.file.core.windows.net:/myshare/ml-workspace-1”. Remember to select the NFS protocol version as 4.1.
8. The Linux VM created above can be deleted if there are no issues with mounting the NFS share.

Related Information

[Azure Files for NFS documentation](#)

[Create a private endpoint](#)

Create Azure NetApp Files Account, Capacity Pool and Volume

Azure NetApp Files is a service on Azure that provides a fully managed, native file share system that is accessible in the Azure cloud. The following procedure is only required if you are using Azure NetApp Files, which is not the recommended NFS service. For more information, see *Quickstart: Set up Azure NetApp Files*.

Procedure

1. In Azure Services, select Azure NetApp Files.
2. In the New NetApp account, create an account.

3. Select the existing Resource Group.
4. In Location, select the region where the CML environment is located.
5. Provision the service. When deployment is done, click Go to Resource
6. Under Storage Service # Capacity Pools, click Add pool.
7. For Capacity, enter 4 TiB.
4 TiB is the minimum size for a capacity pool in Azure NetApp Files. We recommend a Service level of Premium or higher to ensure adequate I/O performance.
8. Click Create.
In the next steps, create a Volume.

Related Information

[Supported Azure regions](#)

[Quickstart: Set up Azure NetApp Files and create an NFS volume](#)

Create a Volume

In the Azure portal, you create an Azure NetApp Files volume to provide a file system within the Capacity Pool.

Procedure

1. In the NetApp account screen, click Volumes in the menu on the left, then click Add Volume.
2. For Quota, set a value in GB. You can set an initial value, then increase it to add additional workspaces, without having to resize the pool itself. We recommend an initial volume capacity of at least 100GB in order to ensure adequate I/O performance.
3. Confirm the value in the Delegated Subnet: the only option should be the 'netapp' subnet, because that is the one that was delegated to this service.
4. On the Protocol tab, choose the protocol version. Version 3 is supported by CML. Ensure that "Root Access" in the "Export policy" section is set to "On", and "Access" is set to "Read & Write".
5. On the Review and Create tab, click Create. The volume is deployed after several minutes.
6. After the volume is deployed, obtain the Mount path.
7. Create a Linux VM in the subnet in which the CML workspace will be created, and verify that the NFS volume can be mounted successfully in the VM. This step ensures that there are no connectivity issues between the NFS server and hosts running in the subnet.
8. Create a sub-directory to be used for the CML workspace project files. For example, if the NFS export path is 10.102.47.132:/netapp-vol and is mounted as /mnt/netapp-vol, run `sudo mkdir /mnt/netapp-vol/ml-workspace-1; sudo chown 8536:8536 /mnt/netapp-vol/ml-workspace-1` to ensure the directory is owned by the cdsu user ID and group ID.
9. When prompted for the NFS path in the CML control plane, enter the path as 10.102.47.132:/netapp-vol/ml-workspace-1.
10. The Linux VM created above can be deleted if there are no issues with mounting the NFS share.

Other NFS Options

As an alternative to Azure Files NFS or Azure NetApp Files, you can configure an NFS server that is external to the CML cluster. This is not the recommended approach.

Make sure to check the following points:

- CML requires an NFS service that is backed by a fully POSIX compliant file system. NFS service implemented over S3-like blob storage, such as NFS v3 over ADLS, is not supported.

- Currently, CML supports NFS versions 4.1 and 3. The NFS client within CML must be able to mount the NFS storage with default options, and also assumes these export options:

```
rw,sync,no_root_squash,no_all_squash,no_subtree_check
```

To prevent data loss on disk failure, it is recommended that the NFS export directory resides on a RAID disk. Before attempting to provision your CML workspace, create a Linux VM in the subnet you will use for CML.



Note: Verify that this VM can mount the directory exported by your NFS server.

- Before creating a CML workspace, the storage administrator must create a directory that will be exported to the cluster for storing ML project files for that workspace. Either a dedicated NFS export path, or a subdirectory in an existing export must be specified for each workspace. If the directory was previously used for another workspace, there may be extraneous files remaining there. Make sure that all such remaining files are deleted.
- Each CML workspace needs a unique directory that does not have files in it from a different or previous workspace. For example, if 10 CML workspaces are expected, the storage administrator will need to create 10 unique directories. Either one NFS export and 10 subdirectories within it need to be created, or 10 unique exports need to be created.

