

Unit 3: Cluster Recovery (Optional)

For this optional unit, we will be needing to SSH into our virtual machines. If you are running *nix of some flavor or MacOS, you already have SSH available in your terminal. Otherwise, you may need to obtain an SSH utility such as [putty](#). There are also browser plugins, and even [thin clients](#) available.

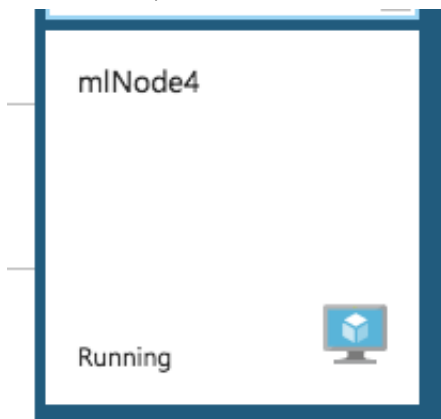
The goal for this unit is to demonstrate how a data drive can be used even after the originating node is unavailable.

Table of Contents

1. [View Existing Disk List](#)
2. [Add a Data Disk](#)
3. [Initialize the new Data Disk](#)
4. [Use the Data Disk](#)
5. [Reuse the Data Disk](#)
6. [Verify the Content](#)
7. [Summary](#)

View Existing Disk List

1. To start with, we'll need to know the full DNS name of our virtual machine. Starting from the *Azure Dashboard*, select *mlNode4*



2. Copy the **DNS name** to the clipboard using the button that appears to the right of the **DNS name** when you hover over it.

SKU
Basic
IP address
104.210.54.65
DNS name
jdwmlnode4.westus.cloudapp.azure.com
Associated to
[mlnode4822](#)
Virtual machine
[mlNode4](#)

Click to copy



- Now, use that DNS name to SSH into the machine, using the credentials we created in Unit 2. For example, `ssh mlwadmin@jdwmlnode4.westus.cloudapp.azure.com`
- Run the command `ls SCSI` and make note of the current drives available.

```
[mlwadmin@mlNode4 ~]$ ls SCSI
[1:0:0:0]    cd/dvd  Msft      Virtual CD/ROM  1.0    /dev/sr0
[2:0:0:0]    disk    Msft      Virtual Disk    1.0    /dev/sda
[3:0:1:0]    disk    Msft      Virtual Disk    1.0    /dev/sdb
[mlwadmin@mlNode4 ~]$
```

- Leave the terminal open.

Add a Data Disk

- Starting from the *Azure Dashboard*, select **mlNode4**
- Under *Settings*, click on *Disks*.
- Click the **+ Add data disk** button.

+ Add data disk

- From the *NAME* drop-down, select `Create disk`.

LUN	NAME
0	No managed disks available. ^
The value should not be greater than 31.	
Create disk	
No managed disks available.	

- In the *Create managed disk* panel enter the following details:
 - Name** - `testDataDisk`
 - Resource group** - `Use existing`, selecting what was specified in Unit 1.

Create managed disk

*

Name

testDataDisk

*

Resource group

Create new

Use existing

mlw

*

Location

West US

Availability zone

None

*

Account type

Premium (SSD)

*

Source type

None (empty disk)

*

Size (GiB)

1023

Create

- Click the **Create** button.
- Once the disk has been created, which may take a couple minutes, you will be returned to the list of disks for your virtual machine. Click **Save** to update the VM configuration.

NOTE: Wait for the update to complete before proceeding

Initialize the new Data Disk

In order to use the new disk, the partition table has to be updated, the file system type has to be specified, and the partition mounted. You can either run a single command line, or you can do it step-by-step.

