

Contents

Windows VMs Documentation

Overview

[About Virtual Machines](#)

Quickstarts

[Create VM - Portal](#)

[Create VM - PowerShell](#)

[Create VM - Azure CLI](#)

Tutorials

[1 - Create / manage a VM](#)

[2 - Create / manage disks](#)

[3 - Automate configuration](#)

[4 - Create VM images](#)

[5 - Highly available VMs](#)

[6 - Create a scale set](#)

[7 - Load balance VMs](#)

[8 - Manage networking](#)

[9 - Backup virtual machines](#)

[10 - Manage VMs](#)

[11 - Track and update VMs](#)

[12 - Monitor VMs](#)

[13 - Manage VM security](#)

[14 - Install a SQL\IIS\.NET stack](#)

[15 - Secure web server with SSL](#)

Samples

[PowerShell index](#)

[PowerShell](#)

[Create](#)

[Quick](#)

[Fully configured VM](#)

- [Highly available VM](#)
- [Custom script example](#)
- [Use DSC](#)
- [VM from VHD upload](#)
- [Attach an OS disk](#)
- [VM from snapshot](#)
- [Manage storage](#)
 - [Create a managed disk from a VHD](#)
 - [Create a managed disk from a snapshot](#)
 - [Copy a managed disk](#)
 - [Export a snapshot as a VHD to a storage account](#)
 - [Export the VHD of a managed disk to a storage account](#)
 - [Create a snapshot from a VHD](#)
 - [Copy a snapshot](#)
- [Secure](#)
 - [Encrypt a VM and its data disks](#)
- [Monitor](#)
 - [Azure Monitor](#)
 - [Collect details about all VMs in a subscription](#)
- [Azure CLI index](#)
- [CLI](#)
 - [Create](#)
 - [Quick](#)
 - [Fully configured VM](#)
 - [Highly available VM](#)
 - [Custom script example](#)
 - [Use DSC configuration](#)
 - [Manage storage](#)
 - [Create managed disk from a VHD](#)
 - [Create a managed disk from a snapshot](#)
 - [Copy managed disk](#)
 - [Export a snapshot as VHD to a storage account](#)

[Export the VHD of a managed disk to a storage account](#)

[Copy snapshot](#)

[Network](#)

[Secure network traffic between VMs](#)

[Secure](#)

[Encrypt a VM and data disks](#)

[Monitor](#)

[Azure Monitor](#)

[Concepts](#)

[Azure Resource Manager](#)

[Regions](#)

[Availability and performance](#)

[Availability](#)

[Co-location](#)

[Network performance](#)

[VM types and sizes](#)

[VM sizes](#)

[Generation 2 VMs](#)

[General purpose](#)

[Overview](#)

[Av2-series](#)

[B-series burstable](#)

[DCv2-series](#)

[Dv2 and DSv2-series](#)

[Dv3 and Dsv3-series](#)

[Dav4 and Dasv4-series](#)

[Compute optimized](#)

[Overview](#)

[Fsv2-series](#)

[Memory optimized](#)

[Overview](#)

[Dv2 and DSv2-series 11-15](#)

[Ev3 and Esv3-series](#)

[Eav4 and Easv4-series](#)

[M-series](#)

[Mv2-series](#)

[Constrained vCPUs](#)

[Storage optimized](#)

[Overview](#)

[Lsv2-series](#)

[Optimize performance](#)

[Accelerated compute](#)

[GPU optimized](#)

[Overview](#)

[NC-series](#)

[NCv2-series](#)

[NCv3-series](#)

[ND-series](#)

[NDv2-series](#)

[NV-series](#)

[NVv3-series](#)

[NVv4-series](#)

[Setup GPU drivers](#)

[Setup AMD GPU drivers](#)

[High performance compute](#)

[Overview](#)

[H-series](#)

[HB-series](#)

[HBv2-series](#)

[HC-series](#)

[Reserved instances](#)

[Prepay for VMs](#)

[What are Azure reservations?](#)

[VM instance size flexibility](#)

- Spot VMs
- Previous generations
- Isolated sizes
- Azure compute units (ACU)
- Benchmark scores
- vCPU quotas
- Reserved instances
 - Prepay for VMs
 - What are Azure reservations?
 - VM instance size flexibility
- Dedicated hosts
- Maintenance and updates
- Disk storage
 - Introduction to managed disks
 - Select a disk type for IaaS VMs
 - Encryption
 - Disk Storage reservations
 - Designing for high performance
 - Disk bursting
 - Scalability targets for disks
 - Backup and disaster recovery for disks
 - Shared disks
 - Ephemeral OS disks
- Networking
- Scale sets
- Infrastructure automation
- Security
 - Security and policy
 - Azure Disk Encryption
 - Built-in security controls
- States and lifecycle
- Monitoring

[Backup and recovery](#)

[Infrastructure example](#)

[How-to guides](#)

[Create VMs](#)

[Use dedicated hosts](#)

[PowerShell](#)

[Portal](#)

[Deploy spot VMs](#)

[Portal](#)

[PowerShell](#)

[Template](#)

[Error codes](#)

[Use C#](#)

[Use specialized disk](#)

[Portal](#)

[PowerShell](#)

[Use a template with C#](#)

[Create VM with Chef](#)

[Use Java](#)

[Use Python](#)

[Use a template](#)

[Connect to a VM](#)

[Use Azure Hybrid Benefit license](#)

[Use Multitenant Hosting Rights for Windows 10](#)

[Migrate from classic to Azure Resource Manager](#)

[Retirement starting March 1, 2023](#)

[Overview](#)

[Deep Dive on migration](#)

[Plan for migration](#)

[Migrate using PowerShell](#)

[Common migration errors](#)

[Community tools for migration](#)

FAQ

Secure VMs

[Recommendations](#)

[Just-in-time access](#)

[Encrypt](#)

[Disk encryption scenarios for Windows](#)

[VM encryption with Azure CLI](#)

[VM encryption with Azure PowerShell](#)

[VM encryption with Azure portal](#)

[Key vault for Azure Disk Encryption](#)

[Disk encryption sample scripts](#)

[Disk encryption troubleshooting](#)

[Disk encryption FAQ](#)

[Disk encryption - previous version \(AAD\)](#)

[Overview](#)

[Key vault for Azure Disk Encryption](#)

[Disk encryption scenarios for Windows](#)

[Use WinRM](#)

[Use access controls](#)

[Use policies](#)

[Create a Key Vault](#)

Protect VMs

[Disaster recovery](#)

[Back up VMs](#)

[Back up a single VM](#)

[Back up multiple VMs](#)

[Restore a disk](#)

[Restore individual files](#)

[Set up disaster recovery for VMs](#)

[Enable disaster recovery for a VM](#)

[Run a disaster recovery drill for a VM](#)

[Fail over a VM to another region](#)

Manage VMs

VM usage

Common PowerShell tasks

Change VM size

Swap the OS disk

Tag a VM

Time sync

Run scripts on a VM

Custom Script Extension

Run Command

Change temp drive letter

Change availability set

Download template

Azure VM agent

Mitigating speculative execution

Monitor metadata

Enable nested virtualization

Platform maintenance

Maintenance notifications

Overview

CLI

Portal

PowerShell

Maintenance control

PowerShell

CLI

Scheduled events

Windows scheduled event service

Monitor VMs

Azure Monitor for VMs

Create metric alerts

Create log alerts

Use Images

Shared images

Overview

PowerShell

Portal

CLI

App registration for sharing

Troubleshoot shared images

Image builder (preview)

Overview

Use Azure CLI

Build for image galleries

Update an existing image

Store scripts

Template reference

Troubleshoot

Find and use images

Prepare VM for upload

Capture VM to image

Use generalized image

Build image with Packer

Use Windows client images

Download existing disk

Availability and scale

Virtual Machine Scale Sets

High availability

Create a proximity placement group

Portal

PowerShell

Vertically scale

PowerShell

Portal

Use automation tools

Chef

Publish Web App from Visual Studio

Run applications

SQL Server

MongoDB

SAP on Azure

MATLAB cluster

Visual Studio

High Performance Computing (HPC)

HPC Pack 2016 cluster

HPC Pack 2016 Azure Active Directory integration

HPC Pack 2012 R2 head node

Submit on-prem jobs to HPC Pack 2012 R2

Excel on HPC Pack

Manage storage

Add a disk

Azure PowerShell

Azure portal

Detach a disk

Deploy disks with template

Enable shared disks

Upload a vhd to a disk - PowerShell

Resize a disk

Use Storage Explorer to manage disks

Snapshot a disk

Reserve Disk Storage

Create an incremental snapshot

Back up unmanaged disks

Migration and conversion

Convert disk between Standard and Premium

Migrate to Premium storage with Azure Site Recovery

- [Migrate to Managed Disks](#)
- [Unmanaged VM to Managed Disks](#)
- [Performance](#)
 - [Enable write accelerator](#)
 - [Using ultra disks](#)
 - [Benchmark a disk](#)
 - [Find unattached disks](#)
- [Disks FAQs](#)
- [Manage networking](#)
 - [Create virtual network](#)
 - [Open ports to a VM](#)
 - [Azure portal](#)
 - [Azure PowerShell](#)
 - [Assign public IP address](#)
 - [Use multiple NICs](#)
 - [Use accelerated networking](#)
 - [Assign public DNS name](#)
 - [DNS resolution](#)
- [Configure managed identities](#)
 - [Portal](#)
 - [CLI](#)
 - [PowerShell](#)
 - [Azure Resource Manager Template](#)
 - [REST](#)
 - [Azure SDKs](#)
- [Use VM extensions](#)
- [Move VMs](#)
 - [Change subscription or resource group](#)
 - [Move regions](#)
 - [Move to an availability zone](#)
- [Migrate AWS and on-premises VMs](#)
- [Upload on-prem VM](#)

[From Amazon Web Services \(AWS\)](#)

[Use Azure Site Recovery](#)

[Troubleshoot](#)

[Reference](#)

[Azure CLI](#)

[Azure PowerShell](#)

[.NET](#)

[Java](#)

[Node.js](#)

[Python](#)

[Compute REST](#)

[Managed Disks REST](#)

[Resource Manager template](#)

[Resources](#)

[Author templates](#)

[Build your skills with Microsoft Learn](#)

[Azure Roadmap](#)

[Community templates](#)

[Pricing](#)

[Regional availability](#)

[Stack Overflow](#)

[Videos](#)

[FAQ](#)

Windows virtual machines in Azure

1/10/2020 • 5 minutes to read • [Edit Online](#)

Azure Virtual Machines (VM) is one of several types of [on-demand, scalable computing resources](#) that Azure offers. Typically, you choose a VM when you need more control over the computing environment than the other choices offer. This article gives you information about what you should consider before you create a VM, how you create it, and how you manage it.

An Azure VM gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the VM by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in various ways. Some examples are:

- **Development and test** – Azure VMs offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- **Applications in the cloud** – Because demand for your application can fluctuate, it might make economic sense to run it on a VM in Azure. You pay for extra VMs when you need them and shut them down when you don't.
- **Extended datacenter** – Virtual machines in an Azure virtual network can easily be connected to your organization's network.

The number of VMs that your application uses can scale up and out to whatever is required to meet your needs.

What do I need to think about before creating a VM?

There are always a multitude of [design considerations](#) when you build out an application infrastructure in Azure. These aspects of a VM are important to think about before you start:

- The names of your application resources
- The location where the resources are stored
- The size of the VM
- The maximum number of VMs that can be created
- The operating system that the VM runs
- The configuration of the VM after it starts
- The related resources that the VM needs

Locations

All resources created in Azure are distributed across multiple [geographical regions](#) around the world. Usually, the region is called **location** when you create a VM. For a VM, the location specifies where the virtual hard disks are stored.

This table shows some of the ways you can get a list of available locations.

METHOD	DESCRIPTION
Azure portal	Select a location from the list when you create a VM.
Azure PowerShell	Use the Get-AzLocation command.

METHOD	DESCRIPTION
REST API	Use the List locations operation.
Azure CLI	Use the <code>az account list-locations</code> operation.

Availability

Azure announced an industry leading single instance virtual machine Service Level Agreement of 99.9% provided you deploy the VM with premium storage for all disks. In order for your deployment to qualify for the standard 99.95% VM Service Level Agreement, you still need to deploy two or more VMs running your workload inside of an availability set. An availability set ensures that your VMs are distributed across multiple fault domains in the Azure data centers as well as deployed onto hosts with different maintenance windows. The full [Azure SLA](#) explains the guaranteed availability of Azure as a whole.

VM size

The [size](#) of the VM that you use is determined by the workload that you want to run. The size that you choose then determines factors such as processing power, memory, and storage capacity. Azure offers a wide variety of sizes to support many types of uses.

Azure charges an [hourly price](#) based on the VM's size and operating system. For partial hours, Azure charges only for the minutes used. Storage is priced and charged separately.

VM Limits

Your subscription has default [quota limits](#) in place that could impact the deployment of many VMs for your project. The current limit on a per subscription basis is 20 VMs per region. Limits can be raised by [filing a support ticket requesting an increase](#).

Operating system disks and images

Virtual machines use [virtual hard disks \(VHDs\)](#) to store their operating system (OS) and data. VHDs are also used for the images you can choose from to install an OS.

Azure provides many [marketplace images](#) to use with various versions and types of Windows Server operating systems. Marketplace images are identified by image publisher, offer, sku, and version (typically version is specified as latest). Only 64-bit operating systems are supported. For more information on the supported guest operating systems, roles, and features, see [Microsoft server software support for Microsoft Azure virtual machines](#).

This table shows some ways that you can find the information for an image.

METHOD	DESCRIPTION
Azure portal	The values are automatically specified for you when you select an image to use.
Azure PowerShell	<code>Get-AzVMImagePublisher -Location <i>location</i></code> <code>Get-AzVMImageOffer -Location <i>location</i> -Publisher <i>publisherName</i></code> <code>Get-AzVMImageSku -Location <i>location</i> -Publisher <i>publisherName</i> -Offer <i>offerName</i></code>
REST APIs	List image publishers List image offers List image skus

METHOD	DESCRIPTION
Azure CLI	<pre>az vm image list-publishers --location <i>location</i> az vm image list-offers --location <i>location</i> --publisher <i>publisherName</i> az vm image list-skus --location <i>location</i> --publisher <i>publisherName</i> --offer <i>offerName</i></pre>

You can choose to [upload and use your own image](#) and when you do, the publisher name, offer, and sku aren't used.

Extensions

VM [extensions](#) give your VM additional capabilities through post deployment configuration and automated tasks.

These common tasks can be accomplished using extensions:

- **Run custom scripts** – The [Custom Script Extension](#) helps you configure workloads on the VM by running your script when the VM is provisioned.
- **Deploy and manage configurations** – The [PowerShell Desired State Configuration \(DSC\) Extension](#) helps you set up DSC on a VM to manage configurations and environments.
- **Collect diagnostics data** – The [Azure Diagnostics Extension](#) helps you configure the VM to collect diagnostics data that can be used to monitor the health of your application.

Related resources

The resources in this table are used by the VM and need to exist or be created when the VM is created.

RESOURCE	REQUIRED	DESCRIPTION
Resource group	Yes	The VM must be contained in a resource group.
Storage account	Yes	The VM needs the storage account to store its virtual hard disks.
Virtual network	Yes	The VM must be a member of a virtual network.
Public IP address	No	The VM can have a public IP address assigned to it to remotely access it.
Network interface	Yes	The VM needs the network interface to communicate in the network.
Data disks	No	The VM can include data disks to expand storage capabilities.

Next steps

Create your first VM!

- [Portal](#)
- [PowerShell](#)

- Azure CLI

Quickstart: Create a Windows virtual machine in the Azure portal

11/6/2019 • 2 minutes to read • [Edit Online](#)

Azure virtual machines (VMs) can be created through the Azure portal. This method provides a browser-based user interface to create VMs and their associated resources. This quickstart shows you how to use the Azure portal to deploy a virtual machine (VM) in Azure that runs Windows Server 2019. To see your VM in action, you then RDP to the VM and install the IIS web server.

If you don't have an Azure subscription, create a [free account](#) before you begin.

Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

Create virtual machine

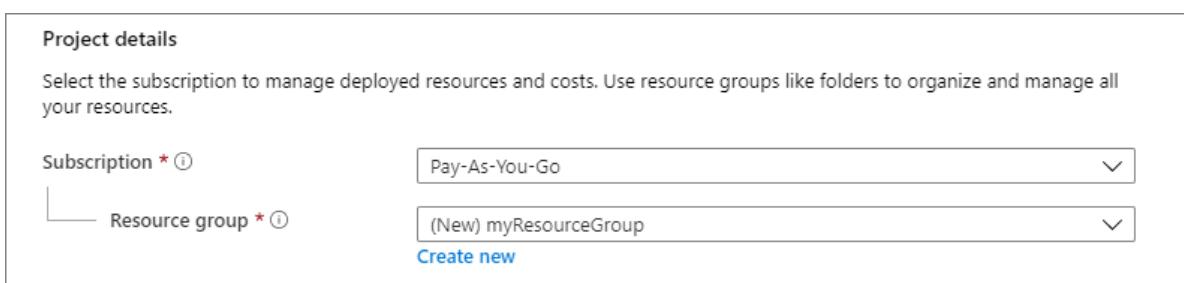
1. Type **virtual machines** in the search.
2. Under **Services**, select **Virtual machines**.
3. In the **Virtual machines** page, select **Add**.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose to **Create new** resource group. Type *myResourceGroup* for the name.

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](#)



5. Under **Instance details**, type *myVM* for the **Virtual machine name** and choose *East US* for your **Region**, and then choose *Windows Server 2019 Datacenter* for the **Image**. Leave the other defaults.

Instance details

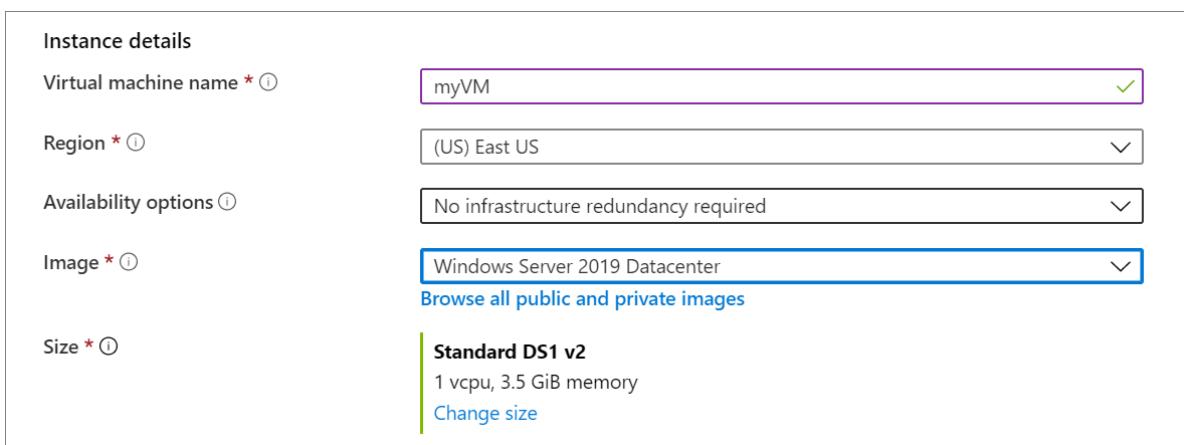
Virtual machine name * ⓘ ✓

Region * ⓘ

Availability options ⓘ

Image * ⓘ [Browse all public and private images](#)

Size * ⓘ
Standard DS1 v2
1 vcpu, 3.5 GiB memory
[Change size](#)



6. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the [defined complexity requirements](#).

Administrator account

Username * ⓘ	azureuser	✓
Password * ⓘ	✓
Confirm password * ⓘ	✓

- Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP (80)** from the drop-down.

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ None Allow selected ports

Select inbound ports *

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

- Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.

Save money

Save up to 49% with a license you already own using Azure Hybrid Benefit. [Learn more](#)

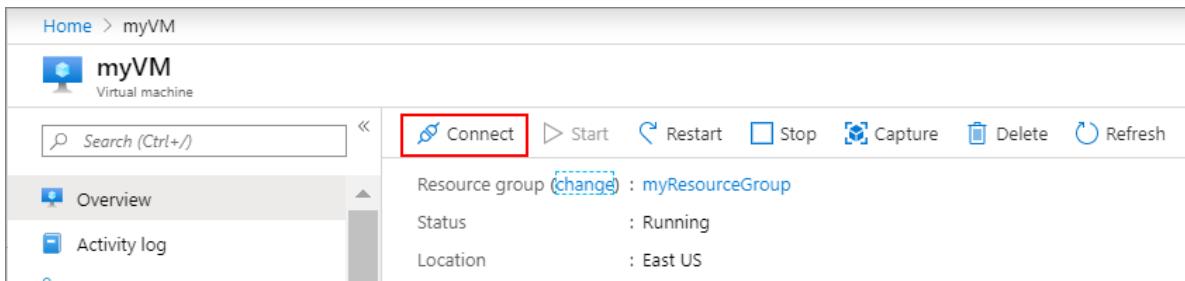
Already have a Windows Server license? * Yes No

Review + create [< Previous](#) [Next : Disks >](#)

Connect to virtual machine

Create a remote desktop connection to the virtual machine. These directions tell you how to connect to your VM from a Windows computer. On a Mac, you need an RDP client such as this [Remote Desktop Client](#) from the Mac App Store.

- Click the **Connect** button on the overview page for your virtual machine.



- In the **Connect to virtual machine** page, keep the default options to connect by IP address, over port 3389, and click **Download RDP file**.
- Open the downloaded RDP file and click **Connect** when prompted.
- In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the

username as **localhost\username**, enter password you created for the virtual machine, and then click **OK**.

5. You may receive a certificate warning during the sign-in process. Click **Yes** or **Continue** to create the connection.

Install web server

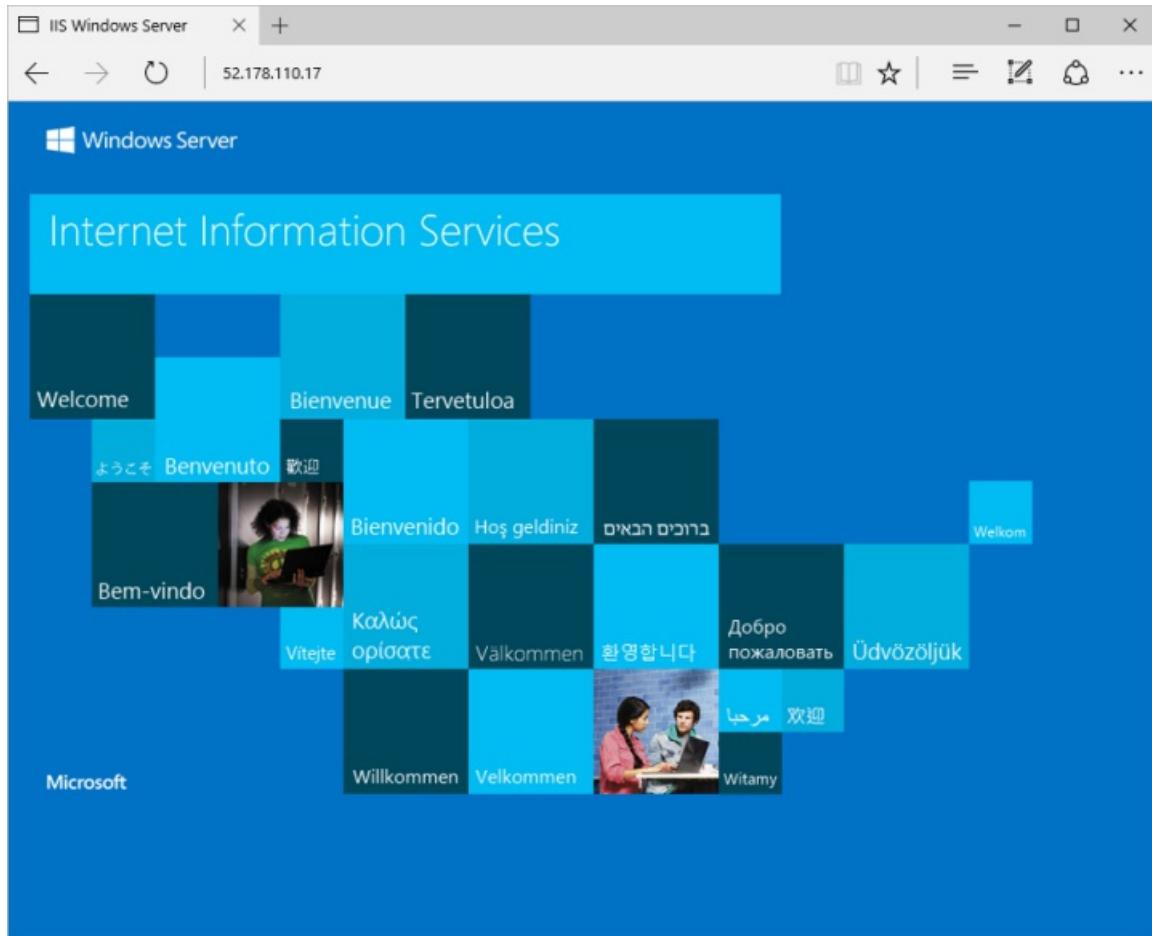
To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

When done, close the RDP connection to the VM.

View the IIS welcome page

In the portal, select the VM and in the overview of the VM, use the **Click to copy** button to the right of the IP address to copy it and paste it into a browser tab. The default IIS welcome page will open, and should look like this:



Clean up resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources.

Select the resource group for the virtual machine, then select **Delete**. Confirm the name of the resource group to finish deleting the resources.

Next steps

In this quickstart, you deployed a simple virtual machine, open a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Windows VMs.

[Azure Windows virtual machine tutorials](#)

Quickstart: Create a Windows virtual machine in Azure with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to deploy a virtual machine (VM) in Azure that runs Windows Server 2016. You will also RDP to the VM and install the IIS web server, to show the VM in action.

If you don't have an Azure subscription, create a [free account](#) before you begin.

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.

Create resource group

Create an Azure resource group with [New-AzResourceGroup](#). A resource group is a logical container into which Azure resources are deployed and managed.

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS
```

Create virtual machine

Create a VM with [New-AzVm](#). Provide names for each of the resources and the `New-AzVm` cmdlet creates if they don't already exist.

When prompted, provide a username and password to be used as the sign-in credentials for the VM:

```
New-AzVm ` 
  -ResourceGroupName "myResourceGroup" ` 
  -Name "myVM" ` 
  -Location "East US" ` 
  -VirtualNetworkName "myVnet" ` 
  -SubnetName "mySubnet" ` 
  -SecurityGroupName "myNetworkSecurityGroup" ` 
  -PublicIpAddressName "myPublicIpAddress" ` 
  -OpenPorts 80,3389
```

Connect to virtual machine

After the deployment has completed, RDP to the VM. To see your VM in action, the IIS web server is then installed.

To see the public IP address of the VM, use the [Get-AzPublicIpAddress](#) cmdlet:

```
Get-AzPublicIpAddress -ResourceGroupName "myResourceGroup" | Select "IpAddress"
```

Use the following command to create a remote desktop session from your local computer. Replace the IP address with the public IP address of your VM.

```
mstsc /v:publicIpAddress
```

In the **Windows Security** window, select **More choices**, and then select **Use a different account**. Type the username as **localhost\username**, enter password you created for the virtual machine, and then click **OK**.

You may receive a certificate warning during the sign-in process. Click **Yes** or **Continue** to create the connection

Install web server

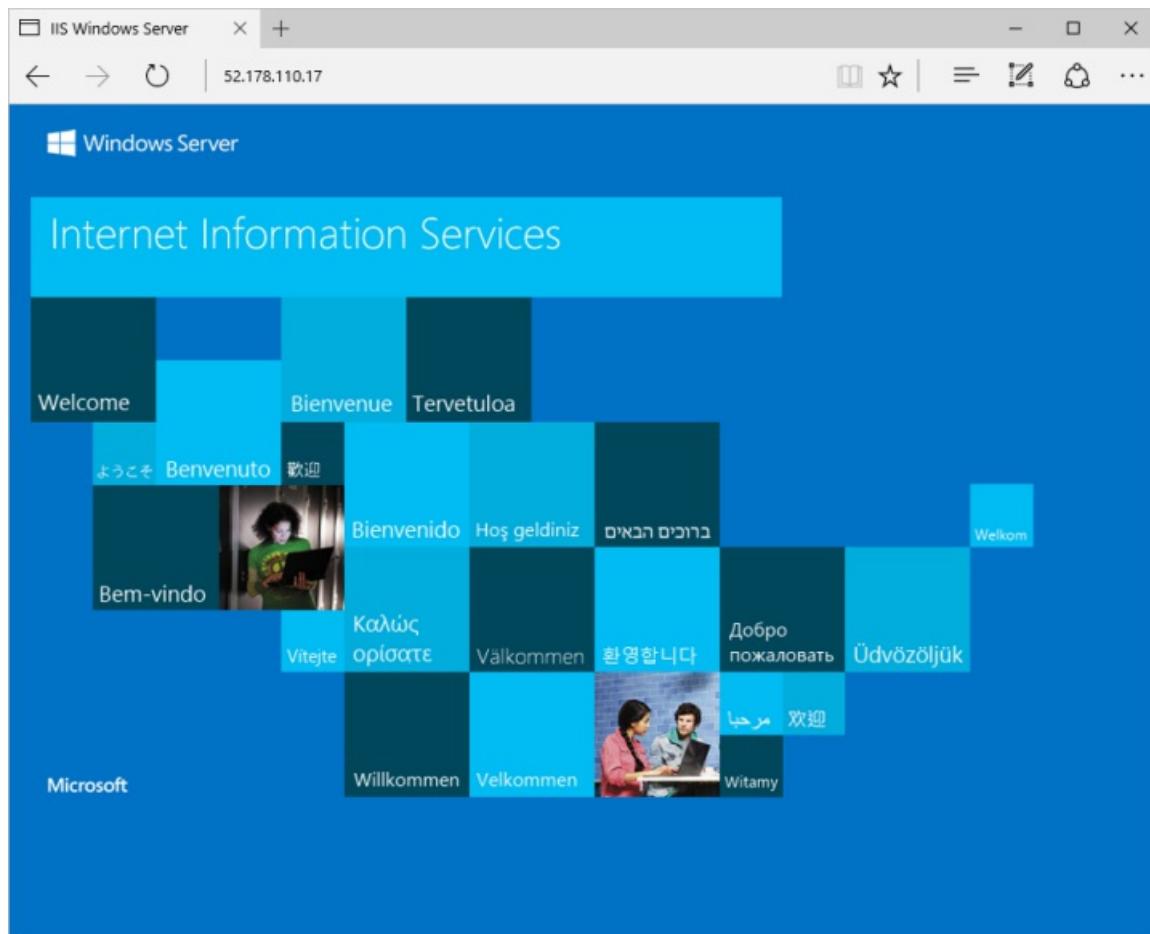
To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

When done, close the RDP connection to the VM.

View the web server in action

With IIS installed and port 80 now open on your VM from the Internet, use a web browser of your choice to view the default IIS welcome page. Use the public IP address of your VM obtained in a previous step. The following example shows the default IIS web site:



Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) cmdlet to remove the resource group, VM, and all related resources:

```
Remove-AzResourceGroup -Name myResourceGroup
```

Next steps

In this quickstart, you deployed a simple virtual machine, open a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Windows VMs.

[Azure Windows virtual machine tutorials](#)

Quickstart: Create a Windows virtual machine with the Azure CLI

11/13/2019 • 3 minutes to read • [Edit Online](#)

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quickstart shows you how to use the Azure CLI to deploy a virtual machine (VM) in Azure that runs Windows Server 2016. To see your VM in action, you then RDP to the VM and install the IIS web server.

If you don't have an Azure subscription, create a [free account](#) before you begin.

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/bash>. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Create a resource group

Create a resource group with the [az group create](#) command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

```
az group create --name myResourceGroup --location eastus
```

Create virtual machine

Create a VM with [az vm create](#). The following example creates a VM named *myVM*. This example uses *azureuser* for an administrative user name.

You must change the value for `--admin-password` or it will fail. Change it to a password that meets the [password requirements for Azure VMs](#). The user name and password will be used later, when you connect to the VM.

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image win2016datacenter \
  --admin-username azureuser \
  --admin-password myPassword
```

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

```
{  
    "fqdns": "",  
    "id":  
        "/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",  
    "location": "eastus",  
    "macAddress": "00-0D-3A-23-9A-49",  
    "powerState": "VM running",  
    "privateIpAddress": "10.0.0.4",  
    "publicIpAddress": "52.174.34.95",  
    "resourceGroup": "myResourceGroup"  
}
```

Note your own `publicIpAddress` in the output from your VM. This address is used to access the VM in the next steps.

Open port 80 for web traffic

By default, only RDP connections are opened when you create a Windows VM in Azure. Use [az vm open-port](#) to open TCP port 80 for use with the IIS web server:

```
az vm open-port --port 80 --resource-group myResourceGroup --name myVM
```

Connect to virtual machine

Use the following command to create a remote desktop session from your local computer. Replace the IP address with the public IP address of your VM. When prompted, enter the credentials used when the VM was created:

```
mstsc /v:publicIpAddress
```

Install web server

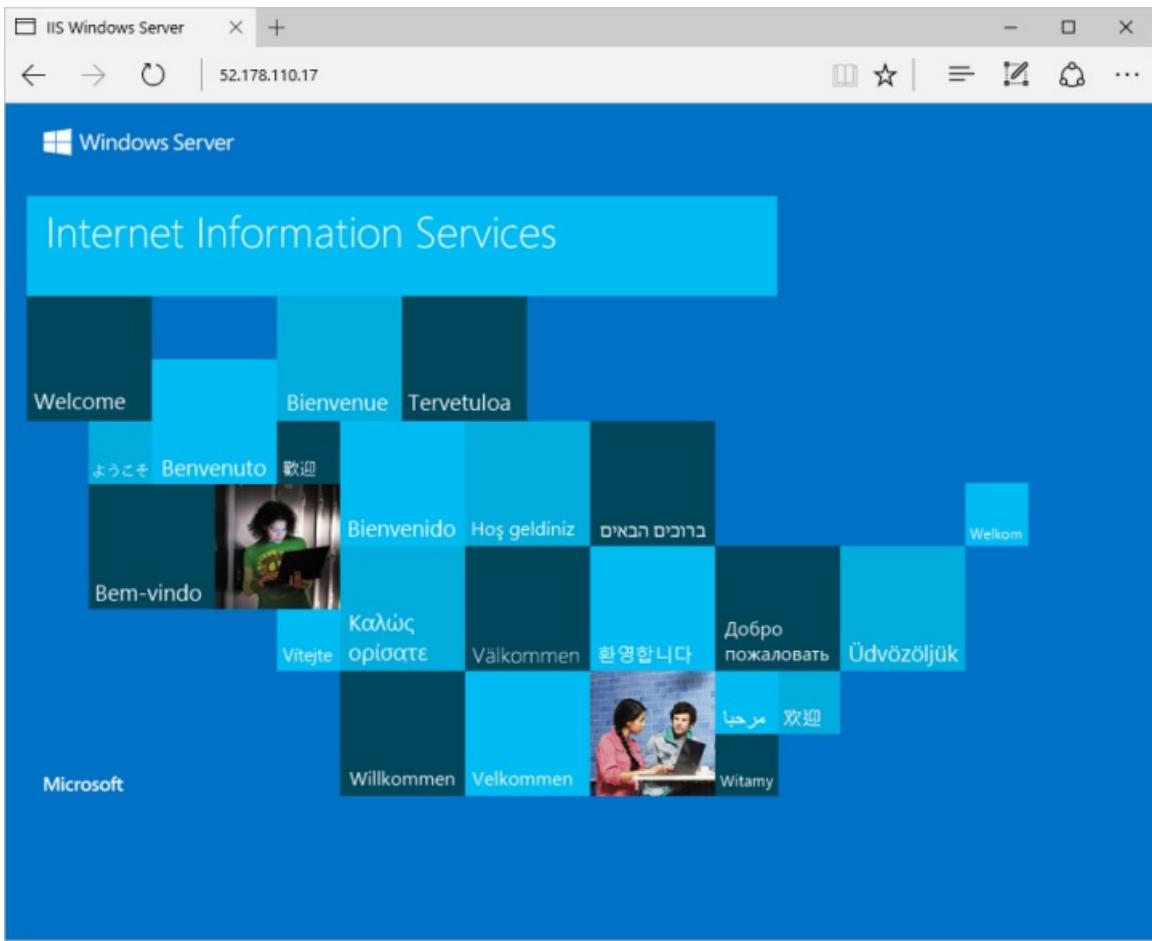
To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

When done, close the RDP connection to the VM.

View the web server in action

With IIS installed and port 80 now open on your VM from the Internet, use a web browser of your choice to view the default IIS welcome page. Use the public IP address of your VM obtained in a previous step. The following example shows the default IIS web site:



Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group, VM, and all related resources:

```
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you deployed a simple virtual machine, open a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Windows VMs.

[Azure Windows virtual machine tutorials](#)

Tutorial: Create and Manage Windows VMs with Azure PowerShell

11/13/2019 • 7 minutes to read • [Edit Online](#)

Azure virtual machines provide a fully configurable and flexible computing environment. This tutorial covers basic Azure virtual machine (VM) deployment tasks like selecting a VM size, selecting a VM image, and deploying a VM. You learn how to:

- Create and connect to a VM
- Select and use VM images
- View and use specific VM sizes
- Resize a VM
- View and understand VM state

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.

Create resource group

Create a resource group with the [New-AzResourceGroup](#) command.

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine. In the following example, a resource group named *myResourceGroupVM* is created in the *EastUS* region:

```
New-AzResourceGroup ` 
    -ResourceGroupName "myResourceGroupVM" ` 
    -Location "EastUS"
```

The resource group is specified when creating or modifying a VM, which can be seen throughout this tutorial.

Create a VM

When creating a VM, several options are available like operating system image, network configuration, and administrative credentials. This example creates a VM named *myVM*, running the default version of Windows Server 2016 Datacenter.

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

```
$cred = Get-Credential
```

Create the VM with [New-AzVM](#).

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroupVM" ` 
    -Name "myVM" ` 
    -Location "EastUS" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIpAddress" ` 
    -Credential $cred
```

Connect to VM

After the deployment has completed, create a remote desktop connection with the VM.

Run the following commands to return the public IP address of the VM. Take note of this IP Address so you can connect to it with your browser to test web connectivity in a future step.

```
Get-AzPublicIpAddress ` 
    -ResourceGroupName "myResourceGroupVM" | Select IpAddress
```

Use the following command, on your local machine, to create a remote desktop session with the VM. Replace the IP address with the *publicIPAddress* of your VM. When prompted, enter the credentials used when creating the VM.

```
mstsc /v:<publicIpAddress>
```

In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username and password you created for the VM and then click **OK**.

Understand marketplace images

The Azure marketplace includes many images that can be used to create a new VM. In the previous steps, a VM was created using the Windows Server 2016 Datacenter image. In this step, the PowerShell module is used to search the marketplace for other Windows images, which can also be used as a base for new VMs. This process consists of finding the publisher, offer, SKU, and optionally a version number to [identify](#) the image.

Use the [Get-AzVMImagePublisher](#) command to return a list of image publishers:

```
Get-AzVMImagePublisher -Location "EastUS"
```

Use the [Get-AzVMImageOffer](#) to return a list of image offers. With this command, the returned list is filtered on the specified publisher named `MicrosoftWindowsServer`:

```
Get-AzVMImageOffer ` 
    -Location "EastUS" ` 
    -PublisherName "MicrosoftWindowsServer"
```

The results will look something like this example:

Offer	PublisherName	Location
Windows-HUB	MicrosoftWindowsServer	EastUS
WindowsServer	MicrosoftWindowsServer	EastUS
WindowsServer-HUB	MicrosoftWindowsServer	EastUS

The [Get-AzVMImageSku](#) command will then filter on the publisher and offer name to return a list of image names.

```
Get-AzVMImageSku ` 
    -Location "EastUS" ` 
    -PublisherName "MicrosoftWindowsServer" ` 
    -Offer "WindowsServer"
```

The results will look something like this example:

Skus	Offer	PublisherName	Location
2008-R2-SP1	WindowsServer	MicrosoftWindowsServer	EastUS
2008-R2-SP1-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2012-Datacenter	WindowsServer	MicrosoftWindowsServer	EastUS
2012-Datacenter-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2012-R2-Datacenter	WindowsServer	MicrosoftWindowsServer	EastUS
2012-R2-Datacenter-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-Server-Core	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-Server-Core-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-with-Containers	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-with-Containers-smalldisk	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Datacenter-with-RDSH	WindowsServer	MicrosoftWindowsServer	EastUS
2016-Nano-Server	WindowsServer	MicrosoftWindowsServer	EastUS

This information can be used to deploy a VM with a specific image. This example deploys a VM using the latest version of a Windows Server 2016 with Containers image.

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroupVM" ` 
    -Name "myVM2" ` 
    -Location "EastUS" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIpAddress2" ` 
    -ImageName "MicrosoftWindowsServer:WindowsServer:2016-Datacenter-with-Containers:latest" ` 
    -Credential $cred ` 
    -AsJob
```

The `-AsJob` parameter creates the VM as a background task, so the PowerShell prompts return to you. You can view details of background jobs with the [Get-Job](#) cmdlet.

Understand VM sizes

The VM size determines the amount of compute resources like CPU, GPU, and memory that are made available to the VM. Virtual machines should be created using a VM size appropriate for the workload. If a workload increases, an existing virtual machine can also be resized.

VM Sizes

The following table categorizes sizes into use cases.

Type	Common Sizes	Description
General purpose	B, Dsv3, Dv3, DSv2, Dv2, Av2, DC	Balanced CPU-to-memory. Ideal for dev / test and small to medium applications and data solutions.
Compute optimized	Fsv2	High CPU-to-memory. Good for medium traffic applications, network appliances, and batch processes.
Memory optimized	Esv3, Ev3, M, DSv2, Dv2	High memory-to-core. Great for relational databases, medium to large caches, and in-memory analytics.
Storage optimized	Lsv2, Ls	High disk throughput and IO. Ideal for Big Data, SQL, and NoSQL databases.
GPU	NV, NVv2, NC, NCv2, NCv3, ND	Specialized VMs targeted for heavy graphic rendering and video editing.
High performance	H	Our most powerful CPU VMs with optional high-throughput network interfaces (RDMA).

Find available VM sizes

To see a list of VM sizes available in a particular region, use the [Get-AzVMSize](#) command.

```
Get-AzVMSize -Location "EastUS"
```

Resize a VM

After a VM has been deployed, it can be resized to increase or decrease resource allocation.

Before resizing a VM, check if the size you want is available on the current VM cluster. The [Get-AzVMSize](#) command returns a list of sizes.

```
Get-AzVMSize -ResourceGroupName "myResourceGroupVM" -VMName "myVM"
```

If the size is available, the VM can be resized from a powered-on state, however it is rebooted during the operation.

```
$vm = Get-AzVM ` 
    -ResourceGroupName "myResourceGroupVM" ` 
    -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_DS3_v2"
Update-AzVM ` 
    -VM $vm ` 
    -ResourceGroupName "myResourceGroupVM"
```

If the size you want isn't available on the current cluster, the VM needs to be deallocated before the resize operation can occur. Deallocating a VM will remove any data on the temp disk, and the public IP address will change unless a static IP address is being used.

```

Stop-AzVM ` 
  -ResourceGroupName "myResourceGroupVM" ` 
  -Name "myVM" -Force
$vm = Get-AzVM ` 
  -ResourceGroupName "myResourceGroupVM" ` 
  -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_E2s_v3"
Update-AzVM -VM $vm ` 
  -ResourceGroupName "myResourceGroupVM"
Start-AzVM ` 
  -ResourceGroupName "myResourceGroupVM" ` 
  -Name $vm.name

```

VM power states

An Azure VM can have one of many power states.

POWER STATE	DESCRIPTION
Starting	The virtual machine is being started.
Running	The virtual machine is running.
Stopping	The virtual machine is being stopped.
Stopped	The VM is stopped. Virtual machines in the stopped state still incur compute charges.
Deallocating	The VM is being deallocated.
Deallocated	Indicates that the VM is removed from the hypervisor but is still available in the control plane. Virtual machines in the Deallocated state do not incur compute charges.
-	The power state of the VM is unknown.

To get the state of a particular VM, use the [Get-AzVM](#) command. Be sure to specify a valid name for a VM and resource group.

```

Get-AzVM ` 
  -ResourceGroupName "myResourceGroupVM" ` 
  -Name "myVM" ` 
  -Status | Select @n="Status"; e={$_.Statuses[1].Code}

```

The output will look something like this example:

```

Status
-----
PowerState/running

```

Management tasks

During the lifecycle of a VM, you may want to run management tasks like starting, stopping, or deleting a VM. Additionally, you may want to create scripts to automate repetitive or complex tasks. Using Azure PowerShell,

many common management tasks can be run from the command line or in scripts.

Stop a VM

Stop and deallocate a VM with [Stop-AzVM](#):

```
Stop-AzVM ` 
-ResourceGroupName "myResourceGroupVM" ` 
-Name "myVM" -Force
```

If you want to keep the VM in a provisioned state, use the `-StayProvisioned` parameter.

Start a VM

```
Start-AzVM ` 
-ResourceGroupName "myResourceGroupVM" ` 
-Name "myVM"
```

Delete resource group

Everything inside of a resource group is deleted when you delete the resource group.

```
Remove-AzResourceGroup ` 
-Name "myResourceGroupVM" ` 
-Force
```

Next steps

In this tutorial, you learned about basic VM creation and management such as how to:

- Create and connect to a VM
- Select and use VM images
- View and use specific VM sizes
- Resize a VM
- View and understand VM state

Advance to the next tutorial to learn about VM disks.

[Create and Manage VM disks](#)

Tutorial - Manage Azure disks with Azure PowerShell

1/19/2020 • 5 minutes to read • [Edit Online](#)

Azure virtual machines use disks to store the VMs operating system, applications, and data. When creating a VM, it's important to choose a disk size and configuration appropriate to the expected workload. This tutorial covers deploying and managing VM disks. You learn about:

- OS disks and temporary disks
- Data disks
- Standard and Premium disks
- Disk performance
- Attaching and preparing data disks

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

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,02 4	2,04 8	4,09 6	8,19 2	16,3 84	32,7 67
IOP S per disk	120	120	120	120	240	500	1,10 0	2,30 0	5,00 0	7,50 0	7,50 0	16,0 00	18,0 00	20,0 00
Thr oug hpu t per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max bur st IOP S per disk **	3,5 00	3,5 00	3,5 00	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0						
Max bur st thro ugh put per disk **	170 MiB /sec													
Max bur st dur atio n**	30 min													

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year					

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

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 :
```

Next steps

In this tutorial, you learned about VM disks topics such as:

- OS disks and temporary disks
- Data disks
- Standard and Premium disks

- Disk performance
- Attaching and preparing data disks

Advance to the next tutorial to learn about automating VM configuration.

[Automate VM configuration](#)

Tutorial - Deploy applications to a Windows virtual machine in Azure with the Custom Script Extension

11/13/2019 • 2 minutes to read • [Edit Online](#)

To configure virtual machines (VMs) in a quick and consistent manner, you can use the [Custom Script Extension for Windows](#). In this tutorial you learn how to:

- Use the Custom Script Extension to install IIS
- Create a VM that uses the Custom Script Extension
- View a running IIS site after the extension is applied

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.

Custom script extension overview

The Custom Script Extension downloads and executes scripts on Azure VMs. This extension is useful for post deployment configuration, software installation, or any other configuration / management task. Scripts can be downloaded from Azure storage or GitHub, or provided to the Azure portal at extension run time.

The Custom Script extension integrates with Azure Resource Manager templates, and can also be run using the Azure CLI, PowerShell, Azure portal, or the Azure Virtual Machine REST API.

You can use the Custom Script Extension with both Windows and Linux VMs.

Create virtual machine

Set the administrator username and password for the VM with [Get-Credential](#):

```
$cred = Get-Credential
```

Now you can create the VM with [New-AzVM](#). The following example creates a VM named *myVM* in the *EastUS* location. If they do not already exist, the resource group *myResourceGroupAutomate* and supporting network resources are created. To allow web traffic, the cmdlet also opens port 80.

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroupAutomate" ` 
    -Name "myVM" ` 
    -Location "East US" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIpAddress" ` 
    -OpenPorts 80 ` 
    -Credential $cred
```

It takes a few minutes for the resources and VM to be created.

Automate IIS install

Use [Set-AzVMExtension](#) to install the Custom Script Extension. The extension runs

```
powershell Add-WindowsFeature Web-Server
```

 to install the IIS webserver and then updates the *Default.htm* page to show the hostname of the VM:

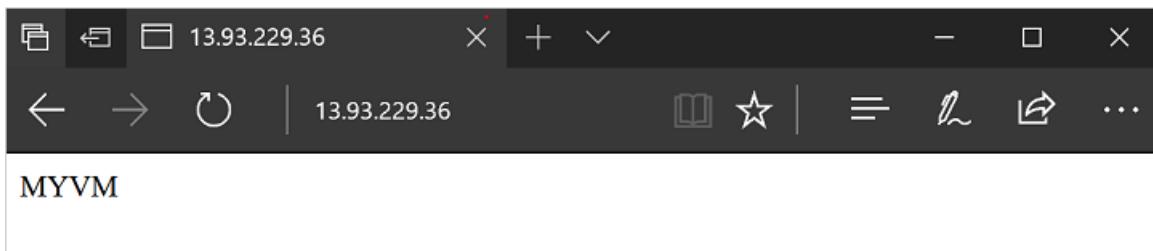
```
Set-AzVMExtension -ResourceGroupName "myResourceGroupAutomate" ` 
    -ExtensionName "IIS" ` 
    -VMName "myVM" ` 
    -Location "EastUS" ` 
    -Publisher Microsoft.Compute ` 
    -ExtensionType CustomScriptExtension ` 
    -TypeHandlerVersion 1.8 ` 
    -SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -Value $($env:computername)"}'
```

Test web site

Obtain the public IP address of your load balancer with [Get-AzPublicIPAddress](#). The following example obtains the IP address for *myPublicIPAddress* created earlier:

```
Get-AzPublicIPAddress ` 
    -ResourceGroupName "myResourceGroupAutomate" ` 
    -Name "myPublicIPAddress" | select IpAddress
```

You can then enter the public IP address in to a web browser. The website is displayed, including the hostname of the VM that the load balancer distributed traffic to as in the following example:



Next steps

In this tutorial, you automated the IIS install on a VM. You learned how to:

- Use the Custom Script Extension to install IIS
- Create a VM that uses the Custom Script Extension
- View a running IIS site after the extension is applied

Advance to the next tutorial to learn how to create custom VM images.

[Create custom VM images](#)

Tutorial: Create a custom image of an Azure VM with Azure PowerShell

1/19/2020 • 4 minutes to read • [Edit Online](#)

Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap deployments and ensure consistency across multiple VMs. In this tutorial, you create your own custom image of an Azure virtual machine using PowerShell. You learn how to:

- Sysprep and generalize VMs
- Create a custom image
- Create a VM from a custom image
- List all the images in your subscription
- Delete an image

In public preview, we have the [Azure VM Image Builder](#) service. Simply describe your customizations in a template, and it will handle the image creation steps in this article. [Try Azure Image Builder \(preview\)](#).

Before you begin

The steps below detail how to take an existing VM and turn it into a re-usable custom image that you can use to create new VM instances.

To complete the example in this tutorial, you must have an existing virtual machine. If needed, this [script sample](#) can create one for you. When working through the tutorial, replace the resource group and VM names where needed.

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.

Prepare VM

To create an image of a virtual machine, you need to prepare the source VM by generalizing it, deallocating, and then marking it as generalized with Azure.

Generalize the Windows VM using Sysprep

Sysprep removes all your personal account information, among other things, and prepares the machine to be used as an image. For details about Sysprep, see [How to Use Sysprep: An Introduction](#).

1. Connect to the virtual machine.
2. Open the Command Prompt window as an administrator. Change the directory to %windir%\system32\sysprep, and then run `sysprep.exe`.
3. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that the **Generalize** check box is selected.
4. In **Shutdown Options**, select **Shutdown** and then click **OK**.

5. When Sysprep completes, it shuts down the virtual machine. **Do not restart the VM.**

Deallocate and mark the VM as generalized

To create an image, the VM needs to be deallocated and marked as generalized in Azure.

Deallocate the VM using [Stop-AzVM](#).