Create a workspace in a dedicated NFS share

For example, to use a dedicated NFS share for a workspace named “workspace1” from NFS server “nfs_server”, do the following:

- Create NFS export directory “/workspace1”.
- Change ownership for the exported directory
 - CML accesses this directory as a user with a UID and GID of 8536. So run `chown 8536:8536 /workspace1`
 - Make the export directory group-writeable and set GID:
`chmod g+srwx /workspace1`
- Provide the NFS export path `nfs_server:/workspace1` when prompted by the CML Control Plane App while creating the workspace.
- To use a subdirectory in an existing NFS share, say `nfs_server:/export`, do the following:
 - Create a subdirectory `/export/workspace1`
 - Change ownership: `chown 8536:8536 /export/workspace1`
 - Set GID and make directory group writeable: `chmod g+srwx /export/workspace1`
 - Provide the export path `nfs_server:/export/workspace1` when prompted by the CML Control Plane App.

Set up minimum permissions

The minimum permissions for Cloudera Machine Learning on Azure govern access control between the ML workspace, Azure resources, and the Azure storage account.

About this task

To set up the minimum permissions, you first create a custom role that contains those permissions, and then assign that role to a credential, called an app registration, in your Azure subscription.

Procedure

Create the custom role.

The following code example creates a custom role for CML and assigns the minimum permissions needed. The permissions are listed in the actions section, so that CML can access resources and operate correctly.

You need to substitute the following values:

- [YOUR-SUBSCRIPTION-ID]: Your subscription ID in use.
- [YOUR-RESTRICTED-ROLE-NAME]: The custom role name which is assigned to the application.
- [YOUR-SINGLE-RESOURCE-GROUP]: The original resource group name.

```
{
  "properties": {
    "roleName": [YOUR-RESTRICTED-ROLE-NAME],
    "description": "Custom restricted role for liftie",
    "isCustom": true,
    "assignableScopes": [
      "/subscriptions/[YOUR-SUBSCRIPTION-ID]/resourceGroups/[YOUR-SINGLE-RESOURCE-GROUP]"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.ContainerService/managedClusters/read",
          "Microsoft.ContainerService/managedClusters/write",
          "Microsoft.ContainerService/managedClusters/agentPools/read",
          "Microsoft.ContainerService/managedClusters/agentPools/write",
          "Microsoft.ContainerService/managedClusters/upgradeProfiles/read",
          "Microsoft.ContainerService/managedClusters/agentPools/delete",
          "Microsoft.ContainerService/managedClusters/delete",
          "Microsoft.ContainerService/managedClusters/accessProfiles/listCredential/action",
          "Microsoft.ContainerService/managedClusters/agentPools/upgradeProfiles/read",
          "Microsoft.Storage/storageAccounts/read",
          "Microsoft.Storage/storageAccounts/write",
          "Microsoft.ManagedIdentity/userAssignedIdentities/assign/action",
          "Microsoft.Compute/virtualMachineScaleSets/write",
          "Microsoft.Network/virtualNetworks/subnets/join/action",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Network/routeTables/read",
          "Microsoft.Network/routeTables/write",
          "Microsoft.Network/routeTables/routes/read",
          "Microsoft.Network/routeTables/routes/write",
          "Microsoft.Insights/diagnosticSettings/write",
          "Microsoft.Insights/metrics/read",
          "Microsoft.Insights/metricDefinitions/read"
        ],
        "notActions": [],
        "dataActions": [],
        "notDataActions": []
      }
    ]
  }
}
```

Related Information

[Prerequisites for the provisioning credential](#)

