



# DH2i DxEnterprise 21.0 Software: SQL Server HA Instances for Linux on Azure Quick Start Guide

**DH2i Company**

Support: +1 (800) 380-5405 ext. 2

<https://dh2i.com/support/>

eFax: +1 970-295-4505

Email: [support@dh2i.com](mailto:support@dh2i.com)

<https://www.dh2i.com>

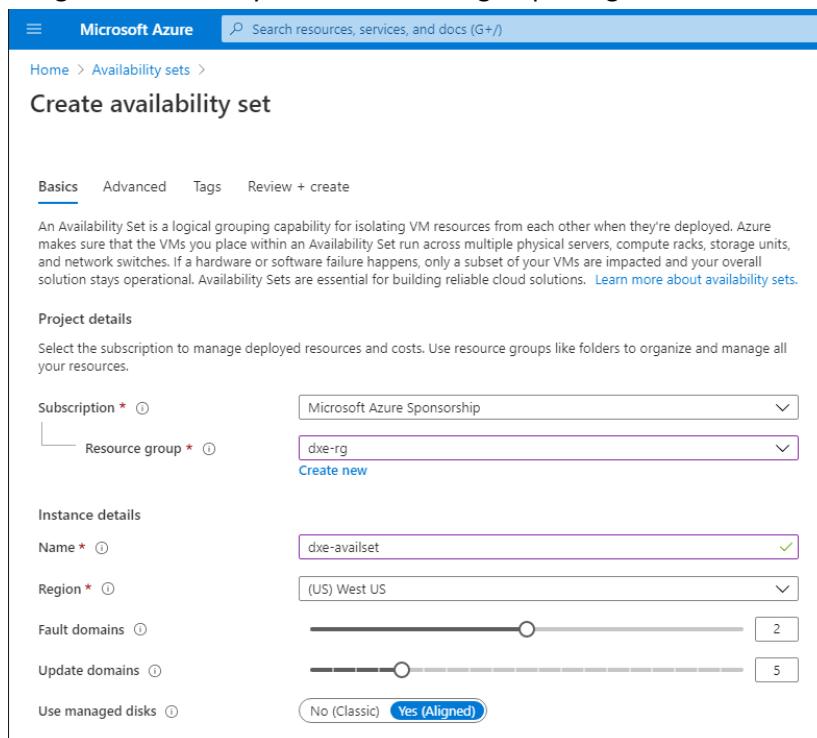
## SQL Server HA Instances for Linux on Azure

This quick start guide describes how to set up and configure a Microsoft SQL Server HA (FCI) with shared disk solution for DxEnterprise running in Azure. Using this guide, the user will create an availability set and virtual machines, configure Azure shared disks, and create and configure an Azure load balancer that will allow access to the resources assigned to the DxEnterprise Vhost.

### Setup Virtual Machines

#### Create the Availability Set and Virtual Machines

1. Login to the Azure Management Portal.
2. Search for **Availability Sets** using the top search bar, then select **Add**.
3. Assign the availability set to a resource group and give it a name.



The screenshot shows the 'Create availability set' form in the Microsoft Azure portal. The form is divided into several sections: 'Project details', 'Instance details', and 'Use managed disks'. The 'Project details' section includes 'Subscription' (Microsoft Azure Sponsorship) and 'Resource group' (dxs-rg). The 'Instance details' section includes 'Name' (dxs-availset), 'Region' ((US) West US), 'Fault domains' (2), 'Update domains' (5), and 'Use managed disks' (Yes (Aligned)).

Microsoft Azure

Search resources, services, and docs (G+)

Home > Availability sets >

### Create availability set

Basics | Advanced | Tags | Review + create

An Availability Set is a logical grouping capability for isolating VM resources from each other when they're deployed. Azure makes sure that the VMs you place within an Availability Set run across multiple physical servers, compute racks, storage units, and network switches. If a hardware or software failure happens, only a subset of your VMs are impacted and your overall solution stays operational. Availability Sets are essential for building reliable cloud solutions. [Learn more about availability sets.](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*  [Create new](#)

Instance details

Name \*

Region \*

Fault domains

Update domains

Use managed disks  Yes (Aligned)  No (Classic)

4. Select **Review + Create** in the bottom-left corner, then select **Create**.
5. Search for **DxEnterprise** in the top search bar and select one of the DxEnterprise offers available under **Marketplace**.
6. Under Select a software plan, choose an operating system and select **Create**.
7. Configure a virtual machine template and assign it to the availability set.
  - a. Assign the VM to the same resource group as the availability set and give it a name.
  - b. Under Instance Details > Availability options, select **Availability set**.
  - c. A new drop-down box will appear. Select the availability set created in step 4.
  - d. Setup an authentication type.
  - e. Under Networking, select **Create New** and set the SKU of the public IP address to standard.