```
Stop-AzVM ` 
  -ResourceGroupName myResourceGroup ` 
  -Name myVM -Force
```

Set the status of the virtual machine to `-Generalized` using [Set-AzVm](#).

```
Set-AzVM ` 
  -ResourceGroupName myResourceGroup ` 
  -Name myVM -Generalized
```

Create the image

Now you can create an image of the VM by using [New-AzImageConfig](#) and [New-AzImage](#). The following example creates an image named *myImage* from a VM named *myVM*.

Get the virtual machine.

```
$vm = Get-AzVM ` 
  -Name myVM ` 
  -ResourceGroupName myResourceGroup
```

Create the image configuration.

```
$image = New-AzImageConfig ` 
  -Location EastUS ` 
  -SourceVirtualMachineId $vm.ID
```

Create the image.

```
New-AzImage ` 
  -Image $image ` 
  -ImageName myImage ` 
  -ResourceGroupName myResourceGroup
```

Create VMs from the image

Now that you have an image, you can create one or more new VMs from the image. Creating a VM from a custom image is similar to creating a VM using a Marketplace image. When you use a Marketplace image, you have to provide the information about the image, image provider, offer, SKU, and version. Using the simplified parameter set for the [New-AzVM](#) cmdlet, you just need to provide the name of the custom image as long as it is in the same resource group.

This example creates a VM named *myVMfromImage* from the *myImage* image, in *myResourceGroup*.

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroup" ` 
    -Name "myVMfromImage" ` 
    -ImageName "myImage" ` 
    -Location "East US" ` 
    -VirtualNetworkName "myImageVnet" ` 
    -SubnetName "myImageSubnet" ` 
    -SecurityGroupName "myImageNSG" ` 
    -PublicIpAddressName "myImagePIP" ` 
    -OpenPorts 3389
```

We recommend that you limit the number of concurrent deployments to 20 VMs from a single image. If you are planning large-scale, concurrent deployments of over 20 VMs from the same custom image, you should use a [Shared Image Gallery](#) with multiple image replicas.

Image management

Here are some examples of common managed image tasks and how to complete them using PowerShell.

List all images by name.

```
$images = Get-AzResource -ResourceType Microsoft.Compute/images
$image.name
```

Delete an image. This example deletes the image named *myImage* from the *myResourceGroup*.

```
Remove-AzImage ` 
    -ImageName myImage ` 
    -ResourceGroupName myResourceGroup
```

Next steps

In this tutorial, you created a custom VM image. You learned how to:

- Sysprep and generalize VMs
- Create a custom image
- Create a VM from a custom image
- List all the images in your subscription
- Delete an image

Advance to the next tutorial to learn about how to create highly available virtual machines.

[Create highly available VMs](#)

Tutorial: Create and deploy highly available virtual machines with Azure PowerShell

11/13/2019 • 4 minutes to read • [Edit Online](#)

In this tutorial, you learn how to increase the availability and reliability of your Virtual Machines (VMs) using Availability Sets. Availability Sets make sure the VMs you deploy on Azure are distributed across multiple, isolated hardware nodes, in a cluster.

In this tutorial, you learn how to:

- Create an availability set
- Create a VM in an availability set
- Check available VM sizes
- Check Azure Advisor

Availability set overview

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.

Let's consider a typical VM-based solution where you might have four front-end web servers and 2 back-end VMs. With Azure, you'd want to define two availability sets before you deploy your VMs: one for the web tier and one for the back tier. When you create a new VM, you specify the availability set as a parameter. Azure makes sure the VMs are isolated across multiple physical hardware resources. If the physical hardware that one of your servers is running on has a problem, you know the other instances of your servers will keep running because they're on different hardware.

Use Availability Sets when you want to deploy reliable VM-based solutions in Azure.

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.

Create an availability set

The hardware in a location is divided into multiple update domains and fault domains. An **update domain** is a group of VMs and underlying physical hardware that can be rebooted at the same time. VMs in the same **fault domain** share common storage as well as a common power source and network switch.

You can create an availability set using [New-AzAvailabilitySet](#). In this example, the number of both update and fault domains is 2 and the availability set is named *myAvailabilitySet*.

Create a resource group.

```
New-AzResourceGroup ` 
  -Name myResourceGroupAvailability ` 
  -Location EastUS
```

Create a managed availability set using [New-AzAvailabilitySet](#) with the `-sku aligned` parameter.

```
New-AzAvailabilitySet ` 
  -Location "EastUS" ` 
  -Name "myAvailabilitySet" ` 
  -ResourceGroupName "myResourceGroupAvailability" ` 
  -Sku aligned ` 
  -PlatformFaultDomainCount 2 ` 
  -PlatformUpdateDomainCount 2
```

Create VMs inside an availability set

VMs must be created within the availability set to make sure they're correctly distributed across the hardware. You can't add an existing VM to an availability set after it's created.

When you create a VM with [New-AzVM](#), you use the `-AvailabilitySetName` parameter to specify the name of the availability set.

First, set an administrator username and password for the VM with [Get-Credential](#):

```
$cred = Get-Credential
```

Now create two VMs with [New-AzVM](#) in the availability set.

```
for ($i=1; $i -le 2; $i++) 
{
    New-AzVm ` 
        -ResourceGroupName "myResourceGroupAvailability" ` 
        -Name "myVM$i" ` 
        -Location "East US" ` 
        -VirtualNetworkName "myVnet" ` 
        -SubnetName "mySubnet" ` 
        -SecurityGroupName "myNetworkSecurityGroup" ` 
        -PublicIpAddressName "myPublicIpAddress$i" ` 
        -AvailabilitySetName "myAvailabilitySet" ` 
        -Credential $cred
}
```

It takes a few minutes to create and configure both VMs. When finished, you have two virtual machines distributed across the underlying hardware.

If you look at the availability set in the portal by going to **Resource Groups > myResourceGroupAvailability > myAvailabilitySet**, you should see how the VMs are distributed across the two fault and update domains.

Resource groups > myResourceGroupAvailability > myAvailabilitySet

Search resources, services and docs

myAvailabilitySet

Availability set

Overview

Activity log

Access control (IAM)

Tags

Virtual machines

Properties

Locks

Automation script

Move Delete

Resource group (change) myResourceGroupAvailability

Location East US

Subscription name (change) Azure

Subscription ID <Subscription ID>

Fault domains 2

Update domains 2

Virtual machines 2

Managed Yes

Search virtual machines

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
myVM1	Running	0	0
myVM2	Running	1	1

Check for available VM sizes

You can add more VMs to the availability set later, but you need to know what VM sizes are available on the hardware. Use [Get-AzVmSize](#) to list all the available sizes on the hardware cluster for the availability set.

```
Get-AzVmSize  
-ResourceGroupName "myResourceGroupAvailability"  
-AvailabilitySetName "myAvailabilitySet"
```

Check Azure Advisor

You can also use Azure Advisor to get more information on how to improve the availability of your VMs. Azure Advisor analyzes your configuration and usage telemetry, then recommends solutions that can help you improve the cost effectiveness, performance, availability, and security of your Azure resources.

Sign in to the [Azure portal](#), select **All services**, and type **Advisor**. The Advisor dashboard shows personalized recommendations for the selected subscription. For more information, see [Get started with Azure Advisor](#).

Next steps

In this tutorial, you learned how to:

- Create an availability set
- Create a VM in an availability set
- Check available VM sizes
- Check Azure Advisor

Advance to the next tutorial to learn about virtual machine scale sets.

[Create a VM scale set](#)

Tutorial: Create a virtual machine scale set and deploy a highly available app on Windows with Azure PowerShell

12/12/2019 • 7 minutes to read • [Edit Online](#)

A virtual machine scale set allows you to deploy and manage a set of identical, autoscaling virtual machines. You can scale the number of VMs in the scale set manually. You can also define rules to autoscale based on resource usage such as CPU, memory demand, or network traffic. In this tutorial, you deploy a virtual machine scale set in Azure and learn how to:

- Use the Custom Script Extension to define an IIS site to scale
- Create a load balancer for your scale set
- Create a virtual machine scale set
- Increase or decrease the number of instances in a scale set
- Create autoscale rules

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.

Scale Set overview

A virtual machine scale set allows you to deploy and manage a set of identical, autoscaling virtual machines. VMs in a scale set are distributed across logic fault and update domains in one or more *placement groups*. Placement groups are groups of similarly configured VMs, similar to [availability sets](#).

VMs are created as needed in a scale set. You define autoscale rules to control how and when VMs are added or removed from the scale set. These rules can trigger based on metrics such as CPU load, memory usage, or network traffic.

Scale sets support up to 1,000 VMs when you use an Azure platform image. For workloads with significant installation or VM customization requirements, you may wish to [Create a custom VM image](#). You can create up to 300 VMs in a scale set when using a custom image.

Create a scale set

Create a virtual machine scale set with [New-AzVms](#). The following example creates a scale set named *myScaleSet* that uses the *Windows Server 2016 Datacenter* platform image. The Azure network resources for virtual network, public IP address, and load balancer are automatically created. When prompted, you can set your own administrative credentials for the VM instances in the scale set:

```
New-AzVmss ` 
-ResourceGroupName "myResourceGroupScaleSet" ` 
-Location "EastUS" ` 
-VMSScaleSetName "myScaleSet" ` 
-VirtualNetworkName "myVnet" ` 
-SubnetName "mySubnet" ` 
-PublicIpAddressName "myPublicIPAddress" ` 
-LoadBalancerName "myLoadBalancer" ` 
-UpgradePolicyMode "Automatic"
```

It takes a few minutes to create and configure all the scale set resources and VMs.

Deploy sample application

To test your scale set, install a basic web application. The Azure Custom Script Extension is used to download and run a script that installs IIS on the VM instances. This extension is useful for post deployment configuration, software installation, or any other configuration / management task. For more information, see the [Custom Script Extension overview](#).

Use the Custom Script Extension to install a basic IIS web server. Apply the Custom Script Extension that installs IIS as follows:

```
# Define the script for your Custom Script Extension to run
$publicSettings = @{
    "fileUris" = (,"https://raw.githubusercontent.com/Azure-Samples/compute-automation-configurations/master/automate-iis.ps1");
    "commandToExecute" = "powershell -ExecutionPolicy Unrestricted -File automate-iis.ps1"
}

# Get information about the scale set
$vmss = Get-AzVmss ` 
-ResourceGroupName "myResourceGroupScaleSet" ` 
-VMSScaleSetName "myScaleSet"

# Use Custom Script Extension to install IIS and configure basic website
Add-AzVmssExtension -VirtualMachineScaleSet $vmss ` 
-Name "customScript" ` 
-Publisher "Microsoft.Compute" ` 
-Type "CustomScriptExtension" ` 
-TypeHandlerVersion 1.8 ` 
-Setting $publicSettings

# Update the scale set and apply the Custom Script Extension to the VM instances
Update-AzVmss ` 
-ResourceGroupName "myResourceGroupScaleSet" ` 
-Name "myScaleSet" ` 
-VirtualMachineScaleSet $vmss
```

Allow traffic to application

To allow access to the basic web application, create a network security group with [New-AzNetworkSecurityRuleConfig](#) and [New-AzNetworkSecurityGroup](#). For more information, see [Networking for Azure virtual machine scale sets](#).

```

# Get information about the scale set
$vmss = Get-AzVmss `

-ResourceGroupName "myResourceGroupScaleSet" `

-VMScaleSetName "myScaleSet"

#Create a rule to allow traffic over port 80
$nsgFrontendRule = New-AzNetworkSecurityRuleConfig `

-Name myFrontendNSGRule `

-Protocol Tcp `

-Direction Inbound `

-Priority 200 `

-SourceAddressPrefix * `

-SourcePortRange * `

-DestinationAddressPrefix * `

-DestinationPortRange 80 `

-Access Allow

#Create a network security group and associate it with the rule
$nsgFrontend = New-AzNetworkSecurityGroup `

-ResourceGroupName "myResourceGroupScaleSet" `

-Location EastUS `

-Name myFrontendNSG `

-SecurityRules $nsgFrontendRule

$vnet = Get-AzVirtualNetwork `

-ResourceGroupName "myResourceGroupScaleSet" `

-Name myVnet

$frontendSubnet = $vnet.Subnets[0]

$frontendSubnetConfig = Set-AzVirtualNetworkSubnetConfig `

-VirtualNetwork $vnet `

-Name mySubnet `

-AddressPrefix $frontendSubnet.AddressPrefix `

-NetworkSecurityGroup $nsgFrontend

Set-AzVirtualNetwork -VirtualNetwork $vnet

# Update the scale set and apply the Custom Script Extension to the VM instances
Update-AzVmss `

-ResourceGroupName "myResourceGroupScaleSet" `

-Name "myScaleSet" `

-VirtualMachineScaleSet $vmss

```

Test your scale set

To see your scale set in action, get the public IP address of your load balancer with [Get-AzPublicIPAddress](#). The following example displays the IP address for *myPublicIP* created as part of the scale set:

```

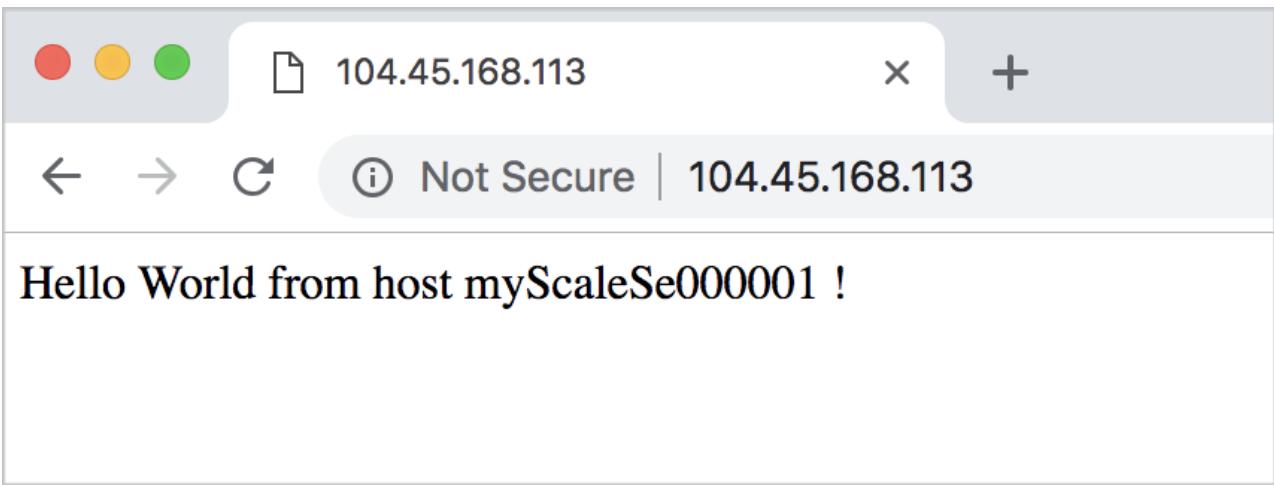
Get-AzPublicIPAddress `

-ResourceGroupName "myResourceGroupScaleSet" `

-Name "myPublicIPAddress" | select IpAddress

```

Enter the public IP address in to a web browser. The web app is displayed, including the hostname of the VM that the load balancer distributed traffic to:



To see the scale set in action, you can force-refresh your web browser to see the load balancer distribute traffic across all the VMs running your app.

Management tasks

Throughout the lifecycle of the scale set, you may need to run one or more management tasks. Additionally, you may want to create scripts that automate various lifecycle-tasks. Azure PowerShell provides a quick way to do those tasks. Here are a few common tasks.

View VMs in a scale set

To view a list of VM instances in a scale set, use [Get-AzVmssVM](#) as follows:

```
Get-AzVmssVM ` 
-ResourceGroupName "myResourceGroupScaleSet" ` 
-VMScaleSetName "myScaleSet"
```

The following example output shows two VM instances in the scale set:

ResourceGroupName	Name	Location	Sku	InstanceID	ProvisioningState
MYRESOURCEGROUPSCALESET	myScaleSet_0	eastus	Standard_DS1_v2	0	Succeeded
MYRESOURCEGROUPSCALESET	myScaleSet_1	eastus	Standard_DS1_v2	1	Succeeded

To view additional information about a specific VM instance, add the `-InstanceId` parameter to [Get-AzVmssVM](#).

The following example views information about VM instance 1:

```
Get-AzVmssVM ` 
-ResourceGroupName "myResourceGroupScaleSet" ` 
-VMScaleSetName "myScaleSet" ` 
-InstanceId "1"
```

Increase or decrease VM instances

To see the number of instances you currently have in a scale set, use [Get-AzVmss](#) and query on `sku.capacity`:

```
Get-AzVmss -ResourceGroupName "myResourceGroupScaleSet" ` 
-VMScaleSetName "myScaleSet" | ` 
Select -ExpandProperty Sku
```

You can then manually increase or decrease the number of virtual machines in the scale set with [Update-AzVmss](#). The following example sets the number of VMs in your scale set to 3:

```
# Get current scale set
$scaleset = Get-AzVmss `

-ResourceGroupName "myResourceGroupScaleSet" `

-VMScaleSetName "myScaleSet"

# Set and update the capacity of your scale set
$scaleset.sku.capacity = 3

Update-AzVmss -ResourceGroupName "myResourceGroupScaleSet" `

-Name "myScaleSet" `

-VirtualMachineScaleSet $scaleset
```

If takes a few minutes to update the specified number of instances in your scale set.

Configure autoscale rules

Rather than manually scaling the number of instances in your scale set, you define autoscale rules. These rules monitor the instances in your scale set and respond accordingly based on metrics and thresholds you define. The following example scales out the number of instances by one when the average CPU load is greater than 60% over a 5-minute period. If the average CPU load then drops below 30% over a 5-minute period, the instances are scaled in by one instance:

```

# Define your scale set information
$mySubscriptionId = (Get-AzSubscription)[0].Id
$myResourceGroup = "myResourceGroupScaleSet"
$myScaleSet = "myScaleSet"
$myLocation = "East US"
$myScaleSetId = (Get-AzVmss -ResourceGroupName $myResourceGroup -VMScaleSetName $myScaleSet).Id

# Create a scale up rule to increase the number instances after 60% average CPU usage exceeded for a 5-minute
period
$myRuleScaleUp = New-AzAutoscaleRule `

    -MetricName "Percentage CPU" `

    -MetricResourceId $myScaleSetId `

    -Operator GreaterThan `

    -MetricStatistic Average `

    -Threshold 60 `

    -TimeGrain 00:01:00 `

    -TimeWindow 00:05:00 `

    -ScaleActionCooldown 00:05:00 `

    -ScaleActionDirection Increase `

    -ScaleActionValue 1

# Create a scale down rule to decrease the number of instances after 30% average CPU usage over a 5-minute
period
$myRuleScaleDown = New-AzAutoscaleRule `

    -MetricName "Percentage CPU" `

    -MetricResourceId $myScaleSetId `

    -Operator LessThan `

    -MetricStatistic Average `

    -Threshold 30 `

    -TimeGrain 00:01:00 `

    -TimeWindow 00:05:00 `

    -ScaleActionCooldown 00:05:00 `

    -ScaleActionDirection Decrease `

    -ScaleActionValue 1

# Create a scale profile with your scale up and scale down rules
$myScaleProfile = New-AzAutoscaleProfile `

    -DefaultCapacity 2 `

    -MaximumCapacity 10 `

    -MinimumCapacity 2 `

    -Rule $myRuleScaleUp,$myRuleScaleDown `

    -Name "autoprofile"

# Apply the autoscale rules
Add-AzAutoscaleSetting `

    -Location $myLocation `

    -Name "autosetting" `

    -ResourceGroup $myResourceGroup `

    -TargetResourceId $myScaleSetId `

    -AutoscaleProfile $myScaleProfile

```

For more design information on the use of autoscale, see [autoscale best practices](#).

Next steps

In this tutorial, you created a virtual machine scale set. You learned how to:

- Use the Custom Script Extension to define an IIS site to scale
- Create a load balancer for your scale set
- Create a virtual machine scale set
- Increase or decrease the number of instances in a scale set
- Create autoscale rules

Advance to the next tutorial to learn more about load balancing concepts for virtual machines.

[Load balance virtual machines](#)

Tutorial: Load balance Windows virtual machines in Azure to create a highly available application with Azure PowerShell

11/13/2019 • 8 minutes to read • [Edit Online](#)

Load balancing provides a higher level of availability by spreading incoming requests across multiple virtual machines. In this tutorial, you learn about the different components of the Azure load balancer that distribute traffic and provide high availability. You learn how to:

- Create an Azure load balancer
- Create a load balancer health probe
- Create load balancer traffic rules
- Use the Custom Script Extension to create a basic IIS site
- Create virtual machines and attach to a load balancer
- View a load balancer in action
- Add and remove VMs from a load balancer

Azure load balancer overview

An Azure load balancer is a Layer-4 (TCP, UDP) load balancer that provides high availability by distributing incoming traffic among healthy VMs. A load balancer health probe monitors a given port on each VM and only distributes traffic to an operational VM.

You define a front-end IP configuration that contains one or more public IP addresses. This front-end IP configuration allows your load balancer and applications to be accessible over the Internet.

Virtual machines connect to a load balancer using their virtual network interface card (NIC). To distribute traffic to the VMs, a back-end address pool contains the IP addresses of the virtual (NICs) connected to the load balancer.

To control the flow of traffic, you define load balancer rules for specific ports and protocols that map to your VMs.

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.

Create Azure load balancer

This section details how you can create and configure each component of the load balancer. Before you can create your load balancer, create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroupLoadBalancer* in the *EastUS* location:

```
New-AzResourceGroup ` 
-ResourceGroupName "myResourceGroupLoadBalancer" ` 
-Location "EastUS"
```

Create a public IP address

To access your app on the Internet, you need a public IP address for the load balancer. Create a public IP address with [New-AzPublicIpAddress](#). The following example creates a public IP address named *myPublicIP* in the *myResourceGroupLoadBalancer* resource group:

```
$publicIP = New-AzPublicIpAddress ` 
-ResourceGroupName "myResourceGroupLoadBalancer" ` 
-Location "EastUS" ` 
-AllocationMethod "Static" ` 
-Name "myPublicIP"
```

Create a load balancer

Create a frontend IP pool with [New-AzLoadBalancerFrontendIpConfig](#). The following example creates a frontend IP pool named *myFrontEndPool* and attaches the *myPublicIP* address:

```
$frontendIP = New-AzLoadBalancerFrontendIpConfig ` 
-Name "myFrontEndPool" ` 
-PublicIpAddress $publicIP
```

Create a backend address pool with [New-AzLoadBalancerBackendAddressPoolConfig](#). The VMs attach to this backend pool in the remaining steps. The following example creates a backend address pool named *myBackEndPool*:

```
$backendPool = New-AzLoadBalancerBackendAddressPoolConfig ` 
-Name "myBackEndPool"
```

Now, create the load balancer with [New-AzLoadBalancer](#). The following example creates a load balancer named *myLoadBalancer* using the frontend and backend IP pools created in the preceding steps:

```
$lb = New-AzLoadBalancer ` 
-ResourceGroupName "myResourceGroupLoadBalancer" ` 
-Name "myLoadBalancer" ` 
-Location "EastUS" ` 
-FrontendIpConfiguration $frontendIP ` 
-BackendAddressPool $backendPool
```

Create a health probe

To allow the load balancer to monitor the status of your app, you use a health probe. The health probe dynamically adds or removes VMs from the load balancer rotation based on their response to health checks. By default, a VM is removed from the load balancer distribution after two consecutive failures at 15-second intervals. You create a health probe based on a protocol or a specific health check page for your app.

The following example creates a TCP probe. You can also create custom HTTP probes for more fine grained health checks. When using a custom HTTP probe, you must create the health check page, such as *healthcheck.aspx*. The probe must return an **HTTP 200 OK** response for the load balancer to keep the host in rotation.

To create a TCP health probe, you use [Add-AzLoadBalancerProbeConfig](#). The following example creates a health probe named *myHealthProbe* that monitors each VM on TCP port 80:

```
Add-AzLoadBalancerProbeConfig ` 
-Name "myHealthProbe" ` 
-LoadBalancer $lb ` 
-Protocol tcp ` 
-Port 80 ` 
-IntervalInSeconds 15 ` 
-ProbeCount 2
```

To apply the health probe, update the load balancer with [Set-AzLoadBalancer](#):

```
Set-AzLoadBalancer -LoadBalancer $lb
```

Create a load balancer rule

A load balancer rule is used to define how traffic is distributed to the VMs. You define the front-end IP configuration for the incoming traffic and the back-end IP pool to receive the traffic, along with the required source and destination port. To make sure only healthy VMs receive traffic, you also define the health probe to use.

Create a load balancer rule with [Add-AzLoadBalancerRuleConfig](#). The following example creates a load balancer rule named *myLoadBalancerRule* and balances traffic on TCP port 80:

```
$probe = Get-AzLoadBalancerProbeConfig -LoadBalancer $lb -Name "myHealthProbe"

Add-AzLoadBalancerRuleConfig ` 
-Name "myLoadBalancerRule" ` 
-LoadBalancer $lb ` 
-FrontendIpConfiguration $lb.FrontendIpConfigurations[0] ` 
-BackendAddressPool $lb.BackendAddressPools[0] ` 
-Protocol Tcp ` 
-FrontendPort 80 ` 
-BackendPort 80 ` 
-Probe $probe
```

Update the load balancer with [Set-AzLoadBalancer](#):

```
Set-AzLoadBalancer -LoadBalancer $lb
```

Configure virtual network

Before you deploy some VMs and can test your balancer, create the supporting virtual network resources. For more information about virtual networks, see the [Manage Azure Virtual Networks](#) tutorial.

Create network resources

Create a virtual network with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVnet* with *mySubnet*:

```

# Create subnet config
$subnetConfig = New-AzVirtualNetworkSubnetConfig ` 
    -Name "mySubnet" ` 
    -AddressPrefix 192.168.1.0/24

# Create the virtual network
$vnet = New-AzVirtualNetwork ` 
    -ResourceGroupName "myResourceGroupLoadBalancer" ` 
    -Location "EastUS" ` 
    -Name "myVnet" ` 
    -AddressPrefix 192.168.0.0/16 ` 
    -Subnet $subnetConfig

```

Virtual NICs are created with [New-AzNetworkInterface](#). The following example creates three virtual NICs. (One virtual NIC for each VM you create for your app in the following steps). You can create additional virtual NICs and VMs at any time and add them to the load balancer:

```

for ($i=1; $i -le 3; $i++)
{
    New-AzNetworkInterface ` 
        -ResourceGroupName "myResourceGroupLoadBalancer" ` 
        -Name myVM$i ` 
        -Location "EastUS" ` 
        -Subnet $vnet.Subnets[0] ` 
        -LoadBalancerBackendAddressPool $lb.BackendAddressPools[0]
}

```

Create virtual machines

To improve the high availability of your app, place your VMs in an availability set.

Create an availability set with [New-AzAvailabilitySet](#). The following example creates an availability set named *myAvailabilitySet*:

```

$availabilitySet = New-AzAvailabilitySet ` 
    -ResourceGroupName "myResourceGroupLoadBalancer" ` 
    -Name "myAvailabilitySet" ` 
    -Location "EastUS" ` 
    -Sku aligned ` 
    -PlatformFaultDomainCount 2 ` 
    -PlatformUpdateDomainCount 2

```

Set an administrator username and password for the VMs with [Get-Credential](#):

```
$cred = Get-Credential
```

Now you can create the VMs with [New-AzVM](#). The following example creates three VMs and the required virtual network components if they do not already exist:

```

for ($i=1; $i -le 3; $i++)
{
    New-AzVm ` 
        -ResourceGroupName "myResourceGroupLoadBalancer" ` 
        -Name "myVM$i" ` 
        -Location "East US" ` 
        -VirtualNetworkName "myVnet" ` 
        -SubnetName "mySubnet" ` 
        -SecurityGroupName "myNetworkSecurityGroup" ` 
        -OpenPorts 80 ` 
        -AvailabilitySetName "myAvailabilitySet" ` 
        -Credential $cred ` 
        -AsJob
}

```

The `-AsJob` parameter creates the VM as a background task, so the PowerShell prompts return to you. You can view details of background jobs with the `Job` cmdlet. It takes a few minutes to create and configure all three VMs.

Install IIS with Custom Script Extension

In a previous tutorial on [How to customize a Windows virtual machine](#), you learned how to automate VM customization with the Custom Script Extension for Windows. You can use the same approach to install and configure IIS on your VMs.

Use [Set-AzVMExtension](#) to install the Custom Script Extension. The extension runs

`powershell Add-WindowsFeature Web-Server` to install the IIS webserver and then updates the *Default.htm* page to show the hostname of the VM:

```

for ($i=1; $i -le 3; $i++)
{
    Set-AzVMExtension ` 
        -ResourceGroupName "myResourceGroupLoadBalancer" ` 
        -ExtensionName "IIS" ` 
        -VMName myVM$i ` 
        -Publisher Microsoft.Compute ` 
        -ExtensionType CustomScriptExtension ` 
        -TypeHandlerVersion 1.8 ` 
        -SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -Value $($env:computername)"}' ` 
        -Location EastUS
}

```

Test load balancer

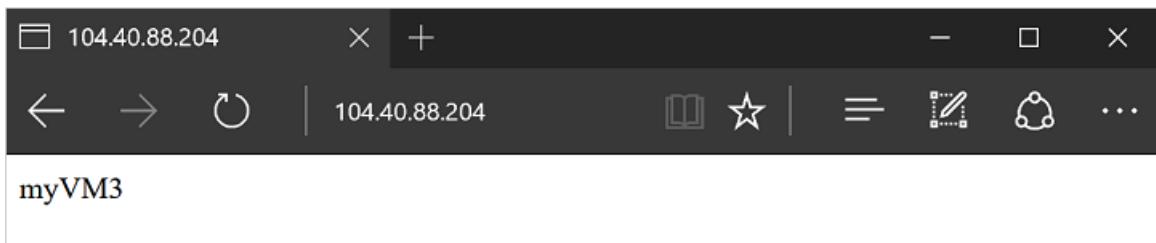
Obtain the public IP address of your load balancer with [Get-AzPublicIPAddress](#). The following example obtains the IP address for *myPublicIP* created earlier:

```

Get-AzPublicIPAddress ` 
    -ResourceGroupName "myResourceGroupLoadBalancer" ` 
    -Name "myPublicIP" | selectIpAddress

```

You can then enter the public IP address in to a web browser. The website is displayed, including the hostname of the VM that the load balancer distributed traffic to as in the following example:



To see the load balancer distribute traffic across all three VMs running your app, you can force-refresh your web browser.

Add and remove VMs

You may need to perform maintenance on the VMs running your app, such as installing OS updates. To deal with increased traffic to your app, you may need to add additional VMs. This section shows you how to remove or add a VM from the load balancer.

Remove a VM from the load balancer

Get the network interface card with [Get-AzNetworkInterface](#), then set the *LoadBalancerBackendAddressPools* property of the virtual NIC to `$null`. Finally, update the virtual NIC.:

```
$nic = Get-AzNetworkInterface `  
    -ResourceGroupName "myResourceGroupLoadBalancer" `  
    -Name "myVM2"  
$nic.IpConfigurations[0].LoadBalancerBackendAddressPools=$null  
Set-AzNetworkInterface -NetworkInterface $nic
```

To see the load balancer distribute traffic across the remaining two VMs running your app you can force-refresh your web browser. You can now perform maintenance on the VM, such as installing OS updates or performing a VM reboot.

Add a VM to the load balancer

After performing VM maintenance, or if you need to expand capacity, set the *LoadBalancerBackendAddressPools* property of the virtual NIC to the *BackendAddressPool* from [Get-AzLoadBalancer](#):

Get the load balancer:

```
$lb = Get-AzLoadBalancer `  
    -ResourceGroupName myResourceGroupLoadBalancer `  
    -Name myLoadBalancer  
$nic.IpConfigurations[0].LoadBalancerBackendAddressPools=$lb.BackendAddressPools[0]  
Set-AzNetworkInterface -NetworkInterface $nic
```

Next steps

In this tutorial, you created a load balancer and attached VMs to it. You learned how to:

- Create an Azure load balancer
- Create a load balancer health probe
- Create load balancer traffic rules
- Use the Custom Script Extension to create a basic IIS site
- Create virtual machines and attach to a load balancer
- View a load balancer in action
- Add and remove VMs from a load balancer

Advance to the next tutorial to learn how to manage VM networking.

[Manage VMs and virtual networks](#)

Tutorial: Create and manage Azure virtual networks for Windows virtual machines with Azure PowerShell

11/13/2019 • 7 minutes to read • [Edit Online](#)

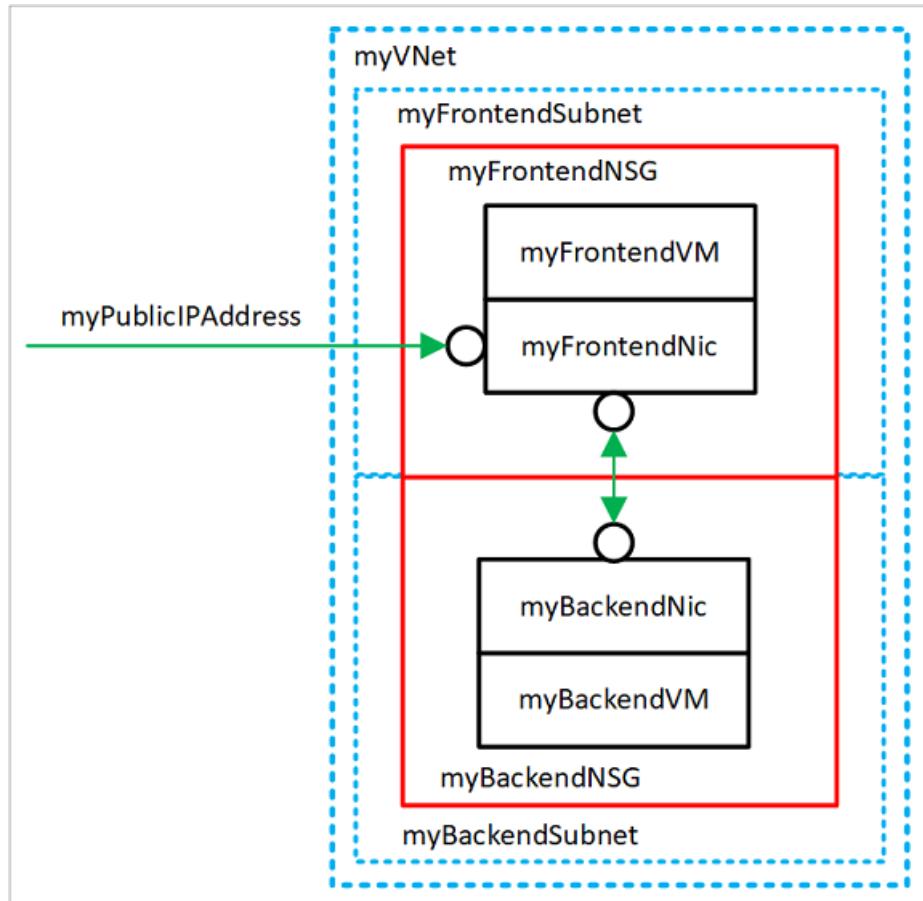
Azure virtual machines use Azure networking for internal and external network communication. This tutorial walks through deploying two virtual machines and configuring Azure networking for these VMs. The examples in this tutorial assume that the VMs are hosting a web application with a database back-end, however an application isn't deployed in the tutorial. In this tutorial, you learn how to:

- Create a virtual network and subnet
- Create a public IP address
- Create a front-end VM
- Secure network traffic
- Create back-end VM

VM networking overview

Azure virtual networks enable secure network connections between virtual machines, the internet, and other Azure services such as Azure SQL database. Virtual networks are broken down into logical segments called subnets. Subnets are used to control network flow, and as a security boundary. When deploying a VM, it generally includes a virtual network interface, which is attached to a subnet.

While completing this tutorial, you can see these resources created:



- *myVNet* - The virtual network that the VMs use to communicate with each other and the internet.

- *myFrontendSubnet* - The subnet in *myVNet* used by the front-end resources.
- *myPublicIPAddress* - The public IP address used to access *myFrontendVM* from the internet.
- *myFrontendNic* - The network interface used by *myFrontendVM* to communicate with *myBackendVM*.
- *myFrontendVM* - The VM used to communicate between the internet and *myBackendVM*.
- *myBackendNSG* - The network security group that controls communication between the *myFrontendVM* and *myBackendVM*.
- *myBackendSubnet* - The subnet associated with *myBackendNSG* and used by the back-end resources.
- *myBackendNic* - The network interface used by *myBackendVM* to communicate with *myFrontendVM*.
- *myBackendVM* - The VM that uses port 1433 to communicate with *myFrontendVM*.

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.

Create subnet

For this tutorial, a single virtual network is created with two subnets. A front-end subnet for hosting a web application, and a back-end subnet for hosting a database server.

Before you can create a virtual network, create a resource group using [New-AzResourceGroup](#). The following example creates a resource group named *myRGNetwork* in the *EastUS* location:

```
New-AzResourceGroup -ResourceGroupName myRGNetwork -Location EastUS
```

Create a subnet configuration named *myFrontendSubnet* using [New-AzVirtualNetworkSubnetConfig](#):

```
$frontendSubnet = New-AzVirtualNetworkSubnetConfig ` 
    -Name myFrontendSubnet ` 
    -AddressPrefix 10.0.0.0/24
```

And, create a subnet configuration named *myBackendSubnet*:

```
$backendSubnet = New-AzVirtualNetworkSubnetConfig ` 
    -Name myBackendSubnet ` 
    -AddressPrefix 10.0.1.0/24
```

Create virtual network

Create a VNET named *myVNet* using *myFrontendSubnet* and *myBackendSubnet* using [New-AzVirtualNetwork](#):

```
$vnet = New-AzVirtualNetwork ` 
    -ResourceGroupName myRGNetwork ` 
    -Location EastUS ` 
    -Name myVNet ` 
    -AddressPrefix 10.0.0.0/16 ` 
    -Subnet $frontendSubnet, $backendSubnet
```

At this point, a network has been created and segmented into two subnets, one for front-end services, and another

for back-end services. In the next section, virtual machines are created and connected to these subnets.

Create a public IP address

A public IP address allows Azure resources to be accessible on the internet. The allocation method of the public IP address can be configured as dynamic or static. By default, a public IP address is dynamically allocated. Dynamic IP addresses are released when a VM is deallocated. This behavior causes the IP address to change during any operation that includes a VM deallocation.

The allocation method can be set to static, which makes sure that the IP address stays assigned to a VM, even during a deallocated state. If you are using a static IP address, the IP address itself can't be specified. Instead, it's allocated from a pool of available addresses.

Create a public IP address named *myPublicIPAddress* using [New-AzPublicIpAddress](#):

```
$pip = New-AzPublicIpAddress `  
    -ResourceGroupName myRGNetwork `  
    -Location EastUS `  
    -AllocationMethod Dynamic `  
    -Name myPublicIPAddress
```

You could change the `-AllocationMethod` parameter to `Static` to assign a static public IP address.

Create a front-end VM

For a VM to communicate in a virtual network, it needs a virtual network interface (NIC). Create a NIC using [New-AzNetworkInterface](#):

```
$frontendNic = New-AzNetworkInterface `  
    -ResourceGroupName myRGNetwork `  
    -Location EastUS `  
    -Name myFrontend `  
    -SubnetId $vnet.Subnets[0].Id `  
    -PublicIpAddressId $pip.Id
```

Set the username and password needed for the administrator account on the VM using [Get-Credential](#). You use these credentials to connect to the VM in additional steps:

```
$cred = Get-Credential
```

Create the VMs using [New-AzVM](#).

```
New-AzVM `  
    -Credential $cred `  
    -Name myFrontend `  
    -PublicIpAddressName myPublicIPAddress `  
    -ResourceGroupName myRGNetwork `  
    -Location "EastUS" `  
    -Size Standard_D1 `  
    -SubnetName myFrontendSubnet `  
    -VirtualNetworkName myVNet
```

Secure network traffic

A network security group (NSG) contains a list of security rules that allow or deny network traffic to resources

connected to Azure Virtual Networks (VNet). NSGs can be associated to subnets or individual network interfaces. An NSG is associated with a network interface only applies to the associated VM. When an NSG is associated to a subnet, the rules apply to all resources connected to the subnet.

Network security group rules

NSG rules define networking ports over which traffic is allowed or denied. The rules can include source and destination IP address ranges so that traffic is controlled between specific systems or subnets. NSG rules also include a priority (between 1—and 4096). Rules are evaluated in the order of priority. A rule with a priority of 100 is evaluated before a rule with priority 200.

All NSGs contain a set of default rules. The default rules can't be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

- **Virtual network** - Traffic originating and ending in a virtual network is allowed both in inbound and outbound directions.
- **Internet** - Outbound traffic is allowed, but inbound traffic is blocked.
- **Load balancer** - Allow Azure's load balancer to probe the health of your VMs and role instances. If you are not using a load balanced set, you can override this rule.

Create network security groups

Create an inbound rule named *myFrontendNSGRule* to allow incoming web traffic on *myFrontendVM* using [New-AzNetworkSecurityRuleConfig](#):

```
$nsgFrontendRule = New-AzNetworkSecurityRuleConfig `  
    -Name myFrontendNSGRule `  
    -Protocol Tcp `  
    -Direction Inbound `  
    -Priority 200 `  
    -SourceAddressPrefix * `  
    -SourcePortRange * `  
    -DestinationAddressPrefix * `  
    -DestinationPortRange 80 `  
    -Access Allow
```

You can limit internal traffic to *myBackendVM* from only *myFrontendVM* by creating an NSG for the back-end subnet. The following example creates an NSG rule named *myBackendNSGRule*:

```
$nsgBackendRule = New-AzNetworkSecurityRuleConfig `  
    -Name myBackendNSGRule `  
    -Protocol Tcp `  
    -Direction Inbound `  
    -Priority 100 `  
    -SourceAddressPrefix 10.0.0.0/24 `  
    -SourcePortRange * `  
    -DestinationAddressPrefix * `  
    -DestinationPortRange 1433 `  
    -Access Allow
```

Add a network security group named *myFrontendNSG* using [New-AzNetworkSecurityGroup](#):

```
$nsgFrontend = New-AzNetworkSecurityGroup `  
    -ResourceGroupName myRGNetwork `  
    -Location EastUS `  
    -Name myFrontendNSG `  
    -SecurityRules $nsgFrontendRule
```

Now, add a network security group named *myBackendNSG* using [New-AzNetworkSecurityGroup](#):

```
$nsgBackend = New-AzNetworkSecurityGroup ` 
-ResourceGroupName myRGNetwork ` 
-Location EastUS ` 
-Name myBackendNSG ` 
-SecurityRules $nsgBackendRule
```

Add the network security groups to the subnets:

```
$vnet = Get-AzVirtualNetwork ` 
-ResourceGroupName myRGNetwork ` 
-Name myVNet
$frontendSubnet = $vnet.Subnets[0]
$backendSubnet = $vnet.Subnets[1]
$frontendSubnetConfig = Set-AzVirtualNetworkSubnetConfig ` 
-VirtualNetwork $vnet ` 
-Name myFrontendSubnet ` 
-AddressPrefix $frontendSubnet.AddressPrefix ` 
-NetworkSecurityGroup $nsgFrontend
$backendSubnetConfig = Set-AzVirtualNetworkSubnetConfig ` 
-VirtualNetwork $vnet ` 
-Name myBackendSubnet ` 
-AddressPrefix $backendSubnet.AddressPrefix ` 
-NetworkSecurityGroup $nsgBackend
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

Create a back-end VM

The easiest way to create the back-end VM for this tutorial is by using a SQL Server image. This tutorial only creates the VM with the database server, but doesn't provide information about accessing the database.

Create *myBackendNic*:

```
$backendNic = New-AzNetworkInterface ` 
-ResourceGroupName myRGNetwork ` 
-Location EastUS ` 
-Name myBackend ` 
-SubnetId $vnet.Subnets[1].Id
```

Set the username and password needed for the administrator account on the VM with *Get-Credential*:

```
$cred = Get-Credential
```

Create *myBackendVM*.

```
New-AzVM ` 
-Credential $cred ` 
-Name myBackend ` 
-ImageName "MicrosoftSQLServer:SQL2016SP1-WS2016:Enterprise:latest" ` 
-ResourceGroupName myRGNetwork ` 
-Location "EastUS" ` 
-SubnetName MyBackendSubnet ` 
-VirtualNetworkName myVNet
```

The image in this example has SQL Server installed, but it isn't used in this tutorial. It's included to show you how you can configure a VM to handle web traffic and a VM to handle database management.

Next steps

In this tutorial, you created and secured Azure networks as related to virtual machines.

- Create a virtual network and subnet
- Create a public IP address
- Create a front-end VM
- Secure network traffic
- Create a back-end VM

Advance to the next tutorial to learn about monitoring securing data on virtual machines using Azure backup.

[Back up Windows virtual machines in Azure](#)

Tutorial: Back up and restore files for Windows virtual machines in Azure

11/13/2019 • 4 minutes to read • [Edit Online](#)

You can protect your data by taking backups at regular intervals. Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore the whole VM or specific files. This article explains how to restore a single file to a VM running Windows Server and IIS. If you don't already have a VM to use, you can create one using the [Windows quickstart](#). In this tutorial you learn how to:

- Create a backup of a VM
- Schedule a daily backup
- Restore a file from a backup

Backup overview

When the Azure Backup service initiates a backup job, it triggers the backup extension to take a point-in-time snapshot. The Azure Backup service uses the *VMSnapshot* extension. The extension is installed during the first VM backup if the VM is running. If the VM is not running, the Backup service takes a snapshot of the underlying storage (since no application writes occur while the VM is stopped).

When taking a snapshot of Windows VMs, the Backup service coordinates with the Volume Shadow Copy Service (VSS) to get a consistent snapshot of the virtual machine's disks. Once the Azure Backup service takes the snapshot, the data is transferred to the vault. To maximize efficiency, the service identifies and transfers only the blocks of data that have changed since the previous backup.

When the data transfer is complete, the snapshot is removed and a recovery point is created.

Create a backup

Create a simple scheduled daily backup to a Recovery Services Vault.

1. Sign in to the [Azure portal](#).
2. In the menu on the left, select **Virtual machines**.
3. From the list, select a VM to back up.
4. On the VM blade, in the **Operations** section, click **Backup**. The **Enable backup** blade opens.
5. In **Recovery Services vault**, click **Create new** and provide the name for the new vault. A new vault is created in the same resource group and location as the virtual machine.
6. Under **Choose backup policy**, keep the default **(New) DailyPolicy**, and then click **Enable Backup**.
7. To create an initial recovery point, on the **Backup** blade click **Backup now**.
8. On the **Backup Now** blade, click the calendar icon, use the calendar control to choose how long the restore point is retained, and click **OK**.
9. In the **Backup** blade for your VM, you'll see the number of restore points that are complete.

Restore points (1)

This list is filtered for last 30 days of restore points. To recover from restore point older than 30 days, [click here](#).

CRASH CONSISTENT	APPLICATION CONSISTENT	FILE-SYSTEM CONSISTENT
0	1	0
TIME	CONSISTENCY	
	10/8/2018 5:41:32 PM	Application Consistent
		...

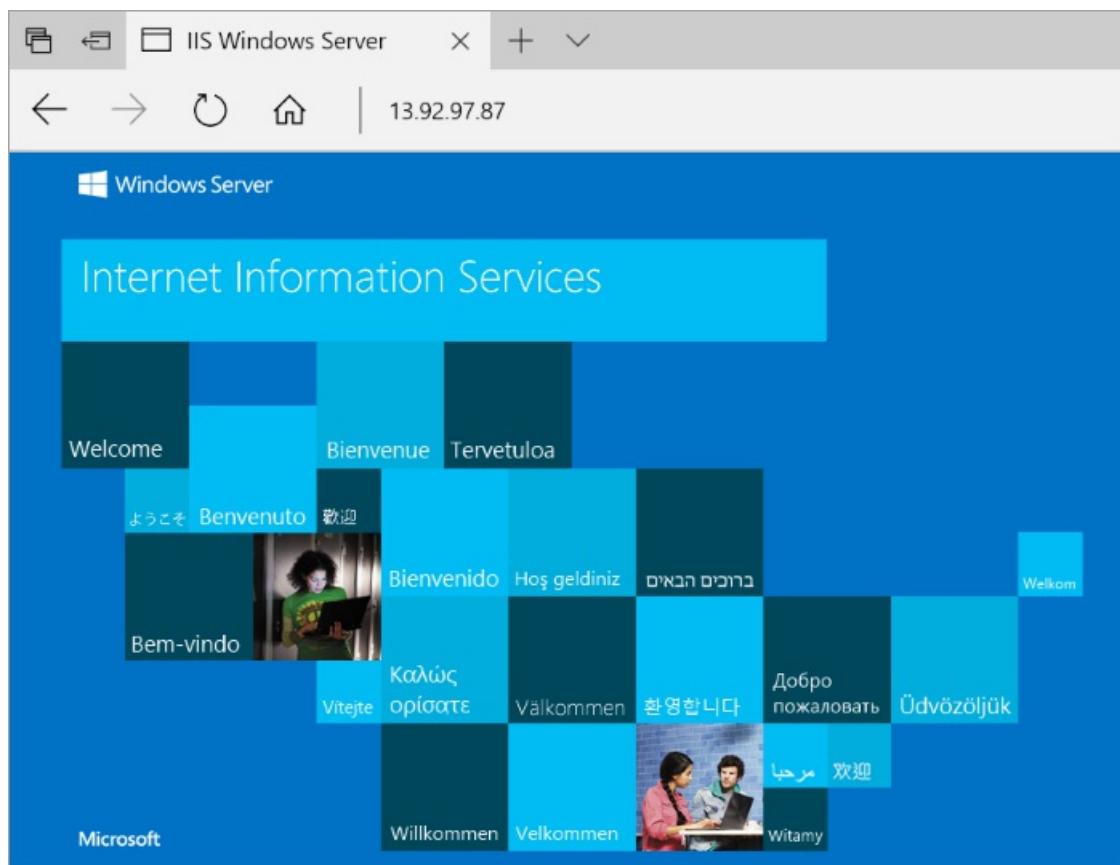
The first backup takes about 20 minutes. Proceed to the next part of this tutorial after your backup is finished.

Recover a file

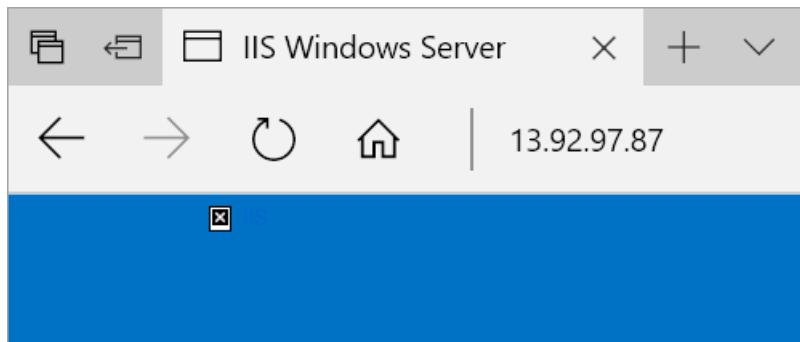
If you accidentally delete or make changes to a file, you can use File Recovery to recover the file from your backup vault. File Recovery uses a script that runs on the VM, to mount the recovery point as local drive. These drives remain mounted for 12 hours so that you can copy files from the recovery point and restore them to the VM.

In this example, we show how to recover the image file that is used in the default web page for IIS.

1. Open a browser and connect to the IP address of the VM to show the default IIS page.



2. Connect to the VM.
3. On the VM, open **File Explorer** and navigate to **\inetpub\wwwroot** and delete the file **iisstart.png**.
4. On your local computer, refresh the browser to see that the image on the default IIS page is gone.



5. On your local computer, open a new tab and go the [Azure portal](#).
6. In the menu on the left, select **Virtual machines** and select the VM from the list.
7. On the VM blade, in the **Operations** section, click **Backup**. The **Backup** blade opens.
8. In the menu at the top of the blade, select **File Recovery**. The **File Recovery** blade opens.
9. In **Step 1: Select recovery point**, select a recovery point from the drop-down.
10. In **Step 2: Download script to browse and recover files**, click the **Download Executable** button. Copy the password for the file and save it somewhere safe.
11. On your local computer, open **File Explorer** and navigate to your **Downloads** folder and copy the downloaded .exe file. The filename is prefixed by your VM name.
12. On your VM (using the RDP connection), paste the .exe file to the Desktop of your VM.
13. Navigate to the desktop of your VM and double-click on the .exe. A command prompt will start. The program mounts the recovery point as a file share that you can access. When it is finished creating the share, type **q** to close the command prompt.
14. On your VM, open **File Explorer** and navigate to the drive letter that was used for the file share.
15. Navigate to **\inetpub\wwwroot** and copy **iisstart.png** from the file share and paste it into **\inetpub\wwwroot**. For example, copy **F:\inetpub\wwwroot\iisstart.png** and paste it into **c:\inetpub\wwwroot** to recover the file.
16. On your local computer, open the browser tab where you are connected to the IP address of the VM showing the IIS default page. Press **CTRL + F5** to refresh the browser page. You should now see that the image has been restored.
17. On your local computer, go back to the browser tab for the Azure portal and in **Step 3: Unmount the disks after recovery** click the **Unmount Disks** button. If you forget to do this step, the connection to the mountpoint is automatically closed after 12 hours. After those 12 hours, you need to download a new script to create a new mount point.

Next steps

In this tutorial, you learned how to:

- Create a backup of a VM
- Schedule a daily backup
- Restore a file from a backup

Advance to the next tutorial to learn about monitoring virtual machines.

[Govern virtual machines](#)

Tutorial: Learn about Windows virtual machine management with Azure PowerShell

1/14/2020 • 10 minutes to read • [Edit Online](#)

When deploying resources to Azure, you have tremendous flexibility when deciding what types of resources to deploy, where they are located, and how to set them up. However, that flexibility may open more options than you would like to allow in your organization. As you consider deploying resources to Azure, you might be wondering:

- How do I meet legal requirements for data sovereignty in certain countries/regions?
- How do I control costs?
- How do I ensure that someone does not inadvertently change a critical system?
- How do I track resource costs and bill it accurately?

This article addresses those questions. Specifically, you:

- Assign users to roles and assign the roles to a scope so users have permission to perform expected actions but not more actions.
- Apply policies that prescribe conventions for resources in your subscription.
- Lock resources that are critical to your system.
- Tag resources so you can track them by values that make sense to your organization.

This article focuses on the tasks you take to implement governance. For a broader discussion of the concepts, see [Governance in Azure](#).

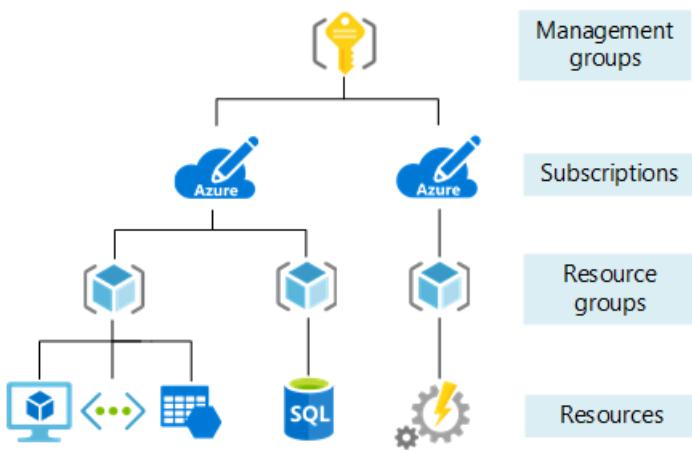
Launch Azure Cloud Shell

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.

Understand scope

Before creating any items, let's review the concept of scope. Azure provides four levels of management: management groups, subscription, resource group, and resource. [Management groups](#) are in a preview release. The following image shows an example of these layers.



You apply management settings at any of these levels of scope. The level you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. When you apply a setting to the subscription, that setting is applied to all resource groups and resources in your subscription. When you apply a setting on the resource group, that setting is applied to the resource group and all its resources. However, another resource group does not have that setting.

Usually, it makes sense to apply critical settings at higher levels and project-specific requirements at lower levels. For example, you might want to make sure all resources for your organization are deployed to certain regions. To accomplish this requirement, apply a policy to the subscription that specifies the allowed locations. As other users in your organization add new resource groups and resources, the allowed locations are automatically enforced.

In this tutorial, you apply all management settings to a resource group so you can easily remove those settings when done.

Let's create that resource group.

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS
```

Currently, the resource group is empty.