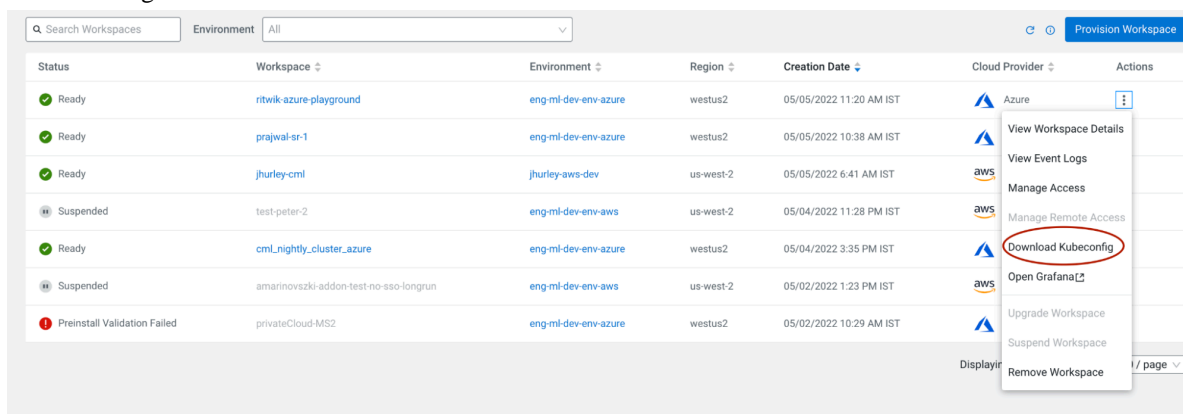
Migrating from generic NFS to Azure Files NFS in CML

CML enables you to migrate from generic NFS solutions (for example NetApp files) to Azure Files NFS for better performance and tighter integration. Migrating to the Azure ecosystem also allows CML to provide the same full fledged backup-restore functionality for Azure, as it does for AWS.

1. Set up a blank Azure file NFS instance.

You would need an empty Azure files volume to boot with. The size of the volume should be equal to or greater than the current NFS size in production use with your workspace. As with Azure files, the IOPS and network bandwidth increases with the capacity being provisioned. For more information on how to provision Azure files, please refer to this doc.

2. Ensure that the workspace is in a steady state by scaling the web pods to 0. This will ensure that the workspace APIs become unavailable, and no read/writes are taking place in the workspace during copying of data.
 - a. Download the kubeconfig from the CML control pane, open a terminal session and set the environment variable “KUBECONFIG” to the path where the kubeconfig is downloaded by export KUBECONFIG=<path to kubeconfig> .



- b. To scale the web pods to 0, enter the following command.

```
kubectl scale deployment/web -n mlx --replicas=0
```

- c. Retrieve the definition for your current NFS Persistent volume (PV) by entering the following command:

```
kubectl get pv projects-share -o yaml
```

3. From the yaml file, note the values marked in bold in the following example:

```
$ kubectl get pv projects-share -o yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    meta.helm.sh/release-name: mlx-mlx
    meta.helm.sh/release-namespace: mlx
    pv.kubernetes.io/bound-by-controller: "yes"
  creationTimestamp: "2022-05-05T09:36:59Z"
  finalizers:
    - kubernetes.io/pv-protection
  labels:
    app.kubernetes.io/managed-by: Helm
  name: projects-share
  resourceVersion: "5319"
  uid: ab4ff905-2c21-4cf6-9c31-6bd91ddc4fc8
```

```
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 50Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: projects-pvc
    namespace: mlx
    resourceVersion: "4747"
    uid: 2f58761d-2b13-4eb8-80eb-f2b97c77407f
  mountOptions:
    - nfsvers=3 # <oldNFSVersion> (this can be either 3 or 4.1)
  nfs:
    path: /eng-ml-nfs-azure/test-fs # <oldNFSPath>
    server: 10.102.47.132 # <oldNFSServer>
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
```

4. Note the server and path values from Azure files that are provisioned. For example, if the Azure Files NFS path is `azurenfsv4.file.core.windows.net/azurenfsv4/test-fs` then:

- `AzureFileNFSServer = azurenfsv4.file.core.windows.net`
- `AzureFileNFSPath = /azurenfsv4/test-fs`
- `AzureFileNFSVersion = 4.1` (This is the only option Azure files offers)

Ideally, you want a pod on which you can mount both NFS directories in a single pod, and copy over the contents. So you need to create PVs and PVCs.