OPTION 1 - Single command

1. If needed, SSH into the virtual machine once again.
2. Run the following command, you should only be prompted for the password once:

```
sudo parted /dev/sdc mklabel msdos -a optimal mkpart primary xfs 0% 100%; sudo mkfs -t xfs /dev/sdc1; sudo mkdir /datadrive; sudo mount /dev/sdc1 /datadrive; sudo chmod go+w /datadrive
```

```
[[mlwadmin@mlNode4 ~]]$ sudo parted /dev/sdc mklabel msdos -a optimal mkpart primary xfs 0% 100%; sudo mkfs -t xfs /dev/sdc1; sudo mkdir /datadrive; sudo mount /dev/sdc1 /datadrive; sudo chmod go+w /datadrive

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[[sudo] password for mlwadmin:
Information: You may need to update /etc/fstab.

meta-data=/dev/sdc1          isize=512    agcount=4, agsize=67043264 blks
                =           sectsz=4096   attr=2, projid32bit=1
                =           crc=1        finobt=0, sparse=0
data      =                 bsize=4096   blocks=268173056, imaxpct=25
                =           sunit=0      swidth=0 blks
naming    =version 2       bsize=4096   ascii-ci=0 ftype=1
log       =internal log   bsize=4096   blocks=130943, version=2
                =           sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none           extsz=4096   blocks=0, rtextents=0
[[mlwadmin@mlNode4 ~]]$
```

OPTION 2 - Step by step

1. If needed, SSH into the virtual machine once again.
2. Run the command `lsscsi` and compare that to what was listed last time. The new drive is the one we added. For example it may be `/dev/sdc`.

```
[[mlwadmin@mlNode4 ~]]$ lsscsi
[1:0:0:0]    cd/dvd  Msft    Virtual CD-ROM  1.0  /dev/sr0
[2:0:0:0]    disk    Msft    Virtual Disk    1.0  /dev/sda
[3:0:1:0]    disk    Msft    Virtual Disk    1.0  /dev/sdb
[[mlwadmin@mlNode4 ~]]$ lsscsi
[1:0:0:0]    cd/dvd  Msft    Virtual CD-ROM  1.0  /dev/sr0
[2:0:0:0]    disk    Msft    Virtual Disk    1.0  /dev/sda
[3:0:1:0]    disk    Msft    Virtual Disk    1.0  /dev/sdb
[5:0:0:0]    disk    Msft    Virtual Disk    1.0  /dev/sdc
[[mlwadmin@mlNode4 ~]]$
```

3. We need to add it as a device so use the `parted` command with the newly added disk. For example, `sudo parted /dev/sdc mklabel msdos -a optimal mkpart primary xfs 0% 100%`.
4. Next we need to set the file system for the new partition using the `mkfs` command. For example, `sudo mkfs -t xfs /dev/sdc1`

5. Now, we'll create a directory to mount the new files system to. `sudo mkdir /datadrive`
6. At this point we can mount the drive to make use of it. `sudo mount /dev/sdc1 /datadrive`
7. With the drive mounted, now we can make it writable with `sudo chmod go+w /datadrive`.
8. Leave the terminal open.

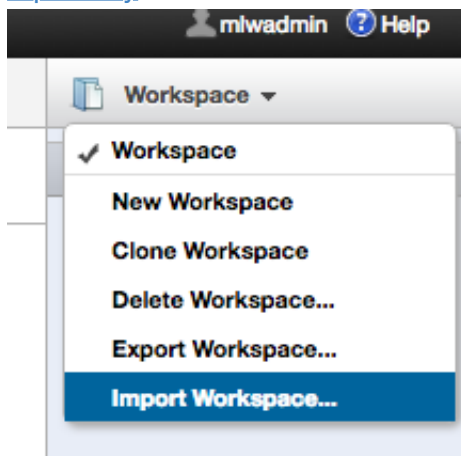
OPTIONAL - Update fstab

To make sure the drive is remounted after a reboot it needs to be added to the `/etc/fstab` file. We don't plan on rebooting during this workshop so it is optional.