Role-based access control

You want to make sure users in your organization have the right level of access to these resources. You don't want to grant unlimited access to users, but you also need to make sure they can do their work. [Role-based access control](#) enables you to manage which users have permission to complete specific actions at a scope.

To create and remove role assignments, users must have `Microsoft.Authorization/roleAssignments/*` access. This access is granted through the Owner or User Access Administrator roles.

For managing virtual machine solutions, there are three resource-specific roles that provide commonly needed access:

- [Virtual Machine Contributor](#)
- [Network Contributor](#)
- [Storage Account Contributor](#)

Instead of assigning roles to individual users, it's often easier to use an Azure Active Directory group that has users who need to take similar actions. Then, assign that group to the appropriate role. For this article, either use an existing group for managing the virtual machine, or use the portal to [create an Azure Active Directory group](#).

After creating a new group or finding an existing one, use the [New-AzRoleAssignment](#) command to assign the Azure Active Directory group to the Virtual Machine Contributor role for the resource group.

```
$adgroup = Get-AzADGroup -DisplayName <your-group-name>

New-AzRoleAssignment -ObjectId $adgroup.id `

    -ResourceGroupName myResourceGroup `

    -RoleDefinitionName "Virtual Machine Contributor"
```

If you receive an error stating **Principal <guid> does not exist in the directory**, the new group hasn't propagated throughout Azure Active Directory. Try running the command again.

Typically, you repeat the process for *Network Contributor* and *Storage Account Contributor* to make sure users are assigned to manage the deployed resources. In this article, you can skip those steps.

Azure Policy

[Azure Policy](#) helps you make sure all resources in subscription meet corporate standards. Your subscription already has several policy definitions. To see the available policy definitions, use the [Get-AzPolicyDefinition](#) command:

```
(Get-AzPolicyDefinition).Properties | Format-Table displayName, policyType
```

You see the existing policy definitions. The policy type is either **BuiltIn** or **Custom**. Look through the definitions for ones that describe a condition you want assign. In this article, you assign policies that:

- Limit the locations for all resources.
- Limit the SKUs for virtual machines.
- Audit virtual machines that don't use managed disks.

In the following example, you retrieve three policy definitions based on the display name. You use the [New-AzPolicyAssignment](#) command to assign those definitions to the resource group. For some policies, you provide parameter values to specify the allowed values.

```

# Values to use for parameters
$locations = "eastus", "eastus2"
$skus = "Standard_DS1_v2", "Standard_E2s_v2"

# Get the resource group
$rg = Get-AzResourceGroup -Name myResourceGroup

# Get policy definitions for allowed locations, allowed SKUs, and auditing VMs that don't use managed disks
$locationDefinition = Get-AzPolicyDefinition | where-object {$_.properties.displayname -eq "Allowed locations"}
$skuDefinition = Get-AzPolicyDefinition | where-object {$_.properties.displayname -eq "Allowed virtual machine SKUs"}
$auditDefinition = Get-AzPolicyDefinition | where-object {$_.properties.displayname -eq "Audit VMs that do not use managed disks"}

# Assign policy for allowed locations
New-AzPolicyAssignment -Name "Set permitted locations" `

    -Scope $rg.ResourceId `

    -PolicyDefinition $locationDefinition `

    -listOfAllowedLocations $locations

# Assign policy for allowed SKUs
New-AzPolicyAssignment -Name "Set permitted VM SKUs" `

    -Scope $rg.ResourceId `

    -PolicyDefinition $skuDefinition `

    -listOfAllowedSKUs $skus

# Assign policy for auditing unmanaged disks
New-AzPolicyAssignment -Name "Audit unmanaged disks" `

    -Scope $rg.ResourceId `

    -PolicyDefinition $auditDefinition

```

Deploy the virtual machine

You have assigned roles and policies, so you're ready to deploy your solution. The default size is Standard_DS1_v2, which is one of your allowed SKUs. When running this step, you're prompted for credentials. The values that you enter are configured as the user name and password for the virtual machine.

```

New-AzVm -ResourceGroupName "myResourceGroup" `

    -Name "myVM" `

    -Location "East US" `

    -VirtualNetworkName "myVnet" `

    -SubnetName "mySubnet" `

    -SecurityGroupName "myNetworkSecurityGroup" `

    -PublicIpAddressName "myPublicIpAddress" `

    -OpenPorts 80,3389

```

After your deployment finishes, you can apply more management settings to the solution.

Lock resources

[Resource locks](#) prevent users in your organization from accidentally deleting or modifying critical resources. Unlike role-based access control, resource locks apply a restriction across all users and roles. You can set the lock level to *CanNotDelete* or *ReadOnly*.

To lock the virtual machine and network security group, use the [New-AzResourceLock](#) command:

```
# Add CanNotDelete lock to the VM
New-AzResourceLock -LockLevel CanNotDelete ` 
    -LockName LockVM ` 
    -ResourceName myVM ` 
    -ResourceType Microsoft.Compute/virtualMachines ` 
    -ResourceGroupName myResourceGroup

# Add CanNotDelete lock to the network security group
New-AzResourceLock -LockLevel CanNotDelete ` 
    -LockName LockNSG ` 
    -ResourceName myNetworkSecurityGroup ` 
    -ResourceType Microsoft.Network/networkSecurityGroups ` 
    -ResourceGroupName myResourceGroup
```

To test the locks, try running the following command:

```
Remove-AzResourceGroup -Name myResourceGroup
```

You see an error stating that the delete operation can't be completed because of a lock. The resource group can only be deleted if you specifically remove the locks. That step is shown in [Clean up resources](#).

Tag resources

You apply [tags](#) to your Azure resources to logically organize them by categories. Each tag consists of a name and a value. For example, you can apply the name "Environment" and the value "Production" to all the resources in production.

To add two tags to a resource group, use the [Set-AzResourceGroup](#) command:

```
Set-AzResourceGroup -Name myResourceGroup -Tag @{ Dept="IT"; Environment="Test" }
```

Let's suppose you want to add a third tag. Every time you apply tags to a resource or a resource group, you overwrite the existing tags on that resource or resource group. To add a new tag without losing the existing tags, you must retrieve the existing tags, add a new tag, and reapply the collection of tags:

```
# Get existing tags and add a new tag
$tags = (Get-AzResourceGroup -Name myResourceGroup).Tags
$tags.Add("Project", "Documentation")

# Reapply the updated set of tags
Set-AzResourceGroup -Tag $tags -Name myResourceGroup
```

Resources don't inherit tags from the resource group. Currently, your resource group has three tags but the resources do not have any tags. To apply all tags from a resource group to its resources, and retain existing tags on resources that are not duplicates, use the following script:

```

# Get the resource group
$group = Get-AzResourceGroup myResourceGroup

if ($group.Tags -ne $null) {
    # Get the resources in the resource group
    $resources = Get-AzResource -ResourceGroupName $group.ResourceGroupName

    # Loop through each resource
    foreach ($r in $resources)
    {
        # Get the tags for this resource
        $resourcetags = (Get-AzResource -ResourceId $r.ResourceId).Tags

        # If the resource has existing tags, add new ones
        if ($resourcetags)
        {
            foreach ($key in $group.Tags.Keys)
            {
                if (-not($resourcetags.ContainsKey($key)))
                {
                    $resourcetags.Add($key, $group.Tags[$key])
                }
            }

            # Reapply the updated tags to the resource
            Set-AzResource -Tag $resourcetags -ResourceId $r.ResourceId -Force
        }
        else
        {
            Set-AzResource -Tag $group.Tags -ResourceId $r.ResourceId -Force
        }
    }
}

```

Alternatively, you can apply tags from the resource group to the resources without keeping the existing tags:

```

# Get the resource group
$g = Get-AzResourceGroup -Name myResourceGroup

# Find all the resources in the resource group, and for each resource apply the tags from the resource group
Get-AzResource -ResourceGroupName $g.ResourceGroupName | ForEach-Object {Set-AzResource -ResourceId
$_._ResourceId -Tag $g.Tags -Force }

```

To combine several values in a single tag, use a JSON string.

```
Set-AzResourceGroup -Name myResourceGroup -Tag @{ CostCenter="{'Dept`":`"IT`","Environment`":`"Test`"}" }
```

To add a new tag with several values without losing the existing tags, you must retrieve the existing tags, use a JSON string for the new tag, and reapply the collection of tags:

```

# Get existing tags and add a new tag
$ResourceGroup = Get-AzResourceGroup -Name myResourceGroup
$Tags = $ResourceGroup.Tags
$Tags.Add("CostCenter", "{'Dept`":`"IT`","Environment`":`"Test`"}" )

# Reapply the updated set of tags
$ResourceGroup | Set-AzResourceGroup -Tag $Tags

```

To remove all tags, you pass an empty hash table.

```
Set-AzResourceGroup -Name myResourceGroup -Tag @{ }
```

To apply tags to a virtual machine, use the [Set-AzResource](#) command:

```
# Get the virtual machine  
$r = Get-AzResource -ResourceName myVM `  
-ResourceGroupName myResourceGroup `  
-ResourceType Microsoft.Compute/virtualMachines  
  
# Apply tags to the virtual machine  
Set-AzResource -Tag @{ Dept="IT"; Environment="Test"; Project="Documentation" } -ResourceId $r.ResourceId -  
Force
```

Find resources by tag

To find resources with a tag name and value, use the [Get-AzResource](#) command:

```
(Get-AzResource -Tag @{ Environment="Test"}).Name
```

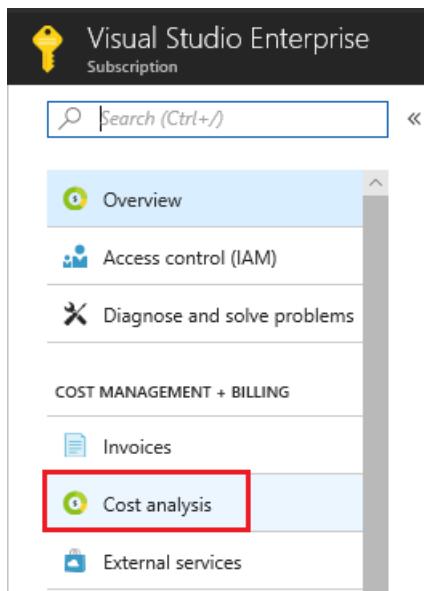
You can use the returned values for management tasks like stopping all virtual machines with a tag value.

```
Get-AzResource -Tag @{ Environment="Test" } | Where-Object {$_._.ResourceType -eq  
"Microsoft.Compute/virtualMachines" } | Stop-AzVM
```

View costs by tag values

After applying tags to resources, you can view costs for resources with those tags. It takes a while for cost analysis to show the latest usage, so you may not see the costs yet. When the costs are available, you can view costs for resources across resource groups in your subscription. Users must have [subscription level access to billing information](#) to see the costs.

To view costs by tag in the portal, select your subscription and select **Cost Analysis**.



Then, filter by the tag value, and select **Apply**.

→ Costs by service

For more cost management and optimization capabilities, try Azure Cost Management →

Subscription	Resource type	Resource group
Visual Studio Enterprise	3 selected	2 selected

Timespan Tag

Current period Environment: Test

Apply Download

 There is a delay between the time when reported here may be delayed. Amount reaches the billing system. Due to this, costs me recent usage. Taxes are not included.

Total cost 0.24 USD

Tag

- Select all
- Project: Documentation
- Dept: IT
- Environment: Test
- No Tags --

You can also use the [Azure Billing APIs](#) to programmatically view costs.

Clean up resources

The locked network security group can't be deleted until the lock is removed. To remove the lock, use the [Remove-AzResourceLock](#) command:

```
Remove-AzResourceLock -LockName LockVM  
-ResourceName myVM  
-ResourceType Microsoft.Compute/virtualMachines  
-ResourceGroupName myResourceGroup  
Remove-AzResourceLock -LockName LockNSG  
-ResourceName myNetworkSecurityGroup  
-ResourceType Microsoft.Network/networkSecurityGroups  
-ResourceGroupName myResourceGroup
```

When no longer needed, you can use the [Remove-AzResourceGroup](#) command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Next steps

In this tutorial, you created a custom VM image. You learned how to:

- Assign users to a role
- Apply policies that enforce standards
- Protect critical resources with locks
- Tag resources for billing and management

Advance to the next tutorial to learn about how to identify changes and manage package updates on a Linux virtual machine.

[Manage virtual machines](#)

Tutorial: Monitor changes and update a Windows virtual machine in Azure

11/13/2019 • 9 minutes to read • [Edit Online](#)

With Azure [Change Tracking](#) and [Update Management](#), you can easily identify changes in your Windows virtual machines in Azure and manage operating system updates for those VMs.

In this tutorial, you learn how to:

- Manage Windows updates.
- Monitor changes and inventory.

Open Azure Cloud Shell

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 Azure account.

To open any code block in Cloud Shell, just select **Try it** from the upper-right corner of that code block.

You can also open Cloud Shell in a separate browser tab by going to <https://shell.azure.com/powershell>. Select **Copy** to copy code blocks, paste them into the Cloud Shell tab, and select the Enter key to run the code.

Create a virtual machine

To configure Azure monitoring and update management in this tutorial, you need a Windows VM in Azure.

First, set an administrator username and password for the VM with [Get-Credential](#):

```
$cred = Get-Credential
```

Next, create the VM with [New-AzVm](#). The following example creates a VM named `myVM` in the `East US` location. If they don't already exist, the resource group `myResourceGroupMonitor` and supporting network resources are created:

```
New-AzVm `‐
    -ResourceGroupName "myResourceGroupMonitor" `‐
    -Name "myVM" `‐
    -Location "East US" `‐
    -Credential $cred
```

It takes a few minutes for the resources and VM to be created.

Manage Windows updates

Update Management helps you manage updates and patches for your Azure Windows VMs. Directly from your VM, you can quickly:

- Assess the status of available updates.
- Schedule installation of required updates.
- Review deployment results to verify updates were successfully applied to the VM.

For pricing information, see [Automation pricing for Update management](#).

Enable Update Management

To enable Update Management for your VM:

1. On the leftmost side of the window, select **Virtual machines**.
2. Choose a VM from the list.
3. In the **Operations** pane of the VM window, select **Update management**.
4. The **Enable Update Management** window opens.

Validation is done to determine if Update Management is enabled for this VM. Validation includes checks for a Log Analytics workspace, for a linked Automation account, and for whether the solution is in the workspace.

You use a [Log Analytics](#) workspace to collect data that is generated by features and services such as Update Management. The workspace provides a single location to review and analyze data from multiple sources.

To perform additional actions on VMs that require updates, you can use Azure Automation to run runbooks against VMs. Such actions include downloading or applying updates.

The validation process also checks to see if the VM is provisioned with the Microsoft Monitoring Agent (MMA) and Automation Hybrid Runbook Worker. You use the agent to communicate with the VM and obtain information about the update status.

In the **Enable Update Management** window, choose the Log Analytics workspace and automation account, and then select **Enable**. The solution takes up to 15 minutes to become enabled.

Any of the following prerequisites that are missing during onboarding are automatically added:

- [Log Analytics](#) workspace
- [Automation](#)
- A [Hybrid runbook worker](#), which is enabled on the VM

After the solution is enabled, the **Update management** window opens. Configure the location, Log Analytics workspace and Automation account to use, and then select **Enable**. If these options appear dimmed, another automation solution is enabled for the VM, and that solution's workspace and Automation account must be used.

The Update Management solution can take up to 15 minutes to become enabled. During this time, don't close the browser window. After the solution is enabled, information about missing updates on the VM flows to Azure Monitor logs. It can take from 30 minutes to 6 hours for the data to become available for analysis.

View an update assessment

After Update Management is enabled, the **Update management** window appears. After the evaluation of updates is finished, you see a list of missing updates on the **Missing updates** tab.

UPDATE NAME	CLASSIFICATION	PUBLISHED DATE	INFORMATION LINK
2019-05 Cumulative Update for Windows Server 2016 for ...	Updates	5/22/2019	KB4499177
Definition Update for Windows Defender Antivirus - KB22...	Definition updates	6/2/2019	KB2267602
Definition Update for Windows Defender Antivirus - KB22...	Definition updates	6/2/2019	KB2267602

Schedule an update deployment

To install updates, schedule a deployment that follows your release schedule and service window. You choose which update types to include in the deployment. For example, you can include critical or security updates and exclude update rollups.

To schedule a new update deployment for the VM, select **Schedule update deployment** at the top of the **Update management** window. In the **New update deployment** window, specify the following information:

OPTION	DESCRIPTION
Name	Enter a unique name to identify the update deployment.

OPTION	DESCRIPTION
Operating system	Select either Linux or Windows .
Groups to update	<p>For VMs hosted on Azure, define a query based on a combination of subscription, resource groups, locations, and tags. This query builds a dynamic group of Azure-hosted VMs to include in your deployment.</p> <p>For VMs not hosted on Azure, select an existing saved search. With this search, you can select a group of these VMs to include in the deployment.</p> <p>To learn more, see Dynamic Groups.</p>
Machines to update	<p>Select Saved search, Imported group, or Machines.</p> <p>If you select Machines, you can choose individual machines from the drop-down list. The readiness of each machine is shown in the UPDATE AGENT READINESS column of the table.</p> <p>To learn about the different methods of creating computer groups in Azure Monitor logs, see Computer groups in Azure Monitor logs</p>
Update classifications	Choose all necessary update classifications.
Include/exclude updates	Select this option to open the Include/Exclude pane. Updates to be included and those to be excluded are on separate tabs. For more information on how inclusion is handled, see Schedule an Update Deployment .
Schedule settings	Choose the time to start, and select either Once or Recurring .
Pre-scripts + Post-scripts	Choose the scripts to run before and after your deployment.
Maintenance window	Enter the number of minutes set for updates. Valid values range from 30 to 360 minutes.
Reboot control	<p>Select how reboots are handled. Available selections are:</p> <ul style="list-style-type: none"> • Reboot if required • Always reboot • Never reboot • Only reboot <p>Reboot if required is the default selection. If you select Only reboot, updates aren't installed.</p>

After you have finished configuring the schedule, click **Create** to return to the status dashboard. The **Scheduled** table shows the deployment schedule you created.

You can also create update deployments programmatically. To learn how to create an update deployment with the REST API, see [Software Update Configurations - Create](#). There's also a sample runbook that you can use to create a weekly update deployment. To learn more about this runbook, see [Create a weekly update deployment for one or more VMs in a resource group](#).

View results of an update deployment

After the scheduled deployment starts, you can see the deployment status in the **Update deployments** tab of the **Update management** window.

If the deployment is currently running, its status shows as "In progress." After successful completion, the status changes to "Succeeded." But if any updates in the deployment fail, the status is "Partially failed."

Select the completed update deployment to see the dashboard for that deployment.

The screenshot shows the 'Update management' window for a VM named 'Marketing'. At the top, it displays the status as 'Succeeded' with a green circle icon. Below this, there's a summary of the deployment results: 5 Updates, 0 Failed, 0 Not attempted, 5 Succeeded, and 0 Not selected. To the right, a detailed table lists individual update names and their statuses:

UPDATE NAME	STATUS
2019-05 Cumulative Update for Windows Server 2016 for x64-based Systems (KB4505052)	Succeeded
Definition Update for Windows Defender Antivirus - KB2267602 (Definition 1.293.1980.0)	Succeeded
2019-05 Servicing Stack Update for Windows Server 2016 for x64-based Systems (KB4498947)	Succeeded
Definition Update for Windows Defender Antivirus - KB2267602 (Definition 1.293.1982.0)	Succeeded
Windows Malicious Software Removal Tool x64 - May 2019 (KB890830)	Succeeded

At the bottom, there are three tabs for 'Diagnostics and Logs': 'All Logs' (selected), 'Output', and 'Errors' (0 errors shown).

The **Update results** tile shows a summary of the total number of updates and deployment results on the VM. The table to the right shows a detailed breakdown of each update and the installation results. Each result has one of the following values:

- **Not attempted:** The update isn't installed. There wasn't enough time available based on the defined maintenance-window duration.
- **Succeeded:** The update succeeded.
- **Failed:** The update failed.

Select **All logs** to see all log entries that the deployment created.

Select the **Output** tile to see the job stream of the runbook responsible for managing the update deployment on the target VM.

Select **Errors** to see detailed information about any deployment errors.

Monitor changes and inventory

You can collect and view an inventory of the software, files, Linux daemons, Windows services, and Windows registry keys on your computers. Tracking the configurations of your machines helps you pinpoint operational issues across your environment and better understand the state of your machines.

Enable change and inventory management

To enable change and inventory management for your VM:

1. On the leftmost side of the window, select **Virtual machines**.
2. Choose a VM from the list.
3. Under **Operations** in the VM window, select either **Inventory** or **Change tracking**.
4. The **Enable Change Tracking and Inventory** pane opens.

Configure the location, Log Analytics workspace, and Automation account to use, and then select **Enable**. If the options appear dimmed, an automation solution is already enabled for the VM. In that case, the already enabled workspace and Automation account must be used.

Even though the solutions appear separately in the menu, they're the same solution. Enabling one enables both for your VM.

myVM - Inventory
Virtual machine

Search (Ctrl+ /)

OPERATIONS

- Auto-shutdown
- Backup
- Disaster recovery (Preview)
- Update management
- Inventory**
- Change tracking

MONITORING

- Metrics
- Alert rules
- Diagnostics settings
- Advisor recommendations

Inventory

Enable consistent control and compliance of this VM with Change Tracking and Inventory.

This service is included with Azure virtual machines. You only pay for logs stored in Log Analytics.

This service requires a Log Analytics workspace and an Automation account. You can use your existing workspace and account or let us configure the nearest workspace and account for use.

Location ⓘ
East US

Log Analytics workspace ⓘ
defaultworkspace

Automation account ⓘ
Automate

Enable

After the solution has been enabled, it might take some time for inventory to be collected on the VM before data appears.

Track changes

On your VM under **OPERATIONS**, select **Change Tracking** and then select **Edit Settings**. The **Change Tracking** pane opens. Select the type of setting you want to track and then select **+ Add** to configure the settings.

The available settings options for Windows are:

- Windows Registry
- Windows Files

For detailed information on Change Tracking, see [Troubleshoot changes on a VM](#).

View inventory

On your VM select **Inventory** under **OPERATIONS**. On the **Software** tab, there's a table that shows the software that had been found. The high-level details for each software record appear in the table. These details include the software name, version, publisher, and last refreshed time.

New software ⓘ
0 files Last 24 hours

Learn more
Inventory
Provide feedback

Software Files Windows Registry Windows Services

Search to filter items...

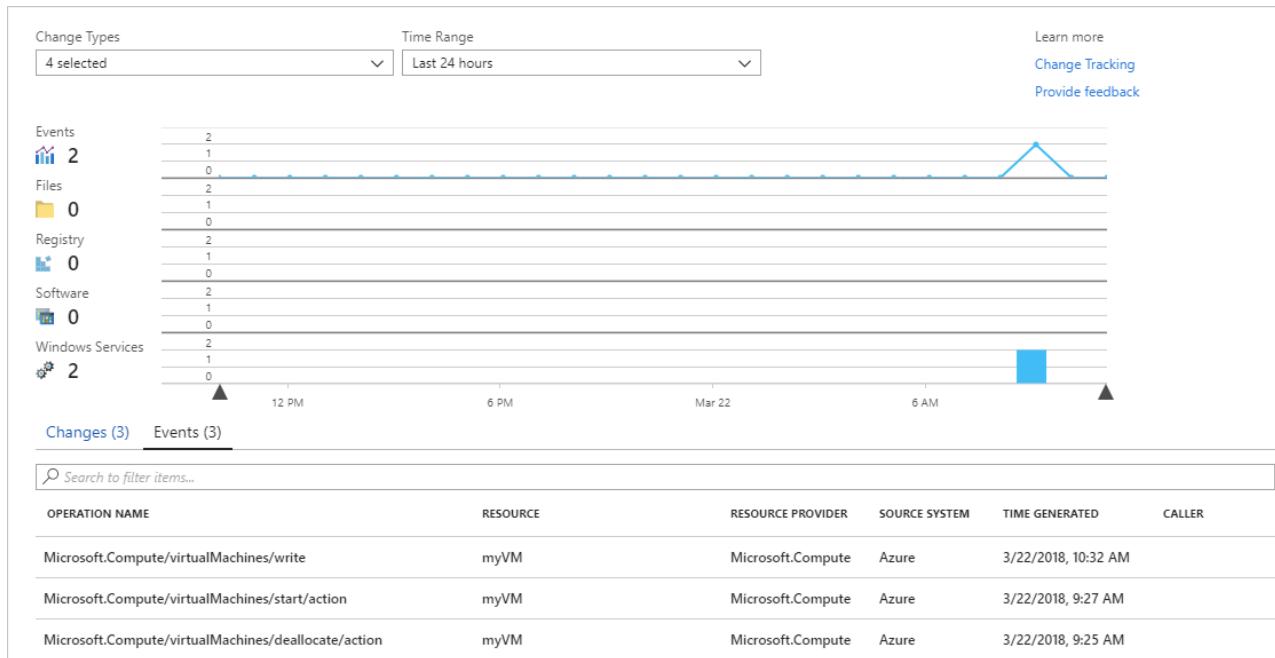
NAME	VERSION	PUBLISHER	LAST REFRESHED TIME
2018-03 Cumulative Update for Windows Server 2016 for x64-based Systems (KB4088787)	Installed	Microsoft Corporation	3/22/2018, 9:08 AM
Definition Update for Windows Defender Antivirus - KB2267602 (Definition 1.263.954.0)	Installed	Microsoft Corporation	3/22/2018, 9:08 AM
Microsoft Monitoring Agent	8.0.11081.0	Microsoft Corporation	3/22/2018, 9:08 AM
Update for Windows Defender antimalware platform - KB4052623 (Version 4.12.17007.18022)	Installed	Microsoft Corporation	3/22/2018, 9:08 AM
Windows Malicious Software Removal Tool x64 - March 2018 (KB890830)	Installed	Microsoft Corporation	3/22/2018, 9:08 AM

Monitor activity logs and changes

From the **Change tracking** window on your VM, select **Manage Activity Log Connection** to open the **Azure Activity log** pane. Select **Connect** to connect Change Tracking to the Azure activity log for your VM.

After Change Tracking is enabled, go to the **Overview** pane for your VM and select **Stop** to stop your VM. When prompted, select **Yes** to stop the VM. After the VM is deallocated, select **Start** to restart your VM.

Stopping and restarting a VM logs an event in its activity log. Go back to the **Change tracking** pane and select the **Events** tab at the bottom of the pane. After a while, the events appear in the chart and the table. You can select each event to view detailed information for that event.



The previous chart shows changes that have occurred over time. After you add an Azure Activity Log connection, the line graph at the top displays Azure Activity Log events.

Each row of bar graphs represents a different trackable change type. These types are Linux daemons, files, Windows registry keys, software, and Windows services. The **Change** tab shows the change details. Changes appear in the order of when each occurred, with the most recent change shown first.

Next steps

In this tutorial, you configured and reviewed Change Tracking and Update Management for your VM. You learned how to:

- Create a resource group and VM.
- Manage Windows updates.
- Monitor changes and inventory.

Go to the next tutorial to learn about monitoring your VM.

[Monitor virtual machines](#)

Tutorial: Monitor a Windows virtual machine in Azure

11/14/2019 • 4 minutes to read • [Edit Online](#)

Azure monitoring uses agents to collect boot and performance data from Azure VMs, store this data in Azure storage, and make it accessible through portal, the Azure PowerShell module, and Azure CLI. Advanced monitoring is delivered with Azure Monitor for VMs by collecting performance metrics, discover application components installed on the VM, and includes performance charts and dependency map.

In this tutorial, you learn how to:

- Enable boot diagnostics on a VM
- View boot diagnostics
- View VM host metrics
- Enable Azure Monitor for VMs
- View VM performance metrics
- Create an alert

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.

Create virtual machine

To configure Azure monitoring and update management in this tutorial, you need a Windows VM in Azure. First, set an administrator username and password for the VM with [Get-Credential](#):

```
$cred = Get-Credential
```

Now create the VM with [New-AzVM](#). The following example creates a VM named *myVM* in the *EastUS* location. If they do not already exist, the resource group *myResourceGroupMonitor* and supporting network resources are created:

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroupMonitor" ` 
    -Name "myVM" ` 
    -Location "East US" ` 
    -Credential $cred
```

It takes a few minutes for the resources and VM to be created.

View boot diagnostics

As Windows virtual machines boot up, the boot diagnostic agent captures screen output that can be used for troubleshooting purpose. This capability is enabled by default. The captured screenshots are stored in an Azure storage account, which is also created by default.

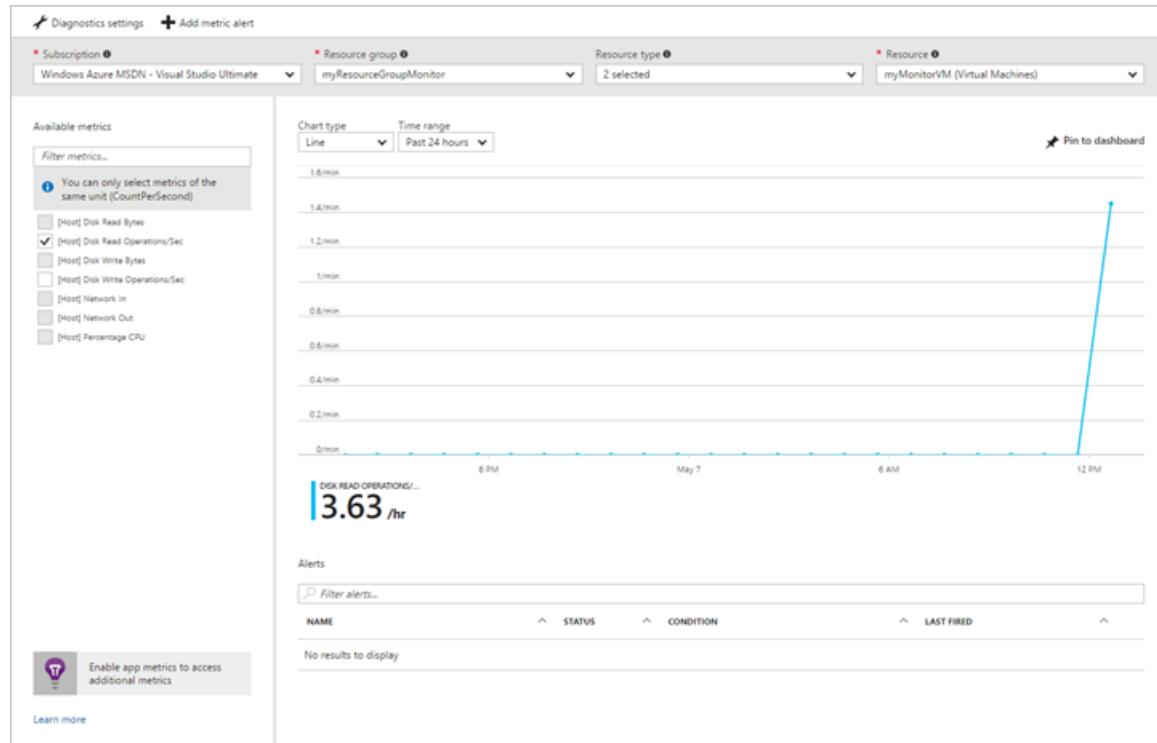
You can get the boot diagnostic data with the [Get-AzVMBootDiagnosticsData](#) command. In the following example, boot diagnostics are downloaded to the root of the *c:* drive.

```
Get-AzVMBootDiagnosticsData -ResourceGroupName "myResourceGroupMonitor" -Name "myVM" -Windows -LocalPath "c:\\"
```

View host metrics

A Windows VM has a dedicated Host VM in Azure that it interacts with. Metrics are automatically collected for the Host and can be viewed in the Azure portal.

1. In the Azure portal, click **Resource Groups**, select **myResourceGroupMonitor**, and then select **myVM** in the resource list.
2. Click **Metrics** on the VM blade, and then select any of the Host metrics under **Available metrics** to see how the Host VM is performing.



Enable advanced monitoring

To enable monitoring of your Azure VM with Azure Monitor for VMs:

1. In the Azure portal, click **Resource Groups**, select **myResourceGroupMonitor**, and then select **myVM** in the resource list.
2. On the VM page, in the **Monitoring** section, select **Insights (preview)**.
3. On the **Insights (preview)** page, select **Try now**.

4. On the **Azure Monitor Insights Onboarding** page, if you have an existing Log Analytics workspace in the same subscription, select it in the drop-down list.

The list preselects the default workspace and location where the VM is deployed in the subscription.

NOTE

To create a new Log Analytics workspace to store the monitoring data from the VM, see [Create a Log Analytics workspace](#). Your Log Analytics workspace must belong to one of the [supported regions](#).

After you've enabled monitoring, you might need to wait several minutes before you can view the performance metrics for the VM.

NOTE

Monitoring data is being collected and routed to Insights. It can take up to 10 minutes to arrive. Please try again in a few minutes.

AZURE MONITOR

Get more visibility into the health and performance of your virtual machines

With an Azure virtual machine you get host CPU, disk and up/down state of your virtual machine out of the box. Enabling additional monitoring capabilities provides insights into the performance, topology, and health for the single VM and across your entire fleet of virtual machines.

You will be billed based on the amount of data ingested and your data retention settings. It can take between 5-10 minutes to get data into virtual machine health and insights.

Have more questions?

[Learn more about virtual machine health and performance monitoring](#)

[Learn more about pricing](#)

View VM performance metrics

Azure Monitor for VMs includes a set of performance charts that target several key performance indicators (KPIs) to help you determine how well a virtual machine is performing. To access from your VM, perform the following steps.

1. In the Azure portal, click **Resource Groups**, select **myResourceGroupMonitor**, and then select **myVM** in the resource list.
2. On the VM page, in the **Monitoring** section, select **Insights (preview)**.
3. Select the **Performance** tab.

This page not only includes performance utilization charts, but also a table showing for each logical disk discovered, its capacity, utilization, and total average by each measure.

Create alerts

You can create alerts based on specific performance metrics. Alerts can be used to notify you when average CPU usage exceeds a certain threshold or available free disk space drops below a certain amount, for example. Alerts are displayed in the Azure portal or can be sent via email. You can also trigger Azure Automation runbooks or Azure Logic Apps in response to alerts being generated.

The following example creates an alert for average CPU usage.

1. In the Azure portal, click **Resource Groups**, select **myResourceGroupMonitor**, and then select **myVM** in the resource list.
2. Click **Alert rules** on the VM blade, then click **Add metric alert** across the top of the alerts blade.
3. Provide a **Name** for your alert, such as *myAlertRule*
4. To trigger an alert when CPU percentage exceeds 1.0 for five minutes, leave all the other defaults selected.
5. Optionally, check the box for *Email owners, contributors, and readers* to send email notification. The default action is to present a notification in the portal.
6. Click the **OK** button.

Next steps

In this tutorial, you configured and viewed performance of your VM. You learned how to:

- Create a resource group and VM
- Enable boot diagnostics on the VM
- View boot diagnostics
- View host metrics
- Enable Azure Monitor for VMs
- View VM metrics
- Create an alert

Advance to the next tutorial to learn about Azure Security Center.

[Manage VM security](#)

Tutorial: Use Azure Security Center to monitor Windows virtual machines

11/13/2019 • 4 minutes to read • [Edit Online](#)

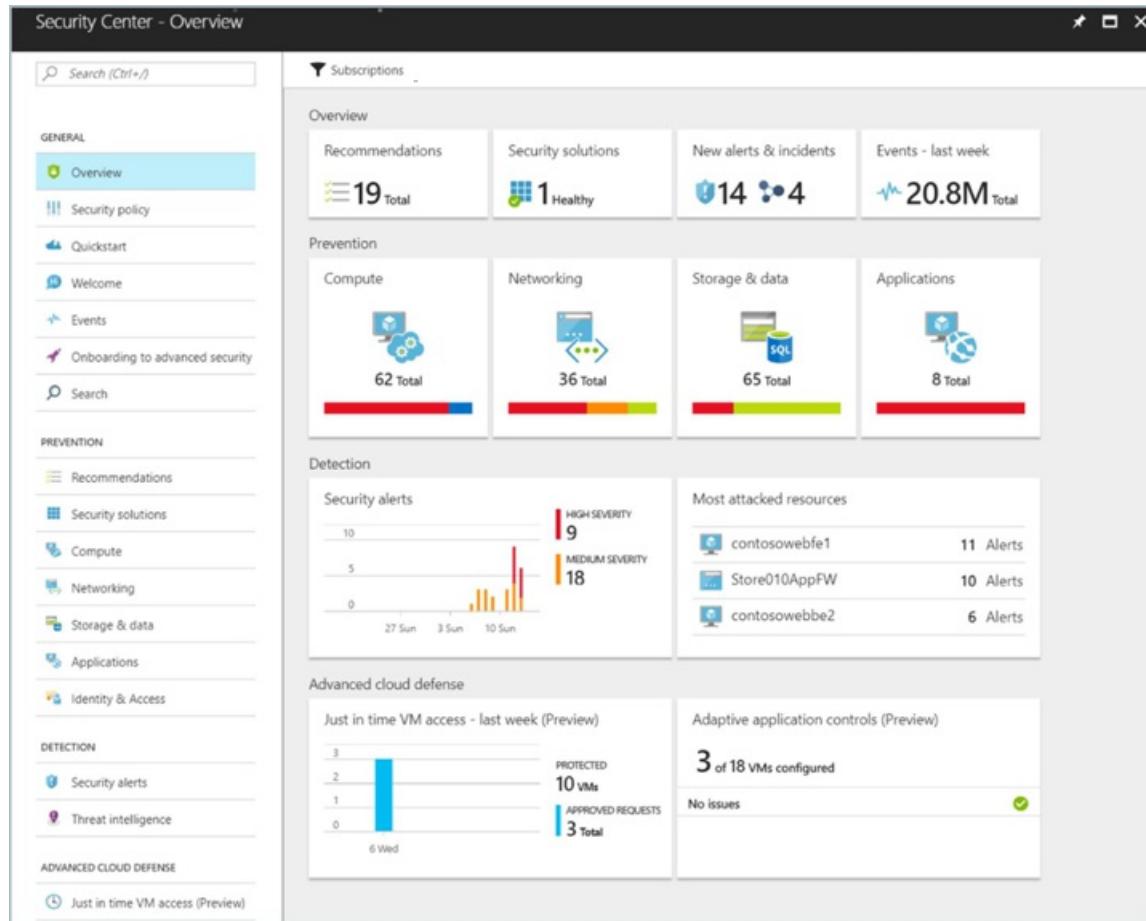
Azure Security Center can help you gain visibility into your Azure resource security practices. Security Center offers integrated security monitoring. It can detect threats that otherwise might go unnoticed. In this tutorial, you learn about Azure Security Center, and how to:

- Set up data collection
- Set up security policies
- View and fix configuration health issues
- Review detected threats

Security Center overview

Security Center identifies potential virtual machine (VM) configuration issues and targeted security threats. These might include VMs that are missing network security groups, unencrypted disks, and brute-force Remote Desktop Protocol (RDP) attacks. The information is shown on the Security Center dashboard in easy-to-read graphs.

To access the Security Center dashboard, in the Azure portal, on the menu, select **Security Center**. On the dashboard, you can see the security health of your Azure environment, find a count of current recommendations, and view the current state of threat alerts. You can expand each high-level chart to see more detail.



Security Center goes beyond data discovery to provide recommendations for issues that it detects. For example, if a VM was deployed without an attached network security group, Security Center displays a recommendation, with

remediation steps you can take. You get automated remediation without leaving the context of Security Center.

Recommendations					X
Filter					
Description	Resource	State	Severity		
Install Endpoint Protection	2016OMSAA	Open	! High	...	
Add a Next Generation Firewall	2 endpoints	Open	! High	...	
Enable Network Security Groups on sub...	2 subnets	Open	! High	...	
Apply disk encryption	3 virtual mac...	Open	! High	...	
Enable encryption for Azure Storage Acc...	9 storage acc...	Open	! High	...	
Restrict access through Internet facing e...	2 virtual mac...	Open	⚠ Medium	...	
Add a vulnerability assessment solution	2016OMSAA	Open	⚠ Medium	...	
Remediate OS vulnerabilities (by Micros...	2016OMSLin...	Open	ℹ Low	...	

Set up data collection

Before you can get visibility into VM security configurations, you need to set up Security Center data collection. This involves turning on data collection which automatically installs the Microsoft Monitoring Agent on all the VMs in your subscription.

1. On the Security Center dashboard, click **Security policy**, and then select your subscription.
2. For **Data collection**, in **Auto Provisioning** select **On**.
3. For **Default workspace configuration** leave it as **Use workspace(s) created by Security Center (default)**.
4. Under **Security Events** keep the default option of **Common**.
5. Click **Save** at the top of the page.

The Security Center data collection agent is then installed on all VMs, and data collection begins.

Set up a security policy

Security policies are used to define the items for which Security Center collects data and makes recommendations. You can apply different security policies to different sets of Azure resources. Although by default Azure resources are evaluated against all policy items, you can turn off individual policy items for all Azure resources or for a resource group. For in-depth information about Security Center security policies, see [Set security policies in Azure Security Center](#).

To set up a security policy for an entire subscription:

1. On the Security Center dashboard, select **Security policy** and then select your subscription.
2. On the **Security policy** blade, select **Security policy**.
3. On the **Security policy - Security policy** blade, turn on or turn off policy items that you want to apply to the subscription.
4. When you're finished selecting your settings, select **Save** at the top of the blade.

Search (Ctrl+ /) Save

POLICY COMPONENTS

Data Collection

Security policy

Email notifications

Pricing tier

Show recommendations for

System updates	On	Off	
Security configurations	On	Off	
Endpoint protection	On	Off	
Disk encryption	On	Off	
Network security groups	On	Off	
Web application firewall	On	Off	
Next generation firewall	On	Off	
Vulnerability Assessment	On	Off	
Storage Encryption	On	Off	
JIT Network Access	On	Off	UPGRADE
Adaptive Application Controls	On	Off	UPGRADE
SQL auditing & Threat detection	On	Off	
SQL Encryption	On	Off	

View VM configuration health

After you've turned on data collection and set a security policy, Security Center begins to provide alerts and recommendations. As VMs are deployed, the data collection agent is installed. Security Center is then populated with data for the new VMs. For in-depth information about VM configuration health, see [Protect your VMs in Security Center](#).

As data is collected, the resource health for each VM and related Azure resource is aggregated. The information is shown in an easy-to-read chart.

To view resource health:

1. On the Security Center dashboard, under **Prevention**, select **Compute**.
2. On the **Compute** blade, select **VMs and computers**. This view provides a summary of the configuration status for all your VMs.

Home > Security Center - Overview > Compute

Compute
SECURITY HEALTH

Add Computers **Filter**

Overview VMs and computers Cloud services

Filtered By: Power State: Running

Search by name

NAME	MONITORED	SYSTEM UPDATES	ENDPOINT PROTECT...	VULNERABILITIES	DISK ENCRYPTION
myVM	✓	!	✓	⚠	!
BackEnd	✓	✓	✓	⚠	!
Database	✓	✓	✓	⚠	!
FrontEnd	✓	✓	✓	⚠	!

To see all recommendations for a VM, select the VM. Recommendations and remediation are covered in more detail in the next section of this tutorial.

Remediate configuration issues

After Security Center begins to populate with configuration data, recommendations are made based on the security policy you set up. For instance, if a VM was set up without an associated network security group, a recommendation is made to create one.

To see a list of all recommendations:

1. On the Security Center dashboard, select **Recommendations**.
2. Select a specific recommendation. A list of all resources for which the recommendation applies appears.
3. To apply a recommendation, select the resource.
4. Follow the instructions for remediation steps.

In many cases, Security Center provides actionable steps you can take to address a recommendation without leaving Security Center. In the following example, Security Center detects a network security group that has an unrestricted inbound rule. On the recommendation page, you can select the **Edit inbound rules** button. The UI that is needed to modify the rule appears.

Restrict access through Internet facing endpoint

FrontEnd

Filter

VIRTUAL MACHINE	IP	STATE	SEVERITY	...
BackEnd	168.62.41.154	Open	⚠ Medium	...
Database	23.96.5.203	Open	⚠ Medium	...
FrontEnd	40.76.193.8	Open	⚠ Medium	...
myVM	168.62.166.52	Open	⚠ Medium	...

Network security group info

NETWORK SECURITY GROUP: FrontEnd

LOCATION: eastus

DESCRIPTION: Your NSG has inbound rules that open access to 'Any' or 'Internet' which might enable attackers to access your resources. We recommend that you edit the below inbound rules to restrict access to a specified set of sources.

Related inbound rules

PRIORITY	NAME	SOURCE	SERVICE	ACTIONS
1000	FrontEnd3389	*	Tcp	Allow
1001	FrontEnd5985	*	Tcp	Allow

Associated with

NAME	VIRTUAL MACHINE
FrontEnd	FrontEnd

As recommendations are remediated, they are marked as resolved.

View detected threats

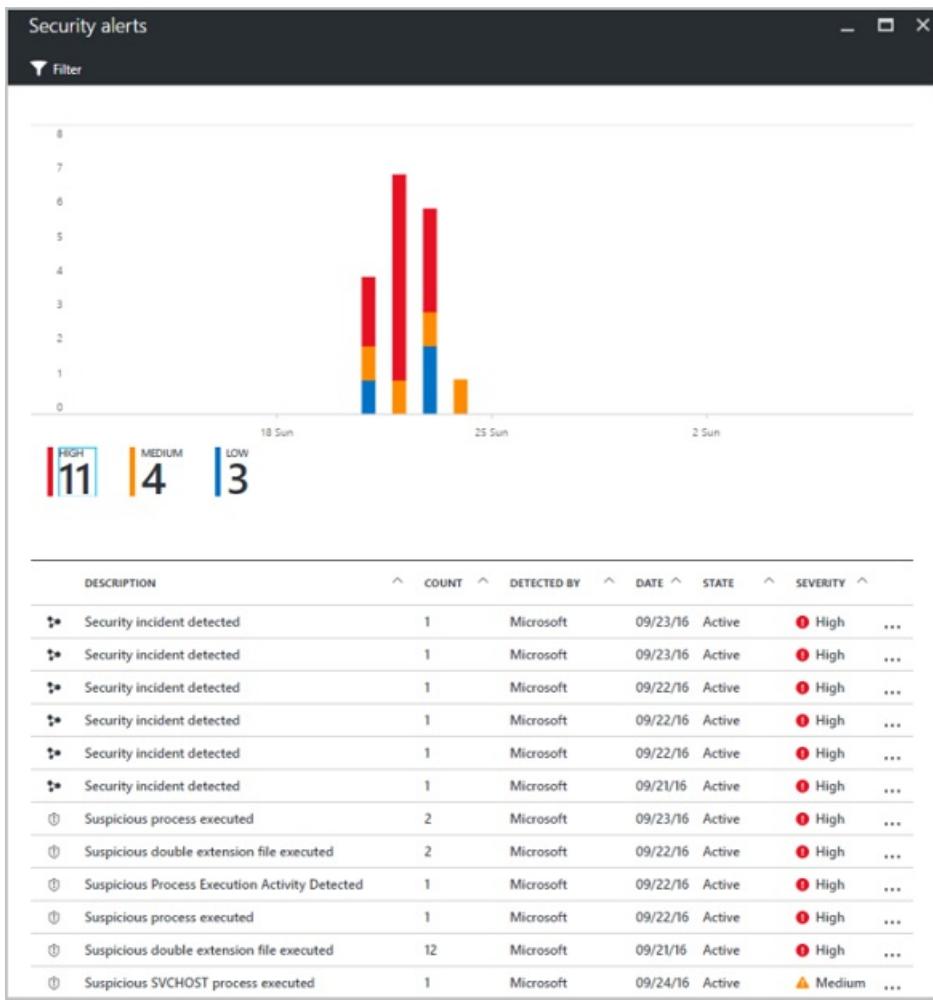
In addition to resource configuration recommendations, Security Center displays threat detection alerts. The security alerts feature aggregates data collected from each VM, Azure networking logs, and connected partner solutions to detect security threats against Azure resources. For in-depth information about Security Center threat detection capabilities, see [How does Security Center detect threats?](#).

The security alerts feature requires the Security Center pricing tier to be increased from *Free* to *Standard*. A **free trial** is available when you move to this higher pricing tier.

To change the pricing tier:

1. On the Security Center dashboard, click **Security policy**, and then select your subscription.
2. Select **Pricing tier**.
3. Select **Standard** and then click **Save** at the top of the blade.

After you've changed the pricing tier, the security alerts graph begins to populate as security threats are detected.



Select an alert to view information. For example, you can see a description of the threat, the detection time, all threat attempts, and the recommended remediation. In the following example, an RDP brute-force attack was detected, with 294 failed RDP attempts. A recommended resolution is provided.

Failed RDP Brute Force Attack	
myVMWindows	
DESCRIPTION	Several Remote Desktop login attempts were detected from 5.9.57.202, none of them succeeded. Event logs analysis shows that in the last 59 minutes there were 198 failed attempts. Some of the failed login attempts aimed at 2 existing user(s).
DETECTION TIME	Saturday, April 29, 2017, 10:59:56 PM
SEVERITY	 Medium
STATE	Active
ATTACKED RESOURCE	myVMWindows
SUBSCRIPTION	Free Trial (248352d0-5fc9-4c2e-8db3-d8b3462a0020)
DETECTED BY	 Microsoft
ACTION TAKEN	Detected
ALERT START TIME (UTC)	04/30/2017 05:00:06
NON-EXISTENT USERS	97
SUCCESSFUL LOGINS	0
FAILED USER LOGONS	server, administrator
REPORTS	Report: RDP Brute Forcing
REMEDIATION STEPS	<ol style="list-style-type: none"> 1. If available, add the source IP to NSG block list for 24 hours (see https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/) 2. Enforce the use of strong passwords and do not reuse them across multiple VMs and services (see http://windows.microsoft.com/en-us/Windows7/Tips-for-creating-strong-passwords-and-passphrases) 3. Create an allow list for RDP access in NSG (see https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/)

Next steps

In this tutorial, you set up Azure Security Center, and then reviewed VMs in Security Center. You learned how to:

- Set up data collection
- Set up security policies
- View and fix configuration health issues
- Review detected threats

Advance to the next tutorial to learn how to install a SQL\IIS\.NET stack on a pair of Windows VMs.

[SQL\IIS\.NET stack](#)

Tutorial: Install the SQL, IIS, .NET stack in a Windows VM with Azure PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

In this tutorial, we install a SQL, IIS, .NET stack using Azure PowerShell. This stack consists of two VMs running Windows Server 2016, one with IIS and .NET and the other with SQL Server.

- Create a VM
- Install IIS and the .NET Core SDK on the VM
- Create a VM running SQL Server
- Install the SQL Server extension

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.

Create an IIS VM

In this example, we use **New-AzVM** cmdlet in the PowerShell Cloud Shell to quickly create a Windows Server 2016 VM and then install IIS and the .NET Framework. The IIS and SQL VMs share a resource group and virtual network, so we create variables for those names.