5. Create a persistent volume (PV) by creating a yaml file and populating it with the following information: Replace the fields marked in bold with the variables defined in the previous step.

```
# nfs-migration-pv.yaml (the file name/location doesn't matter)
apiVersion: v1
kind: PersistentVolume
metadata:
  name: source-pv
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 50Gi
  mountOptions:
    - nfsvers=<oldNFSVersion>
  nfs:
    path: <oldNFSPath>
    server: <oldNFSServer>
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem

---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: destination-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  mountOptions:
```

```

- nfsvers=4.1
nfs:
  Server: <AzureFileNFSServer>
  path: <AzureFileNFSPath>
  volumeMode: Filesystem

```

6. Create a persistent volume claim (PVC) by creating a yaml file and populating it with the information in the following example. Replace the fields marked in bold with the variables defined in the previous step.

```

# nfs-migration-pvc.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: source-pvc
  namespace: mlx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: ""
  volumeMode: Filesystem
  volumeName: source-pv

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: destination-pvc
  namespace: mlx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: ""
  volumeName: destination-pv
  volumeMode: Filesystem

```

7. Apply the kubernetes specs using the following commands:

```

$ kubectl apply -f nfs-migration-pv.yaml
$ kubectl apply -f nfs-migration-pvc.yaml

```

8. Create a helper pod to mount the PVCs created in the previous step and copy data from one NFS workspace to another.

- a. Create a yaml file and populate it with the following information.

```

# nfs-migration-helper-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: static-pod
  namespace: mlx
spec:
  imagePullSecrets:
    - name: jfrog-dev
  restartPolicy: Never
  volumes:

```

```

- name: source-claim
  persistentVolumeClaim:
    claimName: source-pvc
- name: destination-claim
  persistentVolumeClaim:
    claimName: destination-pvc
containers:
- name: static-container
  image: container.repository.cloudera.com/cloudera/cdsw/cdsw-ubi-
minimal:<current CML version>
  command:
    - "/bin/sh"
    - "-c"
  args:
    - "trap : TERM INT; sleep 9999999999d & wait"
  volumeMounts:
    - mountPath: /source
      name: source-claim
    - mountPath: /destination
      name: destination-claim

```

- b. Apply the kubernetes spec using the following command:

```
$ kubectl apply -f nfs-migration-helper-pod.yaml
```

9. After deploying the pod, wait for the pod to be in “running” status, then copy the shell into the pod using the following command:

```
kubectl exec -it -n mlx static-pod -- sh .
```

10. Use rsync to copy the contents from the source (Generic NFS) to the destination (Azure files). Because the image does not come packaged with rsync, install it using `yum install -y rsync`.

After installing rsync, copy the contents by using `rsync -a -r -p --info=progress2 source/ destination`. Please take care of the trailing ‘/’ after “source”.

Now that you have your contents copied, you can proceed to take a backup of the original workspace. Note that the backup only backs up the Azure disks (block volumes), therefore you had to back up the NFS manually.

Backing up workspace

The Azure Back-Up Workspace feature is under entitlement. To enable Back Up Workspace for Azure, you must use the entitlement `ML_ENABLE_BACKUP_RESTORE`. If you have the entitlement, the “Backup Catalog” is visible in the UI. Contact your account manager if you don’t have the required entitlement.

Before you begin

Azure Back-Up has the following prerequisites:

- You must have an Azure resource group “cml-snapshots-<azure region>” preconfigured. Otherwise, the backup operation will throw preflight validation errors. For example, if your ML workspace is in the ‘westus2’ region, the resource group should be named “cml-snapshots-westus2”.