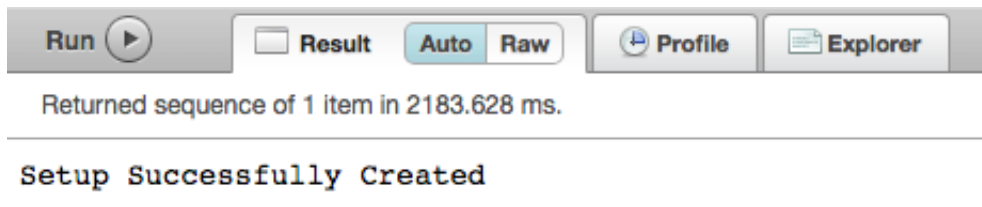
1. We'll reference the partition using its UUID. To find the UUID run `sudo -i blkid` and look for our new drive. Copy and paste it somewhere accessible.
2. Now to add it to the `fstab` file. This is a protected file, so we'll need to edit it using `sudo`.
`sudo vi /etc/fstab`
3. Arrow to the last character in the file and press `a` to begin appending. Add a new line for our drive, `UUID=cbb41429-7b8e-48ad-a638-a297abd6d99b /datadrive ext4 defaults 1 2`, ensuring to set the UUID to what you noted previously.
4. Press the escape key, followed by `:wq`, and then hit enter. (this is performing a write-quit)

Use the Data Disk

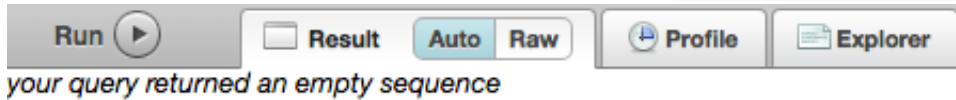
1. Use the full DNS name to bring up the *Query Console* on port 8000.
2. Import the `MLU-workshop-Azure.xml` workspace from the MLW_2018_Cloud_Workshop [GitHub repository](#).



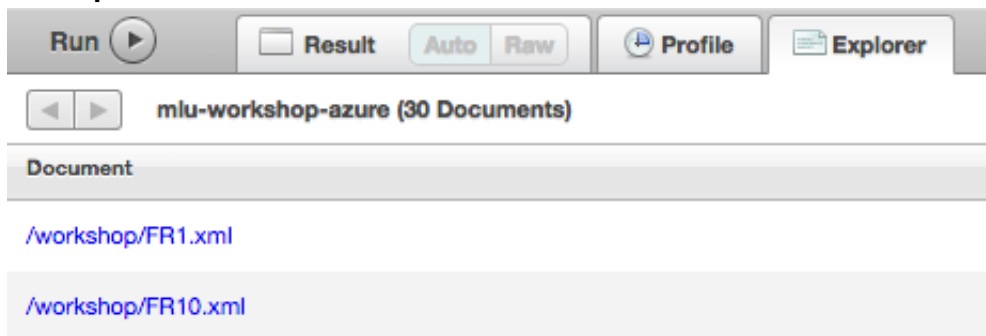
3. Select the *Set the stage* tab. Notice that the forest is being created in the `/datadrive/` directory.
4. Click the **Run** button. Once complete it will return `Setup Successfully Created`.



5. Select the *Create content* tab. This code will randomly generate 30 documents in the database.
6. Click the **Run** button. Once complete it will return `your query returned an empty sequence`.



7. Verify that the content has been created by selecting the `mlu-workshop-azure` database and clicking the **Explore** button.



8. OPTIONAL - You can also verify that content has been created by looking for a stand in the forest on the drive. In the SSH terminal run the command

`ls /datadrive/Forests/mlu-workshop-azure-Forest/` and verify that there is a directory called `00000000`.

```
[mlwadmin@mlNode4 ~]$ ls /datadrive/Forests/mlu-workshop-azure-Forest/
00000000  Journals  Label  Label_1  Large
[mlwadmin@mlNode4 ~]$
```

Reuse the Data Disk

Delete the current virtual machine

1. Starting from the *Azure Dashboard*, select **mlNode4**
2. Click the **Delete** button, and confirm by clicking the **Yes** button. The delete process will take a few minutes.