**NOTE:** *There is an option under network settings to place the virtual machine behind an existing load balancing solution. Do not select this option.*

f. **Select Review + Create, then Create.**

The screenshot shows the 'Create a virtual machine' page in the Microsoft Azure portal, specifically the 'Review + create' tab. The page is titled 'Create a virtual machine' and has a breadcrumb trail: 'Home > DxEnterprise (BYOL) > Create a virtual machine'. Below the title, there are tabs for 'Basics', 'Disks', 'Networking', 'Management', 'Advanced', 'Tags', and 'Review + create'. The 'Review + create' tab is active. The page contains several sections of configuration options:

- Project details:** Subscription (Microsoft Azure Sponsorship), Resource group (dxe-rg).
- Instance details:** Virtual machine name (dxe1), Region ((US) West US), Availability options (Availability set), Availability set (dxe-availset).
- Image:** DxEnterprise on Ubuntu (BYOL).
- Azure Spot instance:** No.
- Size:** Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (\$85.41/month).
- Administrator account:** Authentication type (SSH public key), Username (AzureUser), SSH public key source (Generate new key pair), Key pair name (dxe-key).

8. Repeat steps 7a-f for additional VM(s).
9. The availability set has been created with VMs assigned to it. Return to the Azure homepage by selecting **Microsoft Azure** in the top-left corner.

## Install DxEnterprise, Join Cluster, and Install SQL Server

- Be sure to reference the [Azure Marketplace Image for Linux Quick Start Guide](#) for instructions on starting DxEnterprise, activating the node, and creating a cluster.
- Install Microsoft SQL Server on each node using [Microsoft documentation](#) for Linux.

## Configure Storage

### Present Storage

Present storage to the nodes. DxEnterprise supports any storage that is SCSI-3 Persistent Reservation compliant. The examples provided below are specific to Azure Shared Disks.

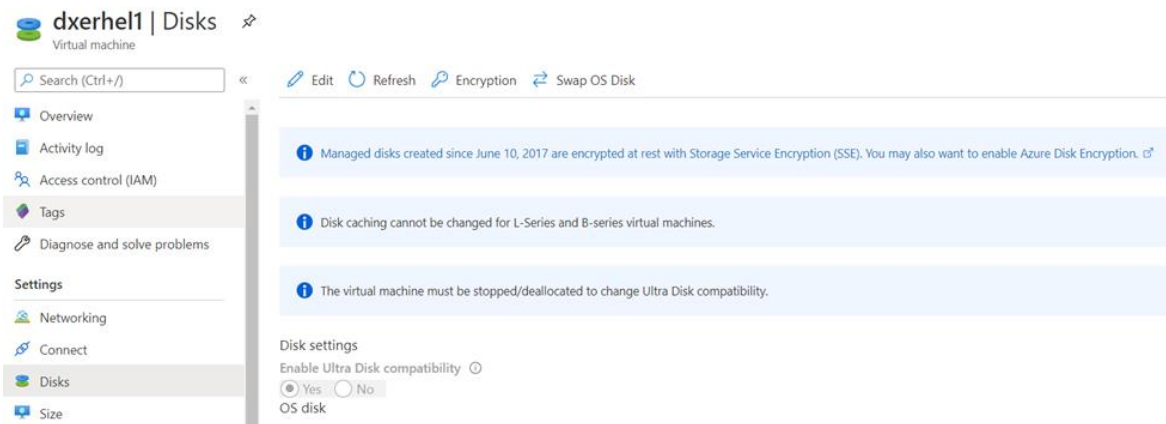
**NOTE:** *Please ensure proper presentation of storage to nodes. If storage not managed by the DxEnterprise cluster is presented to the nodes, it is possible to overwrite or corrupt data held on that storage.*

## Create and Configure Azure Shared Disk

On July 16, 2020, Microsoft announced the general availability of Azure Shared Disks. Azure Shared Disks is the industry's first shared block storage in the cloud. Azure Shared Disks are available both on Ultra Disks and Premium SSDs.

**NOTE: There are various limitations for Azure disks that can be used as shared disks. For example, the VMs and the disks must be residing in the same Availability Zone for regions that support Availability Zones, Availability Sets, or Proximity Placement Groups. It is important to provision resources appropriately as they cannot be changed after creation.**

1. Before an Azure shared disk can be used, the VMs must have its Ultra disk feature enabled. From the Azure Portal, go to the Disk property of the VM and enable **Ultra Disk Compatibility**.