Procedure

1. To create a backup of the Azure workspace, enter the following commands:

```
$ cdpcp-account-helper % cdp ml backup-workspace --workspace-crn
                           crn:cdp:ml:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edb
                           b73d:workspace:8d7ea6ba-718e-45e9-9532-767f8ef0b4a4
                           --backup-name azure-test-backup --profile dev
```

2. Be sure to note the backup CRN that CDPCLI returns:

```
{
  "backupCrn": "crn:cdp:ml:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:workspace_backup:bfc57661-0eb6-4a59-bcb9-ef793740558a"
}
```

3. Wait for the backup to be in a ready state, before proceeding further

Workspace	Workspace CRN	Workspace	CRN	Environment	Last Successful Backup
cml_nightly_cluster_azure	...f0b4e4	cml_nightly_cluster_azure	cm...	eng-ml-dev-env-azure	05/06/2022 3:13 PM IST
prajwal-sr-1	...84d668				
ritwik-azure-playground2	...f7fc3e				

Backup Status	Backup Name	Backup Date	Creator	Version
Ready	azure-test-backup	05/06/2022 3:13 PM IST	Ritwik Saha	2.0.32-b24

4. Bring up the original number of replicas (for example, 3) in order to make the original workspace functional.

```
kubectl scale deployment/web -n mlx --replicas=3
```

Results

At this point, you have the corresponding Azure disks and NFS backed up to be restored into a new workspace.

Restoring old data to a new workspace

You can restore all of the backed up data into a new workspace using the backup CRN. Use your previously defined Azure files NFS as the external NFS being supplied for workspace creation.

Procedure

1. Use the sample CLI command provided below to restore the backup. Change the field information to match with your own requirements. In particular, you must define 'existingNFS', 'nfsVersion' and 'backupCrn' with your specific information.

```
cdpcp-account-helper % cdp ml restore-workspace --cli-input-json '{
  "newWorkspaceParameters": {
    "environmentName": "eng-ml-dev-env-azure",
    "workspaceName": "workspace-with-migrated-nfs",
    "disableTLS": false,
    "usePublicLoadBalancer": false,
    "enableMonitoring": true,
    "enableGovernance": true,
    "enableModelMetrics": true,
    "whitelistAuthorizedIPRanges": false,
    "existingNFS": "nfs://azurenfsv4.file.core.windows.net:/azurenfsv4/test-fs",
    "nfsVersion": "4.1",
    "provisionK8sRequest": {
      "instanceGroups": [
        {
          "instanceType": "Standard_DS3_v2",
```

```

    "rootVolume": {
      "size": 128
    },
    "autoscaling": {
      "minInstances": 1,
      "maxInstances": 10
    }
  },
  "environmentName": "eng-ml-dev-env-azure",
  "tags": [],
  "network": {
    "topology": {
      "subnets": []
    }
  },
  "skipValidation": true
},
"backupCrn": "crn:cdp:ml:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:workspace_backup:97711308-3014-418e-88d7-1f44cca495c7",
"useStaticSubdomain": false
}' --profile dev

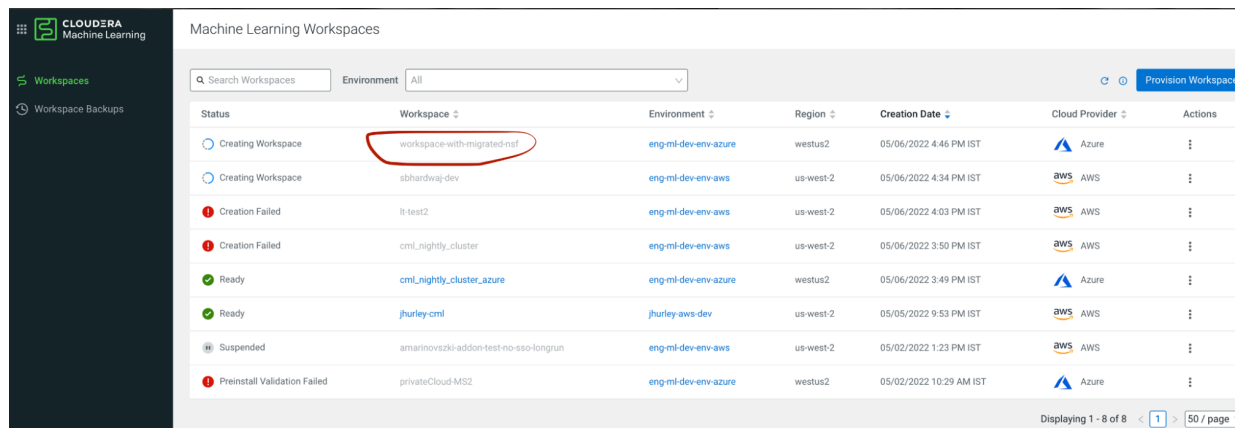
```