```
$vmName = "IISVM"
$vNetName = "myIISSQLvNet"
$resourceGroup = "myIISSQLGroup"
New-AzVm ` 
    -ResourceGroupName $resourceGroup ` 
    -Name $vmName ` 
    -Location "East US" ` 
    -VirtualNetworkName $vNetName ` 
    -SubnetName "myIISSubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -AddressPrefix 192.168.0.0/16 ` 
    -PublicIpAddressName "myIISPublicIpAddress" ` 
    -OpenPorts 80,3389
```

Install IIS and the .NET framework using the custom script extension with the **Set-AzVMExtension** cmdlet.

```
Set-AzVMExtension ` 
    -ResourceGroupName $resourceGroup ` 
    -ExtensionName IIS ` 
    -VMName $vmName ` 
    -Publisher Microsoft.Compute ` 
    -ExtensionType CustomScriptExtension ` 
    -TypeHandlerVersion 1.4 ` 
    -SettingString '{"commandToExecute": "powershell Add-WindowsFeature Web-Server,Web-Asp-Net45,NET-Framework-Features"}' ` 
    -Location EastUS
```

Create another subnet

Create a second subnet for the SQL VM. Get the vNet using [Get-AzVirtualNetwork]
{/powershell/module/az.network/get-azvirtualnetwork}.

```
$vNet = Get-AzVirtualNetwork ` 
    -Name $vNetName ` 
    -ResourceGroupName $resourceGroup
```

Create a configuration for the subnet using [Add-AzVirtualNetworkSubnetConfig](#).

```
Add-AzVirtualNetworkSubnetConfig ` 
    -AddressPrefix 192.168.0.0/24 ` 
    -Name mySQLSubnet ` 
    -VirtualNetwork $vNet ` 
    -ServiceEndpoint Microsoft.Sql
```

Update the vNet with the new subnet information using [Set-AzVirtualNetwork](#)

```
$vNet | Set-AzVirtualNetwork
```

Azure SQL VM

Use a pre-configured Azure marketplace image of a SQL server to create the SQL VM. We first create the VM, then we install the SQL Server Extension on the VM.

```
New-AzVm ` 
    -ResourceGroupName $resourceGroup ` 
    -Name "mySQLVM" ` 
    -ImageName "MicrosoftSQLServer:SQL2016SP1-WS2016:Enterprise:latest" ` 
    -Location eastus ` 
    -VirtualNetworkName $vNetName ` 
    -SubnetName "mySQLSubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "mySQLPublicIpAddress" ` 
    -OpenPorts 3389,1401
```

Use [Set-AzVMSqlServerExtension](#) to add the SQL Server extension to the SQL VM.

```
Set-AzVMSqlServerExtension ` 
    -ResourceGroupName $resourceGroup ` 
    -VMName mySQLVM ` 
    -Name "SQLExtension" ` 
    -Location "EastUS"
```

Next steps

In this tutorial, you installed a SQL\IIS\.NET stack using Azure PowerShell. You learned how to:

- Create a VM
- Install IIS and the .NET Core SDK on the VM
- Create a VM running SQL Server
- Install the SQL Server extension

Advance to the next tutorial to learn how to secure IIS web server with SSL certificates.

[Secure IIS web server with SSL certificates](#)

Tutorial: Secure a web server on a Windows virtual machine in Azure with SSL certificates stored in Key Vault

1/17/2020 • 4 minutes to read • [Edit Online](#)

NOTE

Currently this doc only works for Generalized images. If attempting this tutorial using a Specialized disk you will receive an error.

To secure web servers, a Secure Sockets Layer (SSL) certificate can be used to encrypt web traffic. These SSL certificates can be stored in Azure Key Vault, and allow secure deployments of certificates to Windows virtual machines (VMs) in Azure. In this tutorial you learn how to:

- Create an Azure Key Vault
- Generate or upload a certificate to the Key Vault
- Create a VM and install the IIS web server
- Inject the certificate into the VM and configure IIS with an SSL binding

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.

Overview

Azure Key Vault safeguards cryptographic keys and secrets, such certificates or passwords. Key Vault helps streamline the certificate management process and enables you to maintain control of keys that access those certificates. You can create a self-signed certificate inside Key Vault, or upload an existing, trusted certificate that you already own.

Rather than using a custom VM image that includes certificates baked-in, you inject certificates into a running VM. This process ensures that the most up-to-date certificates are installed on a web server during deployment. If you renew or replace a certificate, you don't also have to create a new custom VM image. The latest certificates are automatically injected as you create additional VMs. During the whole process, the certificates never leave the Azure platform or are exposed in a script, command-line history, or template.

Create an Azure Key Vault

Before you can create a Key Vault and certificates, create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroupSecureWeb* in the *East US* location:

```
$resourceGroup = "myResourceGroupSecureWeb"
$location = "East US"
New-AzResourceGroup -ResourceGroupName $resourceGroup -Location $location
```

Next, create a Key Vault with [New-AzKeyVault](#). Each Key Vault requires a unique name, and should be all lower case. Replace `mykeyvault` in the following example with your own unique Key Vault name:

```
$keyvaultName="mykeyvault"
New-AzKeyVault -VaultName $keyvaultName ` 
    -ResourceGroup $resourceGroup ` 
    -Location $location ` 
    -EnabledForDeployment
```

Generate a certificate and store in Key Vault

For production use, you should import a valid certificate signed by trusted provider with [Import-AzKeyVaultCertificate](#). For this tutorial, the following example shows how you can generate a self-signed certificate with [Add-AzKeyVaultCertificate](#) that uses the default certificate policy from [New-AzKeyVaultCertificatePolicy](#).

```
$policy = New-AzKeyVaultCertificatePolicy ` 
    -SubjectName "CN=www.contoso.com" ` 
    -SecretContentType "application/x-pkcs12" ` 
    -IssuerName Self ` 
    -ValidityInMonths 12

Add-AzKeyVaultCertificate ` 
    -VaultName $keyvaultName ` 
    -Name "mycert" ` 
    -CertificatePolicy $policy
```

Create a virtual machine

Set an administrator username and password for the VM with [Get-Credential](#):

```
$cred = Get-Credential
```

Now you can create the VM with [New-AzVM](#). The following example creates a VM named *myVM* in the *EastUS* location. If they do not already exist, the supporting network resources are created. To allow secure web traffic, the cmdlet also opens port 443.

```

# Create a VM
New-AzVm `

    -ResourceGroupName $resourceGroup `

    -Name "myVM" `

    -Location $location `

    -VirtualNetworkName "myVnet" `

    -SubnetName "mySubnet" `

    -SecurityGroupName "myNetworkSecurityGroup" `

    -PublicIpAddressName "myPublicIpAddress" `

    -Credential $cred `

    -OpenPorts 443

# Use the Custom Script Extension to install IIS
Set-AzVMExtension -ResourceGroupName $resourceGroup `

    -ExtensionName "IIS" `

    -VMName "myVM" `

    -Location $location `

    -Publisher "Microsoft.Compute" `

    -ExtensionType "CustomScriptExtension" `

    -TypeHandlerVersion 1.8 `

    -SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-Server -IncludeManagementTools"}'

```

It takes a few minutes for the VM to be created. The last step uses the Azure Custom Script Extension to install the IIS web server with [Set-AzVmExtension](#).

Add a certificate to VM from Key Vault

To add the certificate from Key Vault to a VM, obtain the ID of your certificate with [Get-AzKeyVaultSecret](#). Add the certificate to the VM with [Add-AzVMSecret](#):

```

$certURL=(Get-AzKeyVaultSecret -VaultName $keyvaultName -Name "mycert").id

$vml=Get-AzVM -ResourceGroupName $resourceGroup -Name "myVM"
$vaultId=(Get-AzKeyVault -ResourceGroupName $resourceGroup -VaultName $keyVaultName).ResourceId
$vm = Add-AzVMSecret -VM $vm -SourceVaultId $vaultId -CertificateStore "My" -CertificateUrl $certURL

Update-AzVM -ResourceGroupName $resourceGroup -VM $vm

```

Configure IIS to use the certificate

Use the Custom Script Extension again with [Set-AzVMExtension](#) to update the IIS configuration. This update applies the certificate injected from Key Vault to IIS and configures the web binding:

```

$PublicSettings = '{
    "fileUris":["https://raw.githubusercontent.com/Azure-Samples/compute-automation-
configurations/master/secure-iis.ps1"],
    "commandToExecute":"powershell -ExecutionPolicy Unrestricted -File secure-iis.ps1"
}'

Set-AzVMExtension -ResourceGroupName $resourceGroup `

    -ExtensionName "IIS" `

    -VMName "myVM" `

    -Location $location `

    -Publisher "Microsoft.Compute" `

    -ExtensionType "CustomScriptExtension" `

    -TypeHandlerVersion 1.8 `

    -SettingString $publicSettings

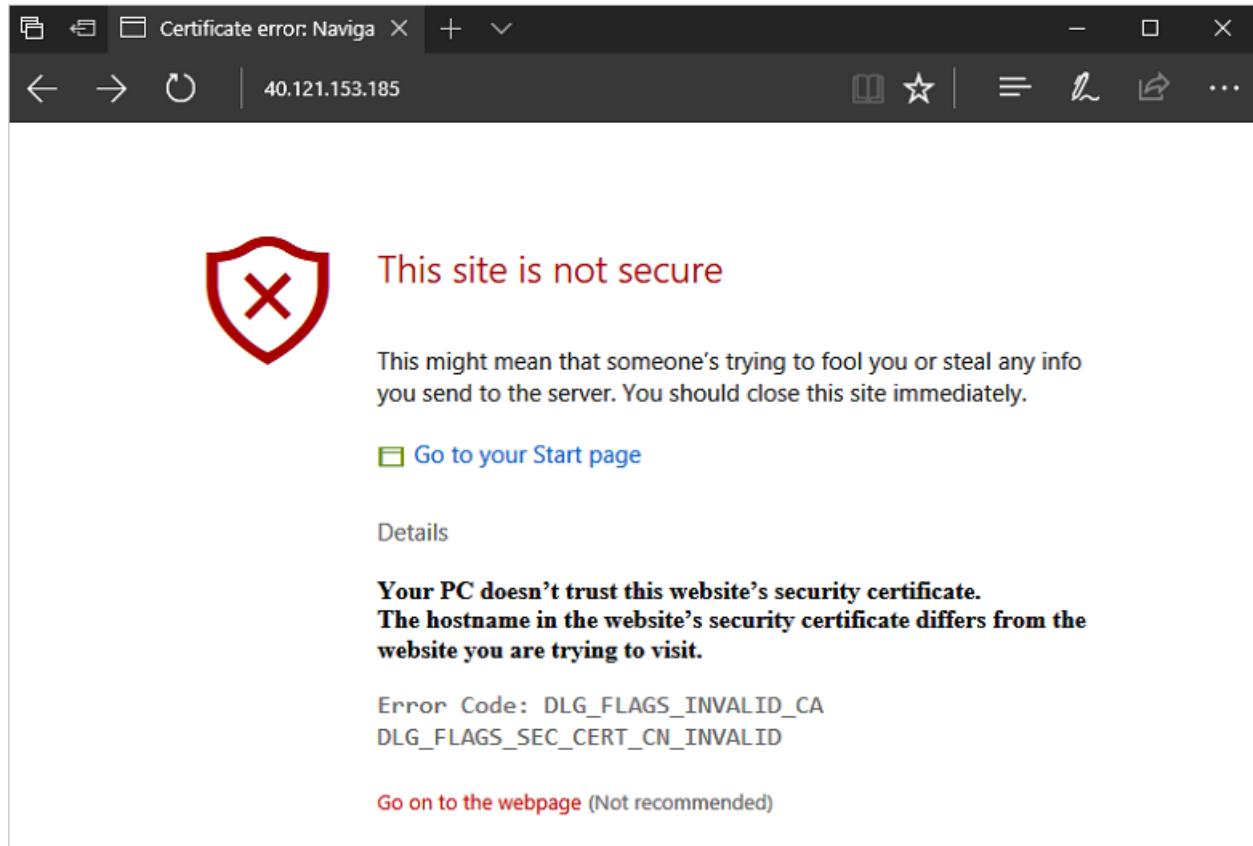
```

Test the secure web app

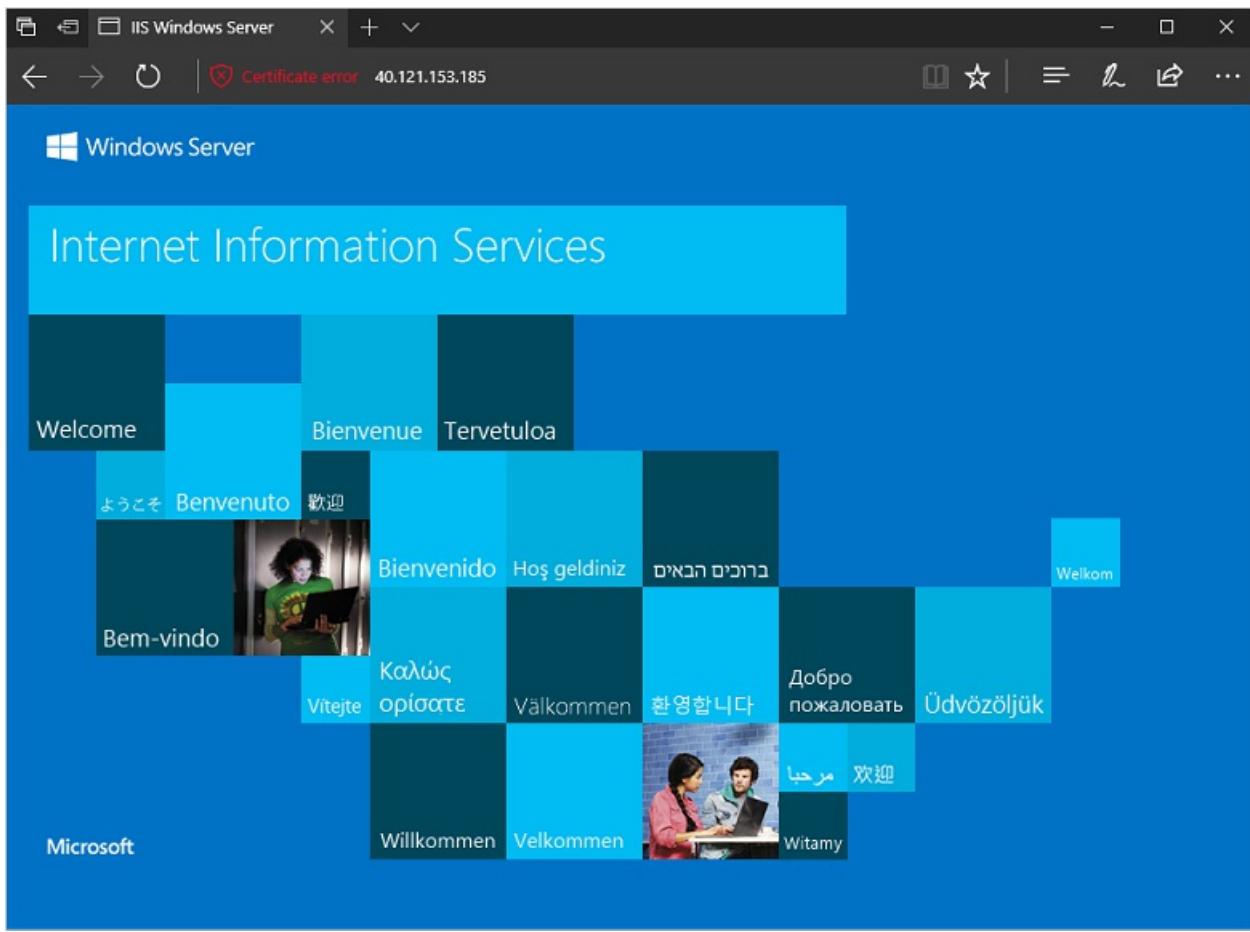
Obtain the public IP address of your VM with [Get-AzPublicIPAddress](#). The following example obtains the IP address for `myPublicIP` created earlier:

```
Get-AzPublicIPAddress -ResourceGroupName $resourceGroup -Name "myPublicIPAddress" | select "IpAddress"
```

Now you can open a web browser and enter `https://<myPublicIP>` in the address bar. To accept the security warning if you used a self-signed certificate, select **Details** and then **Go on to the webpage**:



Your secured IIS website is then displayed as in the following example:



Next steps

In this tutorial, you secured an IIS web server with an SSL certificate stored in Azure Key Vault. You learned how to:

- Create an Azure Key Vault
- Generate or upload a certificate to the Key Vault
- Create a VM and install the IIS web server
- Inject the certificate into the VM and configure IIS with an SSL binding

Follow this link to see pre-built virtual machine script samples.

[Windows virtual machine script samples](#)

Azure Virtual Machine PowerShell samples

11/13/2019 • 2 minutes to read • [Edit Online](#)

The following table provides links to PowerShell script samples that create and manage Windows virtual machines (VMs).

Create virtual machines	
Quickly create a virtual machine	Creates a resource group, a virtual machine, and all related resources, with a minimum of prompts.
Create a fully configured virtual machine	Creates a resource group, a virtual machine, and all related resources.
Create highly available virtual machines	Creates several virtual machines in a highly-available and load-balanced configuration.
Create a VM and run a configuration script	Creates a virtual machine and uses the Azure Custom Script extension to install IIS.
Create a VM and run a DSC configuration	Creates a virtual machine and uses the Azure Desired State Configuration (DSC) extension to install IIS.
Upload a VHD and create VMs	Uploads a local VHD file to Azure, creates an image from the VHD, and then creates a VM from that image.
Create a VM from a managed OS disk	Creates a virtual machine by attaching an existing Managed Disk as OS disk.
Create a VM from a snapshot	Creates a virtual machine from a snapshot by first creating a managed disk from the snapshot and then attaching the new managed disk as OS disk.
Manage storage	
Create a managed disk from a VHD in the same or a different subscription	Creates a managed disk from a specialized VHD as an OS disk, or from a data VHD as a data disk, in the same or a different subscription.
Create a managed disk from a snapshot	Creates a managed disk from a snapshot.
Copy a managed disk to the same or a different subscription	Copies a managed disk to the same or a different subscription that is in the same region as the parent managed disk.
Export a snapshot as a VHD to a storage account	Exports a managed snapshot as a VHD to a storage account in a different region.
Export the VHD of a managed disk to a storage account	Exports the underlying VHD of a managed disk to a storage account in a different region.

Create a snapshot from a VHD	Creates a snapshot from a VHD and then uses that snapshot to create multiple identical managed disks quickly.
Copy a snapshot to the same or a different subscription	Copies snapshot to the same or a different subscription that is in the same region as the parent snapshot.
Secure virtual machines	
Encrypt a VM and its data disks	Creates an Azure key vault, an encryption key, and a service principal, and then encrypts a VM.
Monitor virtual machines	
Monitor a VM with Azure Monitor	Creates a virtual machine, installs the Azure Log Analytics agent, and enrolls the VM in a Log Analytics workspace.
Collect details about all VMs in a subscription with PowerShell	Creates a csv that contains the VM Name, Resource Group Name, Region, Virtual Network, Subnet, Private IP Address, OS Type, and Public IP Address of the VMs in the provided subscription.

Create a virtual machine with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016. After running the script, you can access the virtual machine over RDP.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Variables for common values
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create a virtual machine
New-AzVM ` 
    -ResourceGroupName $resourceGroup ` 
    -Name $vmName ` 
    -Location $location ` 
    -ImageName "Win2016Datacenter" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIp" ` 
    -Credential $cred ` 
    -OpenPorts 3389
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzVM	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also opens port 80 and sets the administrative credentials.

COMMAND	NOTES
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a fully configured virtual machine with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016. After running the script, you can access the virtual machine over RDP.

This sample requires Azure PowerShell Az 1.0 or later. Run `Get-Module -ListAvailable Az` to see which versions are installed. If you need to install, see [Install Azure PowerShell module](#).

Run [Connect-AzAccount](#) to sign in to Azure.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

# Variables for common values
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create a subnet configuration
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24

# Create a virtual network
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location $location ` 
    -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

# Create a public IP address and specify a DNS name
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location $location ` 
    -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4

# Create an inbound network security group rule for port 3389
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp ` 
    -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * ` 
    -DestinationPortRange 3389 -Access Allow

# Create a network security group
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location ` 
    -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP

# Create a virtual network card and associate with public IP address and NSG
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location $location ` 
    -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1 | ` 
    Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | ` 
    Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2016-Datacenter -Version latest | ` 
    Add-AzVMNetworkInterface -Id $nic.Id

# Create a virtual machine
New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.

COMMAND	NOTES
New-AzVirtualNetworkSubnetConfig	Creates a subnet configuration. This configuration is used with the virtual network creation process.
New-AzVirtualNetwork	Creates a virtual network.
New-AzPublicIpAddress	Creates a public IP address.
New-AzNetworkSecurityRuleConfig	Creates a network security group rule configuration. This configuration is used to create an NSG rule when the NSG is created.
New-AzNetworkSecurityGroup	Creates a network security group.
Get-AzVirtualNetworkSubnetConfig	Gets subnet information. This information is used when creating a network interface.
New-AzNetworkInterface	Creates a network interface.
New-AzVMConfig	Creates a VM configuration. This configuration includes information such as VM name, operating system, and administrative credentials. The configuration is used during VM creation.
New-AzVM	Create a virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Load balance traffic between highly available virtual machines

11/13/2019 • 5 minutes to read • [Edit Online](#)

This script sample creates everything needed to run several Windows Server 2016 virtual machines configured in a highly available and load balanced configuration. After running the script, you will have three virtual machines, joined to an Azure Availability Set, and accessible through an Azure Load Balancer.

This sample requires Azure PowerShell Az 1.0 or later. Run `Get-Module -ListAvailable Az` to see which versions are installed. If you need to install, see [Install Azure PowerShell module](#).

Run [Connect-AzAccount](#) to sign in to Azure.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Variables for common values
$rgName='MyResourceGroup'
$location='eastus'

# Create user object
$cred = Get-Credential -Message 'Enter a username and password for the virtual machine.'

# Create a resource group.
New-AzResourceGroup -Name $rgName -Location $location

# Create a virtual network.
$subnet = New-AzVirtualNetworkSubnetConfig -Name 'MySubnet' -AddressPrefix 192.168.1.0/24

$vnet = New-AzVirtualNetwork -ResourceGroupName $rgName -Name 'MyVnet' ` 
    -AddressPrefix 192.168.0.0/16 -Location $location -Subnet $subnet

# Create a public IP address.
$publicIp = New-AzPublicIpAddress -ResourceGroupName $rgName -Name 'myPublicIP' ` 
    -Location $location -AllocationMethod Dynamic

# Create a front-end IP configuration for the website.
$feip = New-AzLoadBalancerFrontendIpConfig -Name 'myFrontEndPool' -PublicIpAddress $publicIp

# Create the back-end address pool.
$bepool = New-AzLoadBalancerBackendAddressPoolConfig -Name 'myBackEndPool'

# Creates a load balancer probe on port 80.
$probe = New-AzLoadBalancerProbeConfig -Name 'myHealthProbe' -Protocol Http -Port 80 ` 
    -RequestPath / -IntervalInSeconds 360 -ProbeCount 5

# Creates a load balancer rule for port 80.
$rule = New-AzLoadBalancerRuleConfig -Name 'myLoadBalancerRuleWeb' -Protocol Tcp ` 
    -Probe $probe -FrontendPort 80 -BackendPort 80 ` 
    -FrontendIpConfiguration $feip -BackendAddressPool $bePool

# Create three NAT rules for port 3389.
$natrule1 = New-AzLoadBalancerInboundNatRuleConfig -Name 'myLoadBalancerRDP1' -FrontendIpConfiguration $feip ` 
    -Protocol tcp -FrontendPort 4221 -BackendPort 3389

$natrule2 = New-AzLoadBalancerInboundNatRuleConfig -Name 'myLoadBalancerRDP2' -FrontendIpConfiguration $feip ` 
    -Protocol tcp -FrontendPort 4222 -BackendPort 3389
```

```

$natrule3 = New-AzLoadBalancerInboundNatRuleConfig -Name 'myLoadBalancerRDP3' -FrontendIpConfiguration $feip ` 
-Protocol tcp -FrontendPort 4223 -BackendPort 3389

# Create a load balancer.
$lb = New-AzLoadBalancer -ResourceGroupName $rgName -Name 'MyLoadBalancer' -Location $location ` 
-FrontendIpConfiguration $feip -BackendAddressPool $bepool ` 
-Probe $probe -LoadBalancingRule $rule -InboundNatRule $natrule1,$natrule2,$natrule3

# Create a network security group rule for port 3389.
$rule1 = New-AzNetworkSecurityRuleConfig -Name 'myNetworkSecurityGroupRuleRDP' -Description 'Allow RDP' ` 
-Access Allow -Protocol Tcp -Direction Inbound -Priority 1000 ` 
-SourceAddressPrefix Internet -SourcePortRange * ` 
-DestinationAddressPrefix * -DestinationPortRange 3389

# Create a network security group rule for port 80.
$rule2 = New-AzNetworkSecurityRuleConfig -Name 'myNetworkSecurityGroupRuleHTTP' -Description 'Allow HTTP' ` 
-Access Allow -Protocol Tcp -Direction Inbound -Priority 2000 ` 
-SourceAddressPrefix Internet -SourcePortRange * ` 
-DestinationAddressPrefix * -DestinationPortRange 80

# Create a network security group
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $RgName -Location $location ` 
-Name 'myNetworkSecurityGroup' -SecurityRules $rule1,$rule2

# Create three virtual network cards and associate with public IP address and NSG.
$nicVM1 = New-AzNetworkInterface -ResourceGroupName $rgName -Location $location ` 
-Name 'MyNic1' -LoadBalancerBackendAddressPool $bepool -NetworkSecurityGroup $nsg ` 
-LoadBalancerInboundNatRule $natrule1 -Subnet $vnet.Subnets[0]

$nicVM2 = New-AzNetworkInterface -ResourceGroupName $rgName -Location $location ` 
-Name 'MyNic2' -LoadBalancerBackendAddressPool $bepool -NetworkSecurityGroup $nsg ` 
-LoadBalancerInboundNatRule $natrule2 -Subnet $vnet.Subnets[0]

$nicVM3 = New-AzNetworkInterface -ResourceGroupName $rgName -Location $location ` 
-Name 'MyNic3' -LoadBalancerBackendAddressPool $bepool -NetworkSecurityGroup $nsg ` 
-LoadBalancerInboundNatRule $natrule3 -Subnet $vnet.Subnets[0]

# Create an availability set.
$as = New-AzAvailabilitySet -ResourceGroupName $rgName -Location $location ` 
-Name 'MyAvailabilitySet' -Sku Aligned -PlatformFaultDomainCount 3 -PlatformUpdateDomainCount 3

# Create three virtual machines.

# ##### VM1 #####
# Create a virtual machine configuration
$vmConfig = New-AzVMConfig -VMName 'myVM1' -VMSize Standard_DS2 -AvailabilitySetId $as.Id | ` 
Set-AzVMOperatingSystem -Windows -ComputerName 'myVM1' -Credential $cred | ` 
Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer ` 
-Skus 2016-Datacenter -Version latest | Add-AzVMNetworkInterface -Id $nicVM1.Id

# Create a virtual machine
$vm1 = New-AzVM -ResourceGroupName $rgName -Location $location -VM $vmConfig

# ##### VM2 #####
# Create a virtual machine configuration
$vmConfig = New-AzVMConfig -VMName 'myVM2' -VMSize Standard_DS2 -AvailabilitySetId $as.Id | ` 
Set-AzVMOperatingSystem -Windows -ComputerName 'myVM2' -Credential $cred | ` 
Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer ` 
-Skus 2016-Datacenter -Version latest | Add-AzVMNetworkInterface -Id $nicVM2.Id

# Create a virtual machine
$vm2 = New-AzVM -ResourceGroupName $rgName -Location $location -VM $vmConfig

# ##### VM3 #####
# Create a virtual machine configuration
$vmConfig = New-AzVMConfig -VMName 'myVM3' -VMSize Standard_DS2 -AvailabilitySetId $as.Id | ` 

```

```

$vmConfig = New-AzVmConfig -VMName 'myVM' -VmSize Standard_D2s -AvailabilitySet $avSet | 
    Set-AzVMOperatingSystem -Windows -ComputerName 'myVM3' -Credential $cred | 
    Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer |
    -Skus 2016-Datacenter -Version latest | Add-AzVMNetworkInterface -Id $nicVM3.Id

# Create a virtual machine
$vm3 = New-AzVM -ResourceGroupName $rgName -Location $location -VM $vmConfig

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzVirtualNetworkSubnetConfig	Creates a subnet configuration. This configuration is used with the virtual network creation process.
New-AzVirtualNetwork	Creates a virtual network.
New-AzPublicIpAddress	Creates a public IP address.
New-AzLoadBalancerFrontendIpConfig	Creates a front-end IP configuration for a load balancer.
New-AzLoadBalancerBackendAddressPoolConfig	Creates a backend address pool configuration for a load balancer.
New-AzLoadBalancerProbeConfig	Creates a probe configuration for a load balancer.
New-AzLoadBalancerRuleConfig	Creates a rule configuration for a load balancer.
New-AzLoadBalancerInboundNatRuleConfig	Creates an inbound NAT rule configuration for a load balancer.
New-AzLoadBalancer	Creates a load balancer.
New-AzNetworkSecurityRuleConfig	Creates a network security group rule configuration. This configuration is used to create an NSG rule when the NSG is created.
New-AzNetworkSecurityGroup	Creates a network security group.
Get-AzVirtualNetworkSubnetConfig	Gets subnet information. This information is used when creating a network interface.
New-AzNetworkInterface	Creates a network interface.

COMMAND	NOTES
New-AzVMConfig	Creates a VM configuration. This configuration includes information such as VM name, operating system, and administrative credentials. The configuration is used during VM creation.
New-AzVM	Create a virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

You can also create the VMs using your own custom managed image. In the VM configuration, for

`Set-AzVMSourceImage` use the `-Id` and `-VM` parameters instead of `-PublisherName`, `-Offer`, `-Skus`, and `-Version`.

For example, creating the VM config would be:

```
$vmConfig = New-AzVMConfig -VMName 'myVM3' -VMSize Standard_DS1_v2 -AvailabilitySetId $as.Id | `  
    Set-AzVMOperatingSystem -Windows -ComputerName 'myVM3' -Credential $cred | `  
    Set-AzVMSourceImage -Id <Image.ID of the custom managed image> | Add-AzVMNetworkInterface -Id $nicVM3.Id
```

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create an IIS VM with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016, and then uses the Azure Virtual Machine Custom Script Extension to install IIS. After running the script, you can access the default IIS website on the public IP address of the virtual machine.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Variables for common values
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create a virtual machine
New-AzVM `

    -ResourceGroupName $resourceGroup `

    -Name $vmName `

    -Location $location `

    -ImageName "Win2016Datacenter" `

    -VirtualNetworkName "myVnet" `

    -SubnetName "mySubnet" `

    -SecurityGroupName "myNetworkSecurityGroup" `

    -PublicIpAddressName "myPublicIp" `

    -Credential $cred `

    -OpenPorts 80

# Install IIS
$PublicSettings = '{"commandToExecute":"powershell Add-WindowsFeature Web-Server"}'

Set-AzVMExtension -ExtensionName "IIS" -ResourceGroupName $resourceGroup -VMName $vmName `

    -Publisher "Microsoft.Compute" -ExtensionType "CustomScriptExtension" -TypeHandlerVersion 1.4 `

    -SettingString $PublicSettings -Location $location
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzVM	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also opens port 80 and sets the administrative credentials.
Set-AzVMExtension	Add a VM extension to the virtual machine. In this sample, the custom script extension is used to install IIS.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create an IIS VM with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016, and then uses the Azure Virtual Machine DSC Extension to install IIS. After running the script, you can access the default IIS website on the public IP address of the virtual machine.

This sample requires Azure PowerShell Az 1.0 or later. Run `Get-Module -ListAvailable Az` to see which versions are installed. If you need to install, see [Install Azure PowerShell module](#).

Run [Connect-AzAccount](#) to sign in to Azure.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Variables for common values
$resourceGroup = "myResourceGroup"
.setLocation = "westeurope"
$vmName = "myVM"

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a virtual machine
New-AzVM ` 
    -ResourceGroupName $resourceGroup ` 
    -Name $vmName ` 
    -Location $location ` 
    -ImageName "Win2016Datacenter" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIp" ` 
    -Credential $cred ` 
    -OpenPorts 80

# Install IIS
$PublicSettings = '{
    "ModulesURL": "https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip",
    "configurationFunction": "ContosoWebsite.ps1\\ContosoWebsite",
    "Properties": { "MachineName": "myVM" }
}'

Set-AzVMExtension -ExtensionName "DSC" -ResourceGroupName $resourceGroup -VMName $vmName ` 
    -Publisher "Microsoft.PowerShell" -ExtensionType "DSC" -TypeHandlerVersion 2.19 ` 
    -SettingString $PublicSettings -Location $location
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzVM	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also opens port 80 and sets the administrative credentials.
Set-AzVMExtension	Add a VM extension to the virtual machine. In this sample, the DSC extension is used to install IIS.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Sample script to upload a VHD to Azure and create a new VM

11/13/2019 • 3 minutes to read • [Edit Online](#)

This script takes a local .vhd file from a generalized VM and uploads it to Azure, creates a Managed Disk image and uses the to create a new VM.

This sample requires Azure PowerShell Az 1.0 or later. Run `Get-Module -ListAvailable Az` to see which versions are installed. If you need to install, see [Install Azure PowerShell module](#).

Run [Connect-AzAccount](#) to sign in to Azure.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Provide values for the variables
$resourceGroup = 'myResourceGroup'
.setLocation = 'EastUS'
$storageaccount = 'mystorageaccount'
$storageType = 'Standard_LRS'
$containername = 'mycontainer'
$localPath = 'C:\Users\Public\Documents\Hyper-V\VHDs\generalized.vhd'
$vmName = 'myVM'
$imageName = 'myImage'
$vhdName = 'myUploadedVhd.vhd'
$diskSizeGB = '128'
$subnetName = 'mySubnet'
$vnetName = 'myVnet'
$ipName = 'myPip'
$nicName = 'myNic'
$nsgName = 'myNsg'
$ruleName = 'myRdpRule'
$computerName = 'myComputerName'
$vmSize = 'Standard_DS1_v2'

# Get the username and password to be used for the administrators account on the VM.
# This is used when connecting to the VM using RDP.

$cred = Get-Credential

# Upload the VHD
New-AzResourceGroup -Name $resourceGroup -Location $location
New-AzStorageAccount -ResourceGroupName $resourceGroup -Name $storageAccount -Location $location ` 
    -SkuName $storageType -Kind "Storage"
$urlOfUploadedImageVhd = ('https://'+$storageaccount+'.blob.core.windows.net/'+$containername+'/'+
    $vhdName)
Add-AzVhd -ResourceGroupName $resourceGroup -Destination $urlOfUploadedImageVhd ` 
    -LocalFilePath $localPath

# Note: Uploading the VHD may take awhile!

# Create a managed image from the uploaded VHD
$imageConfig = New-AzImageConfig -Location $location
$imageConfig = Set-AzImageOsDisk -Image $imageConfig -OsType Windows -OsState Generalized ` 
    -BlobUri $urlOfUploadedImageVhd
$image = New-AzImage -ImageName $imageName -ResourceGroupName $resourceGroup -Image $imageConfig

# Create the networking resources
```

```

$singleSubnet = New-AzVirtualNetworkSubnetConfig -Name $subnetName -AddressPrefix 10.0.0.0/24
$vnet = New-AzVirtualNetwork -Name $vnetName -ResourceGroupName $resourceGroup -Location $location ` 
-AddressPrefix 10.0.0.0/16 -Subnet $singleSubnet
$pip = New-AzPublicIpAddress -Name $ipName -ResourceGroupName $resourceGroup -Location $location ` 
-AllocationMethod Dynamic
$rdpRule = New-AzNetworkSecurityRuleConfig -Name $ruleName -Description 'Allow RDP' -Access Allow ` 
-Protocol Tcp -Direction Inbound -Priority 110 -SourceAddressPrefix Internet -SourcePortRange * ` 
-DestinationAddressPrefix * -DestinationPortRange 3389
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location ` 
-Name $nsgName -SecurityRules $rdpRule
$nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName $resourceGroup -Location $location ` 
-SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id
$vnet = Get-AzVirtualNetwork -ResourceGroupName $resourceGroup -Name $vnetName

# Start building the VM configuration
$vm = New-AzVMConfig -VMName $vmName -VMSize $vmSize

# Set the VM image as source image for the new VM
$vm = Set-AzVMSourceImage -VM $vm -Id $image.Id

# Finish the VM configuration and add the NIC.
$vm = Set-AzVMDisk -VM $vm -DiskSizeInGB $diskSizeGB -CreateOption FromImage -Caching ReadWrite
$vm = Set-AzVMOperatingSystem -VM $vm -Windows -ComputerName $computerName -Credential $cred ` 
-ProvisionVMAgent -EnableAutoUpdate
$vm = Add-AzVMNetworkInterface -VM $vm -Id $nic.Id

# Create the VM
New-AzVM -VM $vm -ResourceGroupName $resourceGroup -Location $location

# Verify that the VM was created
$vmList = Get-AzVM -ResourceGroupName $resourceGroup
$vmList.Name

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name $resourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzStorageAccount	Creates a storage account.
Add-AzVhd	Uploads a virtual hard disk from an on-premises virtual machine to a blob in a cloud storage account in Azure.
New-AzImageConfig	Creates a configurable image object.
Set-AzImageOsDisk	Sets the operating system disk properties on an image object.

COMMAND	NOTES
New-AzImage	Creates a new image.
New-AzVirtualNetworkSubnetConfig	Creates a subnet configuration. This configuration is used with the virtual network creation process.
New-AzVirtualNetwork	Creates a virtual network.
New-AzPublicIpAddress	Creates a public IP address.
New-AzNetworkInterface	Creates a network interface.
New-AzNetworkSecurityRuleConfig	Creates a network security group rule configuration. This configuration is used to create an NSG rule when the NSG is created.
New-AzNetworkSecurityGroup	Creates a network security group.
Get-AzVirtualNetwork	Gets a virtual network in a resource group.
New-AzVMConfig	Creates a VM configuration. This configuration includes information such as VM name, operating system, and administrative credentials. The configuration is used during VM creation.
Set-AzVMSourceImage	Specifies an image for a virtual machine.
Set-AzVMOSDisk	Sets the operating system disk properties on a virtual machine.
Set-AzVMOperatingSystem	Sets the operating system disk properties on a virtual machine.
Add-AzVMNetworkInterface	Adds a network interface to a virtual machine.
New-AzVM	Create a virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a virtual machine using an existing managed OS disk with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a virtual machine by attaching an existing managed disk as OS disk. Use this script in preceding scenarios:

- Create a VM from an existing managed OS disk that was copied from a managed disk in different subscription
- Create a VM from an existing managed disk that was created from a specialized VHD file
- Create a VM from an existing managed OS disk that was created from a snapshot

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group
$resourceGroupName = 'yourResourceGroupName'

#Provide the name of the snapshot that will be used to create OS disk
$snapshotName = 'yourSnapshotName'

#Provide the name of the OS disk that will be created using the snapshot
$osDiskName = 'yourOSDiskName'

#Provide the name of an existing virtual network where virtual machine will be created
$virtualNetworkName = 'yourVNETName'

#Provide the name of the virtual machine
$virtualMachineName = 'yourVMName'

#Provide the size of the virtual machine
#e.g. Standard_DS3
#Get all the vm sizes in a region using below script:
#e.g. Get-AzVMSize -Location westus
$virtualMachineSize = 'Standard_DS3'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshot = Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName

$diskConfig = New-AzDiskConfig -Location $snapshot.Location -SourceResourceId $snapshot.Id -CreateOption Copy

$disk = New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName $osDiskName

#Initialize virtual machine configuration
$VirtualMachine = New-AzVMConfig -VMName $virtualMachineName -VMSize $virtualMachineSize

#Use the Managed Disk Resource Id to attach it to the virtual machine. Please change the OS type to linux if
#OS disk has linux OS
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -ManagedDiskId $disk.Id -CreateOption Attach -Windows

#Create a public IP for the VM
$publicIp = New-AzPublicIpAddress -Name ($VirtualMachineName.ToLower()+'_ip') -ResourceGroupName
$resourceGroupName -Location $snapshot.Location -AllocationMethod Dynamic

#Get the virtual network where virtual machine will be hosted
$vnet = Get-AzVirtualNetwork -Name $virtualNetworkName -ResourceGroupName $resourceGroupName

# Create NIC in the first subnet of the virtual network
$nic = New-AzNetworkInterface -Name ($VirtualMachineName.ToLower()+'_nic') -ResourceGroupName
$resourceGroupName -Location $snapshot.Location -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $publicIp.Id

$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $nic.Id

#Create the virtual machine with Managed Disk
New-AzVM -VM $VirtualMachine -ResourceGroupName $resourceGroupName -Location $snapshot.Location

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to get managed disk properties, attach a managed disk to a new VM and create a VM. Each item in the table links to command specific documentation.

COMMAND	NOTES
Get-AzDisk	Gets disk object based on the name and the resource group of a disk. Id property of the returned disk object is used to attach the disk to a new VM
New-AzVMConfig	Creates a VM configuration. This configuration includes information such as VM name, operating system, and administrative credentials. The configuration is used during VM creation.
Set-AzVMOSDisk	Attaches a managed disk using the Id property of the disk as OS disk to a new virtual machine
New-AzPublicIpAddress	Creates a public IP address.
New-AzNetworkInterface	Creates a network interface.
New-AzVM	Create a virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

For marketplace images use [Set-AzVMPlan](#) to set the plan information.

```
Set-AzVMPlan -VM $VirtualMachine -Publisher $Publisher -Product $Product -Name $Name
```

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a virtual machine from a snapshot with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a virtual machine from a snapshot of an OS disk.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group
$resourceGroupName = 'yourResourceGroupName'

#Provide the name of the snapshot that will be used to create OS disk
$snapshotName = 'yourSnapshotName'

#Provide the name of the OS disk that will be created using the snapshot
$osDiskName = 'yourOSDiskName'

#Provide the name of an existing virtual network where virtual machine will be created
$virtualNetworkName = 'yourVNETName'

#Provide the name of the virtual machine
$virtualMachineName = 'yourVMName'

#Provide the size of the virtual machine
#e.g. Standard_DS3
#Get all the vm sizes in a region using below script:
#e.g. Get-AzVMSize -Location westus
$virtualMachineSize = 'Standard_DS3'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshot = Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName

$diskConfig = New-AzDiskConfig -Location $snapshot.Location -SourceResourceId $snapshot.Id -CreateOption Copy

$disk = New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName $osDiskName

#Initialize virtual machine configuration
$VirtualMachine = New-AzVMConfig -VMName $virtualMachineName -VMSize $virtualMachineSize

#Use the Managed Disk Resource Id to attach it to the virtual machine. Please change the OS type to linux if
OS disk has linux OS
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -ManagedDiskId $disk.Id -CreateOption Attach -Windows

#Create a public IP for the VM
$publicIp = New-AzPublicIpAddress -Name ($VirtualMachineName.ToLower()+'_ip') -ResourceGroupName
$resourceGroupName -Location $snapshot.Location -AllocationMethod Dynamic

#Get the virtual network where virtual machine will be hosted
$vnet = Get-AzVirtualNetwork -Name $virtualNetworkName -ResourceGroupName $resourceGroupName

# Create NIC in the first subnet of the virtual network
$nic = New-AzNetworkInterface -Name ($VirtualMachineName.ToLower()+'_nic') -ResourceGroupName
$resourceGroupName -Location $snapshot.Location -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $publicIp.Id

$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $nic.Id

#Create the virtual machine with Managed Disk
New-AzVM -VM $VirtualMachine -ResourceGroupName $resourceGroupName -Location $snapshot.Location

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to get snapshot properties, create a managed disk from snapshot and create a VM. Each item in the table links to command specific documentation.

COMMAND	NOTES
Get-AzSnapshot	Gets a snapshot using snapshot name.
New-AzDiskConfig	Creates a disk configuration. This configuration is used with the disk creation process.
New-AzDisk	Creates a managed disk.
New-AzVMConfig	Creates a VM configuration. This configuration includes information such as VM name, operating system, and administrative credentials. The configuration is used during VM creation.
Set-AzVMOSDisk	Attaches the managed disk as OS disk to the virtual machine
New-AzPublicIpAddress	Creates a public IP address.
New-AzNetworkInterface	Creates a network interface.
New-AzVM	Creates a virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a managed disk from a VHD file in a storage account in same or different subscription with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a managed disk from a VHD file in a storage account in same or different subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to managed OS disk to create a virtual machine. Also, use it to import a data VHD to managed data disk.

Don't create multiple identical managed disks from a VHD file in small amount of time. To create managed disks from a vhd file, blob snapshot of the vhd file is created and then it is used to create managed disks. Only one blob snapshot can be created in a minute that causes disk creation failures due to throttling. To avoid this throttling, create a [managed snapshot from the vhd file](#) and then use the managed snapshot to create multiple managed disks in short amount of time.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id where Managed Disks will be created
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group where Managed Disks will be created.
$resourceGroupName = 'yourResourceGroupName'

#Provide the name of the Managed Disk
$diskName = 'yourDiskName'

#Provide the size of the disks in GB. It should be greater than the VHD file size.
$diskSize = '128'

#Provide the storage type for Managed Disk. Premium_LRS or Standard_LRS.
$storageType = 'Premium_LRS'

#Provide the Azure region (e.g. westus) where Managed Disk will be located.
#This location should be same as the storage account where VHD file is stored
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Provide the URI of the VHD file (page blob) in a storage account. Please note that this is NOT the SAS URI of
the storage container where VHD file is stored.
#e.g. https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
#Note: VHD file can be deleted as soon as Managed Disk is created.
$sourceVHDURI = 'https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd'

#Provide the resource Id of the storage account where VHD file is stored.
#e.g. /subscriptions/6472s1g8-h217-446b-b509-
314e17e1efb0/resourceGroups/MDDemo/providers/Microsoft.Storage/storageAccounts/contosostorageaccount
#This is an optional parameter if you are creating managed disk in the same subscription
$storageAccountId =
'/subscriptions/yourSubscriptionId/resourceGroups/yourResourceGroupName/providers/Microsoft.Storage/storageAcc
ounts/yourStorageAccountName'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$diskConfig = New-AzDiskConfig -AccountType $storageType -Location $location -CreateOption Import -
StorageAccountId $storageAccountId -SourceUri $sourceVHDURI

New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName $diskName

```

Script explanation

This script uses following commands to create a managed disk from a VHD in different subscription. Each command in the table links to command specific documentation.

COMMAND	NOTES
New-AzDiskConfig	Creates disk configuration that is used for disk creation. It includes storage type, location, resource Id of the storage account where the parent VHD is stored, VHD URI of the parent VHD.
New-AzDisk	Creates a disk using disk configuration, disk name, and resource group name passed as parameters.