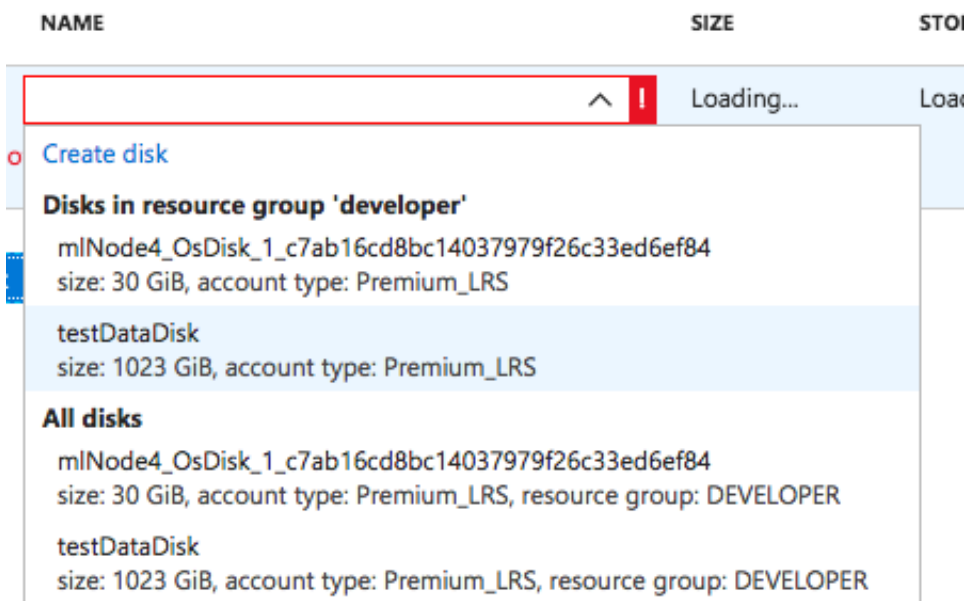
At this point the virtual machine called mlNode4 has been deleted. But, the managed drive that we created *testDataDisk*, still exists and we want to use it again.

Create a new virtual machine

1. Follow the steps in [Unit 2 - Creating the Node](#) to create the mlNode4 virtual machine again, except set the *DNS name label* to be node4new. For example, `jdwm1node4new`.

Attach the existing disk

1. From the *Azure Dashboard* select **mlNode4**.
2. Under *Settings* click on *Disks*.
3. Click the **+ Add data disk** button.
4. From the *NAME* drop-down, select the `testDataDisk`. Note that this is only available now that the original node it was associated with has been deleted and is no longer using it.



5. Click **Save** to update the VM configuration.

NOTE: Wait for the update to complete before proceeding 6. SSH into the new node. If you get a *Host key verification failed* message, you can run `ssh-keygen -R hostname` to clear it.

OPTION 1 - Single command

1. Run the following command, you should only be prompted for the password once:

```
sudo mkdir /datadrive; sudo mount /dev/sdc1 /datadrive; sudo chmod go+w /datadrive
```

OPTION 2 - Step by step

1. First, we'll create a directory to mount the new files system to. `sudo mkdir /datadrive`

2. At this point we can mount the drive to make use of it. `sudo mount /dev/sdc1 /datadrive`
3. You can verify that the forest data is now available on this vm. `ls /datadrive/Forests/`
4. 10. With the drive mounted, now we can make it writable with `sudo chmod go+w /datadrive`.

Verify the content

1. Use the full DNS name to bring up the query console on port 8000.
2. Once again, import the `MLU-workshop-Azure.xml` workspace.
3. Select the *Set the stage* tab. Notice that the forest is being created in the `/datadrive/` (which already has content there).
4. Click the **Run** button.

NOTE: Do not run the *Create content* tab. You should be reusing the content that was previously stored on the disk.

1. Refresh the query console.
2. Select the `mlu-workshop-azure` database and click the **Explore** button. You should see 30 documents listed.

Summary

By establishing our forests on a separate data disk, we can isolate them from the rest of the VM and still potentially make use of them if a VM goes down or is removed.