

How to Manage Azure disks with Azure PowerShell

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to <https://shell.azure.com/powershell>. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Default Azure disks

When an Azure virtual machine is created, two disks are automatically attached to the virtual machine.

Operating system disk - Operating system disks can be sized up to 4 terabytes, and hosts the VMs operating system. If you create a new virtual machine (VM) from an [Azure Marketplace](#) image, the typically 127 GB (but some images have smaller OS disk sizes). The OS disk is assigned a drive letter of *C:* by default. The disk caching configuration of the OS disk is optimized for OS performance. The OS disk **should not** host applications or data. For applications and data, use a data disk, which is detailed later in this article.

Temporary disk - Temporary disks use a solid-state drive that is located on the same Azure host as the VM. Temp disks are highly performant and may be used for operations such as temporary data processing. However, if the VM is moved to a new host, any data stored on a temporary disk is removed. The size of the temporary disk is determined by the [VM size](#). Temporary disks are assigned a drive letter of *D:* by default.

Azure data disks

Additional data disks can be added for installing applications and storing data. Data disks should be used in any situation where durable and responsive data storage is needed. The size of the virtual machine determines how many data disks can be attached to a VM.

VM disk types

Azure provides two types of disks.

Standard disks - backed by HDDs, and delivers cost-effective storage while still being performant. Standard disks are ideal for a cost effective dev and test workload.

Premium disks - backed by SSD-based high-performance, low-latency disk. Perfect for VMs running production workload. Premium Storage supports DS-series, DSV2-series, GS-series, and FS-series VMs. Premium disks come in five types (P10, P20, P30, P40, P50), the size of the disk determines the disk type. When selecting, a disk size the value is rounded up to the next type. For example, if the size is below 128 GB the disk type is P10, or between 129 GB and 512 GB the disk is P20.

Premium disk performance

While the above table identifies max IOPS per disk, a higher level of performance can be achieved by striping multiple data disks. For instance, 64 data disks can be attached to Standard_GS5 VM. If each of these disks is sized as a P30, a maximum of 80,000 IOPS can be achieved. For detailed information on max IOPS per VM, see [VM types and sizes](#).

Create and attach disks

To complete the example in this tutorial, you must have an existing virtual machine. If needed, create a virtual machine with the following commands.

Set the username and password needed for the administrator account on the virtual machine with [Get-Credential](#):

Create the virtual machine with [New-AzVm](#). You'll be prompted to enter a username and password for the administrators account for the VM.

```
New-AzVm `
  -ResourceGroupName "myResourceGroupDisk" `
  -Name "myVM" `
  -Location "East US" `
  -VirtualNetworkName "myVnet" `
  -SubnetName "mySubnet" `
  -SecurityGroupName "myNetworkSecurityGroup" `
  -PublicIpAddressName "myPublicIpAddress"
```

Create the initial configuration with [New-AzDiskConfig](#). The following example configures a disk that is 128 gigabytes in size.

```
$diskConfig = New-AzDiskConfig `
    -Location "EastUS" `
    -CreateOption Empty `
    -DiskSizeGB 128
```

Create the data disk with the [New-AzDisk](#) command.

```
$dataDisk = New-AzDisk `
    -ResourceGroupName "myResourceGroupDisk" `
    -DiskName "myDataDisk" `
    -Disk $diskConfig
```

Get the virtual machine that you want to add the data disk to with the [Get-AzVM](#) command.

```
$vm = Get-AzVM -ResourceGroupName "myResourceGroupDisk" -Name "myVM"
```

Add the data disk to the virtual machine configuration with the [Add-AzVMDataDisk](#) command.

```
$vm = Add-AzVMDataDisk `
    -VM $vm `
    -Name "myDataDisk" `
    -CreateOption Attach `
    -ManagedDiskId $dataDisk.Id `
    -Lun 1
```

Update the virtual machine with the [Update-AzVM](#) command.

```
Update-AzVM -ResourceGroupName "myResourceGroupDisk" -VM $vm
```

Prepare data disks

Once a disk has been attached to the virtual machine, the operating system needs to be configured to use the disk. The following example shows how to manually configure the first disk added to the VM. This process can also be automated using the [custom script extension](#).

Manual configuration

Create an RDP connection with the virtual machine. Open up PowerShell and run this script.

```
Get-Disk | Where partitionstyle -eq 'raw' |  
Initialize-Disk -PartitionStyle MBR -PassThru |  
New-Partition -AssignDriveLetter -UseMaximumSize |  
Format-Volume -FileSystem NTFS -NewFileSystemLabel "myDataDisk" -Confirm:$false
```

Verify the data disk

To verify that the data disk is attached, view the StorageProfile for the attached DataDisks.

```
$vm.StorageProfile.DataDisks
```

The output should look something like this example:

```
Name           : myDataDisk  
DiskSizeGB     : 128  
Lun            : 1  
Caching        : None  
CreateOption   : Attach  
SourceImage    :  
VirtualHardDisk :
```