Next steps

[Create a virtual machine by attaching a managed disk as OS disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a managed disk from a snapshot with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a managed disk from a snapshot. Use it to restore a virtual machine from snapshots of OS and data disks. Create OS and data managed disks from respective snapshots and then create a new virtual machine by attaching managed disks. You can also restore data disks of an existing VM by attaching data disks created from snapshots.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group
$resourceGroupName = 'yourResourceGroupName'

#Provide the name of the snapshot that will be used to create Managed Disks
$snapshotName = 'yourSnapshotName'

#Provide the name of the Managed Disk
$diskName = 'yourManagedDiskName'

#Provide the size of the disks in GB. It should be greater than the VHD file size.
$diskSize = '128'

#Provide the storage type for Managed Disk. PremiumLRS or StandardLRS.
$storageType = 'Premium_LRS'

#Provide the Azure region (e.g. westus) where Managed Disks will be located.
#This location should be same as the snapshot location
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshot = Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -CreateOption Copy -SourceResourceId $snapshot.Id

New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName $diskName
```

Script explanation

This script uses following commands to create a managed disk from a snapshot. Each command in the table links to command specific documentation.

COMMAND	NOTES
Get-AzSnapshot	Gets snapshot properties.
New-AzDiskConfig	Creates disk configuration that is used for disk creation. It includes the resource Id of the parent snapshot, location that is same as the location of parent snapshot and the storage type.
New-AzDisk	Creates a disk using disk configuration, disk name, and resource group name passed as parameters.

Next steps

[Create a virtual machine from a managed disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Copy managed disks in the same subscription or different subscription with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a copy of an existing managed disk in the same subscription or different subscription. The new disk is created in the same region as the parent managed disk.

If needed, install the Azure PowerShell module using the instructions found in the [Azure PowerShell guide](#), and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the `.ssh` directory of your user profile.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id of the subscription where managed disk exists
$sourceSubscriptionId='yourSourceSubscriptionId'

#Provide the name of your resource group where managed disk exists
$sourceResourceGroupName='mySourceResourceGroupName'

#Provide the name of the managed disk
$managedDiskName='myDiskName'

#Set the context to the subscription Id where Managed Disk exists
Select-AzSubscription -SubscriptionId $sourceSubscriptionId

#Get the source managed disk
$managedDisk= Get-AzDisk -ResourceGroupName $sourceResourceGroupName -DiskName $managedDiskName

#Provide the subscription Id of the subscription where managed disk will be copied to
#If managed disk is copied to the same subscription then you can skip this step
$targetSubscriptionId='yourTargetSubscriptionId'

#Name of the resource group where snapshot will be copied to
$targetResourceGroupName='myTargetResourceGroupName'

#Set the context to the subscription Id where managed disk will be copied to
#If snapshot is copied to the same subscription then you can skip this step
Select-AzSubscription -SubscriptionId $targetSubscriptionId

$diskConfig = New-AzDiskConfig -SourceResourceId $managedDisk.Id -Location $managedDisk.Location -CreateOption Copy

#Create a new managed disk in the target subscription and resource group
New-AzDisk -Disk $diskConfig -DiskName $managedDiskName -ResourceGroupName $targetResourceGroupName
```

Script explanation

This script uses following commands to create a new managed disk in the target subscription using the Id of the source managed disk. Each command in the table links to command specific documentation.

COMMAND	NOTES
New-AzDiskConfig	Creates disk configuration that is used for disk creation. It includes the resource Id of the parent disk and location that is same as the location of parent disk.
New-AzDisk	Creates a disk using disk configuration, disk name, and resource group name passed as parameters.

Next steps

[Create a virtual machine from a managed disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Export/Copy managed snapshots as VHD to a storage account in different region with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script exports a managed snapshot to a storage account in different region. It first generates the SAS URI of the snapshot and then uses it to copy it to a storage account in different region. Use this script to maintain backup of your managed disks in different region for disaster recovery.

If needed, install the Azure PowerShell module using the instructions found in the [Azure PowerShell guide](#), and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id of the subscription where snapshot is created
$subscriptionId = "yourSubscriptionId"

#Provide the name of your resource group where snapshot is created
$resourceGroupName ="yourResourceGroupName"

#Provide the snapshot name
$snapshotName = "yourSnapshotName"

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-us/Az.Storage/storage-dotnet-shared-access-signature-part-1
$sasExpiryDuration = "3600"

#Provide storage account name where you want to copy the snapshot.
$storageAccountName = "yourstorageaccountName"

#Name of the storage container where the downloaded snapshot will be stored
$storageContainerName = "yourstoragecontainername"

#Provide the key of the storage account where you want to copy snapshot.
$storageAccountKey = 'yourStorageAccountKey'

#Provide the name of the VHD file to which snapshot will be copied.
$destinationVHDFilename = "yourvhdfilename"

# Set the context to the subscription Id where Snapshot is created
Select-AzSubscription -SubscriptionId $SubscriptionId

#Generate the SAS for the snapshot
$sas = Grant-AzSnapshotAccess -ResourceGroupName $ResourceGroupName -SnapshotName $SnapshotName -DurationInSecond $sasExpiryDuration -Access Read
#Create the context for the storage account which will be used to copy snapshot to the storage account
$destinationContext = New-AzStorageContext -StorageAccountName $storageAccountName -StorageAccountKey $storageAccountKey

#Copy the snapshot to the storage account
Start-AzStorageBlobCopy -AbsoluteUri $sas.AccessSAS -DestContainer $storageContainerName -DestContext $destinationContext -DestBlob $destinationVHDFilename
```

Script explanation

This script uses following commands to generate SAS URI for a managed snapshot and copies the snapshot to a storage account using SAS URI. Each command in the table links to command specific documentation.

COMMAND	NOTES
Grant-AzSnapshotAccess	Generates SAS URI for a snapshot that is used to copy it to a storage account.
New-AzureStorageContext	Creates a storage account context using the account name and key. This context can be used to perform read/write operations on the storage account.
Start-AzureStorageBlobCopy	Copies the underlying VHD of a snapshot to a storage account

Next steps

[Create a managed disk from a VHD](#)

[Create a virtual machine from a managed disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Export/Copy the VHD of a managed disk to a storage account in different region with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script exports the VHD of a managed disk to a storage account in different region. It first generates the SAS URI of the managed disk and then uses it to copy the underlying VHD to a storage account in different region. Use this script to copy managed disks to another region for regional expansion.

If needed, install the Azure PowerShell module using the instructions found in the [Azure PowerShell guide](#), and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id of the subscription where managed disk is created
$subscriptionId = "yourSubscriptionId"

#Provide the name of your resource group where managed is created
$resourceGroupName ="yourResourceGroupName"

#Provide the managed disk name
$diskName = "yourDiskName"

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-us/Az.Storage/storage-dotnet-shared-access-signature-part-1
$sasExpiryDuration = "3600"

#Provide storage account name where you want to copy the underlying VHD of the managed disk.
$storageAccountName = "yourstorageaccountName"

#Name of the storage container where the downloaded VHD will be stored
$storageContainerName = "yourstoragecontainername"

#Provide the key of the storage account where you want to copy the VHD of the managed disk.
$storageAccountKey = 'yourStorageAccountKey'

#Provide the name of the destination VHD file to which the VHD of the managed disk will be copied.
$destinationVHDFilename = "yourvhdfilename"

#Set the value to 1 to use AzCopy tool to download the data. This is the recommended option for faster copy.
#Download AzCopy v10 from the link here: https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10
#Ensure that AzCopy is downloaded in the same folder as this file
#If you set the value to 0 then Start-AzStorageBlobCopy will be used. Azure storage will asynchronously copy
the data.
$useAzCopy = 1

# Set the context to the subscription Id where managed disk is created
Select-AzSubscription -SubscriptionId $SubscriptionId

#Generate the SAS for the managed disk
$sas = Grant-AzDiskAccess -ResourceGroupName $ResourceGroupName -DiskName $diskName -DurationInSecond
$sasExpiryDuration -Access Read

#Create the context of the storage account where the underlying VHD of the managed disk will be copied
$destinationContext = New-AzStorageContext -StorageAccountName $storageAccountName -StorageAccountKey
$storageAccountKey

#Copy the VHD of the managed disk to the storage account
if($useAzCopy -eq 1)
{
    $containerSASURI = New-AzStorageContainerSASToken -Context $destinationContext -ExpiryTime(get-date).AddSeconds($sasExpiryDuration) -FullUri -Name $storageContainerName -Permission rw
    $containername,$sastokenkey = $containerSASURI -split "\?"
    $containerSASURI = "$containername/$destinationVHDFilename`?$sastokenkey"
    azcopy copy $sas.AccessSAS $containerSASURI
}

}else{

    Start-AzStorageBlobCopy -AbsoluteUri $sas.AccessSAS -DestContainer $storageContainerName -DestContext
$destinationContext -DestBlob $destinationVHDFilename
}

```

Script explanation

This script uses the following commands to generate SAS URI of a managed disk and copies the underlying VHD to a storage account using the SAS URI. Each command in the table links to the command specific documentation.

COMMAND	NOTES
Grant-AzDiskAccess	Generates SAS URI for a managed disk that is used to copy the underlying VHD to a storage account.
New-AzureStorageContext	Creates a storage account context using the account name and key. This context can be used to perform read/write operations on the storage account.
Start-AzureStorageBlobCopy	Copies the underlying VHD of a snapshot to a storage account

Next steps

[Create a managed disk from a VHD](#)

[Create a virtual machine from a managed disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create a snapshot from a VHD to create multiple identical managed disks in small amount of time with PowerShell

1/2/2020 • 2 minutes to read • [Edit Online](#)

This script creates a snapshot from a VHD file in a storage account in same or different subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to a snapshot and then use the snapshot to create multiple identical managed disks in small amount of time. Also, use it to import a data VHD to a snapshot and then use the snapshot to create multiple managed disks in small amount of time.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id where snapshot will be created
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group where snapshot will be created.
$resourceGroupName = 'yourResourceGroupName'

#Provide the name of the snapshot
$snapshotName = 'yourSnapshotName'

#Provide the storage type for snapshot. PremiumLRS or StandardLRS.
$storageType = 'StandardLRS'

#Provide the Azure region (e.g. westus) where snapshot will be located.
#This location should be same as the storage account location where VHD file is stored
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Provide the URI of the VHD file (page blob) in a storage account. Please note that this is NOT the SAS URI of
the storage container where VHD file is stored.
#e.g. https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
#Note: VHD file can be deleted as soon as Managed Disk is created.
$sourceVHDURI = 'https://yourStorageAccountName.blob.core.windows.net/vhds/yourVHDName.vhd'

#Provide the resource Id of the storage account where VHD file is stored.
#e.g. /subscriptions/6582b1g7-e212-446b-b509-
314e17e1efb0/resourceGroups/MDDemo/providers/Microsoft.Storage/storageAccounts/contosostorageaccount1
#This is an optional parameter if you are creating snapshot in the same subscription
$storageAccountId =
'/subscriptions/yourSubscriptionId/resourceGroups/yourResourceGroupName/providers/Microsoft.Storage/storageAcc
ounts/yourStorageAccountName'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshotConfig = New-AzSnapshotConfig -AccountType $storageType -Location $location -CreateOption Import -
StorageAccountId $storageAccountId -SourceUri $sourceVHDURI

New-AzSnapshot -Snapshot $snapshotConfig -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName
```

Next steps

[Create a managed disk from snapshot](#)

[Create a virtual machine by attaching a managed disk as OS disk](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Copy snapshot of a managed disk in same subscription or different subscription with PowerShell

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script copies a snapshot of a managed disk to same or different subscription. Use this script for the following scenarios:

1. Migrate a snapshot in Premium storage (Premium_LRS) to Standard storage (Standard_LRS or Standard_ZRS) to reduce your cost.
2. Migrate a snapshot from locally redundant storage (Premium_LRS, Standard_LRS) to zone redundant storage (Standard_ZRS) to benefit from the higher reliability of ZRS storage.
3. Move a snapshot to different subscription in the same region for longer retention.

If needed, install the Azure PowerShell module using the instructions found in the [Azure PowerShell guide](#), and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id of the subscription where snapshot exists
$sourceSubscriptionId='yourSourceSubscriptionId'

#Provide the name of your resource group where snapshot exists
$sourceResourceGroupName='yourResourceGroupName'

#Provide the name of the snapshot
$snapshotName='yourSnapshotName'

#Set the context to the subscription Id where snapshot exists
Select-AzSubscription -SubscriptionId $sourceSubscriptionId

#Get the source snapshot
$snapshot= Get-AzSnapshot -ResourceGroupName $sourceResourceGroupName -Name $snapshotName

#Provide the subscription Id of the subscription where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
$targetSubscriptionId='yourTargetSubscriptionId'

#Name of the resource group where snapshot will be copied to
$targetResourceGroupName='yourTargetResourceGroupName'

#Set the context to the subscription Id where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
Select-AzSubscription -SubscriptionId $targetSubscriptionId

#We recommend you to store your snapshots in Standard storage to reduce cost. Please use Standard_ZRS in
regions where zone redundant storage (ZRS) is available, otherwise use Standard_LRS
#Please check out the availability of ZRS here: https://docs.microsoft.com/en-us/Az.Storage/common/storage-redundancy-zrs#support-coverage-and-regional-availability
$snapshotConfig = New-AzSnapshotConfig -SourceResourceId $snapshot.Id -Location $snapshot.Location -
CreateOption Copy -SkuName Standard_LRS

#Create a new snapshot in the target subscription and resource group
New-AzSnapshot -Snapshot $snapshotConfig -SnapshotName $snapshotName -ResourceGroupName
$targetResourceGroupName

```

Script explanation

This script uses following commands to create a snapshot in the target subscription using the Id of the source snapshot. Each command in the table links to command specific documentation.

COMMAND	NOTES
New-AzSnapshotConfig	Creates snapshot configuration that is used for snapshot creation. It includes the resource Id of the parent snapshot and location that is same as the parent snapshot.
New-AzSnapshot	Creates a snapshot using snapshot configuration, snapshot name, and resource group name passed as parameters.

Next steps

[Create a virtual machine from a snapshot](#)

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Encrypt a Windows virtual machine with Azure PowerShell

11/13/2019 • 3 minutes to read • [Edit Online](#)

This script creates a secure Azure Key Vault, encryption keys, Azure Active Directory service principal, and a Windows virtual machine (VM). The VM is then encrypted using the encryption key from Key Vault and service principal credentials.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# Edit these global variables with your unique Key Vault name, resource group name and location
#Name of the Key Vault
$keyVaultName = "myKeyVault00"
#Resource Group Name
$rgName = "myResourceGroup"
#Region
$location = "East US"
#Password to place w/in the KeyVault
$password = $([guid]::NewGuid()).Guid
$securePassword = ConvertTo-SecureString -String $password -AsPlainText -Force
#Name for the Azure AD Application
$appName = "My App"
#Name for the VM to be encrypted
$vmName = "myEncryptedVM"
#User name for the admin account in the VM being created and then encrypted
$vmAdminName = "encryptedUser"

# Register the Key Vault provider and create a resource group
New-AzResourceGroup -Location $location -Name $rgName

# Create a Key Vault and enable it for disk encryption
New-AzKeyVault `-
    -Location $location `-
    -ResourceGroupName $rgName `-
    -VaultName $keyVaultName `-
    -EnabledForDiskEncryption

# Create a key in your Key Vault
Add-AzKeyVaultKey `-
    -VaultName $keyVaultName `-
    -Name "myKey" `-
    -Destination "Software"

# Put the password in the Key Vault as a Key Vault Secret so we can use it later
# We should never put passwords in scripts.
Set-AzKeyVaultSecret -VaultName $keyVaultName -Name adminCreds -SecretValue $securePassword
Set-AzKeyVaultSecret -VaultName $keyVaultName -Name protectValue -SecretValue $securePassword

# Create Azure Active Directory app and service principal
$app = New-AzADApplication -DisplayName $appName `-
    -HomePage "https://myapp0.contoso.com" `-
    -IdentifierUris "https://contoso.com/myapp0" `-
    -Password (Get-AzKeyVaultSecret -VaultName $keyVaultName -Name adminCreds).SecretValue

New-AzADServicePrincipal -ApplicationId $app.ApplicationId
```

```

# Set permissions to allow your AAD service principal to read keys from Key Vault
Set-AzKeyVaultAccessPolicy -VaultName $keyvaultName `
    -ServicePrincipalName $app.ApplicationId `
    -PermissionsToKeys decrypt,encrypt,unwrapKey,wrapKey,verify,sign,get,list,update `
    -PermissionsToSecrets get,list,set,delete,backup,restore,recover,purge

# Create PSCredential object for VM
$cred = New-Object System.Management.Automation.PSCredential($vmAdminName, (Get-AzKeyVaultSecret -VaultName $keyVaultName -Name adminCreds).SecretValue)

# Create a virtual machine
New-AzVM `
    -ResourceGroupName $rgName `
    -Name $vmName `
    -Location $location `
    -ImageName "Win2016Datacenter" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIp" `
    -Credential $cred `
    -OpenPorts 3389

# Define required information for our Key Vault and keys
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name "myKey").Key.kid;

# Encrypt our virtual machine
Set-AzVmDiskEncryptionExtension `
    -ResourceGroupName $rgName `
    -VMName $vmName `
    -AadClientID $app.ApplicationId `
    -AadClientSecret (Get-AzKeyVaultSecret -VaultName $keyVaultName -Name adminCreds).SecretValueText `
    -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl `
    -DiskEncryptionKeyId $keyVaultResourceId `
    -KeyEncryptionKeyUrl $keyEncryptionKeyUrl `
    -KeyEncryptionKeyId $keyVaultResourceId

# View encryption status
Get-AzVmDiskEncryptionStatus -ResourceGroupName $rgName -VMName $vmName

<#
#clean up
Remove-AzResourceGroup -Name $rgName
#removes all of the Azure AD Applications you created w/ the same name
Remove-AzADApplication -ObjectId $app.ObjectId -Force
#>

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzKeyVault	Creates an Azure Key Vault to store secure data such as encryption keys.
Add-AzKeyVaultKey	Creates an encryption key in Key Vault.
New-AzADServicePrincipal	Creates an Azure Active Directory service principal to securely authenticate and control access to encryption keys.
Set-AzKeyVaultAccessPolicy	Sets permissions on the Key Vault to grant the service principal access to encryption keys.
New-AzVM	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also opens port 80 and sets the administrative credentials.
Get-AzKeyVault	Gets required information on the Key Vault
Set-AzVMDiskEncryptionExtension	Enables encryption on a VM using the service principal credentials and encryption key.
Get-AzVmDiskEncryptionStatus	Shows the status of the VM encryption process.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Create an Azure Monitor VM with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine, installs the Log Analytics agent, and enrolls the system with a Log Analytics workspace. Once the script has run, the virtual machine will be visible in Azure Monitor. Also, you need to update the Log Analytics workspace ID and workspace key.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
# OMS ID and OMS key
$omsId = "<Replace with your OMS ID>"
$omsKey = "<Replace with your OMS key>

# Variables for common values
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

# Create a user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create a virtual machine
New-AzVM ` 
    -ResourceGroupName $resourceGroup ` 
    -Name $vmName ` 
    -Location $location ` 
    -ImageName "Win2016Datacenter" ` 
    -VirtualNetworkName "myVnet" ` 
    -SubnetName "mySubnet" ` 
    -SecurityGroupName "myNetworkSecurityGroup" ` 
    -PublicIpAddressName "myPublicIp" ` 
    -Credential $cred ` 
    -OpenPorts 3389

# Install and configure the OMS agent
$publicSettings = New-Object psobject | Add-Member -PassThru NoteProperty workspaceId $omsId | ConvertTo-Json
$protectedSettings = New-Object psobject | Add-Member -PassThru NoteProperty workspaceKey $omsKey | ConvertTo-Json

Set-AzVMExtension -ExtensionName "OMS" -ResourceGroupName $resourceGroup -VMName $vmName ` 
    -Publisher "Microsoft.EnterpriseCloud.Monitoring" -ExtensionType "MicrosoftMonitoringAgent" ` 
    -TypeHandlerVersion 1.0 -SettingString $publicSettings -ProtectedSettingString $protectedSettings ` 
    -Location $location
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Script explanation

This script uses the following commands to create the deployment. Each item in the table links to command specific documentation.

COMMAND	NOTES
New-AzResourceGroup	Creates a resource group in which all resources are stored.
New-AzVM	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also opens port 80 and sets the administrative credentials.
Set-AzVMExtension	Add a VM extension to the virtual machine.
Remove-AzResourceGroup	Removes a resource group and all resources contained within.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Collect details about all VMs in a subscription with PowerShell

12/6/2019 • 2 minutes to read • [Edit Online](#)

This script creates a csv that contains the VM Name, Resource Group Name, Region, Virtual Network, Subnet, Private IP Address, OS Type, and Public IP Address of the VMs in the provided subscription.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

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.

Sample script

```
#Provide the subscription Id where the VMs reside
$subscriptionId = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

#Provide the name of the csv file to be exported
$reportName = "myReport.csv"

Select-AzSubscription $subscriptionId
$report = @()
$vms = Get-AzVM
$publicIps = Get-AzPublicIpAddress
$nics = Get-AzNetworkInterface | ?{ $_.VirtualMachine -NE $null}
foreach ($nic in $nics) {
    $info = "" | Select VmName, ResourceGroupName, Region, VirtualNetwork, Subnet, PrivateIpAddress, OsType, PublicIPAddress
    $vm = $vms | ? -Property Id -eq $nic.VirtualMachine.id
    foreach($publicIp in $publicIps) {
        if($nic.IpConfigurations.id -eq $publicIp.ipconfiguration.Id) {
            $info.PublicIPAddress = $publicIp.ipaddress
        }
    }
    $info.OsType = $vm.StorageProfile.OsDisk.OsType
    $info.VMName = $vm.Name
    $info.ResourceGroupName = $vm.ResourceGroupName
    $info.Region = $vm.Location
    $info.VirtualNetwork = $nic.IpConfigurations.subnet.Id.Split("/")[-3]
    $info.Subnet = $nic.IpConfigurations.subnet.Id.Split("/")[-1]
    $info.PrivateIpAddress = $nic.IpConfigurations.PrivateIpAddress
    $report+=$info
}
$report | ft VmName, ResourceGroupName, Region, VirtualNetwork, Subnet, PrivateIpAddress, OsType, PublicIPAddress
$report | Export-Csv "$home/$reportName"
```

Script explanation

This script uses following commands to create a csv export of the details of VMs in a subscription. Each command in the table links to command specific documentation.

COMMAND	NOTES
Select-AzSubscription	Sets the tenant, subscription, and environment for cmdlets to use in the current session.
Get-AzVM	Gets the properties of a virtual machine.
Get-AzPublicIpAddress	Gets a public IP address.
Get-AzNetworkInterface	Gets a network interface.

Next steps

For more information on the Azure PowerShell module, see [Azure PowerShell documentation](#).

Additional virtual machine PowerShell script samples can be found in the [Azure Windows VM documentation](#).

Azure CLI Samples for Windows virtual machines

11/13/2019 • 2 minutes to read • [Edit Online](#)

The following table includes links to bash scripts built using the Azure CLI that deploy Windows virtual machines.

Create virtual machines	
Create a virtual machine	Creates a Windows virtual machine with minimal configuration.
Create a fully configured virtual machine	Creates a resource group, virtual machine, and all related resources.
Create highly available virtual machines	Creates several virtual machines in a highly available and load balanced configuration.
Create a VM and run configuration script	Creates a virtual machine and uses the Azure Custom Script extension to install IIS.
Create a VM and run DSC configuration	Creates a virtual machine and uses the Azure Desired State Configuration (DSC) extension to install IIS.
Manage storage	
Create managed disk from a VHD	Creates a managed disk from a specialized VHD as an OS disk or from a data VHD as data disk.
Create a managed disk from a snapshot	Creates a managed disk from a snapshot.
Copy managed disk to same or different subscription	Copies managed disk to same or different subscription but in the same region as the parent managed disk.
Export a snapshot as VHD to a storage account	Exports a managed snapshot as VHD to a storage account in different region.
Export the VHD of a managed disk to a storage account	Exports the underlying VHD of a managed disk to a storage account in different region.
Copy snapshot to same or different subscription	Copies snapshot to same or different subscription but in the same region as the parent snapshot.
Network virtual machines	
Secure network traffic between virtual machines	Creates two virtual machines, all related resources, and an internal and external network security groups (NSG).
Secure virtual machines	

Encrypt a VM and data disks	Creates an Azure Key Vault, encryption key, and service principal, then encrypts a VM.
Monitor virtual machines	
Monitor a VM with Azure Monitor	Creates a virtual machine, installs the Log Analytics agent, and enrolls the VM in a Log Analytics workspace.

Quick Create a virtual machine with the Azure CLI

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016. After running the script, you can access the virtual machine through a Remote Desktop connection.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westus

# Create a virtual machine.
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password $AdminPassword \
    --no-wait
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also specifies the virtual machine image to be used and administrative credentials.

COMMAND	NOTES
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Create a virtual machine with the Azure CLI

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016. After running the script, you can access the virtual machine through a Remote Desktop connection.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westeurope

# Create a virtual network.
az network vnet create --resource-group myResourceGroup --name myVnet --subnet-name mySubnet

# Create a public IP address.
az network public-ip create --resource-group myResourceGroup --name myPublicIP

# Create a network security group.
az network nsg create --resource-group myResourceGroup --name myNetworkSecurityGroup

# Create a virtual network card and associate with public IP address and NSG.
az network nic create \
    --resource-group myResourceGroup \
    --name myNic \
    --vnet-name myVnet \
    --subnet mySubnet \
    --network-security-group myNetworkSecurityGroup \
    --public-ip-address myPublicIP

# Create a virtual machine.
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --location westeurope \
    --nics myNic \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password $AdminPassword

# Open port 3389 to allow RDP traffic to host.
az vm open-port --port 3389 --resource-group myResourceGroup --name myVM
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az network vnet create	Creates an Azure virtual network and subnet.
az network public-ip create	Creates a public IP address with a static IP address and an associated DNS name.
az network nsg create	Creates a network security group (NSG), which is a security boundary between the internet and the virtual machine.
az network nic create	Creates a virtual network card and attaches it to the virtual network, subnet, and NSG.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used, and administrative credentials.
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Load balance traffic between highly available virtual machines

11/13/2019 • 4 minutes to read • [Edit Online](#)

This script sample creates everything needed to run several Ubuntu virtual machines configured in a highly available and load balanced configuration. After running the script, you will have three virtual machines, joined to an Azure Availability Set, and accessible through an Azure Load Balancer.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westus

# Create a virtual network.
az network vnet create --resource-group myResourceGroup --name myVnet \
--address-prefix 192.168.0.0/16 --subnet-name mySubnet --subnet-prefix 192.168.1.0/24

# Create a public IP address.
az network public-ip create --resource-group myResourceGroup --name myPublicIP

# Create an Azure Load Balancer.
az network lb create --resource-group myResourceGroup --name myLoadBalancer --public-ip-address myPublicIP \
--frontend-ip-name myFrontEndPool --backend-pool-name myBackEndPool

# Creates an LB probe on port 80.
az network lb probe create --resource-group myResourceGroup --lb-name myLoadBalancer \
--name myHealthProbe --protocol tcp --port 80

# Creates an LB rule for port 80.
az network lb rule create --resource-group myResourceGroup --lb-name myLoadBalancer --name
myLoadBalancerRuleWeb \
--protocol tcp --frontend-port 80 --backend-port 80 --frontend-ip-name myFrontEndPool \
--backend-pool-name myBackEndPool --probe-name myHealthProbe

# Create three NAT rules for port 3389.
for i in `seq 1 3`; do
    az network lb inbound-nat-rule create \
        --resource-group myResourceGroup --lb-name myLoadBalancer \
        --name myLoadBalancerRuleSSH$i --protocol tcp \
        --frontend-port 422$i --backend-port 3389 \
        --frontend-ip-name myFrontEndPool
done

# Create a network security group
az network nsg create --resource-group myResourceGroup --name myNetworkSecurityGroup
```

```

# Create a network security group rule for port 3389.
az network nsg rule create --resource-group myResourceGroup --nsg-name myNetworkSecurityGroup --name myNetworkSecurityGroupRuleSSH \
    --protocol tcp --direction inbound --source-address-prefix '*' --source-port-range '*' \
    --destination-address-prefix '*' --destination-port-range 3389 --access allow --priority 1000

# Create a network security group rule for port 80.
az network nsg rule create --resource-group myResourceGroup --nsg-name myNetworkSecurityGroup --name myNetworkSecurityGroupRuleHTTP \
    --protocol tcp --direction inbound --priority 1001 --source-address-prefix '*' --source-port-range '*' \
    --destination-address-prefix '*' --destination-port-range 80 --access allow --priority 2000

# Create three virtual network cards and associate with public IP address and NSG.
for i in `seq 1 3`; do
    az network nic create \
        --resource-group myResourceGroup --name myNic$i \
        --vnet-name myVnet --subnet mySubnet \
        --network-security-group myNetworkSecurityGroup --lb-name myLoadBalancer \
        --lb-address-pools myBackEndPool --lb-inbound-nat-rules myLoadBalancerRuleSSH$i
done

# Create an availability set.
az vm availability-set create --resource-group myResourceGroup --name myAvailabilitySet --platform-fault-domain-count 3 --platform-update-domain-count 3

# Create three virtual machines.
for i in `seq 1 3`; do
    az vm create \
        --resource-group myResourceGroup \
        --name myVM$i \
        --availability-set myAvailabilitySet \
        --nics myNic$i \
        --image win2016datacenter \
        --admin-password $AdminPassword \
        --admin-username azureuser \
        --no-wait
done

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, availability set, load balancer, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az network vnet create	Creates an Azure virtual network and subnet.
az network public-ip create	Creates a public IP address with a static IP address and an associated DNS name.
az network lb create	Creates an Azure Network Load Balancer (NLB).

COMMAND	NOTES
<code>az network lb probe create</code>	Creates an NLB probe. An NLB probe is used to monitor each VM in the NLB set. If any VM becomes inaccessible, traffic is not routed to the VM.
<code>az network lb rule create</code>	Creates an NLB rule. In this sample, a rule is created for port 80. As HTTP traffic arrives at the NLB, it is routed to port 80 one of the VMs in the NLB set.
<code>az network lb inbound-nat-rule create</code>	Creates an NLB Network Address Translation (NAT) rule. NAT rules map a port of the NLB to a port on a VM. In this sample, a NAT rule is created for SSH traffic to each VM in the NLB set.
<code>az network nsg create</code>	Creates a network security group (NSG), which is a security boundary between the internet and the virtual machine.
<code>az network nsg rule create</code>	Creates an NSG rule to allow inbound traffic. In this sample, port 22 is opened for SSH traffic.
<code>az network nic create</code>	Creates a virtual network card and attaches it to the virtual network, subnet, and NSG.
<code>az vm availability-set create</code>	Creates an availability set. Availability sets ensure application uptime by spreading the virtual machines across physical resources such that if failure occurs, the entire set is not effected.
<code>az vm create</code>	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used and administrative credentials.
<code>az group delete</code>	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Quick Create a virtual machine with the Azure CLI

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine running Windows Server 2016, and uses the Azure Virtual Machine Custom Script Extension to install IIS. After running the script, you can access the default IIS website on the public IP address of the virtual machine.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westeurope

# Create a virtual machine.
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password $AdminPassword

# Open port 80 to allow web traffic to host.
az vm open-port --port 80 --resource-group myResourceGroup --name myVM

# Use CustomScript extension to install IIS.
az vm extension set \
    --publisher Microsoft.Compute \
    --version 1.8 \
    --name CustomScriptExtension \
    --vm-name myVM \
    --resource-group myResourceGroup \
    --settings '{"commandToExecute":"powershell.exe Install-WindowsFeature -Name Web-Server"}'
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources.

Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and network security group. This command also specifies the virtual machine image to be used and administrative credentials.
az vm open-port	Creates a network security group rule to allow inbound traffic. In this sample, port 80 is opened for HTTP traffic.
azure vm extension set	Adds and runs a virtual machine extension to a VM. In this sample, the custom script extension is used to install IIS.
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Create a VM with IIS using DSC

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates a virtual machine, and uses the Azure Virtual Machine DSC custom script extension to install and configure IIS.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westus

# Create a VM
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password $AdminPassword

# Start a CustomScript extension to use a simple bash script to update, download and install WordPress and MySQL
az vm extension set \
    --name DSC \
    --publisher Microsoft.Powershell \
    --version 2.19 \
    --vm-name myVM \
    --resource-group myResourceGroup \
    --settings '{"ModulesURL": "https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip", "configurationFunction": "ContosoWebsite.ps1\\ContosoWebsite", "Properties": {"MachineName": "myVM"} }'

# open port 80 to allow web traffic to host
az vm open-port \
    --port 80 \
    --resource-group myResourceGroup \
    --name myVM
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used, and administrative credentials.
az vm extension set	Add the Custom Script Extension to the virtual machine which invokes a script to install IIS.
az vm open-port	Creates a network security group rule to allow inbound traffic. In this sample, port 80 is opened for HTTP traffic.
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Create a managed disk from a VHD file in a storage account in the same subscription with CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a managed disk from a VHD file in a storage account in the same subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to managed OS disk to create a virtual machine. Or, use it to import a data VHD to managed data disk.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id
subscriptionId=mySubscriptionId

#Provide the name of your resource group.
#Ensure that resource group is already created
resourceGroupName=myResourceGroupName

#Provide the name of the Managed Disk
diskName=myDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file size.
diskSize=128

#Provide the URI of the VHD file that will be used to create Managed Disk.
# VHD file can be deleted as soon as Managed Disk is created.
# e.g. https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
vhdUri=https://contosostorageaccount1.blob.core.windows.net/vhds/contosoumd78620170425131836.vhd

#Provide the storage type for the Managed Disk. Premium_LRS or Standard_LRS.
storageType=Premium_LRS

#Provide the Azure location (e.g. westus) where Managed Disk will be located.
#The location should be same as the location of the storage account where VHD file is stored.
#Get all the Azure location supported for your subscription using command below:
#az account list-locations
location=westus

#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Create the Managed disk from the VHD file
az disk create --resource-group $resourceGroupName --name $diskName --sku $storageType --location $location --size-gb $diskSize --source $vhdUri
```

Script explanation

This script uses following commands to create a managed disk from a VHD. Each command in the table links to command specific documentation.

COMMAND	NOTES
az disk create	Creates a managed disk using URI of a VHD in a storage account in the same subscription

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Create a managed disk from a snapshot with CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script creates a managed disk from a snapshot. Use it to restore a virtual machine from snapshots of OS and data disks. Create OS and data managed disks from respective snapshots and then create a new virtual machine by attaching managed disks. You can also restore data disks of an existing VM by attaching data disks created from snapshots.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id of the subscription where you want to create Managed Disks
subscriptionId=dd80b94e-0463-4a65-8d04-c94f403879dc

#Provide the name of your resource group
resourceGroupName=myResourceGroupName

#Provide the name of the snapshot that will be used to create Managed Disks
snapshotName=mySnapshotName

#Provide the name of the new Managed Disks that will be create
diskName=myDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file size.
diskSize=128

#Provide the storage type for Managed Disk. Premium_LRS or Standard_LRS.
storageType=Premium_LRS

#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group $resourceGroupName --query [id] -o tsv)

#Create a new Managed Disks using the snapshot Id
#Note that managed disk will be created in the same location as the snapshot
az disk create --resource-group $resourceGroupName --name $diskName --sku $storageType --size-gb $diskSize --
source $snapshotId
```

Script explanation

This script uses following commands to create a managed disk from a snapshot. Each command in the table links to command specific documentation.

COMMAND	NOTES
<code>az snapshot show</code>	Gets all the properties of a snapshot using the name and resource group properties of the snapshot. Id property is used to create managed disk.
<code>az disk create</code>	Creates a managed disk using snapshot Id of a managed snapshot

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Copy managed disks to same or different subscription with CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script copies a managed disk to same or different subscription but in the same region. The copy works only when the subscriptions are part of same AAD tenant.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id of the subscription where managed disk exists
sourceSubscriptionId=dd80b94e-0463-4a65-8d04-c94f403879dc

#Provide the name of your resource group where managed disk exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the managed disk
managedDiskName=myDiskName

#Set the context to the subscription Id where managed disk exists
az account set --subscription $sourceSubscriptionId

#Get the managed disk Id
managedDiskId=$(az disk show --name $managedDiskName --resource-group $sourceResourceGroupName --query [id] -o tsv)

#If managedDiskId is blank then it means that managed disk does not exist.
echo 'source managed disk Id is: ' $managedDiskId

#Provide the subscription Id of the subscription where managed disk will be copied to
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where managed disk will be copied to
targetResourceGroupName=mytargetResourceGroupName

#Set the context to the subscription Id where managed disk will be copied to
az account set --subscription $targetSubscriptionId

#Copy managed disk to different subscription using managed disk Id
az disk create --resource-group $targetResourceGroupName --name $managedDiskName --source $managedDiskId
```

Script explanation

This script uses following commands to create a new managed disk in the target subscription using the Id of the source managed disk. Each command in the table links to command specific documentation.

COMMAND	NOTES
az disk show	Gets all the properties of a managed disk using the name and resource group properties of the managed disk. Id property is used to copy the managed disk to different subscription.
az disk create	Copies a managed disk by creating a new managed disk in different subscription using Id and name the parent managed disk.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Export/Copy a snapshot to a storage account in different region with CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script exports a managed snapshot to a storage account in different region. It first generates the SAS URI of the snapshot and then uses it to copy it to a storage account in different region. Use this script to maintain backup of your managed disks in different region for disaster recovery.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id where snapshot is created
subscriptionId=dd80b94e-0463-4a65-8d04-c94f403879dc

#Provide the name of your resource group where snapshot is created
resourceGroupName=myResourceGroupName

#Provide the snapshot name
snapshotName=mySnapshotName

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-shared-access-signature-part-1
sasExpiryDuration=3600

#Provide storage account name where you want to copy the snapshot.
storageAccountName=mystorageaccountname

#Name of the storage container where the downloaded snapshot will be stored
storageContainerName=mystoragecontainername

#Provide the key of the storage account where you want to copy snapshot.
storageAccountKey=mystorageaccountkey

#Provide the name of the VHD file to which snapshot will be copied.
destinationVHDFileName=myvhdfilename

az account set --subscription $subscriptionId

sas=$(az snapshot grant-access --resource-group $resourceGroupName --name $snapshotName --duration-in-seconds $sasExpiryDuration --query [accessSas] -o tsv)

az storage blob copy start --destination-blob $destinationVHDFileName --destination-container $storageContainerName --account-name $storageAccountName --account-key $storageAccountKey --source-uri $sas
```

Script explanation

This script uses following commands to generate SAS URI for a managed snapshot and copies the snapshot to a storage account using SAS URI. Each command in the table links to command specific documentation.

COMMAND	NOTES
az snapshot grant-access	Generates read-only SAS that is used to copy underlying VHD file to a storage account or download it to on-premises
az storage blob copy start	Copies a blob asynchronously from one storage account to another

Next steps

[Create a managed disk from a VHD](#)

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Export/Copy a managed disk to a storage account using the Azure CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script exports the underlying VHD of a managed disk to a storage account in same or different region. It first generates the SAS URI of the managed disk and then uses it to copy the VHD to a storage account. Use this script to copy your managed disks for regional expansion.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#Provide the subscription Id where managed disk is created
subscriptionId=yourSubscriptionId

#Provide the name of your resource group where managed disk is created
resourceGroupName=myResourceGroupName

#Provide the managed disk name
diskName=myDiskName

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-shared-access-signature-part-1
sasExpiryDuration=3600

#Provide storage account name where you want to copy the underlying VHD file of the managed disk.
storageAccountName=mystorageaccountname

#Name of the storage container where the downloaded VHD will be stored
storageContainerName=mystoragecontainername

#Provide the key of the storage account where you want to copy the VHD
storageAccountKey=mystorageaccountkey

#Provide the name of the destination VHD file to which the VHD of the managed disk will be copied.
destinationVHDFilename=myvhdfilename.vhd

az account set --subscription $subscriptionId

sas=$(az disk grant-access --resource-group $resourceGroupName --name $diskName --duration-in-seconds $sasExpiryDuration --query [accessSas] -o tsv)

az storage blob copy start --destination-blob $destinationVHDFilename --destination-container $storageContainerName --account-name $storageAccountName --account-key $storageAccountKey --source-uri $sas
```

Script explanation

This script uses following commands to generate the SAS URI for a managed disk and copies the underlying VHD to a storage account using the SAS URI. Each command in the table links to command specific documentation.

COMMAND	NOTES
az disk grant-access	Generates read-only SAS that is used to copy the underlying VHD file to a storage account or download it to on-premises
az storage blob copy start	Copies a blob asynchronously from one storage account to another

Next steps

[Create a managed disk from a VHD](#)

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Copy snapshot of a managed disk to same or different subscription with CLI

12/10/2019 • 2 minutes to read • [Edit Online](#)

This script copies a snapshot of a managed disk to same or different subscription. Use this script for the following scenarios:

1. Migrate a snapshot in Premium storage (Premium_LRS) to Standard storage (Standard_LRS or Standard_ZRS) to reduce your cost.
2. Migrate a snapshot from locally redundant storage (Premium_LRS, Standard_LRS) to zone redundant storage (Standard_ZRS) to benefit from the higher reliability of ZRS storage.
3. Move a snapshot to different subscription in the same region for longer retention.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#Provide the subscription Id of the subscription where snapshot exists
sourceSubscriptionId=dd80b94e-0463-4a65-8d04-c94f403879dc

#Provide the name of your resource group where snapshot exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the snapshot
snapshotName=mySnapshotName

#Set the context to the subscription Id where snapshot exists
az account set --subscription $sourceSubscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group $sourceResourceGroupName --query [id] -o tsv)

#If snapshotId is blank then it means that snapshot does not exist.
echo 'source snapshot Id is: ' $snapshotId

#Provide the subscription Id of the subscription where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where snapshot will be copied to
targetResourceGroupName=mytargetResourceGroupName

#Set the context to the subscription Id where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
az account set --subscription $targetSubscriptionId

#Copy snapshot to different subscription using the snapshot Id
#We recommend you to store your snapshots in Standard storage to reduce cost. Please use Standard_ZRS in
regions where zone redundant storage (ZRS) is available, otherwise use Standard_LRS
#Please check out the availability of ZRS here: https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy-zrs#support-coverage-and-regional-availability
az snapshot create --resource-group $targetResourceGroupName --name $snapshotName --source $snapshotId --sku Standard_LRS

```

Script explanation

This script uses following commands to create a snapshot in the target subscription using the Id of the source snapshot. Each command in the table links to command specific documentation.

COMMAND	NOTES
az snapshot show	Gets all the properties of a snapshot using the name and resource group properties of the snapshot. Id property is used to copy the snapshot to different subscription.
az snapshot create	Copies a snapshot by creating a snapshot in different subscription using the Id and name of the parent snapshot.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine and managed disks CLI script samples can be found in the [Azure Windows VM documentation](#).

Secure network traffic between virtual machines

11/13/2019 • 3 minutes to read • [Edit Online](#)

This script creates two virtual machines and secures incoming traffic to both. One virtual machine is accessible on the internet and has a network security group (NSG) configured to allow traffic on port 3389 and port 80. The second virtual machine is not accessible on the internet, and has an NSG configured to only allow traffic from the first virtual machine.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#!/bin/bash

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# Create a resource group.
az group create --name myResourceGroup --location westeurope

# Create a virtual network and front-end subnet.
az network vnet create --resource-group myResourceGroup --name myVnet --address-prefix 10.0.0.0/16 \
--subnet-name mySubnetFrontEnd --subnet-prefix 10.0.1.0/24

# Create a back-end subnet and associate with virtual network.
az network vnet subnet create --resource-group myResourceGroup --vnet-name myVnet \
--name mySubnetBackEnd --address-prefix 10.0.2.0/24

# Create a front-end virtual machine.
az vm create --resource-group myResourceGroup --name myVMFrontEnd --image win2016datacenter \
--admin-username azureuser --admin-password $AdminPassword --vnet-name myVnet --subnet mySubnetFrontEnd \
--nsg myNetworkSecurityGroupFrontEnd --no-wait

# Create a back-end virtual machine without a public IP address.
az vm create --resource-group myResourceGroup --name myVMBackEnd --image win2016datacenter \
--admin-username azureuser --admin-password $AdminPassword --public-ip-address "" --vnet-name myVnet \
--subnet mySubnetBackEnd --nsg myNetworkSecurityGroupBackEnd

# Create front-end NSG rule to allow traffic on port 80.
az network nsg rule create --resource-group myResourceGroup --nsg-name myNetworkSecurityGroupFrontEnd \
--name http --access allow --protocol Tcp --direction Inbound --priority 200 \
--source-address-prefix "*" --source-port-range "*" --destination-address-prefix "*" --destination-port-range 80

# Get nsg rule name.
nsgrule=$(az network nsg rule list --resource-group myResourceGroup --nsg-name myNetworkSecurityGroupBackEnd -q [0].name -o tsv)

# Update back-end network security group rule to limit SSH to source prefix (priority 100).
az network nsg rule update --resource-group myResourceGroup --nsg-name myNetworkSecurityGroupBackEnd \
--name $nsgrule --protocol tcp --direction inbound --priority 100 \
--source-address-prefix 10.0.1.0/24 --source-port-range '*' --destination-address-prefix '*' \
--destination-port-range 22 --access allow

# Create backend NSG rule to block all incoming traffic (priority 200).
az network nsg rule create --resource-group myResourceGroup --nsg-name myNetworkSecurityGroupBackEnd \
--name denyAll --access Deny --protocol Tcp --direction Inbound --priority 200 \
--source-address-prefix "*" --source-port-range "*" --destination-address-prefix "*" --destination-port-range "*"

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
<code>az group create</code>	Creates a resource group in which all resources are stored.
<code>az network vnet create</code>	Creates an Azure virtual network and subnet.
<code>az network vnet subnet create</code>	Creates a subnet.
<code>az vm create</code>	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used, and administrative credentials.
<code>az network nsg rule update</code>	Updates an NSG rule. In this sample, the back-end rule is updated to pass through traffic only from the front-end subnet.
<code>az network nsg rule list</code>	Returns information about a network security group rule. In this sample, the rule name is stored in a variable for use later in the script.
<code>az group delete</code>	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Encrypt a Windows virtual machine in Azure

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates a secure Azure Key Vault, encryption keys, Azure Active Directory service principal, and a Windows virtual machine (VM). The VM is then encrypted using the encryption key from Key Vault and service principal credentials.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```

#!/bin/bash

# Provide your own unique Key Vault name
keyvault_name=<your_unique_keyvault_name>

# Register the Key Vault provider and create a resource group.
az provider register -n Microsoft.KeyVault
az group create --name myResourceGroup --location eastus

# Create a Key Vault for storing keys and enabled for disk encryption.
az keyvault create --name $keyvault_name --resource-group myResourceGroup --location eastus \
--enabled-for-disk-encryption True

# Create a key within the Key Vault.
az keyvault key create --vault-name $keyvault_name --name myKey --protection software

# Create an Azure Active Directory service principal for authenticating requests to Key Vault.
# Read in the service principal ID and password for use in later commands.
read sp_id sp_password <<< $(az ad sp create-for-rbac --query [appId,password] -o tsv)

# Grant permissions on the Key Vault to the AAD service principal.
az keyvault set-policy --name $keyvault_name \
--spn $sp_id \
--key-permissions wrapKey \
--secret-permissions set

# Create a virtual machine.
az vm create \
--resource-group myResourceGroup \
--name myVM \
--name myVM --image win2016datacenter \
--admin-username azureuser \
--admin-password myPassword12

# Encrypt the VM disks.
az vm encryption enable --resource-group myResourceGroup --name myVM \
--aad-client-id $sp_id \
--aad-client-secret $sp_password \
--disk-encryption-keyvault $keyvault_name \
--key-encryption-key myKey \
--volume-type all

# Output how to monitor the encryption status and next steps.
echo "The encryption process can take some time. View status with:

az vm encryption show --resource-group myResourceGroup --name myVM --query [osDisk] -o tsv

When encryption status shows \`Encrypted\`, restart the VM with:

az vm restart --resource-group myResourceGroup --name myVM"

```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup
```

Script explanation

This script uses the following commands to create a resource group, Azure Key Vault, service principal, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az keyvault create	Creates an Azure Key Vault to store secure data such as encryption keys.
az keyvault key create	Creates an encryption key in Key Vault.
az ad sp create-for-rbac	Creates an Azure Active Directory service principal to securely authenticate and control access to encryption keys.
az keyvault set-policy	Sets permissions on the Key Vault to grant the service principal access to encryption keys.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used, and administrative credentials.
az vm encryption enable	Enables encryption on a VM using the service principal credentials and encryption key.
az vm encryption show	Shows the status of the VM encryption process.
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Monitor a VM with Azure Monitor logs

11/13/2019 • 2 minutes to read • [Edit Online](#)

This script creates an Azure Virtual Machine, installs the Log Analytics agent, and enrolls the system with a Log Analytics workspace. Once the script has run, the virtual machine will be visible in Azure Monitoring.

To run this sample, install the latest version of the [Azure CLI](#). To start, run `az login` to create a connection with Azure.

Samples for the Azure CLI are written for the `bash` shell. To run this sample in Windows PowerShell or Command Prompt, you may need to change elements of the script.

If you don't have an [Azure subscription](#), create a [free account](#) before you begin.

Sample script

```
#!/bin/sh

# Update for your admin password
AdminPassword=ChangeYourAdminPassword1

# OMS Id and OMS key.
omsid=<Replace with your OMS Id>
omskey=<Replace with your OMS key>

# Create a resource group.
az group create --name myResourceGroup --location westeurope

# Create a virtual machine.
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password $AdminPassword

# Install and configure the OMS agent.
az vm extension set \
    --resource-group myResourceGroup \
    --vm-name myVM --name MicrosoftMonitoringAgent \
    --publisher Microsoft.EnterpriseCloud.Monitoring \
    --version 1.0 --protected-settings '{"workspaceKey": """$omskey"""}' \
    --settings '{"workspaceId": """$omsid"""}'
```

Clean up deployment

Run the following command to remove the resource group, VM, and all related resources.

```
az group delete --name myResourceGroup --yes
```

Script explanation

This script uses the following commands to create a resource group, virtual machine, and all related resources. Each command in the table links to command specific documentation.

COMMAND	NOTES
az group create	Creates a resource group in which all resources are stored.
az vm create	Creates the virtual machine and connects it to the network card, virtual network, subnet, and NSG. This command also specifies the virtual machine image to be used, and administrative credentials.
azure vm extension set	Runs a VM extension against a virtual machine.
az group delete	Deletes a resource group including all nested resources.

Next steps

For more information on the Azure CLI, see [Azure CLI documentation](#).

Additional virtual machine CLI script samples can be found in the [Azure Windows VM documentation](#).

Azure Resource Manager overview

12/23/2019 • 5 minutes to read • [Edit Online](#)

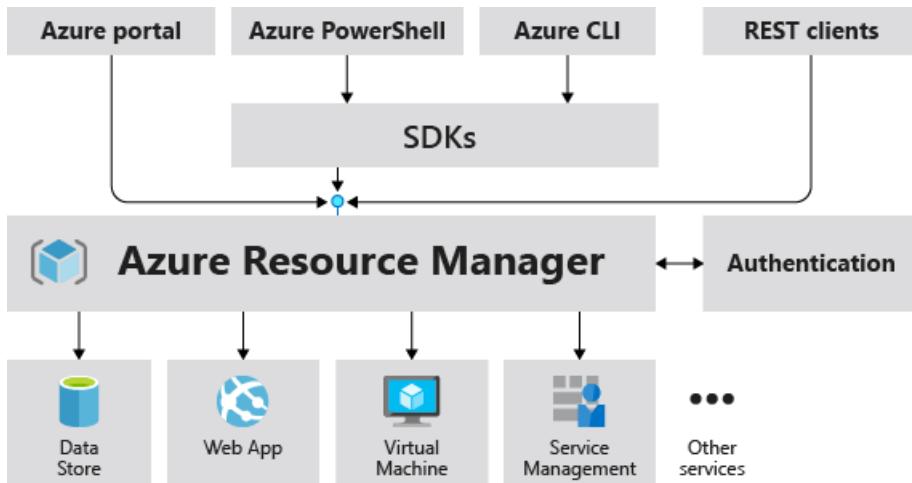
Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure subscription. You use management features, like access control, locks, and tags, to secure and organize your resources after deployment.

To learn about Azure Resource Manager templates, see [Template deployment overview](#).

Consistent management layer

When a user sends a request from any of the Azure tools, APIs, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.

The following image shows the role Azure Resource Manager plays in handling Azure requests.



All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs, and client SDKs. Functionality initially released through APIs will be represented in the portal within 180 days of initial release.

Terminology

If you're new to Azure Resource Manager, there are some terms you might not be familiar with.

- **resource** - A manageable item that is available through Azure. Virtual machines, storage accounts, web apps, databases, and virtual networks are examples of resources.
- **resource group** - A container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for your organization. See [Resource groups](#).
- **resource provider** - A service that supplies Azure resources. For example, a common resource provider is Microsoft.Compute, which supplies the virtual machine resource. Microsoft.Storage is another common resource provider. See [Resource providers and types](#).
- **Resource Manager template** - A JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group or subscription. The template can be used to deploy the resources consistently and repeatedly. See [Template deployment overview](#).