The CLI returns the following output:

```

{
  "workspaceCrn": "crn:cdp:ml:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edb73d:workspace:e5520051-2e5f-406d-a003-ale9dabbelc4"
}

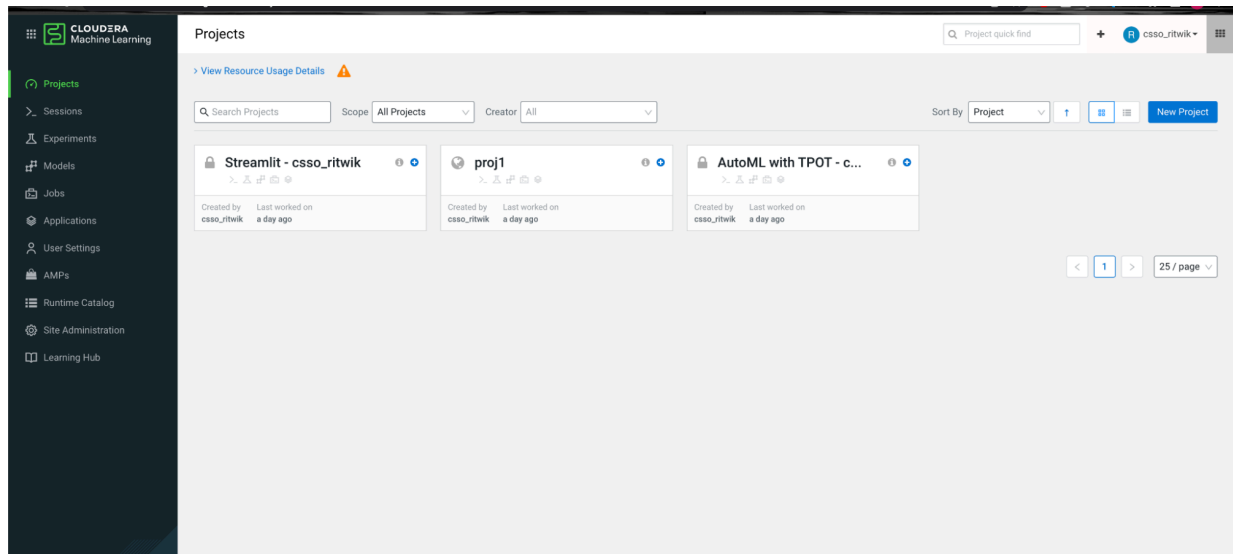
```



Status	Workspace	Environment	Region	Creation Date	Cloud Provider	Actions
Creating Workspace	workspace-with-migrated-nfs	eng-ml-dev-env-azure	westus2	05/06/2022 4:46 PM IST	Azure	
Creating Workspace	abhardway-dev	eng-ml-dev-env-aws	us-west-2	05/06/2022 4:34 PM IST	AWS	
Creation Failed	lt-test2	eng-ml-dev-env-aws	us-west-2	05/06/2022 4:03 PM IST	AWS	
Creation Failed	cml_nightly_cluster	eng-ml-dev-env-aws	us-west-2	05/06/2022 3:50 PM IST	AWS	
Ready	cml_nightly_cluster_azure	eng-ml-dev-env-azure	westus2	05/06/2022 3:49 PM IST	Azure	
Ready	jhurley-cml	jhurley-aws-dev	us-west-2	05/05/2022 9:53 PM IST	AWS	
Suspended	amarinovszki-addon-test-no-ssu-longrun	eng-ml-dev-env-aws	us-west-2	05/02/2022 1:23 PM IST	AWS	
Preinstall Validation Failed	privateCloud-MSS2	eng-ml-dev-env-azure	westus2	05/02/2022 10:29 AM IST	Azure	

You can see the new workspace spinning up in the UI.

2. Once the workspace is in ready state, log into it. You should be able to see all the previous projects in the same state that you left off.



3. Try going into one of the projects and see if the files are still visible and accessible.