The Ultra Disk feature can also be enabled via the Azure CLI:

- `az vm update -g <resource_group> -n <vm_name> --set additionalCapabilities.ultraSSDEnabled=1`
2. To deploy an Azure disk (e.g. Ultra Disk) as a shared disk that is shareable across multiple VMs, modify the `max-shares` parameter to a value greater than 1.
    - `az disk create -g <resource_group> -n <disk_name> --size-gb <desired_size> -l <location> --sku UltraSSD_LRS --max-shares <number_of_VMs_sharing>`
  3. Once the shared disk is created with `max-shares` set to a value greater than 1, it can be attached to the VMs via the Azure CLI:
    - `az vm disk attach --resource-group <resource_group> --vm-name <vm_name> --name <disk_name>`

## Configure Disk(s) in DxEnterprise

DxEnterprise leverages native file system (NTFS/ext4/xfs) and shared storage technology to coordinate access to a pool of disk resources in the cluster. Managing a disk puts that disk under DxEnterprise control.

1. To manage a disk, use the `sudo dxcli add-disk` command. The `sudo dxcli get-disks` command can be used to retrieve a list of disk IDs.

### Syntax

```
dxcli add-disk <disk_id> [label]
```

### Parameters

| Name    | Description                    | Required |
|---------|--------------------------------|----------|
| disk_id | The ID of the disk.            | True     |
| label   | The unique label for the disk. | False    |

### Example

```
sudo dxcli add-disk 50842140-be32-d3a7-45d4-3999bf3ad3a8 "Cluster Disk 1"
```

2. Create a volume on the managed disk by using the `sudo dxcli create-volume` command.

### Syntax

```
dxcli create-volume <disk_id>,<size_in_bytes>
```

### Parameters

| Name          | Description                                | Required |
|---------------|--|----------|
| disk_id       | The ID of the disk.                        | True     |
| size_in_bytes | The size of the volume to create in bytes. | True     |

### Example

```
sudo dxcli create-volume 50842140-be32-d3a7-45d4-3999bf3ad3a8,1073741824
```

3. Retrieve the volume ID using the `sudo dxcli get-disk-detail` command.

### Syntax

```
dxcli get-disk-detail <disk_id>
```

### Parameters

| Name    | Description        | Required |
|---------|--------------------|----------|
| disk_id | The ID of the disk | True     |

### Example

```
dxcli get-disk-detail 50842140-be32-d3a7-45d4-3999bf3ad3a8
```

4. Format the volume on the managed disk using the `sudo dxcli format-volume` command.

### Syntax

```
dxcli format-volume <volume_id> <fstype> <label> <block_size>  
<quick_format:true|false> <compression:true|false>  
[optional_parameters]
```

## Parameters

| Name                    | Description   | Required |
|-------------------------|---|----------|
| volume_id               | The ID of the volume.   | True     |
| fstype                  | The file system type.<br>[FAT FAT32 exFAT EXT3 EXT4 NTFS UDF ReFS XFS]  | True     |
| label                   | The label for the volume.   | True     |
| block_size              | The block size in bytes.<br>[512 1024 2048 4096 8192 16384 32768 65536] | True     |
| quick_format:true false | Whether or not to perform a quick format.                               | True     |
| compression:true false  | Whether or not to enable compression.                                   | True     |
| optional_parameters     | Optional parameters from format.com.                                    | False    |

## Example

```
sudo dxcli format-volume 3409ed39-60c1-4f49-8186-dfface26e2a1 EXT4  
Volume1 4096 quick_format:true compression:false
```

5. Assign a mount point for the volume using the `sudo dxcli set-mountpoint` command.

## Syntax

```
dxcli set-mountpoint <volume_id> <mount_point>
```

## Parameters

| Name        | Description           | Required |
|-------------|-----------------------|----------|
| volume_id   | The ID of the volume. | True     |
| mount_point | The mount path.       | True     |

## Example

```
sudo dxcli set-mountpoint 3409ed39-60c1-4f49-8186-dfface26e2a1  
/mnt/volume1
```

## Create and Configure a DxEnterprise Vhost

### Create a Vhost

DxEnterprise uses Virtual Hosts (Vhosts) to provide failover support and high availability. A Vhost virtualizes the network name and IP address associated to a particular SQL Server Instance, file share, and/or service. Instead of using the network name and IP address of a physical server, a Vhost is created and assigned a unique name/virtual IP-address pair. Clients access the databases associated with an instance via the Vhost name or IP address; they do not need to know which node is running the SQL instance. When a Vhost is configured, the user will need to specify at least one node to participate in the Vhost. It is recommended to create A and PTR records for each Vhost in DNS for resolution as well as add Vhost entries to each node's local hosts file.