- **declarative syntax** - Syntax that lets you state "Here is what I intend to create" without having to write the sequence of programming commands to create it. The Resource Manager template is an example of declarative syntax. In the file, you define the properties for the infrastructure to deploy to Azure. See [Template deployment overview](#).

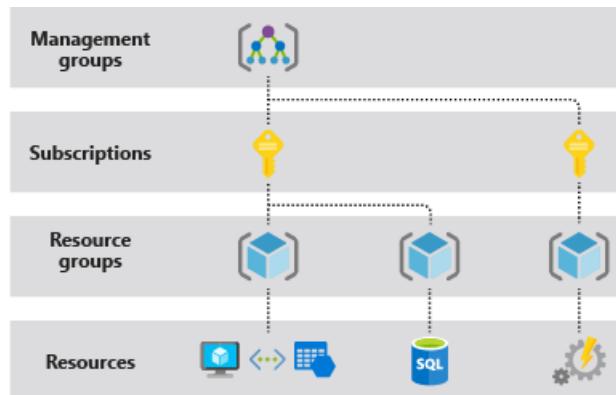
The benefits of using Resource Manager

With Resource Manager, you can:

- Manage your infrastructure through declarative templates rather than scripts.
- Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
- Redeploy your solution throughout the development lifecycle and have confidence your resources are deployed in a consistent state.
- Define the dependencies between resources so they're deployed in the correct order.
- Apply access control to all services in your resource group because Role-Based Access Control (RBAC) is natively integrated into the management platform.
- Apply tags to resources to logically organize all the resources in your subscription.
- Clarify your organization's billing by viewing costs for a group of resources sharing the same tag.

Understand scope

Azure provides four levels of scope: [management groups](#), subscriptions, [resource groups](#), and resources. The following image shows an example of these layers.



You apply management settings at any of these levels of scope. The level you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For example, when you apply a [policy](#) to the subscription, the policy is applied to all resource groups and resources in your subscription. When you apply a policy on the resource group, that policy is applied to the resource group and all its resources. However, another resource group doesn't have that policy assignment.

You can deploy templates to management groups, subscriptions, or resource groups.

Resource groups

There are some important factors to consider when defining your resource group:

- All the resources in your group should share the same lifecycle. You deploy, update, and delete them together. If one resource, such as a database server, needs to exist on a different deployment cycle it should be in another resource group.

- Each resource can only exist in one resource group.
- You can add or remove a resource to a resource group at any time.
- You can move a resource from one resource group to another group. For more information, see [Move resources to new resource group or subscription](#).
- A resource group can contain resources that are located in different regions.
- A resource group can be used to scope access control for administrative actions.
- A resource can interact with resources in other resource groups. This interaction is common when the two resources are related but don't share the same lifecycle (for example, web apps connecting to a database).

When creating a resource group, you need to provide a location for that resource group. You may be wondering, "Why does a resource group need a location? And, if the resources can have different locations than the resource group, why does the resource group location matter at all?" The resource group stores metadata about the resources. When you specify a location for the resource group, you're specifying where that metadata is stored. For compliance reasons, you may need to ensure that your data is stored in a particular region.

If the resource group's region is temporarily unavailable, you can't update resources in the resource group because the metadata is unavailable. The resources in other regions will still function as expected, but you can't update them. For more information about building reliable applications, see [Designing reliable Azure applications](#).

Resiliency of Azure Resource Manager

The Azure Resource Manager service is designed for resiliency and continuous availability. Resource Manager and control plane operations (requests sent to management.azure.com) in the REST API are:

- Distributed across regions. Some services are regional.
- Distributed across Availability Zones (as well regions) in locations that have multiple Availability Zones.
- Not dependent on a single logical data center.
- Never taken down for maintenance activities.

This resiliency applies to services that receive requests through Resource Manager. For example, Key Vault benefits from this resiliency.

Next steps

- For all the operations offered by resource providers, see the [Azure REST APIs](#).
- To learn about moving resources, see [Move resources to new resource group or subscription](#).
- To learn about tagging resources, see [Use tags to organize your Azure resources](#).
- To learn about locking resources, see [Lock resources to prevent unexpected changes](#).
- For information about creating templates for deployments, see [Template deployment overview](#).

Regions for virtual machines in Azure

1/19/2020 • 4 minutes to read • [Edit Online](#)

It is important to understand how and where your virtual machines (VMs) operate in Azure, along with your options to maximize performance, availability, and redundancy. This article provides you with an overview of the availability and redundancy features of Azure.

What are Azure regions?

Azure operates in multiple datacenters around the world. These datacenters are grouped in to geographic regions, giving you flexibility in choosing where to build your applications.

You create Azure resources in defined geographic regions like 'West US', 'North Europe', or 'Southeast Asia'. You can review the [list of regions and their locations](#). Within each region, multiple datacenters exist to provide for redundancy and availability. This approach gives you flexibility as you design applications to create VMs closest to your users and to meet any legal, compliance, or tax purposes.

Special Azure regions

Azure has some special regions that you may wish to use when building out your applications for compliance or legal purposes. These special regions include:

- **US Gov Virginia and US Gov Iowa**

- A physical and logical network-isolated instance of Azure for US government agencies and partners, operated by screened US persons. Includes additional compliance certifications such as [FedRAMP](#) and [DISA](#). Read more about [Azure Government](#).

- **China East and China North**

- These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft does not directly maintain the datacenters. See more about [Azure China 21Vianet](#).

- **Germany Central and Germany Northeast**

- These regions are available via a data trustee model whereby customer data remains in Germany under control of T-Systems, a Deutsche Telekom company, acting as the German data trustee.

Region pairs

Each Azure region is paired with another region within the same geography (such as US, Europe, or Asia). This approach allows for the replication of resources, such as VM storage, across a geography that should reduce the likelihood of natural disasters, civil unrest, power outages, or physical network outages affecting both regions at once. Additional advantages of region pairs include:

- In the event of a wider Azure outage, one region is prioritized out of every pair to help reduce the time to restore for applications.
- Planned Azure updates are rolled out to paired regions one at a time to minimize downtime and risk of application outage.
- Data continues to reside within the same geography as its pair (except for Brazil South) for tax and law enforcement jurisdiction purposes.

Examples of region pairs include:

PRIMARY	SECONDARY
West US	East US
North Europe	West Europe
Southeast Asia	East Asia

You can see the full [list of regional pairs here](#).

Feature availability

Some services or VM features are only available in certain regions, such as specific VM sizes or storage types. There are also some global Azure services that do not require you to select a particular region, such as [Azure Active Directory](#), [Traffic Manager](#), or [Azure DNS](#). To assist you in designing your application environment, you can check the [availability of Azure services across each region](#). You can also [programmatically query the supported VM sizes and restrictions in each region](#).

Storage availability

Understanding Azure regions and geographies becomes important when you consider the available storage replication options. Depending on the storage type, you have different replication options.

Azure Managed Disks

- Locally redundant storage (LRS)
 - Replicates your data three times within the region in which you created your storage account.

Storage account-based disks

- Locally redundant storage (LRS)
 - Replicates your data three times within the region in which you created your storage account.
- Zone redundant storage (ZRS)
 - Replicates your data three times across two to three facilities, either within a single region or across two regions.
- Geo-redundant storage (GRS)
 - Replicates your data to a secondary region that is hundreds of miles away from the primary region.
- Read-access geo-redundant storage (RA-GRS)
 - Replicates your data to a secondary region, as with GRS, but also then provides read-only access to the data in the secondary location.

The following table provides a quick overview of the differences between the storage replication types:

REPLICATION STRATEGY	LRS	ZRS	GRS	RA-GRS
Data is replicated across multiple facilities.	No	Yes	Yes	Yes
Data can be read from the secondary location and from the primary location.	No	No	No	Yes

REPLICATION STRATEGY	LRS	ZRS	GRS	RA-GRS
Number of copies of data maintained on separate nodes.	3	3	6	6

You can read more about [Azure Storage replication options here](#). For more information about managed disks, see [Azure Managed Disks overview](#).

Storage costs

Prices vary depending on the storage type and availability that you select.

Azure Managed Disks

- Premium Managed Disks are backed by Solid-State Drives (SSDs) and Standard Managed Disks are backed by regular spinning disks. Both Premium and Standard Managed Disks are charged based on the provisioned capacity for the disk.

Unmanaged disks

- Premium storage is backed by Solid-State Drives (SSDs) and is charged based on the capacity of the disk.
- Standard storage is backed by regular spinning disks and is charged based on the in-use capacity and desired storage availability.
 - For RA-GRS, there is an additional Geo-Replication Data Transfer charge for the bandwidth of replicating that data to another Azure region.

See [Azure Storage Pricing](#) for pricing information on the different storage types and availability options.

Availability options for virtual machines in Azure

1/19/2020 • 5 minutes to read • [Edit Online](#)

This article provides you with an overview of the availability features of Azure virtual machines (VMs).

High availability

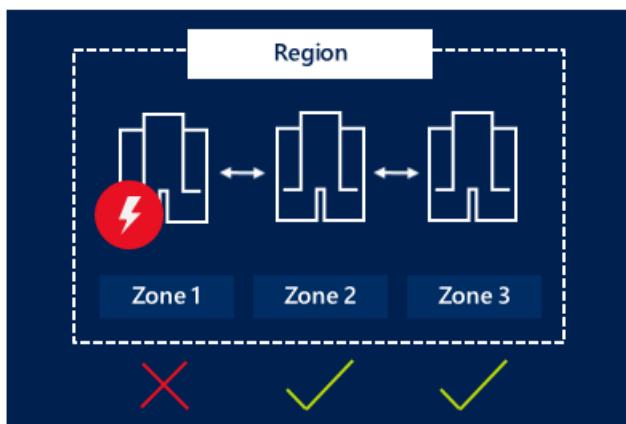
Workloads are typically spread across different virtual machines to gain high throughput, performance, and to create redundancy in case a VM is impacted due to an update or other event.

There are few options that Azure provides to achieve High Availability. First let's talk about basic constructs.

Availability zones

[Availability zones](#) expand the level of control you have to maintain the availability of the applications and data on your VMs. An Availability Zone is a physically separate zone, within an Azure region. There are three Availability Zones per supported Azure region.

Each Availability Zone has a distinct power source, network, and cooling. By architecting your solutions to use replicated VMs in zones, you can protect your apps and data from the loss of a datacenter. If one zone is compromised, then replicated apps and data are instantly available in another zone.



Learn more about deploying a [Windows](#) or [Linux](#) VM in an Availability Zone.

Fault domains

A fault domain is a logical group of underlying hardware that share a common power source and network switch, similar to a rack within an on-premises datacenter.

Update domains

An update domain is a logical group of underlying hardware that can undergo maintenance or be rebooted at the same time.

This approach ensures that at least one instance of your application always remains running as the Azure platform undergoes periodic maintenance. The order of update domains being rebooted may not proceed sequentially during maintenance, but only one update domain is rebooted at a time.

Virtual Machines Scale Sets

Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many VMs. We

recommended that two or more VMs are created within a scale set to provide for a highly available application and to meet the [99.95% Azure SLA](#). There is no cost for the scale set itself, you only pay for each VM instance that you create. When a single VM is using [Azure premium SSDs](#), the Azure SLA applies for unplanned maintenance events. Virtual machines in a scale set can be deployed across multiple update domains and fault domains to maximize availability and resilience to outages due to data center outages, and planned or unplanned maintenance events. Virtual machines in a scale set can also be deployed into a single Availability zone, or regionally. Availability zone deployment options may differ based on the orchestration mode.

Preview: Orchestration mode Preview

Virtual machines scale sets allow you to specify orchestration mode. With the virtual machine scale set orchestration mode (preview), you can now choose whether the scale set should orchestrate virtual machines which are created explicitly outside of a scale set configuration model, or virtual machine instances created implicitly based on the configuration model. Choose the orchestration mode that VM orchestration model allows you group explicitly defined Virtual Machines together in a region or in an availability zone. Virtual machines deployed in an Availability Zone provides zonal isolation to VMs as they are bound to the availability zone boundary and are not subjected to any failures that may occur in other availability zone in the region.

	"ORCHESTRATIONMODE": "VM" (VIRTUALMACHINE)	"ORCHESTRATIONMODE": "SCALESETVM" (VIRTUALMACHINESCALESET VM)
VM configuration model	None. VirtualMachineProfile is undefined in the scale set model.	Required. VirtualMachineProfile is populated in the scale set model.
Adding new VM to Scale Set	VMs are explicitly added to the scale set when the VM is created.	VMs are implicitly created and added to the scale set based on the VM configuration model, instance count, and AutoScaling rules.
Availability Zones	Supports regional deployment or VMs in one Availability Zone	Supports regional deployment or multiple Availability Zones; Can define the zone balancing strategy
Fault domains	Can define fault domains count. 2 or 3 based on regional support and 5 for Availability zone. The assigned VM fault domain will persist with VM lifecycle, including deallocate and restart.	Can define 1, 2, or 3 fault domains for non-zonal deployments, and 5 for Availability zone deployments. The assigned VM fault domain does not persist with VM lifecycle, virtual machines are assigned a fault domain at time of allocation.
Update domains	N/A. Update domains are automatically mapped to fault domains	N/A. Update domains are automatically mapped to fault domains

Fault domains and update domains

Virtual machine scale sets simplify designing for high availability by aligning fault domains and update domains. You will only have to define fault domains count for the scale set. The number of fault domains available to the scale sets may vary by region. See [Manage the availability of virtual machines in Azure](#).

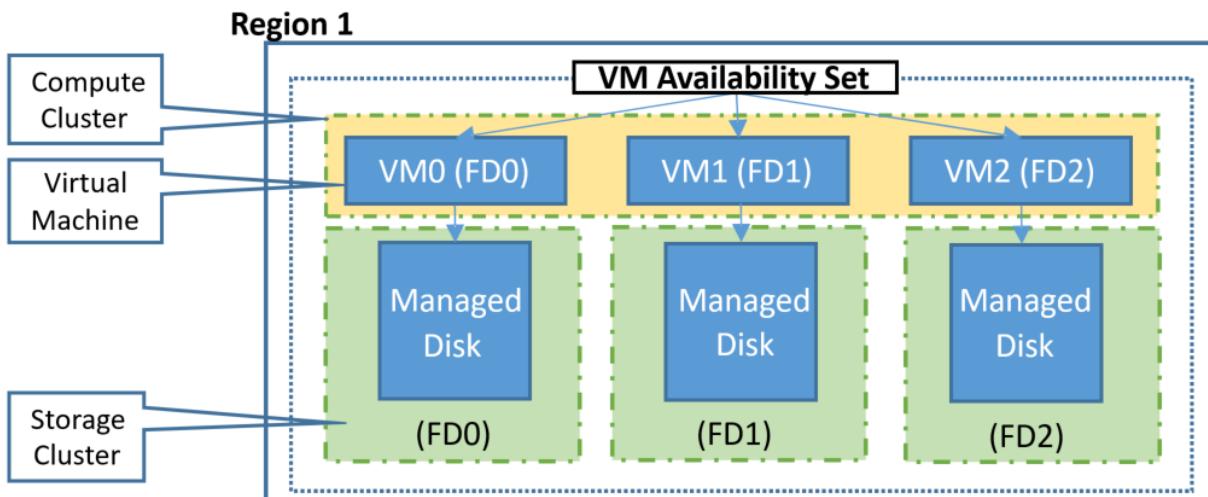
Availability sets

An availability set is a logical grouping of VMs within a datacenter that allows Azure to understand how your application is built to provide for redundancy and availability. We recommended that two or more VMs are created within an availability set to provide for a highly available application and to meet the [99.95% Azure SLA](#). There is no cost for the Availability Set itself, you only pay for each VM instance that you create. When a single VM is using [Azure premium SSDs](#), the Azure SLA applies for unplanned maintenance events.

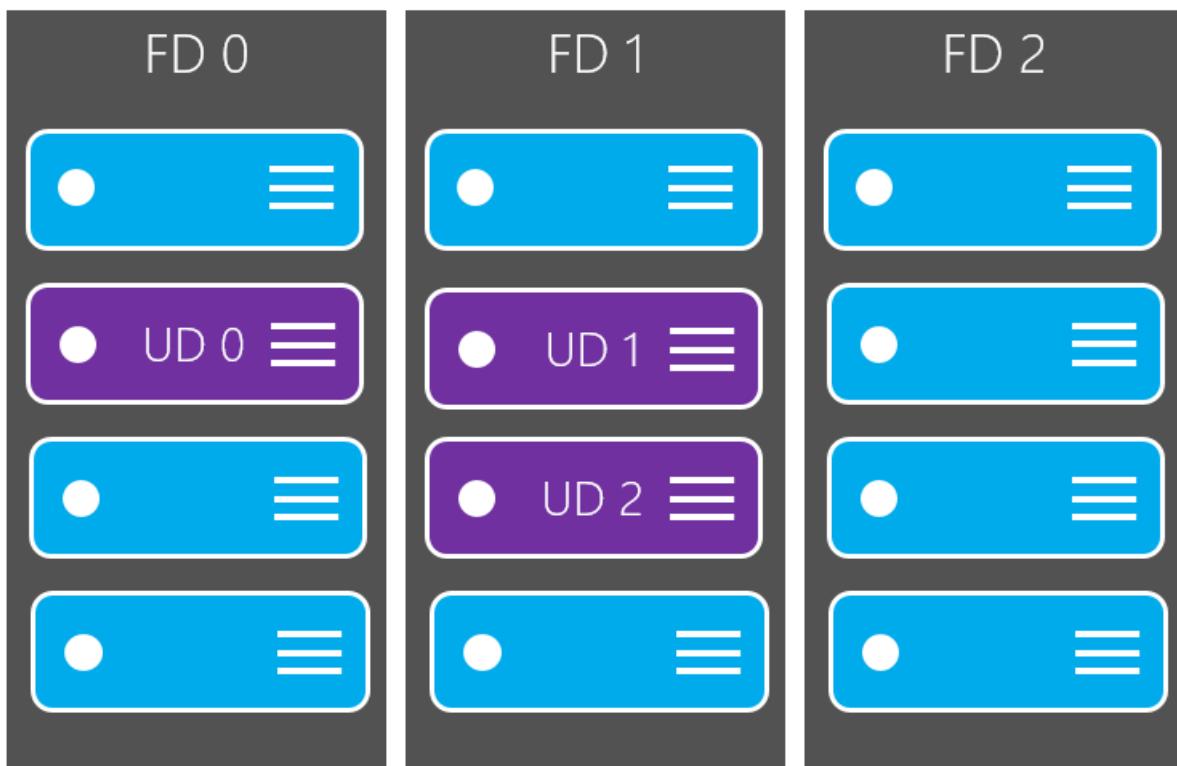
In an availability set, VMs are automatically distributed across these fault domains. This approach limits the impact of potential physical hardware failures, network outages, or power interruptions.

For VMs using [Azure Managed Disks](#), VMs are aligned with managed disk fault domains when using a managed availability set. This alignment ensures that all the managed disks attached to a VM are within the same managed disk fault domain.

Only VMs with managed disks can be created in a managed availability set. The number of managed disk fault domains varies by region - either two or three managed disk fault domains per region. You can read more about these managed disk fault domains for [Linux VMs](#) or [Windows VMs](#).



VMs within an availability set are also automatically distributed across update domains.



Next steps

You can now start to use these availability and redundancy features to build your Azure environment. For best practices information, see [Azure availability best practices](#).

Co-locate resource for improved latency

10/30/2019 • 3 minutes to read • [Edit Online](#)

When deploying your application in Azure, spreading instances across regions or availability zones creates network latency, which may impact the overall performance of your application.

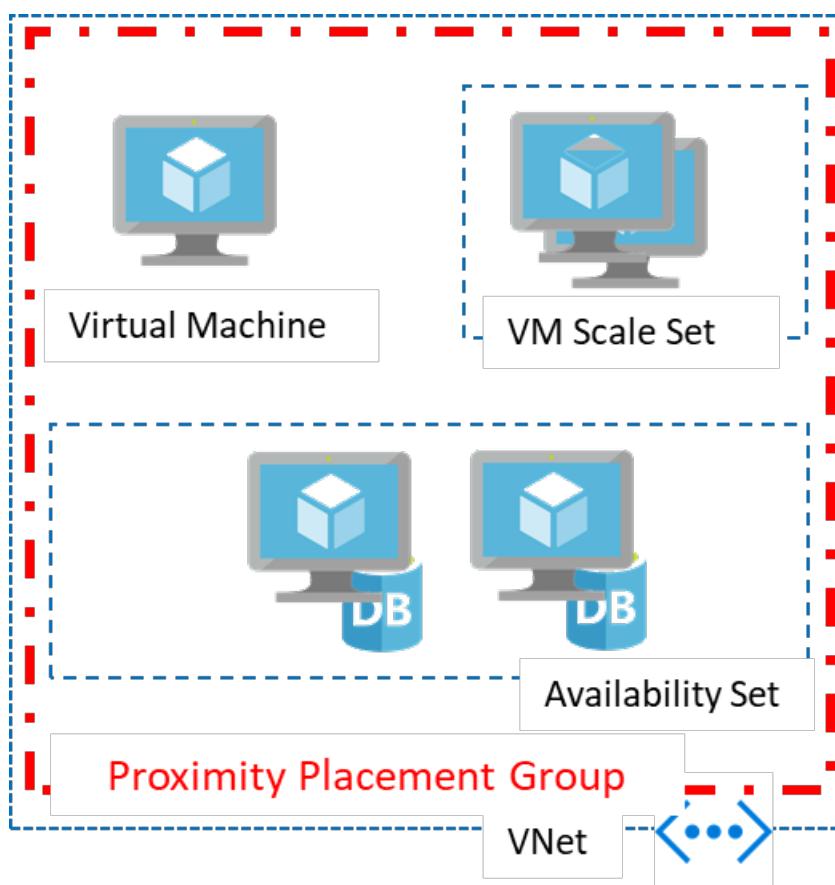
Proximity placement groups

Placing VMs in a single region reduces the physical distance between the instances. Placing them within a single availability zone will also bring them physically closer together. However, as the Azure footprint grows, a single availability zone may span multiple physical data centers, which may result in a network latency impacting your application.

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group.

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

- Low latency between stand-alone VMs.
- Low Latency between VMs in a single availability set or a virtual machine scale set.
- Low latency between stand-alone VMs, VMs in multiple Availability Sets, or multiple scale sets. You can have multiple compute resources in a single placement group to bring together a multi-tiered application.
- Low latency between multiple application tiers using different hardware types. For example, running the backend using M-series in an availability set and the front end on a D-series instance, in a scale set, in a single proximity placement group.



Using Proximity Placement Groups

A proximity placement group is a new resource type in Azure. You need to create one before using it with other resources. Once created, it could be used with virtual machines, availability sets, or virtual machine scale sets. You specify a proximity placement group when creating compute resources providing the proximity placement group ID.

You can also move an existing resource into a proximity placement group. When moving a resource into a proximity placement group, you should stop (deallocate) the asset first since it will be redeployed potentially into a different data center in the region so satisfy the colocation constraint.

In the case of availability sets and virtual machine scale sets, you should set the proximity placement group at the resource level rather than the individual virtual machines.

A proximity placement group is a colocation constraint rather than a pinning mechanism. It is pinned to a specific data center with the deployment of the first resource to use it. Once all resources using the proximity placement group have been stopped (deallocated) or deleted, it is no longer pinned. Therefore, when using a proximity placement group with multiple VM series, it is important to specify all the required types upfront in a template when possible or follow a deployment sequence which will improve your chances for a successful deployment. If your deployment fails, restart the deployment with the VM size which has failed as the first size to be deployed.

Best practices

- For the lowest latency, use proximity placement groups together with accelerated networking. For more information, see [Create a Linux virtual machine with Accelerated Networking](#) or [Create a Windows virtual machine with Accelerated Networking](#).
- Deploy all VM sizes in a single template. In order to avoid landing on hardware that doesn't support all the VM SKUs and sizes you require, include all of the application tiers in a single template so that they will all be deployed at the same time.
- If you are scripting your deployment using PowerShell, CLI or the SDK, you may get an allocation error `OverconstrainedAllocationRequest`. In this case, you should stop/deallocate all the existing VMs, and change the sequence in the deployment script to begin with the VM SKU/sizes that failed.
- When reusing an existing placement group from which VMs were deleted, wait for the deletion to fully complete before adding VMs to it.
- If latency is your first priority, put VMs in a proximity placement group and the entire solution in an availability zone. But, if resiliency is your top priority, spread your instances across multiple availability zones (a single proximity placement group cannot span zones).

Next steps

Deploy a VM to a [proximity placement group](#) using Azure PowerShell.

Learn how to [test network latency](#).

Learn how to [optimize network throughput](#).

Learn how to [use proximity placement groups with SAP applications](#).

Optimize network throughput for Azure virtual machines

1/6/2020 • 3 minutes to read • [Edit Online](#)

Azure virtual machines (VM) have default network settings that can be further optimized for network throughput. This article describes how to optimize network throughput for Microsoft Azure Windows and Linux VMs, including major distributions such as Ubuntu, CentOS, and Red Hat.

Windows VM

If your Windows VM supports [Accelerated Networking](#), enabling that feature would be the optimal configuration for throughput. For all other Windows VMs, using Receive Side Scaling (RSS) can reach higher maximal throughput than a VM without RSS. RSS may be disabled by default in a Windows VM. To determine whether RSS is enabled, and enable it if it's currently disabled, complete the following steps:

1. See if RSS is enabled for a network adapter with the `Get-NetAdapterRss` PowerShell command. In the following example output returned from the `Get-NetAdapterRss`, RSS is not enabled.

```
Name      : Ethernet
InterfaceDescription : Microsoft Hyper-V Network Adapter
Enabled    : False
```

2. To enable RSS, enter the following command:

```
Get-NetAdapter | % {Enable-NetAdapterRss -Name $_.Name}
```

The previous command does not have an output. The command changed NIC settings, causing temporary connectivity loss for about one minute. A Reconnecting dialog box appears during the connectivity loss. Connectivity is typically restored after the third attempt.

3. Confirm that RSS is enabled in the VM by entering the `Get-NetAdapterRss` command again. If successful, the following example output is returned:

```
Name      : Ethernet
InterfaceDescription : Microsoft Hyper-V Network Adapter
Enabled    : True
```

Linux VM

RSS is always enabled by default in an Azure Linux VM. Linux kernels released since October 2017 include new network optimizations options that enable a Linux VM to achieve higher network throughput.

Ubuntu for new deployments

The Ubuntu Azure kernel provides the best network performance on Azure and has been the default kernel since September 21, 2017. In order to get this kernel, first install the latest supported version of 16.04-LTS, as follows:

```
"Publisher": "Canonical",
"Offer": "UbuntuServer",
"Sku": "16.04-LTS",
"Version": "latest"
```

After the creation is complete, enter the following commands to get the latest updates. These steps also work for VMs currently running the Ubuntu Azure kernel.

```
#run as root or preface with sudo
apt-get -y update
apt-get -y upgrade
apt-get -y dist-upgrade
```

The following optional command set may be helpful for existing Ubuntu deployments that already have the Azure kernel but that have failed to further updates with errors.

```
#optional steps may be helpful in existing deployments with the Azure kernel
#run as root or preface with sudo
apt-get -f install
apt-get --fix-missing install
apt-get clean
apt-get -y update
apt-get -y upgrade
apt-get -y dist-upgrade
```

Ubuntu Azure kernel upgrade for existing VMs

Significant throughput performance can be achieved by upgrading to the Azure Linux kernel. To verify whether you have this kernel, check your kernel version.

```
#Azure kernel name ends with "-azure"
uname -r

#sample output on Azure kernel:
#4.13.0-1007-azure
```

If your VM does not have the Azure kernel, the version number usually begins with "4.4." If the VM does not have the Azure kernel, run the following commands as root:

```
#run as root or preface with sudo
apt-get update
apt-get upgrade -y
apt-get dist-upgrade -y
apt-get install "linux-azure"
reboot
```

CentOS

In order to get the latest optimizations, it is best to create a VM with the latest supported version by specifying the following parameters:

```
"Publisher": "OpenLogic",
"Offer": "CentOS",
"Sku": "7.4",
"Version": "latest"
```

New and existing VMs can benefit from installing the latest Linux Integration Services (LIS). The throughput

optimization is in LIS, starting from 4.2.2-2, although later versions contain further improvements. Enter the following commands to install the latest LIS:

```
sudo yum update  
sudo reboot  
sudo yum install microsoft-hyper-v
```

Red Hat

In order to get the optimizations, it is best to create a VM with the latest supported version by specifying the following parameters:

```
"Publisher": "RedHat"  
"Offer": "RHEL"  
"Sku": "7-RAW"  
"Version": "latest"
```

New and existing VMs can benefit from installing the latest Linux Integration Services (LIS). The throughput optimization is in LIS, starting from 4.2. Enter the following commands to download and install LIS:

```
wget https://aka.ms/lis  
tar xvf lis  
cd LISISO  
sudo ./install.sh #or upgrade.sh if prior LIS was previously installed
```

Learn more about Linux Integration Services Version 4.2 for Hyper-V by viewing the [download page](#).

Next steps

- See the optimized result with [Bandwidth/Throughput testing Azure VM](#) for your scenario.
- Read about how [bandwidth is allocated to virtual machines](#)
- Learn more with [Azure Virtual Network frequently asked questions \(FAQ\)](#)

Sizes for Windows virtual machines in Azure

2/25/2020 • 2 minutes to read • [Edit Online](#)

This article describes the available sizes and options for the Azure virtual machines you can use to run your Windows apps and workloads. It also provides deployment considerations to be aware of when you're planning to use these resources. This article is also available for [Linux virtual machines](#).

Type	Sizes	Description
General purpose	B, Dsv3, Dv3, Dasv4, Dav4, DSv2, Dv2, Av2, DC	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	Fsv2	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	Esv3, Ev3, Easv4, Eav4, Mv2, M, DSv2, Dv2	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	Lsv2	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases.
GPU	NC, NCv2, NCv3, ND, NDv2 (Preview), NV, NVv3, NVv4	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
High performance compute	HB, HC, H	Our fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

- For information about pricing of the various sizes, see [Virtual Machines Pricing](#).
- To see general limits on Azure VMs, see [Azure subscription and service limits, quotas, and constraints](#).
- Storage costs are calculated separately based on used pages in the storage account. For details, [Azure Storage Pricing](#).
- Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

REST API

For information on using the REST API to query for VM sizes, see the following:

- [List available virtual machine sizes for resizing](#)
- [List available virtual machine sizes for a subscription](#)

- [List available virtual machine sizes in an availability set](#)

ACU

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Benchmark scores

Learn more about compute performance for Windows VMs using the [CoreMark benchmark scores](#).

Next steps

Learn more about the different VM sizes that are available:

- [General purpose](#)
- [Compute optimized](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- Check the [Previous generation](#) page for A Standard, Dv1 (D1-4 and D11-14 v1), and A8-A11 series

Support for generation 2 VMs on Azure

2/28/2020 • 6 minutes to read • [Edit Online](#)

Support for generation 2 virtual machines (VMs) is now available on Azure. You can't change a virtual machine's generation after you've created it, so review the considerations on this page before you choose a generation.

Generation 2 VMs support key features that aren't supported in generation 1 VMs. These features include increased memory, Intel Software Guard Extensions (Intel SGX), and virtualized persistent memory (vPMEM). Generation 2 VMs running on-premises, have some features that aren't supported in Azure yet. For more information, see the [Features and capabilities](#) section.

Generation 2 VMs use the new UEFI-based boot architecture rather than the BIOS-based architecture used by generation 1 VMs. Compared to generation 1 VMs, generation 2 VMs might have improved boot and installation times. For an overview of generation 2 VMs and some of the differences between generation 1 and generation 2, see [Should I create a generation 1 or 2 virtual machine in Hyper-V?](#).

Generation 2 VM sizes

Generation 1 VMs are supported by all VM sizes in Azure (except for Mv2-series VMs). Azure now offers generation 2 support for the following selected VM series:

- [B-series](#)
- [DC-series](#)
- [DSv2-series](#) and [Dsv3-series](#)
- [Esv3-series](#)
- [Fsv2-series](#)
- [GS-series](#)
- [HB-series](#)
- [HC-series](#)
- [Ls-series](#) and [Lsv2-series](#)
- [Mv2-series](#)
- [NCv2-series](#) and [NCv3-series](#)
- [ND-series](#)
- [NVv3-series](#)

NOTE

The usage of generation 2 VM images for Mv2-series VMs is generally available since the Mv2-series works with generation 2 VM images exclusively. Generation 1 VM images are not supported on Mv2-series VMs.

Generation 2 VM images in Azure Marketplace

Generation 2 VMs support the following Marketplace images:

- Windows Server 2019, 2016, 2012 R2, 2012
- Windows 10
- SUSE Linux Enterprise Server 15 SP1
- SUSE Linux Enterprise Server 12 SP4

- Ubuntu Server 16.04, 18.04, 19.04, 19.10
- RHEL 8.1, 8.0, 7.7, 7.6, 7.5, 7.4, 7.0
- Cent OS 8.0, 7.7, 7.6, 7.5, 7.4
- Oracle Linux 7.7, 7.7-CI

On-premises vs. Azure generation 2 VMs

Azure doesn't currently support some of the features that on-premises Hyper-V supports for generation 2 VMs.

GENERATION 2 FEATURE	ON-PREMISES HYPER-V	AZURE
Secure boot	✓□	□
Shielded VM	✓□	□
vTPM	✓□	□
Virtualization-based security (VBS)	✓□	□
VHDX format	✓□	□

Features and capabilities

Generation 1 vs. generation 2 features

FEATURE	GENERATION 1	GENERATION 2
Boot	PCAT	UEFI
Disk controllers	IDE	SCSI
VM sizes	All VM sizes	Only VMs that support premium storage

Generation 1 vs. generation 2 capabilities

CAPABILITY	GENERATION 1	GENERATION 2
OS disk > 2 TB	□	✓□
Custom disk/image/swap OS	✓□	✓□
Virtual machine scale set support	✓□	✓□
Azure Site Recovery	✓□	✓□
Backup/restore	✓□	✓□
Shared image gallery	✓□	✓□
Azure disk encryption	✓□	□

Creating a generation 2 VM

Marketplace image

In the Azure portal or Azure CLI, you can create generation 2 VMs from a Marketplace image that supports UEFI boot.

Azure portal

Below are the steps to create a generation 2 (Gen2) VM in Azure portal.

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. Select **Create a resource**.
3. Click **See all** from the Azure Marketplace on the left.
4. Select an image which supports Gen2.
5. Click **Create**.
6. In the **Advanced** tab, under the **VM generation** section, select the **Gen 2** option.
7. In the **Basics** tab, Under **Instance details**, go to **Size** and open the **Select a VM size** blade.
8. Select a [supported generation 2 VM](#).
9. Go through the [Azure portal creation flow](#) to finish creating the VM.



PowerShell

You can also use PowerShell to create a VM by directly referencing the generation 1 or generation 2 SKU.

For example, use the following PowerShell cmdlet to get a list of the SKUs in the `WindowsServer` offer.

```
Get-AzVMImageSku -Location westus2 -PublisherName MicrosoftWindowsServer -Offer WindowsServer
```

Alternatively, you can use the Azure CLI to see any available generation 2 images, listed by **Publisher**.

```
az vm image list --publisher Canonical --sku gen2 --output table --all
```

If you're creating a VM with Windows Server 2012 as the OS, then you will select either the generation 1 (BIOS) or generation 2 (UEFI) VM SKU, which looks like this:

```
2012-Datacenter  
2012-datacenter-gensecond
```

See the [Features and capabilities](#) section for a current list of supported Marketplace images.

Managed image or managed disk

You can create a generation 2 VM from a managed image or managed disk in the same way you would create a generation 1 VM.

Virtual machine scale sets

You can also create generation 2 VMs by using virtual machine scale sets. In the Azure CLI, use Azure scale sets to create generation 2 VMs.

Frequently asked questions

- **Are generation 2 VMs available in all Azure regions?**

Yes. But not all [generation 2 VM sizes](#) are available in every region. The availability of the generation 2 VM depends on the availability of the VM size.

- **Is there a price difference between generation 1 and generation 2 VMs?**

No.

- **I have a .vhf file from my on-premises generation 2 VM. Can I use that .vhf file to create a generation 2 VM in Azure?** Yes, you can bring your generation 2 .vhf file to Azure and use that to create a generation 2 VM. Use the following steps to do so:

1. Upload the .vhf to a storage account in the same region where you'd like to create your VM.
2. Create a managed disk from the .vhf file. Set the Hyper-V Generation property to V2. The following PowerShell commands set Hyper-V Generation property when creating managed disk.

```
$sourceUri = 'https://xyzstorage.blob.core.windows.net/vhd/abcd.vhd'. #<Provide location to your
uploaded .vhf file>
$osDiskName = 'gen2Diskfrmgenvhf' #<Provide a name for your disk>
$diskconfig = New-AzDiskConfig -Location '<location>' -DiskSizeGB 127 -AccountType Standard_LRS -
OsType Windows -HyperVGeneration "V2" -SourceUri $sourceUri -CreateOption 'Import'
New-AzDisk -DiskName $osDiskName -ResourceGroupName '<Your Resource Group>' -Disk $diskconfig
```

3. Once the disk is available, create a VM by attaching this disk. The VM created will be a generation 2 VM. When the generation 2 VM is created, you can optionally generalize the image of this VM. By generalizing the image, you can use it to create multiple VMs.

- **How do I increase the OS disk size?**

OS disks larger than 2 TB are new to generation 2 VMs. By default, OS disks are smaller than 2 TB for generation 2 VMs. You can increase the disk size up to a recommended maximum of 4 TB. Use the Azure CLI or the Azure portal to increase the OS disk size. For information about how to expand disks programmatically, see [Resize a disk](#).

To increase the OS disk size from the Azure portal:

1. In the Azure portal, go to the VM properties page.
2. To shut down and deallocate the VM, select the **Stop** button.
3. In the **Disk**s section, select the OS disk you want to increase.
4. In the **Disk**s section, select **Configuration**, and update the **Size** to the value you want.
5. Go back to the VM properties page and **Start** the VM.

You might see a warning for OS disks larger than 2 TB. The warning doesn't apply to generation 2 VMs. However, OS disk sizes larger than 4 TB are *not recommended*.

- **Do generation 2 VMs support accelerated networking?**

Yes. For more information, see [Create a VM with accelerated networking](#).

- **Is VHDX supported on generation 2?**

No, generation 2 VMs support only VHD.

- **Do generation 2 VMs support Azure Ultra Disk Storage?**

Yes.

- **Can I migrate a VM from generation 1 to generation 2?**

No, you can't change the generation of a VM after you create it. If you need to switch between VM generations, create a new VM of a different generation.

- **Why is my VM size not enabled in the size selector when I try to create a Gen2 VM?**

This may be solved by doing the following:

1. Verify that the **VM generation** property is set to **Gen 2** in the **Advanced** tab.
2. Verify you are searching for a [VM size which supports Gen2 VMs](#).

Next steps

- Learn about [generation 2 virtual machines in Hyper-V](#).
- Learn how to [prepare a VHD](#) to upload from on-premises systems to Azure.

General purpose virtual machine sizes

2/25/2020 • 2 minutes to read • [Edit Online](#)

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers. This article provides information about the offerings for general purpose computing.

- The [Av2-series](#) VMs can be deployed on a variety of hardware types and processors. A-series VMs have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled, based upon the hardware, to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on. To determine the physical hardware on which this size is deployed, query the virtual hardware from within the Virtual Machine. Example use cases include development and test servers, low traffic web servers, small to medium databases, proof-of-concepts, and code repositories.
- **B-series burstable** VMs are ideal for workloads that do not need the full performance of the CPU continuously, like web servers, small databases and development and test environments. These workloads typically have burstable performance requirements. The B-Series provides these customers the ability to purchase a VM size with a price conscious baseline performance that allows the VM instance to build up credits when the VM is utilizing less than its base performance. When the VM has accumulated credit, the VM can burst above the VM's baseline using up to 100% of the CPU when your application requires the higher CPU performance.
- **Dav4 and Dasv4-series** are new sizes utilizing AMD's 2.35Ghz EPYC™ 7452 processor in a multi-threaded configuration with up to 256 MB L3 cache dedicating 8 GB of that L3 cache to every 8 cores increasing customer options for running their general purpose workloads. The Dav4-series and Dasv4-series have the same memory and disk configurations as the D & Dsv3-series.
- The [DCv2-series](#) can help protect the confidentiality and integrity of your data and code while it's processed in the public cloud. These machines are backed by the latest generation of Intel XEON E-2288G Processor with SGX technology. With the Intel Turbo Boost Technology these machines can go up to 5.0GHz. DCv2 series instances enable customers to build secure enclave-based applications to protect their code and data while it's in use.
- **Dv2 and Dsv2-series** VMs, a follow-on to the original D-series, features a more powerful CPU and optimal CPU-to-memory configuration making them suitable for most production workloads. The Dv2-series is about 35% faster than the D-series. Dv2-series runs on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.
- **Dv3 and Dsv3-series** VMs run on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads. Memory has been expanded (from ~3.5 GiB/vCPU to 4 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Dv3-series no longer has the high memory VM sizes of the D/Dv2-series, those have been moved to the memory optimized [Ev3 and Esv3-series](#).

Example D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Other sizes

- [Compute optimized](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Av2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The Av2-series VMs can be deployed on a variety of hardware types and processors. Av2-series VMs have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on. To determine the physical hardware on which this size is deployed, query the virtual hardware from within the Virtual Machine. Some example use cases include development and test servers, low traffic web servers, small to medium databases, proof-of-concepts, and code repositories.

ACU: 100

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_A1_v2	1	2	10	1000/20/10	2/2x500	2/250
Standard_A2_v2	2	4	20	2000/40/20	4/4x500	2/500
Standard_A4_v2	4	8	40	4000/80/40	8/8x500	4/1000
Standard_A8_v2	8	16	80	8000/160/80	16/16x500	8/2000
Standard_A2_m_v2	2	16	20	2000/40/20	4/4x500	2/500
Standard_A4_m_v2	4	32	40	4000/80/40	8/8x500	4/1000
Standard_A8_m_v2	8	64	80	8000/160/80	16/16x500	8/2000

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode

is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.

- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

B-series burstable virtual machine sizes

2/20/2020 • 6 minutes to read • [Edit Online](#)

The B-series VMs are ideal for workloads that do not need the full performance of the CPU continuously, like web servers, proof of concepts, small databases and development build environments. These workloads typically have burstable performance requirements. The B-series provides you with the ability to purchase a VM size with baseline performance and the VM instance builds up credits when it is using less than its baseline. When the VM has accumulated credit, the VM can burst above the baseline using up to 100% of the vCPU when your application requires higher CPU performance.

The B-series comes in the following VM sizes:

Premium Storage: Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GIB	TEMP STORAGE (SSD) GIB	BASE CPU PERF OF VM	MAX CPU PERF OF VM	INITIAL CREDITS	CREDITS BANKED/HOUR	MAX BANKED CREDITS	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICS
Standard_B_1ls ¹	1	0.5	4	5%	100%	30	3	72	2	200/10	160/10	2
Standard_B_1s	1	1	4	10%	100%	30	6	144	2	400/10	320/10	2
Standard_B_1ms	1	2	4	20%	100%	30	12	288	2	800/10	640/10	2
Standard_B_2s	2	4	8	40%	200%	60	24	576	4	1600/15	1280/15	3
Standard_B_2ms	2	8	16	60%	200%	60	36	864	4	2400/22.5	1920/22.5	3
Standard_B_4ms	4	16	32	90%	400%	120	54	1296	8	3600/35	2880/35	4
Standard_B_8ms	8	32	64	135%	800%	240	81	1944	16	4320/50	4320/50	4

SIZE	VCPU	MEM ORY: GiB	TEMP STOR AGE (SSD) GiB	BASE CPU PERF OF VM	MAX CPU PERF OF VM	INITI AL CREDI TS	CREDI TS BANK ED/H OUR	MAX BANK ED CREDI TS	MAX DATA DISKS	MAX CACH ED AND TEMP STOR AGE	MAX UNCA CHED DISK	MAX NICS
										THRO UGHP UT:	THRO UGHP UT:	
Standard_B 12ms	12	48	96	202%	1200 %	360	121	2909	16	6480 /75	4320 /50	6
Standard_B 16ms	16	64	128	270%	1600 %	480	162	3888	32	8640 /100	4320 /50	8
Standard_B 20ms	20	80	160	337%	2000 %	600	203	4860	32	1080 0/12 5	4320 /50	8

¹ B11s is supported only on Linux

Workload example

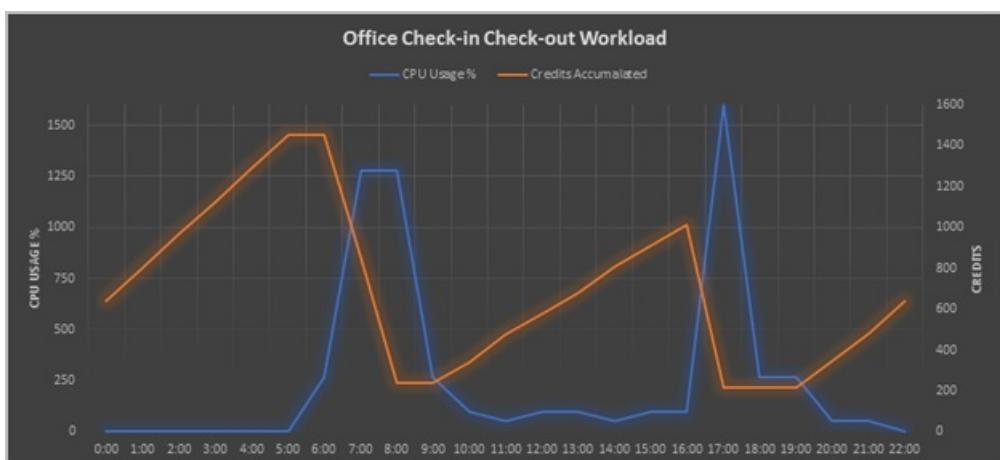
Consider an office check-in/out application. The application needs CPU bursts during business hours, but not a lot of computing power during off hours. In this example, the workload requires a 16vCPU virtual machine with 64GiB of RAM to work efficiently.

The table shows the hourly traffic data and the chart is a visual representation of that traffic.

B16 characteristics:

Max CPU perf: 16vCPU * 100% = 1600%

Baseline: 270%



SCENARIO	TIME	CPU USAGE (%)	CREDITS ACCUMULATED ¹	CREDITS AVAILABLE
B16ms Deployment	Deployment	Deployment	480 (Initial Credits)	480
No traffic	0:00	0	162	642

SCENARIO	TIME	CPU USAGE (%)	CREDITS ACCUMULATED	CREDITS AVAILABLE
No traffic	1:00	0	162	804
No traffic	2:00	0	162	966
No traffic	3:00	0	162	1128
No traffic	4:00	0	162	1290
No traffic	5:00	0	162	1452
Low Traffic	6:00	270	0	1452
Employees come to office (app needs 80% vCPU)	7:00	1280	-606	846
Employees continue coming to office (app needs 80% vCPU)	8:00	1280	-606	240
Low Traffic	9:00	270	0	240
Low Traffic	10:00	100	102	342
Low Traffic	11:00	50	132	474
Low Traffic	12:00	100	102	576
Low Traffic	13:00	100	102	678
Low Traffic	14:00	50	132	810
Low Traffic	15:00	100	102	912
Low Traffic	16:00	100	102	1014
Employees checking out (app needs 100% vCPU)	17:00	1600	-798	216
Low Traffic	18:00	270	0	216
Low Traffic	19:00	270	0	216
Low Traffic	20:00	50	132	348
Low Traffic	21:00	50	132	480
No traffic	22:00	0	162	642
No traffic	23:00	0	162	804

¹ Credits accumulated/credits used in an hour is equivalent to:

$$((\text{Base CPU perf of VM} - \text{CPU Usage}) / 100) * 60 \text{ minutes}$$

For a D16s_v3 which has 16 vCPUs and 64 GiB of memory the hourly rate is \$0.936 per hour (monthly \$673.92) and for B16ms with 16 vCPUs and 64 GiB memory the rate is \$0.794 per hour (monthly \$547.86). **This results in 15% savings!**

Q & A

Q: How do you get 135% baseline performance from a VM?

A: The 135% is shared amongst the 8 vCPU's that make up the VM size. For example, if your application uses 4 of the 8 cores working on batch processing and each of those 4 vCPU's are running at 30% utilization the total amount of VM CPU performance would equal 120%. Meaning that your VM would be building credit time based on the 15% delta from your baseline performance. But it also means that when you have credits available that same VM can use 100% of all 8 vCPU's giving that VM a Max CPU performance of 800%.

Q: How can I monitor my credit balance and consumption

A: We will be introducing 2 new metrics in the coming weeks, the **Credit** metric will allow you to view how many credits your VM has banked and the **ConsumedCredit** metric will show how many CPU credits your VM has consumed from the bank. You will be able to view these metrics from the metrics pane in the portal or programmatically through the Azure Monitor APIs.

For more information on how to access the metrics data for Azure, see [Overview of metrics in Microsoft Azure](#).

Q: How are credits accumulated?

A: The VM accumulation and consumption rates are set such that a VM running at exactly its base performance level will have neither a net accumulation or consumption of bursting credits. A VM will have a net increase in credits whenever it is running below its base performance level and will have a net decrease in credits whenever the VM is utilizing the CPU more than its base performance level.

Example: I deploy a VM using the B1ms size for my small time and attendance database application. This size allows my application to use up to 20% of a vCPU as my baseline, which is 0.2 credits per minute I can use or bank.

My application is busy at the beginning and end of my employees work day, between 7:00-9:00 AM and 4:00 - 6:00PM. During the other 20 hours of the day, my application is typically at idle, only using 10% of the vCPU. For the non-peak hours, I earn 0.2 credits per minute but only consume 0.1 credits per minute, so my VM will bank 0.1 x 60 = 6 credits per hour. For the 20 hours that I am off-peak, I will bank 120 credits.

During peak hours my application averages 60% vCPU utilization, I still earn 0.2 credits per minute but I consume 0.6 credits per minute, for a net cost of 0.4 credits a minute or 0.4 x 60 = 24 credits per hour. I have 4 hours per day of peak usage, so it costs 4 x 24 = 96 credits for my peak usage.

If I take the 120 credits I earned off-peak and subtract the 96 credits I used for my peak times, I bank an additional 24 credits per day that I can use for other bursts of activity.

Q: How can I calculate credits accumulated and used?

A: You can use the following formula:

$$(\text{Base CPU perf of VM} - \text{CPU Usage}) / 100 = \text{Credits bank or use per minute}$$

e.g in above instance your baseline is 20% and if you use 10% of the CPU you are accumulating $(20\% - 10\%) / 100 = 0.1$ credit per minute.

Q: Does the B-Series support Premium Storage data disks?

A: Yes, all B-Series sizes support Premium Storage data disks.

Q: Why is my remaining credit set to 0 after a redeploy or a stop/start?

A : When a VM is “REDPLOYED” and the VM moves to another node, the accumulated credit is lost. If the VM is stopped/started, but remains on the same node, the VM retains the accumulated credit. Whenever the VM starts fresh on a node, it gets an initial credit, for Standard_B8ms it is 240 mins.

Q: What happens if I deploy an unsupported OS image on B1ls?

A : B1ls only supports Linux images and if you deploy any another OS image you might not get the best customer experience.

Other sizes

- [General purpose](#)
- [Compute optimized](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Preview: DCv2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The DCv2-series can help protect the confidentiality and integrity of your data and code while it's processed in the public cloud. These machines are backed by the latest generation of Intel XEON E-2288G Processor with SGX technology. With the Intel Turbo Boost Technology these machines can go up to 5.0GHz. DCv2 series instances enable customers to build secure enclave-based applications to protect their code and data while it's in use.

Example use cases include confidential multiparty data sharing, fraud detection, anti-money laundering, blockchain, confidential usage analytics, intelligence analysis and confidential machine learning.

Premium Storage: Supported*

Premium Storage caching: Supported*

*Except for Standard_DC8_v2

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUHPUT: IOPS / MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUHPUT: IOPS / MBPS	MAX NICS / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D_C1s_v2	1	4	50	1	2000/16 (21)	1600/24	2
Standard_D_C2s_v2	2	8	100	2	4000/32 (43)	3200/48	2
Standard_D_C4s_v2	4	16	200	4	8000/64 (86)	6400/96	2
Standard_D_C8_v2	8	32	400	8	16000/128 (172)	12800/192	2

- DCv2-series VMs are [generation 2 VMs](#) and only support [Gen2](#) images.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Dv2 and DSv2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The Dv2 and DSv2-series, a follow-on to the original D-series, feature a more powerful CPU and optimal CPU-to-memory configuration making them suitable for most production workloads. The Dv2-series is about 35% faster than the D-series. Dv2-series runs on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.

Dv2-series

Dv2-series sizes run on the Intel® Xeon® 8171M 2.1GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with Intel Turbo Boost Technology 2.0.

ACU: 210-250

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS	THROUGHPUT UT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D1_v2	1	3.5	50	3000/46/23	4	4x500	2/750
Standard_D2_v2	2	7	100	6000/93/46	8	8x500	2/1500
Standard_D3_v2	4	14	200	12000/187/93	16	16x500	4/3000
Standard_D4_v2	8	28	400	24000/375/187	32	32x500	8/6000
Standard_D5_v2	16	56	800	48000/750/375	64	64x500	8/12000

DSv2-series

DSv2-series sizes run on the Intel® Xeon® 8171M 2.1GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with Intel Turbo Boost Technology 2.0 and use premium storage.

ACU: 210-250

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GiB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D1_v2	1	3.5	7	4	4000/32 (43)	3200/48	2/750
Standard_D2_v2	2	7	14	8	8000/64 (86)	6400/96	2/1500
Standard_D3_v2	4	14	28	16	16000/128 (172)	12800/192	4/3000
Standard_D4_v2	8	28	56	32	32000/256 (344)	25600/384	8/6000
Standard_D5_v2	16	56	112	64	64000/512 (688)	51200/768	8/12000

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)

- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Dv3 and Dsv3-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The Dv3-series runs on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads. Memory has been expanded (from ~3.5 GiB/vCPU to 4 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Dv3-series no longer has the high memory VM sizes of the D/Dv2-series, those have been moved to the memory optimized [Ev3 and Esv3-series](#).

Example D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Dv3-series

Dv3-series sizes run on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with Intel Turbo Boost Technology 2.0. The Dv3-series sizes offer a combination of vCPU, memory, and temporary storage for most production workloads.

Data disk storage is billed separately from virtual machines. To use premium storage disks, use the Dsv3 sizes. The pricing and billing meters for Dsv3 sizes are the same as Dv3-series.

Dv3-series VMs feature Intel® Hyper-Threading Technology.

ACU: 160-190

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX NICs/NETWORK BANDWIDTH
Standard_D2_v3	2	8	50	4	3000/46/23	2/1000
Standard_D4_v3	4	16	100	8	6000/93/46	2/2000
Standard_D8_v3	8	32	200	16	12000/187/93	4/4000
Standard_D16_v3	16	64	400	32	24000/375/187	8/8000
Standard_D32_v3	32	128	800	32	48000/750/375	8/16000

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX NICS/NETWORK BANDWIDTH
Standard_D48_v3	48	192	1200	32	96000/1000/500	8/24000
Standard_D64_v3	64	256	1600	32	96000/1000/500	8/30000

Dsv3-series

Dsv3-series sizes run on the Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with Intel Turbo Boost Technology 2.0 and use premium storage. The Dsv3-series sizes offer a combination of vCPU, memory, and temporary storage for most production workloads.

Dsv3-series VMs feature Intel® Hyper-Threading Technology.

ACU: 160-190

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D2s_v3	2	8	16	4	4000/32 (50)	3200/48	2/1000
Standard_D4s_v3	4	16	32	8	8000/64 (100)	6400/96	2/2000
Standard_D8s_v3	8	32	64	16	16000/128 (200)	12800/192	4/4000
Standard_D16s_v3	16	64	128	32	32000/256 (400)	25600/384	8/8000
Standard_D32s_v3	32	128	256	32	64000/512 (800)	51200/768	8/16000
Standard_D48s_v3	48	192	384	32	96000/768 (1200)	76800/1152	8/24000
Standard_D64s_v3	64	256	512	32	128000/1024 (1600)	80000/1200	8/30000

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Dav4 and Dasv4-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The Dav4-series and Dasv4-series are new sizes utilizing AMD's 2.35Ghz EPYC™ 7452 processor in a multi-threaded configuration with up to 256 MB L3 cache dedicating 8 GB of that L3 cache to every 8 cores increasing customer options for running their general purpose workloads. The Dav4-series and Dasv4-series have the same memory and disk configurations as the D & Dsv3-series.

Dav4-series

ACU: 230-260

Premium Storage: Not Supported

Premium Storage caching: Not Supported

Dav4-series sizes are based on the 2.35Ghz AMD EPYC™ 7452 processor that can achieve a boosted maximum frequency of 3.35GHz. The Dav4-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads. Data disk storage is billed separately from virtual machines. To use premium SSD, use the Dasv4 sizes. The pricing and billing meters for Dasv4 sizes are the same as the Dav4-series.

SIZE	VCPUs	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS / READ MBPS / WRITE MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D2a_v4	2	8	50	4	3000 / 46 / 23	2 / 1000
Standard_D4a_v4	4	16	100	8	6000 / 93 / 46	2 / 2000
Standard_D8a_v4	8	32	200	16	12000 / 187 / 93	4 / 4000
Standard_D16a_v4	16	64	400	32	24000 / 375 / 187	8 / 8000
Standard_D32a_v4	32	128	800	32	48000 / 750 / 375	8 / 16000
Standard_D48a_v4 **	48	192	1200	32		
Standard_D64a_v4 **	64	256	1600	32		
Standard_D96a_v4 **	96	384	2400	32		

** These sizes are in Preview. If you are interested in trying out these larger sizes, sign up at <https://aka.ms/AzureAMDLargeVMPreview>.

Dasv4-series

ACU: 230-260

Premium Storage: Supported

Premium Storage caching: Supported

Dasv4-series sizes are based on the 2.35Ghz AMD EPYC™ 7452 processor that can achieve a boosted maximum frequency of 3.35GHz and use premium SSD. The Dasv4-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads.

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS / MBPS (CACHE SIZE IN GiB)	MAX UNCACHED DISK THROUGHPUT: IOPS / MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D_2as_v4	2	8	16	4	4000 / 32 (50)	3200 / 48	2 / 1000
Standard_D_4as_v4	4	16	32	8	8000 / 64 (100)	6400 / 96	2 / 2000
Standard_D_8as_v4	8	32	64	16	16000 / 128 (200)	12800 / 192	4 / 4000
Standard_D_16as_v4	16	64	128	32	32000 / 255 (400)	25600 / 384	8 / 8000
Standard_D_32as_v4	32	128	256	32	64000 / 510 (800)	51200 / 768	8 / 16000
Standard_D_48as_v4 **	48	192	384	32			
Standard_D_64as_v4 **	64	256	512	32			
Standard_D_96as_v4 **	96	384	768	32			

** These sizes are in Preview. If you are interested in trying out these larger sizes, sign up at <https://aka.ms/AzureAMDLargeVMPreview>.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode

is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.

- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Compute optimized virtual machine sizes

2/20/2020 • 2 minutes to read • [Edit Online](#)

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers. This article provides information about the number of vCPUs, data disks, and NICs. It also includes information about storage throughput and network bandwidth for each size in this grouping.

The [Fsv2-series](#) is based on the Intel® Xeon® Platinum 8168 processor. It features a sustained all core Turbo clock speed of 3.4 GHz and a maximum single-core turbo frequency of 3.7 GHz. Intel® AVX-512 instructions are new on Intel Scalable Processors. These instructions provide up to a 2X performance boost to vector processing workloads on both single and double precision floating point operations. In other words, they're really fast for any computational workload.

At a lower per-hour list price, the Fsv2-series is the best value in price-performance in the Azure portfolio based on the Azure Compute Unit (ACU) per vCPU.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Fsv2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The Fsv2-series is based on the Intel® Xeon® Platinum 8168 processor. It features a sustained all core Turbo clock speed of 3.4 GHz and a maximum single-core turbo frequency of 3.7 GHz. Intel® AVX-512 instructions are new on Intel Scalable Processors. These instructions provide up to a 2X performance boost to vector processing workloads on both single and double precision floating point operations. In other words, they're really fast for any computational workload.

Fsv2-series VMs feature Intel® Hyper-Threading Technology.

ACU: 195 - 210

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU'S	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUHPUT: IOPS/MBPS	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_F_2s_v2	2	4	16	4	4000/31 (32)	3200/47	2/875
Standard_F_4s_v2	4	8	32	8	8000/63 (64)	6400/95	2/1750
Standard_F_8s_v2	8	16	64	16	16000/127 (128)	12800/190	4/3500
Standard_F_16s_v2	16	32	128	32	32000/255 (256)	25600/380	4/7000
Standard_F_32s_v2	32	64	256	32	64000/512 (512)	51200/750	8/14000
Standard_F_48s_v2	48	96	384	32	96000/768 (768)	76800/1100	8/21000
Standard_F_64s_v2	64	128	512	32	128000/1024 (1024)	80000/1100	8/28000
Standard_F_72s_v2 ^{1, 2}	72	144	576	32	144000/1152 (1520)	80000/1100	8/30000

¹ The use of more than 64 vCPU require one of these supported guest operating systems:

- Windows Server 2016 or later
- Ubuntu 16.04 LTS or later, with Azure tuned kernel (4.15 kernel or later)

- SLES 12 SP2 or later
- RHEL or CentOS version 6.7 through 6.10, with Microsoft-provided LIS package 4.3.1 (or later) installed
- RHEL or CentOS version 7.3, with Microsoft-provided LIS package 4.2.1 (or later) installed
- RHEL or CentOS version 7.6 or later
- Oracle Linux with UEK4 or later
- Debian 9 with the backports kernel, Debian 10 or later
- CoreOS with a 4.14 kernel or later

² Instance is isolated to hardware dedicated to a single customer.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Memory optimized virtual machine sizes

2/20/2020 • 2 minutes to read • [Edit Online](#)

Memory optimized VM sizes offer a high memory-to-CPU ratio that are great for relational database servers, medium to large caches, and in-memory analytics. This article provides information about the number of vCPUs, data disks and NICs as well as storage throughput and network bandwidth for each size in this grouping.

- [Dv2 and DSv2-series](#), a follow-on to the original D-series, features a more powerful CPU. The Dv2-series is about 35% faster than the D-series. It runs on the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors, and with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.

Dv2 and DSv2-series are ideal for applications that demand faster vCPUs, better temporary storage performance, or have higher memory demands. They offer a powerful combination for many enterprise-grade applications.

- The [Eav4 and Easv4-series](#) utilize AMD's 2.35Ghz EPYC™ 7452 processor in a multi-threaded configuration with up to 256MB L3 cache, increasing options for running most memory optimized workloads. The Eav4-series and Easv4-series have the same memory and disk configurations as the Ev3 & Esv3-series.
- The [Ev3 and Esv3-series](#) Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads, and bringing the Ev3 into alignment with the general purpose VMs of most other clouds. Memory has been expanded (from 7 GiB/vCPU to 8 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyper-threading. The Ev3 is the follow up to the high memory VM sizes of the D/Dv2 families.
- The [M-series](#) offers a high vCPU count (up to 128 vCPUs) and a large amount of memory (up to 3.8 TiB). It's also ideal for extremely large databases or other applications that benefit from high vCPU counts and large amounts of memory.
- The [Mv2-series](#) offers the highest vCPU count (up to 416 vCPUs) and largest memory (up to 8.19 TiB) of any VM in the cloud. It's ideal for extremely large databases or other applications that benefit from high vCPU counts and large amounts of memory.

Azure Compute offers virtual machine sizes that are Isolated to a specific hardware type and dedicated to a single customer. These virtual machine sizes are best suited for workloads that require a high degree of isolation from other customers for workloads involving elements like compliance and regulatory requirements. Customers can also choose to further subdivide the resources of these Isolated virtual machines by using [Azure support for nested virtual machines](#). See the pages for virtual machine families below for your isolated VM options.

Other sizes

- [General purpose](#)
- [Compute optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)

- High performance compute
- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Memory optimized Dv2 and DSv2-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

Dv2 and DSv2-series, a follow-on to the original D-series, features a more powerful CPU. DSv2-series sizes run on the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors. The Dv2-series has the same memory and disk configurations as the D-series.

Dv2-series 11-15

Dv2-series sizes run on the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors.

ACU: 210 - 250

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D11_v2	2	14	100	6000/93/46	8/8x500	2/1500
Standard_D12_v2	4	28	200	12000/187/93	16/16x500	4/3000
Standard_D13_v2	8	56	400	24000/375/187	32/32x500	8/6000
Standard_D14_v2	16	112	800	48000/750/375	64/64x500	8/12000
Standard_D15_v2 ¹	20	140	1000	60000/937/468	64/64x500	8/25000 ²

¹ Instance is isolated to hardware dedicated to a single customer. ² 25000 Mbps with Accelerated Networking.

DSv2-series 11-15

DSv2-series sizes run on the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors.

ACU: 210 - 250¹

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D S11_v2 ³	2	14	28	8	8000/64 (72)	6400/96	2/1500
Standard_D S12_v2 ³	4	28	56	16	16000/128 (144)	12800/192	4/3000
Standard_D S13_v2 ³	8	56	112	32	32000/256 (288)	25600/384	8/6000
Standard_D S14_v2 ³	16	112	224	64	64000/512 (576)	51200/768	8/12000
Standard_D S15_v2 ²	20	140	280	64	80000/640 (720)	64000/960	8/25000 ⁴

¹ The maximum disk throughput (IOPS or MBps) possible with a DSv2 series VM may be limited by the number, size and striping of the attached disk(s). For details, see [Designing for high performance](#). ² Instance is isolated to the Intel Haswell based hardware and dedicated to a single customer.

³ Constrained core sizes available.

⁴ 25000 Mbps with Accelerated Networking.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Ev3 and Esv3-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The Ev3 and Esv3-series feature the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads, and bringing the Ev3 into alignment with the general purpose VMs of most other clouds. Memory has been expanded (from 7 GiB/vCPU to 8 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Ev3 is the follow up to the high memory VM sizes of the D/Dv2 families.

Ev3-series

Ev3-series instances are based on feature the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor and Intel Turbo Boost Technology 2.0. Ev3-series instances are ideal for memory-intensive enterprise applications.

Data disk storage is billed separately from virtual machines. To use premium storage disks, use the ESv3 sizes. The pricing and billing meters for ESv3 sizes are the same as Ev3-series.

Ev3-series VM's feature Intel® Hyper-Threading Technology.

ACU: 160 - 190

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS / READ MBPS / WRITE MBPS	MAX NICs / NETWORK BANDWIDTH
Standard_E2_v3	2	16	50	4	3000/46/23	2/1000
Standard_E4_v3	4	32	100	8	6000/93/46	2/2000
Standard_E8_v3	8	64	200	16	12000/187/93	4/4000
Standard_E16_v3	16	128	400	32	24000/375/187	8/8000
Standard_E20_v3	20	160	500	32	30000/469/234	8/10000
Standard_E32_v3	32	256	800	32	48000/750/375	8/16000
Standard_E48_v3	48	384	1200	32	96000/1000/500	8/24000

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS / READ MBPS / WRITE MBPS	MAX NICs / NETWORK BANDWIDTH
Standard_E64_v3	64	432	1600	32	96000/1000/500	8/30000
Standard_E64i_v3 ^{1,2}	64	432	1600	32	96000/1000/500	8/30000

¹ Constrained core sizes available.

² Instance is isolated to hardware dedicated to a single customer.

Esv3-series

Esv3-series instances are based on feature the Intel® Xeon® 8171M 2.1 GHz (Skylake) or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor, Intel Turbo Boost Technology 2.0, and use premium storage. Esv3-series instances are ideal for memory-intensive enterprise applications.

Esv3-series VM's feature Intel® Hyper-Threading Technology.

ACU: 160-190

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_E2s_v3	2	16	32	4	4000/32 (50)	3200/48	2/1000
Standard_E4s_v3 ¹	4	32	64	8	8000/64 (100)	6400/96	2/2000
Standard_E8s_v3 ¹	8	64	128	16	16000/128 (200)	12800/192	4/4000
Standard_E16s_v3 ¹	16	128	256	32	32000/256 (400)	25600/384	8/8000
Standard_E20s_v3	20	160	320	32	40000/320 (400)	32000/480	8/10000
Standard_E32s_v3 ¹	32	256	512	32	64000/512 (800)	51200/768	8/16000
Standard_E48s_v3 ¹	48	384	768	32	96000/768 (1200)	76800/1152	8/24000

SIZE	VCPU	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GiB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
------	------	----------------	------------------------------	-------------------	--	---	---

Standard_E 64s_v3 ¹	64	432	864	32	128000/10 24 (1600)	80000/120 0	8/30000
Standard_E 64is_v3 ²	64	432	864	32	128000/10 24 (1600)	80000/120 0	8/30000

¹ Constrained core sizes available.

² Instance is isolated to hardware dedicated to a single customer.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Eav4 and Easv4-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The Eav4-series and Easv4-series utilize AMD's 2.35Ghz EPYC™ 7452 processor in a multi-threaded configuration with up to 256MB L3 cache, increasing options for running most memory optimized workloads. The Eav4-series and Easv4-series have the same memory and disk configurations as the Ev3 & Esv3-series.

Eav4-series

ACU: 230 - 260

Premium Storage: Not Supported

Premium Storage caching: Not Supported

Eav4-series sizes are based on the 2.35Ghz AMD EPYC™ 7452 processor that can achieve a boosted maximum frequency of 3.35GHz and use premium SSD. The Eav4-series sizes are ideal for memory-intensive enterprise applications. Data disk storage is billed separately from virtual machines. To use premium SSD, use the Easv4-series sizes. The pricing and billing meters for Easv4 sizes are the same as the Eav3-series.

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT: IOPS / READ MBPS / WRITE MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_E2a_v4	2	16	50	4	3000 / 46 / 23	2 / 1000
Standard_E4a_v4	4	32	100	8	6000 / 93 / 46	2 / 2000
Standard_E8a_v4	8	64	200	16	12000 / 187 / 93	4 / 4000
Standard_E16a_v4	16	128	400	32	24000 / 375 / 187	8 / 8000
Standard_E20a_v4	20	160	500	32	30000 / 468 / 234	8 / 10000
Standard_E32a_v4	32	256	800	32	48000 / 750 / 375	8 / 16000
Standard_E48a_v4 **	48	384	1200	32		
Standard_E64a_v4 **	64	512	1600	32		
Standard_E96a_v4 **	96	672	2400	32		

** These sizes are in Preview. If you are interested in trying out these larger sizes, sign up at <https://aka.ms/AzureAMDLargeVMPreview>.

Easv4-series

ACU: 230 - 260

Premium Storage: Supported

Premium Storage caching: Supported

Easv4-series sizes are based on the 2.35Ghz AMD EPYC™ 7452 processor that can achieve a boosted maximum frequency of 3.35GHz and use premium SSD. The Easv4-series sizes are ideal for memory-intensive enterprise applications.

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS / MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS / MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_E_2as_v4	2	16	32	4	4000 / 32 (50)	3200 / 48	2 / 1000
Standard_E_4as_v4	4	32	64	8	8000 / 64 (100)	6400 / 96	2 / 2000
Standard_E_8as_v4	8	64	128	16	16000 / 128 (200)	12800 / 192	4 / 4000
Standard_E_16as_v4	16	128	256	32	32000 / 255 (400)	25600 / 384	8 / 8000
Standard_E_20as_v4	20	160	320	32	40000 / 320 (500)	32000 / 480	8 / 10000
Standard_E_32as_v4	32	256	512	32	64000 / 510 (800)	51200 / 768	8 / 16000
Standard_E_48as_v4 **	48	384	768	32			
Standard_E_64as_v4 **	64	512	1024	32			
Standard_E_96as_v4 **	96	672	1344	32			

** These sizes are in Preview. If you are interested in trying out these larger sizes, sign up at <https://aka.ms/AzureAMDLargeVMPreview>.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may

appear smaller. For example, 1023 GiB = 1098.4 GB.

- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

M-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The M-series offers a high vCPU count (up to 128 vCPUs) and a large amount of memory (up to 3.8 TiB). It's also ideal for extremely large databases or other applications that benefit from high vCPU counts and large amounts of memory. M-series sizes are based on the Intel® Xeon® CPU E7-8890 v3 @ 2.50GHz

M-series VM's feature Intel® Hyper-Threading Technology

ACU: 160-180

Premium Storage: Supported

Premium Storage caching: Supported

Write Accelerator: [Supported](#)

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_M 8ms ²	8	218.75	256	8	10000/100 (793)	5000/125	4/2000
Standard_M 16ms ²	16	437.5	512	16	20000/200 (1587)	10000/250	8/4000
Standard_M 32ts	32	192	1024	32	40000/400 (3174)	20000/500	8/8000
Standard_M 32ls	32	256	1024	32	40000/400 (3174)	20000/500	8/8000
Standard_M 32ms ²	32	875	1024	32	40000/400 (3174)	20000/500	8/8000
Standard_M 64s	64	1024	2048	64	80000/800 (6348)	40000/1000	8/16000
Standard_M 64ls	64	512	2048	64	80000/800 (6348)	40000/1000	8/16000
Standard_M 64ms ³	64	1792	2048	64	80000/800 (6348)	40000/1000	8/16000
Standard_M 128s ¹	128	2048	4096	64	160000/1600 (12696)	80000/2000	8/30000

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GiB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECT ED NETWORK BANDWIDTH (MBPS)
Standard_M 128ms ^{1,2,3}	128	3892	4096	64	160000/16 00 (12696)	80000/200 0	8/30000
Standard_M 64	64	1024	7168	64	80000/800 (1228)	40000/100 0	8/16000
Standard_M 64m	64	1792	7168	64	80000/800 (1228)	40000/100 0	8/16000
Standard_M 128 ¹	128	2048	14336	64	250000/16 00 (2456)	80000/200 0	8/32000
Standard_M 128m ¹	128	3892	14336	64	250000/16 00 (2456)	80000/200 0	8/32000

¹ More than 64 vCPU's require one of these supported guest OSes: Windows Server 2016, Ubuntu 16.04 LTS, SLES 12 SP2, and Red Hat Enterprise Linux, CentOS 7.3 or Oracle Linux 7.3 with LIS 4.2.1.

² Constrained core sizes available.

³ Instance is isolated to hardware dedicated to a single customer.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Mv2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The Mv2-series features high throughput, low latency platform running on a hyper-threaded Intel® Xeon® Platinum 8180M 2.5GHz (Skylake) processor with an all core base frequency of 2.5 GHz and a max turbo frequency of 3.8 GHz. All Mv2-series virtual machine sizes can use both standard and premium persistent disks. Mv2-series instances are memory optimized VM sizes providing unparalleled computational performance to support large in-memory databases and workloads, with a high memory-to-CPU ratio that is ideal for relational database servers, large caches, and in-memory analytics.

Mv2-series VM's feature Intel® Hyper-Threading Technology

Premium Storage: Supported

Premium Storage caching: Supported

Write Accelerator: [Supported](#)

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS / MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS / MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_M_208ms_v2 ¹	208	5700	4096	64	80000 / 800 (7040)	40000 / 1000	8 / 16000
Standard_M_208s_v2 ¹	208	2850	4096	64	80000 / 800 (7040)	40000 / 1000	8 / 16000
Standard_M_416ms_v2 ^{1, 2}	416	11400	8192	64	250000 / 1600 (14080)	80000 / 2000	8 / 32000
Standard_M_416s_v2 ^{1, 2}	416	5700	8192	64	250000 / 1600 (14080)	80000 / 2000	8 / 32000

¹ Mv2-series VMs are generation 2 only. If you're using Linux, see [Support for generation 2 VMs on Azure](#) for instructions on how to find and select an image.

² For the M416ms_v2 and M416s_v2 sizes, note that there is initial support for the following image only: "GEN2: SUSE Linux Enterprise Server (SLES) 12 SP4 for SAP Applications."

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.

- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Constrained vCPU capable VM sizes

2/28/2020 • 2 minutes to read • [Edit Online](#)

Some database workloads like SQL Server or Oracle require high memory, storage, and I/O bandwidth, but not a high core count. Many database workloads are not CPU-intensive. Azure offers certain VM sizes where you can constrain the VM vCPU count to reduce the cost of software licensing, while maintaining the same memory, storage, and I/O bandwidth.

The vCPU count can be constrained to one half or one quarter of the original VM size. These new VM sizes have a suffix that specifies the number of active vCPUs to make them easier for you to identify.

For example, the current VM size Standard_GS5 comes with 32 vCPUs, 448 GB RAM, 64 disks (up to 256 TB), and 80,000 IOPs or 2 GB/s of I/O bandwidth. The new VM sizes Standard_GS5-16 and Standard_GS5-8 comes with 16 and 8 active vCPUs respectively, while maintaining the rest of the specs of the Standard_GS5 for memory, storage, and I/O bandwidth.

The licensing fees charged for SQL Server or Oracle are constrained to the new vCPU count, and other products should be charged based on the new vCPU count. This results in a 50% to 75% increase in the ratio of the VM specs to active (billable) vCPUs. These new VM sizes allow customer workloads to use the same memory, storage, and I/O bandwidth while optimizing their software licensing cost. At this time, the compute cost, which includes OS licensing, remains the same one as the original size. For more information, see [Azure VM sizes for more cost-effective database workloads](#).

NAME	VCPU	SPECS
Standard_M8-2ms	2	Same as M8ms
Standard_M8-4ms	4	Same as M8ms
Standard_M16-4ms	4	Same as M16ms
Standard_M16-8ms	8	Same as M16ms
Standard_M32-8ms	8	Same as M32ms
Standard_M32-16ms	16	Same as M32ms
Standard_M64-32ms	32	Same as M64ms
Standard_M64-16ms	16	Same as M64ms
Standard_M128-64ms	64	Same as M128ms
Standard_M128-32ms	32	Same as M128ms
Standard_E4-2s_v3	2	Same as E4s_v3
Standard_E8-4s_v3	4	Same as E8s_v3
Standard_E8-2s_v3	2	Same as E8s_v3

NAME	VCPUs	SPECS
Standard_E16-8s_v3	8	Same as E16s_v3
Standard_E16-4s_v3	4	Same as E16s_v3
Standard_E32-16s_v3	16	Same as E32s_v3
Standard_E32-8s_v3	8	Same as E32s_v3
Standard_E64-32s_v3	32	Same as E64s_v3
Standard_E64-16s_v3	16	Same as E64s_v3
Standard_GS4-8	8	Same as GS4
Standard_GS4-4	4	Same as GS4
Standard_GS5-16	16	Same as GS5
Standard_GS5-8	8	Same as GS5
Standard_DS11-1_v2	1	Same as DS11_v2
Standard_DS12-2_v2	2	Same as DS12_v2
Standard_DS12-1_v2	1	Same as DS12_v2
Standard_DS13-4_v2	4	Same as DS13_v2
Standard_DS13-2_v2	2	Same as DS13_v2
Standard_DS14-8_v2	8	Same as DS14_v2
Standard_DS14-4_v2	4	Same as DS14_v2

Other sizes

- [Compute optimized](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU](#)
- [High performance compute](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Storage optimized virtual machine sizes

2/28/2020 • 2 minutes to read • [Edit Online](#)

Storage optimized VM sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis. This article provides information about the number of vCPUs, data disks, and NICs as well as local storage throughput and network bandwidth for each optimized size.

The [Lsv2-series](#) features high throughput, low latency, directly mapped local NVMe storage running on the [AMD EPYC™ 7551 processor](#) with an all core boost of 2.55GHz and a max boost of 3.0GHz. The Lsv2-series VMs come in sizes from 8 to 80 vCPU in a simultaneous multi-threading configuration. There is 8 GiB of memory per vCPU, and one 1.92TB NVMe SSD M.2 device per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s v2.

Other sizes

- [General purpose](#)
- [Compute optimized](#)
- [Memory optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Learn how to optimize performance on the Lsv2-series virtual machines for [Windows](#) or [Linux](#).

Lsv2-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The Lsv2-series features high throughput, low latency, directly mapped local NVMe storage running on the [AMD EPYC™ 7551 processor](#) with an all core boost of 2.55GHz and a max boost of 3.0GHz. The Lsv2-series VMs come in sizes from 8 to 80 vCPU in a simultaneous multi-threading configuration. There is 8 GiB of memory per vCPU, and one 1.92TB NVMe SSD M.2 device per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s v2.

NOTE

The Lsv2-series VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks. This allows for greater IOPs / throughput for your workloads. The Lsv2 and Ls-series do not support the creation of a local cache to increase the IOPs achievable by durable data disks.

The high throughput and IOPs of the local disk makes the Lsv2-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB which replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

To learn more, see Optimize performance on the Lsv2-series virtual machines for [Windows](#) or [Linux](#).

ACU: 150-175

Premium Storage: Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY (GiB)	TEMP DISK ¹ (GiB)	NVME DISKS ²	NVME DISK THROUGHPUT ³ (READ IOPS/MBPS)	MAX UNCACHE D DATA DISK THROUGHPUT (IOPS/MBPS) ⁴	MAX DATA DISKS	MAX NICs / EXPECTED NETWORK BANDWIDTH (Mbps)
Standard_L8s_v2	8	64	80	1x1.92 TB	400000/2000	8000/160	16	2 / 3200
Standard_L16s_v2	16	128	160	2x1.92 TB	800000/4000	16000/320	32	4 / 6400
Standard_L32s_v2	32	256	320	4x1.92 TB	1.5M/8000	32000/640	32	8 / 12800
Standard_L48s_v2	48	384	480	6x1.92 TB	2.2M/14000	48000/960	32	8 / 16000+
Standard_L64s_v2	64	512	640	8x1.92 TB	2.9M/16000	64000/1280	32	8 / 16000+
Standard_L80s_v2 ⁵	80	640	800	10x1.92TB	3.8M/20000	80000/1400	32	8 / 16000+

¹ Lsv2-series VMs have a standard SCSI based temp resource disk for OS paging/swap file use (D: on Windows, /dev/sdb on Linux). This disk provides 80 GiB of storage, 4,000 IOPS, and 80 MBps transfer rate for every 8

vCPUs (e.g. Standard_L80s_v2 provides 800 GiB at 40,000 IOPS and 800 MBPS). This ensures the NVMe drives can be fully dedicated to application use. This disk is Ephemeral, and all data will be lost on stop/deallocate.

² Local NVMe Disks are ephemeral, data will be lost on these disks if you stop/deallocate your VM.

³ Hyper-V NVMe Direct technology provides unthrottled access to local NVMe drives mapped securely into the guest VM space. Achieving maximum performance requires using either the latest WS2019 build or Ubuntu 18.04 or 16.04 from the Azure Marketplace. Write performance varies based on IO size, drive load, and capacity utilization.

⁴ Lsv2-series VMs do not provide host cache for data disk as it does not benefit the Lsv2 workloads. However, Lsv2 VMs can accommodate Azure's Ephemeral VM OS disk option (up to 30 GiB).

⁵ VMs with more than 64 vCPUs require one of these supported guest operating systems:

- Windows Server 2016 or later
- Ubuntu 16.04 LTS or later, with Azure tuned kernel (4.15 kernel or later)
- SLES 12 SP2 or later
- RHEL or CentOS version 6.7 through 6.10, with Microsoft-provided LIS package 4.3.1 (or later) installed
- RHEL or CentOS version 7.3, with Microsoft-provided LIS package 4.2.1 (or later) installed
- RHEL or CentOS version 7.6 or later
- Oracle Linux with UEK4 or later
- Debian 9 with the backports kernel, Debian 10 or later
- CoreOS with a 4.14 kernel or later

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When comparing disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- If you want to get the best performance for your VMs, you should limit the number of data disks to 2 disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated [bandwidth allocated per VM type](#) across all NICs, for all destinations. Upper limits are not guaranteed, but are intended to provide guidance for selecting the right VM type for the intended application. Actual network performance will depend on a variety of factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimizing network throughput for Windows and Linux](#). To achieve the expected network performance on Linux or Windows, it may be necessary to select a specific version or optimize your VM. For more information, see [How to reliably test for virtual machine throughput](#).

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Optimize performance on the Lsv2-series virtual machines

2/28/2020 • 5 minutes to read • [Edit Online](#)

Lsv2-series virtual machines support a variety of workloads that need high I/O and throughput on local storage across a wide range of applications and industries. The Lsv2-series is ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases, including Cassandra, MongoDB, Cloudera, and Redis.

The design of the Lsv2-series Virtual Machines (VMs) maximizes the AMD EPYC™ 7551 processor to provide the best performance between the processor, memory, NVMe devices, and the VMs. In addition to maximizing the hardware performance, Lsv2-series VMs are designed to work with the needs of Windows and Linux operating systems for better performance with the hardware and the software.

Tuning the software and hardware resulted in the optimized version of [Windows Server 2019 Datacenter](#), released in early December 2018 to the Azure Marketplace, which supports maximum performance on the NVMe devices in Lsv2-series VMs.

This article provides tips and suggestions to ensure your workloads and applications achieve the maximum performance designed into the VMs. The information on this page will be continuously updated as more Lsv2 optimized images are added to the Azure Marketplace.

AMD EYPC™ chipset architecture

Lsv2-series VMs use AMD EYPC™ server processors based on the Zen microarchitecture. AMD developed Infinity Fabric (IF) for EYPC™ as scalable interconnect for its NUMA model that could be used for on-die, on-package, and multi-package communications. Compared with QPI (Quick-Path Interconnect) and UPI (Ultra-Path Interconnect) used on Intel modern monolithic-die processors, AMD's many-NUMA small-die architecture may bring both performance benefits as well as challenges. The actual impact of memory bandwidth and latency constraints could vary depending on the type of workloads running.

Tips for maximizing performance

- The hardware that powers the Lsv2-series VMs utilizes NVMe devices with eight I/O Queue Pairs (QP)s. Every NVMe device I/O queue is actually a pair: a submission queue and a completion queue. The NVMe driver is set up to optimize the utilization of these eight I/O QPs by distributing I/O's in a round robin schedule. To gain max performance, run eight jobs per device to match.
- Avoid mixing NVMe admin commands (for example, NVMe SMART info query, etc.) with NVMe I/O commands during active workloads. Lsv2 NVMe devices are backed by Hyper-V NVMe Direct technology, which switches into "slow mode" whenever any NVMe admin commands are pending. Lsv2 users could see a dramatic performance drop in NVMe I/O performance if that happens.
- Lsv2 users should not rely on device NUMA information (all 0) reported from within the VM for data drives to decide the NUMA affinity for their apps. The recommended way for better performance is to spread workloads across CPUs if possible.
- The maximum supported queue depth per I/O queue pair for Lsv2 VM NVMe device is 1024 (vs. Amazon i3 QD 32 limit). Lsv2 users should limit their (synthetic) benchmarking workloads to queue depth 1024 or lower to avoid triggering queue full conditions, which can reduce performance.

Utilizing local NVMe storage

Local storage on the 1.92 TB NVMe disk on all Lsv2 VMs is ephemeral. During a successful standard reboot of the VM, the data on the local NVMe disk will persist. The data will not persist on the NVMe if the VM is redeployed, de-allocated, or deleted. Data will not persist if another issue causes the VM, or the hardware it is running on, to become unhealthy. When this happens, any data on the old host is securely erased.

There will also be cases when the VM needs to be moved to a different host machine, for example, during a planned maintenance operation. Planned maintenance operations and some hardware failures can be anticipated with [Scheduled Events](#). Scheduled Events should be used to stay updated on any predicted maintenance and recovery operations.

In the case that a planned maintenance event requires the VM to be recreated on a new host with empty local disks, the data will need to be resynchronized (again, with any data on the old host being securely erased). This occurs because Lsv2-series VMs do not currently support live migration on the local NVMe disk.

There are two modes for planned maintenance.

Standard VM customer-controlled maintenance

- The VM is moved to an updated host during a 30-day window.
- Lsv2 local storage data could be lost, so backing-up data prior to the event is recommended.

Automatic maintenance

- Occurs if the customer does not execute customer-controlled maintenance, or in the event of emergency procedures such as a security zero-day event.
- Intended to preserve customer data, but there is a small risk of a VM freeze or reboot.
- Lsv2 local storage data could be lost, so backing-up data prior to the event is recommended.

For any upcoming service events, use the controlled maintenance process to select a time most convenient to you for the update. Prior to the event, you may back up your data in premium storage. After the maintenance event completes, you can return your data to the refreshed Lsv2 VMs local NVMe storage.

Scenarios that maintain data on local NVMe disks include:

- The VM is running and healthy.
- The VM is rebooted in place (by you or Azure).
- The VM is paused (stopped without de-allocation).
- The majority of the planned maintenance servicing operations.

Scenarios that securely erase data to protect the customer include:

- The VM is redeployed, stopped (de-allocated), or deleted (by you).
- The VM becomes unhealthy and has to service heal to another node due to a hardware issue.
- A small number of the planned maintenance servicing operations that requires the VM to be reallocated to another host for servicing.

To learn more about options for backing up data in local storage, see [Backup and disaster recovery for Azure IaaS disks](#).

Frequently asked questions

• How do I start deploying Lsv2-series VMs?

Much like any other VM, use the [Portal](#), [Azure CLI](#), or [PowerShell](#) to create a VM.

• Will a single NVMe disk failure cause all VMs on the host to fail?

If a disk failure is detected on the hardware node, the hardware is in a failed state. When this occurs, all VMs

on the node are automatically de-allocated and moved to a healthy node. For Lsv2-series VMs, this means that the customer's data on the failing node is also securely erased and will need to be recreated by the customer on the new node. As noted, before live migration becomes available on Lsv2, the data on the failing node will be proactively moved with the VMs as they are transferred to another node.

- **Do I need to make polling adjustments in Windows in Windows Server 2012 or Windows Server 2016?**

NVMe polling is only available on Windows Server 2019 on Azure.

- **Can I switch back to a traditional interrupt service routine (ISR) model?**

Lsv2-series VMs are optimized for NVMe polling. Updates are continuously provided to improve polling performance.

- **Can I adjust the polling settings in Windows Server 2019?**

The polling settings are not user adjustable.

Next steps

- See specifications for all VMs optimized for storage performance on Azure

2 minutes to read

NC-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

NC-series VMs are powered by the [NVIDIA Tesla K80](#) card and the Intel Xeon E5-2690 v3 (Haswell) processor. Users can crunch through data faster by leveraging CUDA for energy exploration applications, crash simulations, ray traced rendering, deep learning, and more. The NC24r configuration provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads.

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	GPU	GPU MEMORY: GiB	MAX DATA DISKS	MAX NICs
Standard_N_C6	6	56	340	1	12	24	1
Standard_N_C12	12	112	680	2	24	48	2
Standard_N_C24	24	224	1440	4	48	64	4
Standard_N_C24r*	24	224	1440	4	48	64	4

1 GPU = one-half K80 card.

*RDMA capable

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or

Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

NCv2-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

NCv2-series VMs are powered by [NVIDIA Tesla P100](#) GPUs. These GPUs can provide more than 2x the computational performance of the NC-series. Customers can take advantage of these updated GPUs for traditional HPC workloads such as reservoir modeling, DNA sequencing, protein analysis, Monte Carlo simulations, and others. In addition to the GPUs, the NCv2-series VMs are also powered by Intel Xeon E5-2690 v4 (Broadwell) CPUs.

The NC24rs v2 configuration provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads.

Premium Storage: Supported

Premium Storage caching: Supported

IMPORTANT

For this VM series, the vCPU (core) quota in your subscription is initially set to 0 in each region. [Request a vCPU quota increase](#) for this series in an [available region](#).

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	GPU	GPU MEMORY: GiB	MAX DATA DISKS	MAX UNCACHE D DISK THROUGH PUT: IOPS/MBPS	MAX NICs
Standard_NC6s_v2	6	112	736	1	16	12	20000/200	4
Standard_NC12s_v2	12	224	1474	2	32	24	40000/400	8
Standard_NC24s_v2	24	448	2948	4	64	32	80000/800	8
Standard_NC24rs_v2*	24	448	2948	4	64	32	80000/800	8

1 GPU = one P100 card.

*RDMA capable

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.

- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

NCv3-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

NCv3-series VMs are powered by [NVIDIA Tesla V100](#) GPUs. These GPUs can provide 1.5x the computational performance of the NCv2-series. Customers can take advantage of these updated GPUs for traditional HPC workloads such as reservoir modeling, DNA sequencing, protein analysis, Monte Carlo simulations, and others. The NC24rs v3 configuration provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads. In addition to the GPUs, the NCv3-series VMs are also powered by Intel Xeon E5-2690 v4 (Broadwell) CPUs.

Premium Storage: Supported

Premium Storage caching: Supported

IMPORTANT

For this VM series, the vCPU (core) quota in your subscription is initially set to 0 in each region. [Request a vCPU quota increase](#) for this series in an [available region](#).

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	GPU	GPU MEMORY: GIB	MAX DATA DISKS	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs
Standard_NC6s_v3	6	112	736	1	16	12	20000/200	4
Standard_NC12s_v3	12	224	1474	2	32	24	40000/400	8
Standard_NC24s_v3	24	448	2948	4	64	32	80000/800	8
Standard_NC24rs_v3*	24	448	2948	4	64	32	80000/800	8

1 GPU = one V100 card.

*RDMA capable

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode

is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.

- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

ND-series

2/28/2020 • 3 minutes to read • [Edit Online](#)

The ND-series virtual machines are a new addition to the GPU family designed for AI, and Deep Learning workloads. They offer excellent performance for training and inference. ND instances are powered by [NVIDIA Tesla P40](#) GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. These instances provide excellent performance for single-precision floating point operations, for AI workloads utilizing Microsoft Cognitive Toolkit, TensorFlow, Caffe, and other frameworks. The ND-series also offers a much larger GPU memory size (24 GB), enabling to fit much larger neural net models. Like the NC-series, the ND-series offers a configuration with a secondary low-latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

Premium Storage: Supported

Premium Storage caching: Supported

IMPORTANT

For this VM series, the vCPU (core) quota per region in your subscription is initially set to 0. [Request a vCPU quota increase](#) for this series in an [available region](#).

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	GPU	GPU MEMORY: GIB	MAX DATA DISKS	MAX UNCACHE D DISK THROUGHPUT: IOPS/MBPS	MAX NICs
Standard_ND6s	6	112	736	1	24	12	20000/200	4
Standard_ND12s	12	224	1474	2	48	24	40000/400	8
Standard_ND24s	24	448	2948	4	96	32	80000/800	8
Standard_ND24rs*	24	448	2948	4	96	32	80000/800	8

1 GPU = one P40 card.

*RDMA capable

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.

- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Updated NDv2-series (Preview)

2/28/2020 • 3 minutes to read • [Edit Online](#)

The NDv2-series virtual machine is a new addition to the GPU family designed for the needs of the most demanding GPU-accelerated AI, machine learning, simulation, and HPC workloads.

NDv2 is powered by 8 NVIDIA Tesla V100 NVLINK-connected GPUs, each with 32 GB of GPU memory. Each NDv2 VM also has 40 non-HyperThreaded Intel Xeon Platinum 8168 (Skylake) cores and 672 GiB of system memory.

NDv2 instances provide excellent performance for HPC and AI workloads utilizing CUDA GPU-optimized computation kernels, and the many AI, ML, and analytics tools that support GPU acceleration 'out-of-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks.

Critically, the NDv2 is built for both computationally intense scale-up (harnessing 8 GPUs per VM) and scale-out (harnessing multiple VMs working together) workloads. The NDv2 series now supports 100-Gigabit InfiniBand EDR backend networking, similar to that available on the HB series of HPC VM, to allow high-performance clustering for parallel scenarios including distributed training for AI and ML. This backend network supports all major InfiniBand protocols, including those employed by NVIDIA's NCCL2 libraries, allowing for seamless clustering of GPUs.

NOTE

When enabling [InfiniBand](#) on the ND40rs_v2 VM, please use the 4.7-1.0.0.1 Mellanox OFED driver.

Due to increased GPU memory, the new ND40rs_v2 VM requires the use of [Generation 2 VMs](#) and marketplace images.

[Sign-up to request early access to the NDv2 virtual machine preview.](#)

Please note: The ND40s_v2 featuring 16 GB of per-GPU memory is no longer available for preview and has been superceded by the updated ND40rs_v2.

Premium Storage: Supported

Premium Storage caching: Supported

InfiniBand: Supported

SIZE	VCPU	MEMORY : GIB	TEMP STORAGE (SSD): GIB	GPU	GPU MEMORY : GIB	MAX DATA DISKS	MAX UNCACHED DISK THROUROUGHPUT: IOPS / MBPS	MAX NETWORK BANDWIDTH: DTH	MAX NICs
Standard_ND40rs_v2	40	672	2948	8 V100 32 GB (NVLink)	16	32	80000 / 800	24000 Mbps	8

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may

appear smaller. For example, 1023 GiB = 1098.4 GB.

- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

NV-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The NV-series virtual machines are powered by [NVIDIA Tesla M60](#) GPUs and NVIDIA GRID technology for desktop accelerated applications and virtual desktops where customers are able to visualize their data or simulations. Users are able to visualize their graphics intensive workflows on the NV instances to get superior graphics capability and additionally run single precision workloads such as encoding and rendering. NV-series VMs are also powered by Intel Xeon E5-2690 v3 (Haswell) CPUs.

Each GPU in NV instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY : GIB	TEMP STORAGE (SSD) GIB	GPU	GPU MEMORY : GIB	MAX DATA DISKS	MAX NICS	VIRTUAL WORKSTATIONS	VIRTUAL APPLICATIONS
Standard_NV6	6	56	340	1	8	24	1	1	25
Standard_NV12	12	112	680	2	16	48	2	2	50
Standard_NV24	24	224	1440	4	32	64	4	4	100

1 GPU = one-half M60 card.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize](#)

network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

NVv3-series

2/28/2020 • 2 minutes to read • [Edit Online](#)

The NVv3-series virtual machines are powered by [NVIDIA Tesla M60](#) GPUs and NVIDIA GRID technology with Intel E5-2690 v4 (Broadwell) CPUs and Intel Hyper-Threading Technology. These virtual machines are targeted for GPU accelerated graphics applications and virtual desktops where customers want to visualize their data, simulate results to view, work on CAD, or render and stream content. Additionally, these virtual machines can run single precision workloads such as encoding and rendering. NVv3 virtual machines support Premium Storage and come with twice the system memory (RAM) when compared with its predecessor NV-series.

Each GPU in NVv3 instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

Premium Storage caching: Supported

SIZE	vCPU	MEMORY: GiB	TEMP STORAGE (SSD) GiB	GPU	GPU MEMORY: GiB	MAX DATA DISKS	MAX UNCACHED DISK THROUGHPUT: IOPS/Mbps	MAX NICs	VIRTUAL WORKSTATIONS	VIRTUAL APPLICATIONS
Standard_NV1_2s_v3	12	112	320	1	8	12	20000/200	4	1	25
Standard_NV2_4s_v3	24	224	640	2	16	24	40000/400	8	2	50
Standard_NV4_8s_v3	48	448	1280	4	32	32	80000/800	8	4	100

1 GPU = one-half M60 card.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all

NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs, NVIDIA GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) or [N-series GPU driver setup for Linux](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

NVv4-series (Preview)

2/28/2020 • 2 minutes to read • [Edit Online](#)

The NVv4-series virtual machines are powered by [AMD Radeon Instinct MI25](#) GPUs and AMD EPYC 7V12(Rome) CPUs. With NVv4-series Azure is introducing virtual machines with partial GPUs. Pick the right sized virtual machine for GPU accelerated graphics applications and virtual desktops starting at 1/8th of a GPU with 2 GiB frame buffer to a full GPU with 16 GiB frame buffer. NVv4 virtual machines currently support only Windows guest operating system.

[Sign-up and get access to these machines during preview.](#)

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GiB	TEMP STORAGE (SSD) GiB	GPU	GPU MEMORY: GiB	MAX DATA DISKS	MAX NICs
Standard_N_V4as_v4	4	14	88	1/8	2	4	2
Standard_N_V8as_v4	8	28	176	1/4	4	8	4
Standard_N_V16as_v4	16	56	352	1/2	8	16	8
Standard_N_V32as_v4	32	112	704	1	16	32	8

¹ NVv4-series VMs feature AMD Simultaneous multithreading Technology

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize](#)

network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Supported operating systems and drivers

To take advantage of the GPU capabilities of Azure N-series VMs running Windows, NVIDIA or AMD GPU drivers must be installed.

The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on a Windows N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps. For general information about VM extensions, see [Azure virtual machine extensions and features](#).

If you choose to install NVIDIA GPU drivers manually, see [N-series GPU driver setup for Windows](#) for supported operating systems, drivers, installation, and verification steps.

To install AMD GPU drivers manually, see [N-series AMD GPU driver setup for Windows](#) for supported operating systems, drivers, installation, and verification steps.

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Install NVIDIA GPU drivers on N-series VMs running Windows

2/28/2020 • 3 minutes to read • [Edit Online](#)

To take advantage of the GPU capabilities of Azure N-series VMs running Windows, NVIDIA GPU drivers must be installed. The [NVIDIA GPU Driver Extension](#) installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the [NVIDIA GPU Driver Extension documentation](#) for supported operating systems and deployment steps.

If you choose to install GPU drivers manually, this article provides supported operating systems, drivers, and installation and verification steps. Manual driver setup information is also available for [Linux VMs](#).

For basic specs, storage capacities, and disk details, see [GPU Windows VM sizes](#).

Supported operating systems and drivers

NVIDIA Tesla (CUDA) drivers

NVIDIA Tesla (CUDA) drivers for NC, NCv2, NCv3, ND, and NDv2-series VMs (optional for NV-series) are supported only on the operating systems listed in the following table. Driver download links are current at time of publication. For the latest drivers, visit the [NVIDIA](#) website.

TIP

As an alternative to manual CUDA driver installation on a Windows Server VM, you can deploy an Azure [Data Science Virtual Machine](#) image. The DSVM editions for Windows Server 2016 pre-install NVIDIA CUDA drivers, the CUDA Deep Neural Network Library, and other tools.

OS	Driver
Windows Server 2016	398.75 (.exe)
Windows Server 2012 R2	398.75 (.exe)

NVIDIA GRID drivers

Microsoft redistributes NVIDIA GRID driver installers for NV and NVv3-series VMs used as virtual workstations or for virtual applications. Install only these GRID drivers on Azure NV-series VMs, only on the operating systems listed in the following table. These drivers include licensing for GRID Virtual GPU Software in Azure. You do not need to set up a NVIDIA vGPU software license server.

Please note that the Nvidia extension will always install the latest driver. We provide links to the previous version here for customers, who have dependency on an older version.