To add a Vhost, use the `sudo dxcli cluster-add-vhost` command.

## Syntax

```
dxcli cluster-add-vhost <vhost> <vips> <nodes> [autofailback] [priority[1-5]] [ilb_ports]
```

## Parameters

| Name         | Description  | Required |
|--------------|--|----------|
| vhost        | The name of the Vhost.   | True     |
| vips         | The virtual IP(s) for the Vhost (comma separated list for multiples). The use of a loopback address (127.0.0.1) is supported, but must be preceded by an asterisk (*). | True     |
| nodes        | The node(s) to add to the Vhost (comma separated list for multiples).  | True     |
| autofailback | Set autofailback option, or leave blank if autofailback is not desired.  | False    |
| priority     | The priority order of failover between Vhosts (1 is the highest and 5 is the lowest).  | False    |
| ilb_ports    | Port(s) to use for internal load balancer probing (comma-separated list for multiples).  | False    |

## Example

```
sudo dxcli cluster-add-vhost vhost1 192.168.1.10 dxemssql1,dxemssql2
```

## Manage a Vhost Diskgroup

A diskgroup is a logical set of disks that are added to a Vhost. When a disk is added to a Vhost diskgroup, the disk will be set online on the active node and offline on all other nodes. If there is a failure of a disk within the diskgroup, the Vhost will failover onto the next available node in the cluster.

To assign a disk to a Vhost, use the `sudo dxcli vhost-set-diskgroup` command.

## Syntax

```
dxcli vhost-set-diskgroup <vhost> <disk_ids>
```

## Parameters

| Name     | Description   | Required |
|----------|---|----------|
| vhost    | The name of the Vhost.  | True     |
| disk_ids | A full list of pipe ( ) delimited disk IDs to add to the diskgroup. | True     |

## Example

```
sudo dxcli vhost-set-diskgroup VHOST1 50842140-be32-d3a7-45d4-3999bf3ad3a8
```

## Add a SQL Instance to a Vhost

When a SQL Server instance is added to a Vhost, DxEnterprise virtualizes the network name and IP address associated with the SQL Server instance creating a Virtual SQL Server instance. Clients can then access the Virtual SQL Server instance via the Vhost\instance name.

1. OPTIONAL: This step is only required if the SQL Server sysadmin (SA) account credentials will be provided in step 2. Encrypt the SQL Server sysadmin password using `sudo dxcli encrypt-text` command.

## Syntax

```
dxcli encrypt-text <value>
```

### Parameters

| Name  | Description               | Required |
|-------|---------------------------|----------|
| value | The text to be encrypted. | True     |

### Example

```
dxcli encrypt-text Passw0rd
```

2. Assigning a SQL Server instance to a Vhost creates a managed instance. To add a SQL instance to a Vhost, use the `sudo dxcli add-instance` command. Insert the encrypted SQL sysadmin password from the previous step for the `sql_pass` parameter.

### Syntax

```
add-instance <vhost>\<instance> <port> <sql_data_path>  
<sql_log_path> [sql_login] [sql_pass] [keep]
```

### Parameters

| Name          | Description   | Required |
|---------------|---|----------|
| vhost         | The name of the Vhost.  | True     |
| instance      | The name of the SQL instance.   | True     |
| port          | The port to be used.  | True     |
| sql_data_path | The path for the SQL data.  | True     |
| sql_log_path  | The path for the SQL logs.  | True     |
| sql_login     | A sysadmin user for the SQL instance. If using Windows authentication, the credential must be supplied in UPN format (e.g. <a href="#">user@domain.com</a> ). | False    |
| sql_pass      | The encrypted password for the SQL sysadmin user.   | False    |
| keep          | Specify this parameter to keep the current SQL data.  | False    |

### Example

```
sudo dxcli add-instance vhost1\mssqlserver 1433 /mnt/volume1/data  
/mnt/volume1/log sa Gks+GJplFmUbTLLBy4wPmw==
```

## (Optional) Configure Azure Load Balancer

To configure an Azure Load Balancer, please view the [quick start guide](#).

## References

- [DH2i Support Portal](#)
- [DxEnterprise Documentation](#)
- [DxEnterprise DxCli Guide](#)
- [Microsoft – Share an Azure managed disk](#)
- [Microsoft – Enable shared disk](#)
- [Microsoft – Install SQL Server on Linux](#)