For Windows Server 2019, Windows Server 2016, and Windows 10(up to build 1909):

- [GRID 10.1 \(442.06\) \(.exe\)](#)
- [GRID 10.0 \(441.66\) \(.exe\)](#)

For Windows Server 2012 R2, Windows Server 2008 R2, Windows 8, and Windows 7:

- [GRID 10.1 \(442.06\) \(.exe\)](#)
- [GRID 10.0 \(441.66\) \(.exe\)](#)

Driver installation

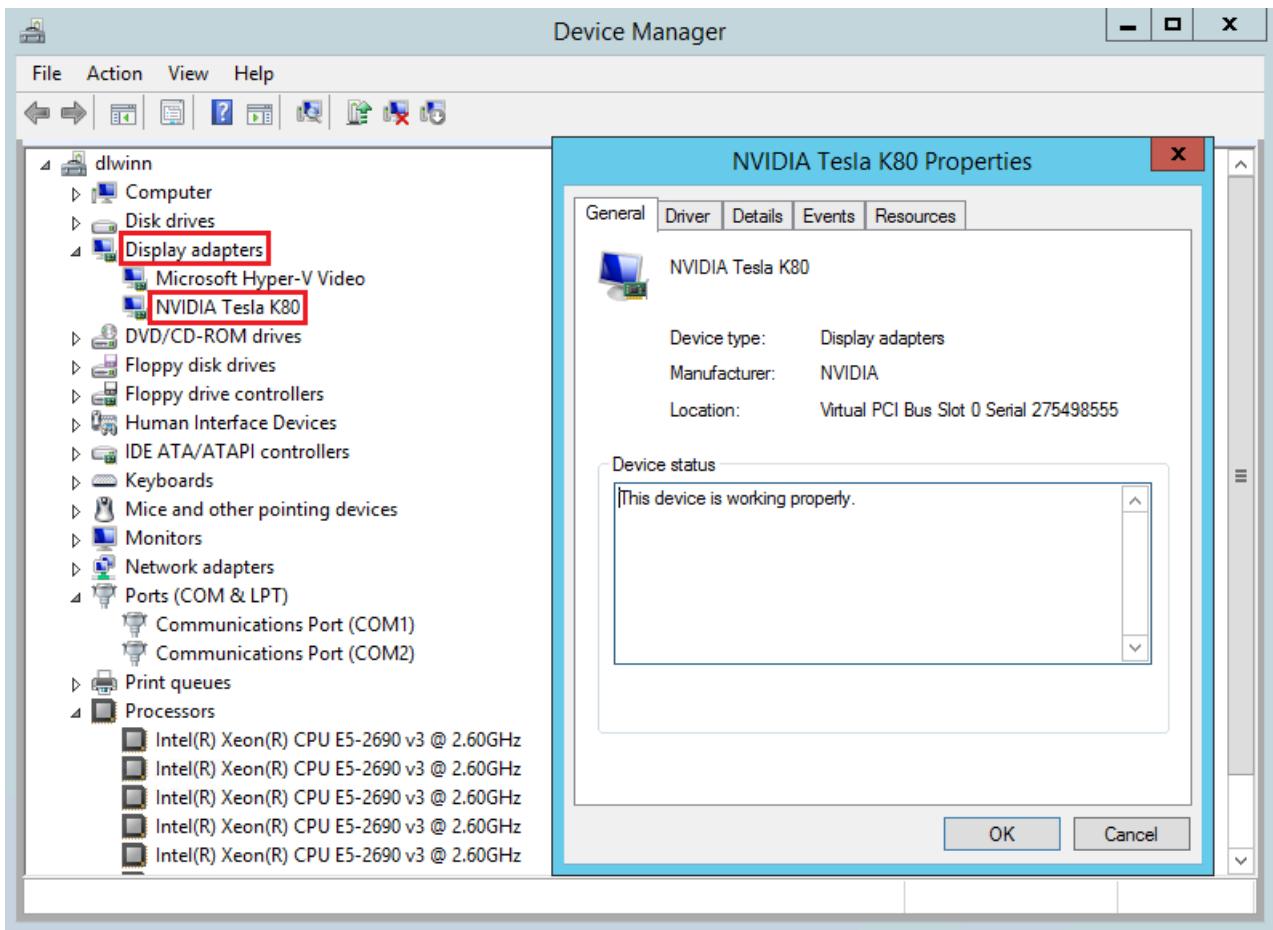
1. Connect by Remote Desktop to each N-series VM.
2. Download, extract, and install the supported driver for your Windows operating system.

After GRID driver installation on a VM, a restart is required. After CUDA driver installation, a restart is not required.

Verify driver installation

Please note that the Nvidia Control panel is only accessible with the GRID driver installation. If you have installed CUDA drivers then the Nvidia control panel will not be visible.

You can verify driver installation in Device Manager. The following example shows successful configuration of the Tesla K80 card on an Azure NC VM.



To query the GPU device state, run the `nvidia-smi` command-line utility installed with the driver.

1. Open a command prompt and change to the **C:\Program Files\NVIDIA Corporation\NVSMI** directory.
2. Run `nvidia-smi`. If the driver is installed, you will see output similar to the following. The **GPU-Util** shows **0%** unless you are currently running a GPU workload on the VM. Your driver version and GPU details may be different from the ones shown.

```
C:\Program Files\NVIDIA Corporation\NVSMI>nvidia-smi
Wed Nov 23 20:49:33 2016
+-----+-----+-----+
| NVIDIA-SMI 369.73 | Driver Version: 369.73 |
+-----+-----+-----+
| GPU  Name    TCC/WDDM | Bus-Id     Disp.A  | Volatile Uncorr. ECC |
| Fan  Temp   Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+-----+-----+-----+-----+
| 0  Tesla K80    TCC | B794:00:00.0  Off |        0MiB / 11423MiB |      0%       Default |
| N/A   56C   P8    28W / 149W |                |               |             |
+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                               GPU Memory |
| GPU PID  Type  Process name        Usage  |
|-----+-----+-----+-----|
| No running processes found            |
+-----+
```

RDMA network connectivity

RDMA network connectivity can be enabled on RDMA-capable N-series VMs such as NC24r deployed in the same availability set or in a single placement group in a virtual machine scale set. The HpcVmDrivers extension must be added to install Windows network device drivers that enable RDMA connectivity. To add the VM extension to an RDMA-enabled N-series VM, use [Azure PowerShell](#) cmdlets for Azure Resource Manager.

To install the latest version 1.1 HpcVMDRivers extension on an existing RDMA-capable VM named myVM in the West US region:

```
Set-AzVMExtension -ResourceGroupName "myResourceGroup" -Location "westus" -VMName "myVM" -ExtensionName "HpcVmDrivers" -Publisher "Microsoft.HpcCompute" -Type "HpcVmDrivers" -TypeHandlerVersion "1.1"
```

For more information, see [Virtual machine extensions and features for Windows](#).

The RDMA network supports Message Passing Interface (MPI) traffic for applications running with [Microsoft MPI](#) or Intel MPI 5.x.

Next steps

- Developers building GPU-accelerated applications for the NVIDIA Tesla GPUs can also download and install the latest [CUDA Toolkit](#). For more information, see the [CUDA Installation Guide](#).

Install AMD GPU drivers on N-series VMs running Windows

2/12/2020 • 2 minutes to read • [Edit Online](#)

To take advantage of the GPU capabilities of the new Azure NVv4 series VMs running Windows, AMD GPU drivers must be installed. The AMD driver extension will be available in the coming weeks. This article provides supported operating systems, drivers, and manual installation and verification steps.

For basic specs, storage capacities, and disk details, see [GPU Windows VM sizes](#).

Supported operating systems and drivers

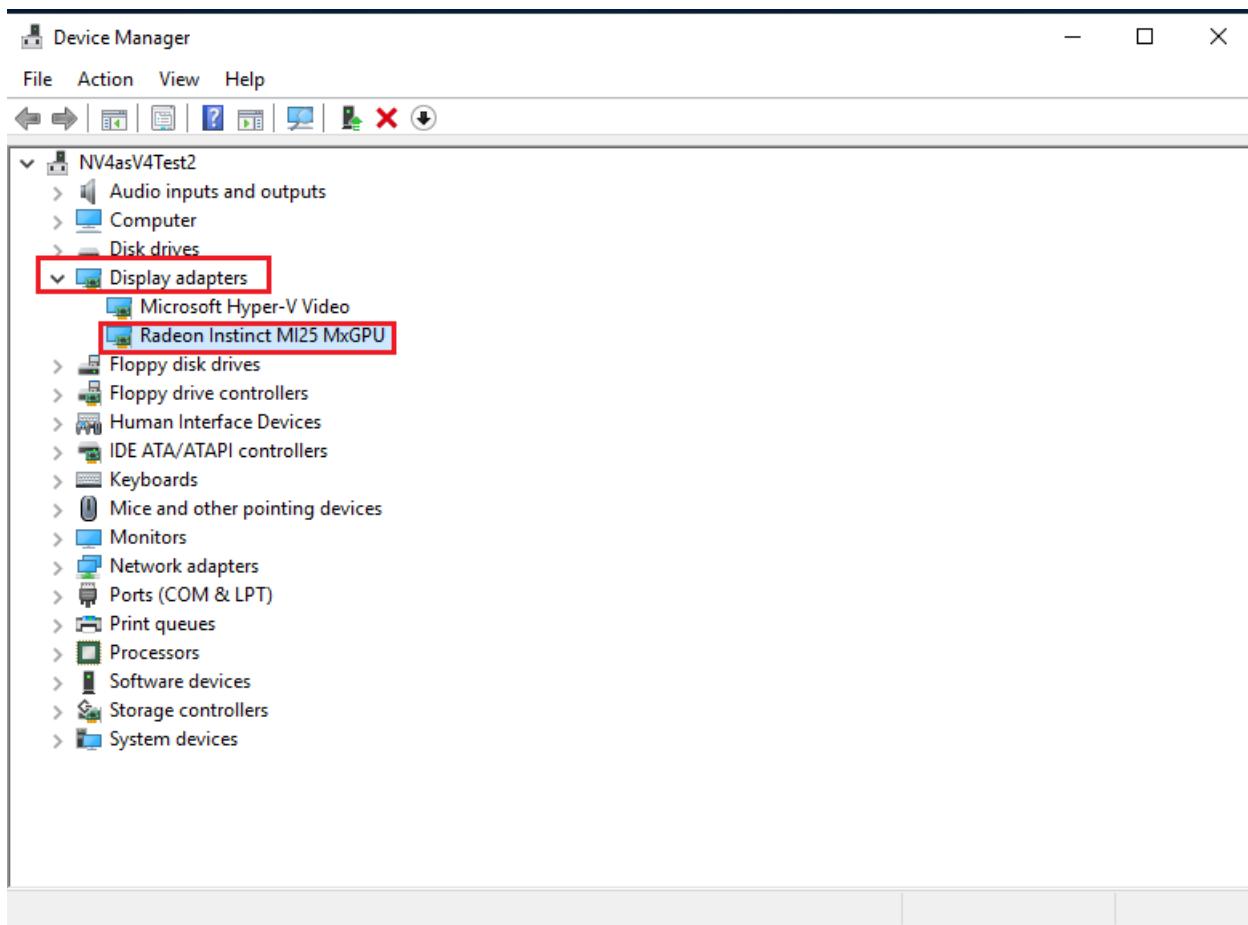
os	DRIVER
Windows 10 EVD - Build 1903	19.Q4.1 (.exe)
Windows 10 - Build 1809	
Windows Server 2016	
Windows Server 2019	

Driver installation

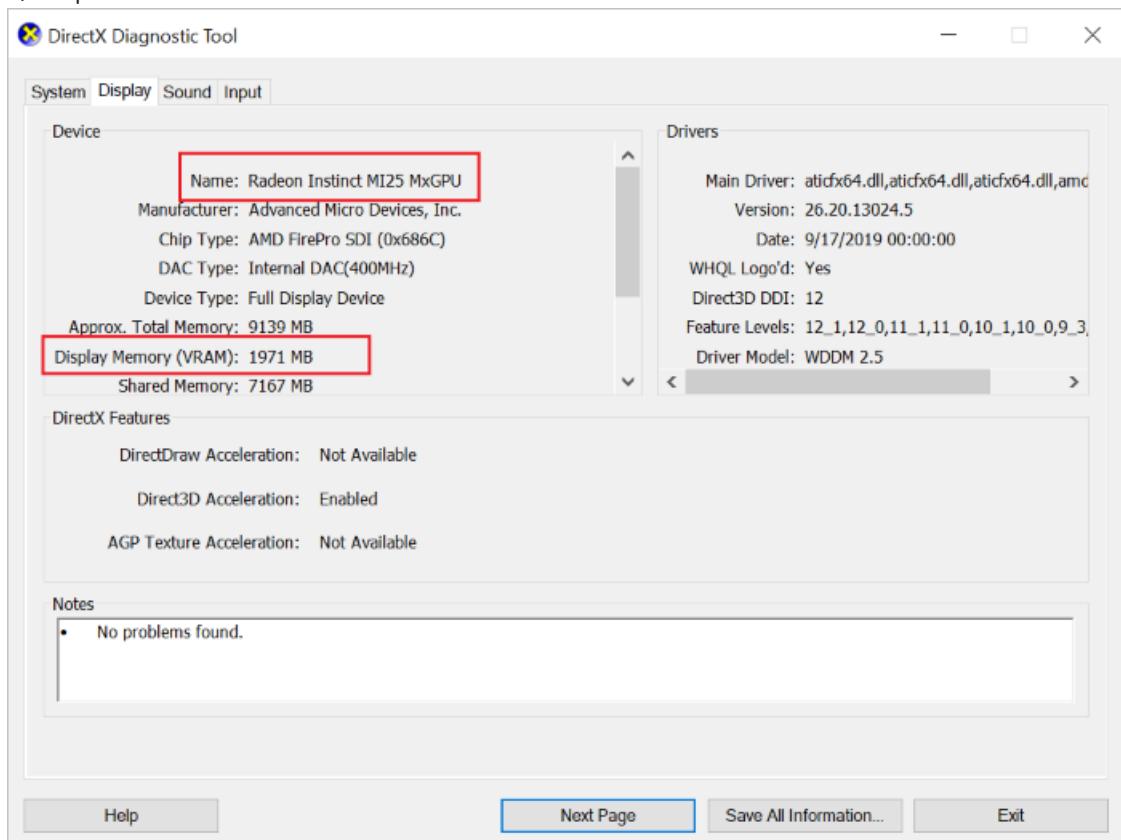
1. Connect by Remote Desktop to each NVv4-series VM.
2. Download and extract the driver setup files. Navigate to the folder and run 'setup.exe' to install the supported driver for your Windows operating system.

Verify driver installation

You can verify driver installation in Device Manager. The following example shows successful configuration of the Radeon Instinct MI25 card on an Azure NVv4 VM.



You can use dxdiag to verify the GPU display properties including the video RAM. The following example shows a 1/8th partition of the Radeon Instinct MI25 card on an Azure NVv4 VM.



High performance compute VM sizes

2/27/2020 • 6 minutes to read • [Edit Online](#)

Azure H-series virtual machines (VMs) are designed to deliver leadership-class performance, MPI scalability, and cost efficiency for a variety of real-world HPC workloads.

HBv2-series VMs feature 200 Gb/sec Mellanox HDR InfiniBand, while both HB and HC-series VMs feature 100 Gb/sec Mellanox EDR InfiniBand. Each of these VM types are connected in a non-blocking fat tree for optimized and consistent RDMA performance. HBv2 VMs support Adaptive Routing and the Dynamic Connected Transport (DCT, in addition to standard RC and UD transports). These features enhance application performance, scalability, and consistency, and usage of them is strongly recommended.

HB-series VMs are optimized for applications driven by memory bandwidth, such as fluid dynamics, explicit finite element analysis, and weather modeling. HB VMs feature 60 AMD EPYC 7551 processor cores, 4 GB of RAM per CPU core, and no hyperthreading. The AMD EPYC platform provides more than 260 GB/sec of memory bandwidth.

HC-series VMs are optimized for applications driven by dense computation, such as implicit finite element analysis, molecular dynamics, and computational chemistry. HC VMs feature 44 Intel Xeon Platinum 8168 processor cores, 8 GB of RAM per CPU core, and no hyperthreading. The Intel Xeon Platinum platform supports Intel's rich ecosystem of software tools such as the Intel Math Kernel Library.

H-series VMs are optimized for applications driven by high CPU frequencies or large memory per core requirements. H-series VMs feature 8 or 16 Intel Xeon E5 2667 v3 processor cores, 7 or 14 GB of RAM per CPU core, and no hyperthreading. H-series features 56 Gb/sec Mellanox FDR InfiniBand in a non-blocking fat tree configuration for consistent RDMA performance. H-series VMs support Intel MPI 5.x and MS-MPI.

Deployment considerations

- **Azure subscription** – To deploy more than a few compute-intensive instances, consider a pay-as-you-go subscription or other purchase options. If you're using an [Azure free account](#), you can use only a limited number of Azure compute cores.
- **Pricing and availability** - These VM sizes are offered only in the Standard pricing tier. Check [Products available by region](#) for availability in Azure regions.
- **Cores quota** – You might need to increase the cores quota in your Azure subscription from the default value. Your subscription might also limit the number of cores you can deploy in certain VM size families, including the H-series. To request a quota increase, [open an online customer support request](#) at no charge. (Default limits may vary depending on your subscription category.)

NOTE

Contact Azure Support if you have large-scale capacity needs. Azure quotas are credit limits, not capacity guarantees. Regardless of your quota, you are only charged for cores that you use.

- **Virtual network** – An Azure [virtual network](#) is not required to use the compute-intensive instances. However, for many deployments you need at least a cloud-based Azure virtual network, or a site-to-site connection if you need to access on-premises resources. When needed, create a new virtual network to deploy the instances. Adding compute-intensive VMs to a virtual network in an affinity group is not supported.

- **Resizing** – Because of their specialized hardware, you can only resize compute-intensive instances within the same size family (H-series or compute-intensive A-series). For example, you can only resize an H-series VM from one H-series size to another. In addition, resizing from a non-compute-intensive size to a compute-intensive size is not supported.

RDMA-capable instances

A subset of the compute-intensive instances (A8, A9, H16r, H16mr, HB and HC) feature a network interface for remote direct memory access (RDMA) connectivity. Selected N-series sizes designated with 'r' such as the NC24rs configurations (NC24rs_v2 and NC24rs_v3) are also RDMA-capable. This interface is in addition to the standard Azure network interface available to other VM sizes.

This interface allows the RDMA-capable instances to communicate over an InfiniBand (IB) network, operating at EDR rates for HB, HC, FDR rates for H16r, H16mr, and RDMA-capable N-series virtual machines, and QDR rates for A8 and A9 virtual machines. These RDMA capabilities can boost the scalability and performance of certain Message Passing Interface (MPI) applications. For more information on speed, see the details in the tables on this page.

NOTE

In Azure, IP over IB is only supported on the SR-IOV enabled VMs (SR-IOV for InfiniBand, currently HB and HC). RDMA over IB is supported for all RDMA-capable instances.

- **Operating system** - Windows Server 2016 on all the above HPC series VMs. Windows Server 2012 R2, Windows Server 2012 are also supported on the non-SR-IOV enabled VMs (hence excluding HB and HC).
- **MPI** - The SR-IOV enabled VM sizes on Azure (HB, HC) allow almost any flavor of MPI to be used with Mellanox OFED. On non-SR-IOV enabled VMs, supported MPI implementations use the Microsoft Network Direct (ND) interface to communicate between instances. Hence, only Microsoft MPI (MS-MPI) 2012 R2 or later and Intel MPI 5.x versions are supported. Later versions (2017, 2018) of the Intel MPI runtime library may or may not be compatible with the Azure RDMA drivers.
- **InfiniBandDriverWindows VM extension** - On RDMA-capable VMs, add the InfiniBandDriverWindows extension to enable InfiniBand. This Windows VM extension installs Windows Network Direct drivers (on non-SR-IOV VMs) or Mellanox OFED drivers (on SR-IOV VMs) for RDMA connectivity. In certain deployments of A8 and A9 instances, the HpcVmDrivers extension is added automatically. Note that the HpcVmDrivers VM extension is being deprecated; it will not be updated. To add the VM extension to a VM, you can use [Azure PowerShell](#) cmdlets.

The following command installs the latest version 1.0 InfiniBandDriverWindows extension on an existing RDMA-capable VM named *myVM* deployed in the resource group named *myResourceGroup* in the *West US* region:

```
Set-AzVMExtension -ResourceGroupName "myResourceGroup" -Location "westus" -VMName "myVM" -ExtensionName "InfiniBandDriverWindows" -Publisher "Microsoft.HpcCompute" -Type "InfiniBandDriverWindows" -TypeHandlerVersion "1.0"
```

Alternatively, VM extensions can be included in Azure Resource Manager templates for easy deployment, with the following JSON element:

```
"properties":{  
    "publisher": "Microsoft.HpcCompute",  
    "type": "InfiniBandDriverWindows",  
    "typeHandlerVersion": "1.0",  
}
```

The following command installs the latest version 1.0 InfiniBandDriverWindows extension on all RDMA-capable VMs in an existing VM scale set named *myVMSS* deployed in the resource group named *myResourceGroup*:

```
$VMSS = Get-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS"  
Add-AzVmssExtension -VirtualMachineScaleSet $VMSS -Name "InfiniBandDriverWindows" -Publisher  
"Microsoft.HpcCompute" -Type "InfiniBandDriverWindows" -TypeHandlerVersion "1.0"  
Update-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "MyVMSS" -  
VirtualMachineScaleSet $VMSS  
Update-AzVmssInstance -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS" -InstanceId  
"**"
```

For more information, see [Virtual machine extensions and features](#). You can also work with extensions for VMs deployed in the [classic deployment model](#).

- **RDMA network address space** - The RDMA network in Azure reserves the address space 172.16.0.0/16. To run MPI applications on instances deployed in an Azure virtual network, make sure that the virtual network address space does not overlap the RDMA network.

Cluster configuration options

Azure provides several options to create clusters of Windows HPC VMs that can communicate using the RDMA network, including:

- **Virtual machines** - Deploy the RDMA-capable HPC VMs in the same availability set (when you use the Azure Resource Manager deployment model). If you use the classic deployment model, deploy the VMs in the same cloud service.
- **Virtual machine scale sets** - In a virtual machine scale set, ensure that you limit the deployment to a single placement group. For example, in a Resource Manager template, set the `singlePlacementGroup` property to `true`.
- **MPI among virtual machines** - If MPI communication is required between virtual machines (VMs), ensure that the VMs are in the same availability set or the virtual machine same scale set.
- **Azure CycleCloud** - Create an HPC cluster in [Azure CycleCloud](#) to run MPI jobs on Windows nodes.
- **Azure Batch** - Create an [Azure Batch](#) pool to run MPI workloads on Windows Server compute nodes. For more information, see [Use RDMA-capable or GPU-enabled instances in Batch pools](#). Also see the [Batch Shipyard](#) project, for running container-based workloads on Batch.
- **Microsoft HPC Pack** - [HPC Pack](#) includes a runtime environment for MS-MPI that uses the Azure RDMA network when deployed on RDMA-capable Linux VMs. For example deployments, see [Set up a Linux RDMA cluster with HPC Pack to run MPI applications](#).

Other sizes

- [General purpose](#)
- [Compute optimized](#)

- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [Previous generations](#)

Next steps

- For checklists to use the compute-intensive instances with HPC Pack on Windows Server, see [Set up a Linux RDMA cluster with HPC Pack to run MPI applications](#).
- To use compute-intensive instances when running MPI applications with Azure Batch, see [Use multi-instance tasks to run Message Passing Interface \(MPI\) applications in Azure Batch](#).
- Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

H-series

2/20/2020 • 2 minutes to read • [Edit Online](#)

H-series VMs are optimized for applications driven by high CPU frequencies or large memory per core requirements. H-series VMs feature 8 or 16 Intel Xeon E5 2667 v3 processor cores, up to 14 GB of RAM per CPU core, and no hyperthreading. H-series features 56 Gb/sec Mellanox FDR InfiniBand in a non-blocking fat tree configuration for consistent RDMA performance. H-series VMs support Intel MPI 5.x and MS-MPI.

ACU: 290-300

Premium Storage: Not Supported

Premium Storage Caching: Not Supported

SIZE	VCPU	PROC ESSO R	MEM ORY (GB)	MEM ORY BAND WIDT H GB/S	BASE CPU FREQ Y (GHZ)	ALL- CORE S FREQ UENC Y (GHZ, PEAK)	SINGL E- CORE FREQ UENC Y (GHZ, PEAK)	RDM A PERF ORM ANCE (GB/S)	MPI SUPP ORT	TEMP STOR AGE (GB)	MAX DATA DISKS	MAX ETHE RNET NICS
Stand ard_ H8	8	Intel Xeon E5 2667 v3	56	40	3.2	3.3	3.6	-	Intel 5.x, MS- MPI	1000	32	2
Stand ard_ H16	16	Intel Xeon E5 2667 v3	112	80	3.2	3.3	3.6	-	Intel 5.x, MS- MPI	2000	64	4
Stand ard_ H8m	8	Intel Xeon E5 2667 v3	112	40	3.2	3.3	3.6	-	Intel 5.x, MS- MPI	1000	32	2
Stand ard_ H16 m	16	Intel Xeon E5 2667 v3	224	80	3.2	3.3	3.6	-	Intel 5.x, MS- MPI	2000	64	4
Stand ard_ H16r 1	16	Intel Xeon E5 2667 v3	112	80	3.2	3.3	3.6	56	Intel 5.x, MS- MPI	2000	64	4

SIZE	VCPU	PROC ESSO R	MEM ORY (GB)	MEM BAND WIDT H GB/S	BASE CPU FREQ UENC Y (GHZ)	ALL-CORE S FREQ UENC Y (GHZ, PEAK)	SINGL E-CORE FREQ UENC Y (GHZ, PEAK)	RDMA PERFORM ANCE (GB/S)	MPI SUPP ORT	TEMP STOR AGE (GB)	MAX DATA DISKS	MAX ETHERNET NICs
Standard_H16_mr ¹	16	Intel Xeon E5 2667 v3	224	80	3.2	3.3	3.6	56	Intel 5.x, MS-MPI	2000	64	4

¹ For MPI applications, dedicated RDMA backend network is enabled by FDR InfiniBand network.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

HB-series

2/20/2020 • 2 minutes to read • [Edit Online](#)

HB-series VMs are optimized for applications driven by memory bandwidth, such as fluid dynamics, explicit finite element analysis, and weather modeling. HB VMs feature 60 AMD EPYC 7551 processor cores, 4 GB of RAM per CPU core, and no simultaneous multithreading. An HB VM provides up to 260 GB/sec of memory bandwidth.

ACU: 199-216

Premium Storage: Supported

Premium Storage Caching: Supported

SIZE	VCPU	PROC ESSO R	MEM ORY (GB)	MEM BAND WIDT H GB/S	BASE FREQ UENC Y (GHZ)	ALL-CORE S FREQ UENC Y (GHZ, PEAK)	SINGL E-CORE FREQ UENC Y (GHZ, PEAK)	RDM A PERFORM ANCE (GB/S)	MPI SUPP ORT	TEMP STOR AGE (GB)	MAX DATA DISKS	MAX ETHERNET NICs
Standard_HB60rs	60	AMD EPYC 7551	240	263	2.0	2.55	2.55	100	All	700	4	1

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTTCP\)](#).

Other sizes

- [General purpose](#)

- [Memory optimized](#)
- [Storage optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Previous generations](#)

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

HBv2-series

2/20/2020 • 2 minutes to read • [Edit Online](#)

HBv2-series VMs are optimized for applications driven by memory bandwidth, such as fluid dynamics, finite element analysis, and reservoir simulation. HBv2 VMs feature 120 AMD EPYC 7742 processor cores, 4 GB of RAM per CPU core, and no simultaneous multithreading. Each HBv2 VM provides up to 340 GB/sec of memory bandwidth, and up to 4 teraFLOPS of FP64 compute.

Premium Storage: Supported

SIZE	vCPU	PROC ESSO R	MEM ORY (GB)	MEM BAND WIDTH GB/S	BASE CPU FREQ Y (GHZ)	ALL-CORE S FREQ Y (GHZ, PEAK)	SINGL E-CORE FREQ Y (GHZ, PEAK)	RDM A PERFOR MANCE (GB/S)	MPI SUPP ORT	TEMP STOR AGE (GB)	MAX DATA DISKS	MAX ETHER NET NICs
Standard_HB12_0rs_v2	120	AMD EPYC 7V12	480	350	2.45	3.1	3.3	200	All	480 + 960	8	1

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- [General purpose](#)
- [Memory optimized](#)

- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

HC-series

2/20/2020 • 2 minutes to read • [Edit Online](#)

HC-series VMs are optimized for applications driven by dense computation, such as implicit finite element analysis, molecular dynamics, and computational chemistry. HC VMs feature 44 Intel Xeon Platinum 8168 processor cores, 8 GB of RAM per CPU core, and no hyperthreading. The Intel Xeon Platinum platform supports Intel's rich ecosystem of software tools such as the Intel Math Kernel Library.

ACU: 297-315

Premium Storage: Supported

Premium Storage Caching: Supported

SIZE	vCPU	PROC ESSO R	MEM ORY (GB)	MEM ORY BAND WIDT H GB/S	BASE CPU FREQ Y (GHZ)	ALL-CORE S FREQ Y (GHZ, PEAK)	SINGL E-CORE FREQ Y (GHZ, PEAK)	RDM A PERFOR MANCE (GB/S)	MPI SUPP ORT	TEMP STOR AGE (GB)	MAX DATA DISKS	MAX ETHERNET NICs
Standard_HC4_4rs	44	Intel Xeon Platinum 8168	352	191	2.7	3.4	3.7	100	All	700	4	1

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Other sizes

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Save costs with Azure Reserved VM Instances

11/12/2019 • 7 minutes to read • [Edit Online](#)

When you commit to an Azure reserved VM instance you can save money. The reservation discount is applied automatically to the number of running virtual machines that match the reservation scope and attributes. You don't need to assign a reservation to a virtual machine to get the discounts. A reserved instance purchase covers only the compute part of your VM usage. For Windows VMs, the usage meter is split into two separate meters. There's a compute meter, which is same as the Linux meter, and a Windows IP meter. The charges that you see when you make the purchase are only for the compute costs. Charges don't include Windows software costs. For more information about software costs, see [Software costs not included with Azure Reserved VM Instances](#).

Determine the right VM size before you buy

Before you buy a reservation, you should determine the size of the VM that you need. The following sections will help you determine the right VM size.

Use reservation recommendations

You can use reservation recommendations to help determine the reservations you should purchase.

- Purchase recommendations and recommended quantity are shown when you purchase a VM reserved instance in the Azure portal.
- Azure Advisor provides purchase recommendations for individual subscriptions.
- You can use the APIs to get purchase recommendations for both shared scope and single subscription scope. For more information, see [Reserved instance purchase recommendation APIs for enterprise customers](#).
- For Enterprise Agreement (EA) and Microsoft Customer Agreement (MCA) customers, purchase recommendations for shared and single subscription scopes are available with the [Azure Consumption Insights Power BI content pack](#).

Services that get VM reservation discounts

Your VM reservations can apply to VM usage emitted from multiple services - not just for your VM deployments. Resources that get reservation discounts change depending on the instance size flexibility setting.

Instance size flexibility setting

The instance size flexibility setting determines which services get the reserved instance discounts.

Whether the setting is on or off, reservation discounts automatically apply to any matching VM usage when the *ConsumedService* is `Microsoft.Compute`. So, check your usage data for the *ConsumedService* value. Some examples include:

- Virtual machines
- Virtual machine scale sets
- Container service
- Azure Batch deployments (in user subscriptions mode)
- Azure Kubernetes Service (AKS)
- Service Fabric

When the setting is on, reservation discounts automatically apply to matching VM usage when the *ConsumedService* is any of the following items:

- `Microsoft.Compute`
- `Microsoft.ClassicCompute`

- Microsoft.Batch
- Microsoft.MachineLearningServices
- Microsoft.Kusto

Check the *ConsumedService* value in your usage data to determine if the usage is eligible for reservation discounts.

For more information about instance size flexibility, see [Virtual machine size flexibility with Reserved VM Instances](#).

Analyze your usage information

Analyze your usage information to help determine which reservations you should purchase.

Usage data is available in the usage file and APIs. Use them together to determine which reservation to purchase. Check for VM instances that have high usage on daily basis to determine the quantity of reservations to purchase.

Avoid the `Meter` subcategory and `Product` fields in usage data. They don't distinguish between VM sizes that use premium storage. If you use these fields to determine the VM size for reservation purchase, you may buy the wrong size. Then you won't get the reservation discount you expect. Instead, refer to the `AdditionalInfo` field in your usage file or usage API to determine the correct VM size.

Purchase restriction considerations

Reserved VM Instances are available for most VM sizes with some exceptions. Reservation discounts don't apply for the following VMs:

- **VM series** - A-series, Av2-series, or G-series.
- **Preview or Promo VMs** - Any VM-series or size that is in preview or uses promotional meter.
- **Clouds** - Reservations aren't available for purchase in Germany or China regions.
- **Insufficient quota** - A reservation that is scoped to a single subscription must have vCPU quota available in the subscription for the new RI. For example, if the target subscription has a quota limit of 10 vCPUs for D-Series, then you can't buy a reservation for 11 Standard_D1 instances. The quota check for reservations includes the VMs already deployed in the subscription. For example, if the subscription has a quota of 10 vCPUs for D-Series and has two standard_D1 instances deployed, then you can buy a reservation for 10 standard_D1 instances in this subscription. You can [create quote increase request](#) to resolve this issue.
- **Capacity restrictions** - In rare circumstances, Azure limits the purchase of new reservations for subset of VM sizes, because of low capacity in a region.

Buy a Reserved VM Instance

You can buy a reserved VM instance in the [Azure portal](#). Pay for the reservation [up front or with monthly payments](#). These requirements apply to buying a reserved VM instance:

- You must be in an Owner role for at least one EA subscription or a subscription with a pay-as-you-go rate.
- For EA subscriptions, the **Add Reserved Instances** option must be enabled in the [EA portal](#). Or, if that setting is disabled, you must be an EA Admin for the subscription.
- For the Cloud Solution Provider (CSP) program, only the admin agents or sales agents can buy reservations.

To buy an instance:

1. Sign in to the [Azure portal](#).
2. Select **All services > Reservations**.
3. Select **Add** to purchase a new reservation and then click **Virtual machine**.
4. Enter required fields. Running VM instances that match the attributes you select qualify to get the reservation discount. The actual number of your VM instances that get the discount depend on the scope and quantity selected.

If you have an EA agreement, you can use the **Add more option** to quickly add additional instances. The option isn't available for other subscription types.

FIELD	DESCRIPTION
Subscription	The subscription used to pay for the reservation. The payment method on the subscription is charged the costs for the reservation. The subscription type must be an enterprise agreement (offer numbers: MS-AZR-0017P or MS-AZR-0148P) or Microsoft Customer Agreement or an individual subscription with pay-as-you-go rates (offer numbers: MS-AZR-0003P or MS-AZR-0023P). The charges are deducted from the monetary commitment balance, if available, or charged as overage. For a subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.
Scope	The reservation's scope can cover one subscription or multiple subscriptions (shared scope). If you select: <ul style="list-style-type: none"> • Single resource group scope — Applies the reservation discount to the matching resources in the selected resource group only. • Single subscription scope — Applies the reservation discount to the matching resources in the selected subscription. • Shared scope — Applies the reservation discount to matching resources in eligible subscriptions that are in the billing context. For EA customers, the billing context is the enrollment. For individual subscriptions with pay-as-you-go rates, the billing scope is all eligible subscriptions created by the account administrator.
Region	The Azure region that's covered by the reservation.
VM Size	The size of the VM instances.
Optimize for	VM instance size flexibility is selected by default. Click Advanced settings to change the instance size flexibility value to apply the reservation discount to other VMs in the same VM size group . Capacity priority prioritizes data center capacity for your deployments. It offers additional confidence in your ability to launch the VM instances when you need them. Capacity priority is only available when the reservation scope is single subscription.
Term	One year or three years.
Quantity	The number of instances being purchased within the reservation. The quantity is the number of running VM instances that can get the billing discount. For example, if you are running 10 Standard_D2 VMs in the East US, then you would specify quantity as 10 to maximize the benefit for all running VMs.

Usage data and reservation utilization

Your usage data has an effective price of zero for the usage that gets a reservation discount. You can see which VM

instance received the reservation discount for each reservation.

For more information about how reservation discounts appear in usage data, see [Understand Azure reservation usage for your Enterprise enrollment](#) if you are an EA customer. If you have an individual subscription, see [Understand Azure reservation usage for your Pay-As-You-Go subscription](#).

Change a reservation after purchase

You can make the following types of changes to a reservation after purchase:

- Update reservation scope
- Instance size flexibility (if applicable)
- Ownership

You can also split a reservation into smaller chunks and merge already split reservations. None of the changes cause a new commercial transaction or change the end date of the reservation.

You can't make the following types of changes after purchase, directly:

- An existing reservation's region
- SKU
- Quantity
- Duration

However, you can *exchange* a reservation if you want to make changes.

Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations with certain limitations. For more information, see [Self-service exchanges and refunds for Azure Reservations](#).

Need help? Contact us.

If you have questions or need help, [create a support request](#).

Next steps

- To learn how to manage a reservation, see [Manage Azure Reservations](#).
- To learn more about Azure Reservations, see the following articles:
 - [What are Azure Reservations?](#)
 - [Manage Reservations in Azure](#)
 - [Understand how the reservation discount is applied](#)
 - [Understand reservation usage for a subscription with pay-as-you-go rates](#)
 - [Understand reservation usage for your Enterprise enrollment](#)
 - [Windows software costs not included with reservations](#)
 - [Azure Reservations in Partner Center Cloud Solution Provider \(CSP\) program](#)

2 minutes to read

Virtual machine size flexibility with Reserved VM Instances

2/19/2020 • 2 minutes to read • [Edit Online](#)

When you buy a Reserved VM Instance, you can choose to optimize for instance size flexibility or capacity priority. For more information about setting or changing the optimize setting for reserved VM instances, see [Change the optimize setting for reserved VM instances](#).

With a reserved virtual machine instance that's optimized for instance size flexibility, the reservation you buy can apply to the virtual machines (VMs) sizes in the same instance size flexibility group. For example, if you buy a reservation for a VM size that's listed in the DSv2 Series, like Standard_DS5_v2, the reservation discount can apply to the other four sizes that are listed in that same instance size flexibility group:

- Standard_DS1_v2
- Standard_DS2_v2
- Standard_DS3_v2
- Standard_DS4_v2

But that reservation discount doesn't apply to VMs sizes that are listed in different instance size flexibility groups, like SKUs in DSv2 Series High Memory: Standard_DS11_v2, Standard_DS12_v2, and so on.

Within the instance size flexibility group, the number of VMs the reservation discount applies to depends on the VM size you pick when you buy a reservation. It also depends on the sizes of the VMs that you have running. The ratio column compares the relative footprint for each VM size in that instance size flexibility group. Use the ratio value to calculate how the reservation discount applies to the VMs you have running.

Examples

The following examples use the sizes and ratios in the DSv2-series table.

You buy a reserved VM instance with the size Standard_DS4_v2 where the ratio or relative footprint compared to the other sizes in that series is 8.

- Scenario 1: Run eight Standard_DS1_v2 sized VMs with a ratio of 1. Your reservation discount applies to all eight of those VMs.
- Scenario 2: Run two Standard_DS2_v2 sized VMs with a ratio of 2 each. Also run a Standard_DS3_v2 sized VM with a ratio of 4. The total footprint is $2+2+4=8$. So your reservation discount applies to all three of those VMs.
- Scenario 3: Run one Standard_DS5_v2 with a ratio of 16. Your reservation discount applies to half that VM's compute cost.

The following sections show what sizes are in the same size series group when you buy a reserved VM instance optimized for instance size flexibility.

Instance size flexibility ratio for VMs

CSV below has the instance size flexibility groups, ArmSkuName and the ratios.

Instance size flexibility ratios

We will keep the file URL and the schema fixed so you can consume this file programmatically. The data will also be available through API soon.

Preview: Use Spot VMs in Azure

12/3/2019 • 4 minutes to read • [Edit Online](#)

Using Spot VMs allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Spot VMs. Therefore, Spot VMs are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

The amount of available capacity can vary based on size, region, time of day, and more. When deploying Spot VMs, Azure will allocate the VMs if there is capacity available, but there is no SLA for these VMs. A Spot VM offers no high availability guarantees. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Spot VMs with 30 seconds notice.

IMPORTANT

Spot instances are currently in public preview. This preview version is not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Eviction policy

VMs can be evicted based on capacity or the max price you set. For virtual machines, the eviction policy is set to *Deallocate* which moves your evicted VMs to the stopped-deallocated state, allowing you to redeploy the evicted VMs at a later time. However, reallocating Spot VMs will be dependent on there being available Spot capacity. The deallocated VMs will count against your spot vCPU quota and you will be charged for your underlying disks.

Users can opt-in to receive in-VM notifications through [Azure Scheduled Events](#). This will notify you if your VMs are being evicted and you will have 30 seconds to finish any jobs and perform shutdown tasks prior to the eviction.

OPTION	OUTCOME
Max price is set to \geq the current price.	VM is deployed if capacity and quota are available.
Max price is set to $<$ the current price.	The VM is not deployed. You will get an error message that the max price needs to be \geq current price.
Restarting a stop/deallocate VM if the max price is \geq the current price	If there is capacity and quota, then the VM is deployed.
Restarting a stop/deallocate VM if the max price is $<$ the current price	You will get an error message that the max price needs to be \geq current price.
Price for the VM has gone up and is now $>$ the max price.	The VM gets evicted. You get a 30s notification before actual eviction.
After eviction the price for the VM goes back to being $<$ the max price.	The VM will not be automatically re-started. You can restart the VM yourself, and it will be charged at the current price.

OPTION	OUTCOME
If the max price is set to <code>-1</code>	The VM will not be evicted for pricing reasons. The max price will be the current price, up to the price for standard VMs. You will never be charged above the standard price.
Changing the max price	You need to deallocate the VM to change the max price. Deallocate the VM, set a new max price, then update the VM.

Limitations

The following VM sizes are not supported for Spot VMs:

- B-series
- Promo versions of any size (like Dv2, NV, NC, H promo sizes)

Spot VMs can't currently use ephemeral OS disks.

Spot VMs can be deployed to any region, except Microsoft Azure China 21Vianet.

Pricing

Pricing for Spot VMs is variable, based on region and SKU. For more information, see VM pricing for [Linux](#) and [Windows](#).

With variable pricing, you have option to set a max price, in US dollars (USD), using up to 5 decimal places. For example, the value `0.98765` would be a max price of \$0.98765 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, whichever is less, as long as there is capacity and quota available.

Frequently asked questions

Q: Once created, is a Spot VM the same as regular standard VM?

A: Yes, except there is no SLA for Spot VMs and they can be evicted at any time.

Q: What to do when you get evicted, but still need capacity?

A: We recommend you use standard VMs instead of Spot VMs if you need capacity right away.

Q: How is quota managed for Spot VMs?

A: Spot VMs will have a separate quota pool. Spot quota will be shared between VMs and scale-set instances. For more information, see [Azure subscription and service limits, quotas, and constraints](#).

Q: Can I request for additional quota for Spot?

A: Yes, you will be able to submit the request to increase your quota for Spot VMs through the [standard quota request process](#).

Q: What channels support Spot VMs?

A: See the table below for Spot VM availability.

AZURE CHANNELS	AZURE SPOT VMs AVAILABILITY
Enterprise Agreement	Yes

AZURE CHANNELS	AZURE SPOT VMs AVAILABILITY
Pay As You Go	Yes
Cloud Service Provider (CSP)	Contact your partner
Benefits	Not available
Sponsored	Not available
Free Trial	Not available

Q: Where can I post questions?

A: You can post and tag your question with `azure-spot` at [Q&A](#).

Next steps

Use the [portal](#), [CLI](#) or [PowerShell](#) to deploy Spot VMs.

You can also deploy a [scale set with Spot VM instances](#).

If you encounter an error, see [Error codes](#).

Previous generations of virtual machine sizes

2/28/2020 • 11 minutes to read • [Edit Online](#)

This section provides information on previous generations of virtual machine sizes. These sizes can still be used, but there are newer generations available.

F-series

F-series is based on the 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell) processor, which can achieve clock speeds as high as 3.1 GHz with the Intel Turbo Boost Technology 2.0. This is the same CPU performance as the Dv2-series of VMs.

F-series VMs are an excellent choice for workloads that demand faster CPUs but do not need as much memory or temporary storage per vCPU. Workloads such as analytics, gaming servers, web servers, and batch processing will benefit from the value of the F-series.

ACU: 210 - 250

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_F1	1	2	16	3000/46/23	4/4x500	2/750
Standard_F2	2	4	32	6000/93/46	8/8x500	2/1500
Standard_F4	4	8	64	12000/187/93	16/16x500	4/3000
Standard_F8	8	16	128	24000/375/187	32/32x500	8/6000
Standard_F16	16	32	256	48000/750/375	64/64x500	8/12000

Fs-series ¹

The Fs-series provides all the advantages of the F-series, in addition to Premium storage.

ACU: 210 - 250

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_F1s	1	2	4	4	4000/32 (12)	3200/48	2/750
Standard_F2s	2	4	8	8	8000/64 (24)	6400/96	2/1500
Standard_F4s	4	8	16	16	16000/128 (48)	12800/192	4/3000
Standard_F8s	8	16	32	32	32000/256 (96)	25600/384	8/6000
Standard_F16s	16	32	64	64	64000/512 (192)	51200/768	8/12000

MBps = 10^6 bytes per second, and GiB = 1024^3 bytes.

¹ The maximum disk throughput (IOPS or MBps) possible with a Fs series VM may be limited by the number, size, and striping of the attached disk(s). For details, see designing for high performance for [Windows](#) or [Linux](#).

NVv2-series

Newer size recommendation: [NVv3-series](#)

The NVv2-series virtual machines are powered by [NVIDIA Tesla M60](#) GPUs and NVIDIA GRID technology with Intel Broadwell CPUs. These virtual machines are targeted for GPU accelerated graphics applications and virtual desktops where customers want to visualize their data, simulate results to view, work on CAD, or render and stream content. Additionally, these virtual machines can run single precision workloads such as encoding and rendering. NVv2 virtual machines support Premium Storage and come with twice the system memory (RAM) when compared with its predecessor NV-series.

Each GPU in NVv2 instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	GPU	GPU MEMORY: GIB	MAX DATA DISKS	MAX NICs	VIRTUAL WORKSTATIONS	VIRTUAL APPLICATIONS
Standard_NV6s_v2	6	112	320	1	8	12	4	1	25
Standard_NV12s_v2	12	224	640	2	16	24	8	2	50

SIZE	VCPU	MEMORY: GiB	TEMP STORAGE (SSD) GiB	GPU	GPU MEMORY: GiB	MAX DATA DISKS	MAX NICs	VIRTUAL WORKSTATIONS	VIRTUAL APPLICATIONS
Standard_NV2_4s_v2	24	448	1280	4	32	32	8	4	100

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- If you want to get the best performance for your VMs, you should limit the number of data disks to two disks per vCPU.
- Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see [Virtual machine network bandwidth](#).

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see [Optimize network throughput for Azure virtual machines](#). To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see [Bandwidth/Throughput testing \(NTTCP\)](#).

Older generations of virtual machine sizes

This section provides information on older generations of virtual machine sizes. These sizes are still supported but will not receive additional capacity. There are newer or alternative sizes that are generally available. Please refer to [Sizes for Linux virtual machines in Azure](#) to choose the VM sizes that will best fit your need.

For more information on resizing a Windows VM, see [Resize a Linux VM](#).

Basic A

Newer size recommendation: [Av2-series](#)

Premium Storage: Not Supported

Premium Storage caching: Not Supported

The basic tier sizes are primarily for development workloads and other applications that don't require load balancing, auto-scaling, or memory-intensive virtual machines.

SIZE – SIZE\NAME	VCPU	MEMORY	NICS (MAX)	MAX TEMPORARY DISK SIZE	MAX. DATA DISKS (1023 GB EACH)	MAX. IOPS (300 PER DISK)
A0\Basic_A0	1	768 MB	2	20 GB	1	1x300
A1\Basic_A1	1	1.75 GB	2	40 GB	2	2x300
A2\Basic_A2	2	3.5 GB	2	60 GB	4	4x300
A3\Basic_A3	4	7 GB	2	120 GB	8	8x300
A4\Basic_A4	8	14 GB	2	240 GB	16	16x300

Standard A0 - A4 using CLI and PowerShell

In the classic deployment model, some VM size names are slightly different in CLI and PowerShell:

- Standard_A0 is ExtraSmall
- Standard_A1 is Small
- Standard_A2 is Medium
- Standard_A3 is Large
- Standard_A4 is ExtraLarge

A-series

Newer size recommendation: [Av2-series](#)

ACU: 50-100

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (HDD): GIB	MAX DATA DISKS	MAX DATA DISK THROUGHPUT: IOPS	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_A0_1	1	0.768	20	1	1x500	2/100
Standard_A1	1	1.75	70	2	2x500	2/500
Standard_A2	2	3.5	135	4	4x500	2/500
Standard_A3	4	7	285	8	8x500	2/1000
Standard_A4	8	14	605	16	16x500	4/2000
Standard_A5	2	14	135	4	4x500	2/500
Standard_A6	4	28	285	8	8x500	2/1000
Standard_A7	8	56	605	16	16x500	4/2000

¹ The A0 size is over-subscribed on the physical hardware. For this specific size only, other customer deployments may impact the performance of your running workload. The relative performance is outlined below as the expected baseline, subject to an approximate variability of 15 percent.

A-series - compute-intensive instances

Newer size recommendation: [Av2-series](#)

ACU: 225

Premium Storage: Not Supported

Premium Storage caching: Not Supported

The A8-A11 and H-series sizes are also known as *compute-intensive instances*. The hardware that runs these sizes is designed and optimized for compute-intensive and network-intensive applications, including high-performance computing (HPC) cluster applications, modeling, and simulations. The A8-A11 series uses Intel Xeon E5-2670 @ 2.6 GHZ and the H-series uses Intel Xeon E5-2667 v3 @ 3.2 GHz.

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (HDD): GIB	MAX DATA DISKS	MAX DATA DISK THROUGHPUT: IOPS	MAX NICs
Standard_A8 ¹	8	56	382	32	32x500	2
Standard_A9 ¹	16	112	382	64	64x500	4
Standard_A10	8	56	382	32	32x500	2
Standard_A11	16	112	382	64	64x500	4

¹For MPI applications, dedicated RDMA backend network is enabled by FDR InfiniBand network, which delivers ultra-low-latency and high bandwidth.

D-series

Newer size recommendation: [Dv3-series](#)

ACU: 160-250¹

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D1	1	3.5	50	3000/46/23	4/4x500	2/500

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D2	2	7	100	6000/93/46	8/8x500	2/1000
Standard_D3	4	14	200	12000/187/93	16/16x500	4/2000
Standard_D4	8	28	400	24000/375/187	32/32x500	8/4000

¹ VM Family can run on one of the following CPU's: 2.2 GHz Intel Xeon® E5-2660 v2, 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell) or 2.3 GHz Intel XEON® E5-2673 v4 (Broadwell)

D-series - memory optimized

Newer size recommendation: [Dv3-series](#)

ACU: 160-250 ¹

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_D1 1	2	14	100	6000/93/46	8/8x500	2/1000
Standard_D1 2	4	28	200	12000/187/93	16/16x500	4/2000
Standard_D1 3	8	56	400	24000/375/187	32/32x500	8/4000
Standard_D1 4	16	112	800	48000/750/375	64/64x500	8/8000

¹ VM Family can run on one of the following CPU's: 2.2 GHz Intel Xeon® E5-2660 v2, 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell) or 2.3 GHz Intel XEON® E5-2673 v4 (Broadwell)

Preview: DC-series

Premium Storage: Supported

Premium Storage caching: Supported

The DC-series uses the latest generation of 3.7GHz Intel XEON E-2176G Processor with SGX technology,

and with the Intel Turbo Boost Technology can go up to 4.7GHz.

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS / MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS / MBPS	MAX NICs / EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_DC2s	2	8	100	2	4000 / 32 (43)	3200 /48	2 / 1500
Standard_DC4s	4	16	200	4	8000 / 64 (86)	6400 /96	2 / 3000

IMPORTANT

DC-series VMs are [generation 2 VMs](#) and only support [Gen2](#) images.

DS-series

Newer size recommendation: [Dsv3-series](#)

ACU: 160-250 ¹

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_DS1	1	3.5	7	4	4000/32 (43)	3200/32	2/500
Standard_DS2	2	7	14	8	8000/64 (86)	6400/64	2/1000
Standard_DS3	4	14	28	16	16000/128 (172)	12800/128	4/2000
Standard_DS4	8	28	56	32	32000/256 (344)	25600/256	8/4000

¹ VM Family can run on one of the following CPU's: 2.2 GHz Intel Xeon® E5-2660 v2, 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell) or 2.3 GHz Intel XEON® E5-2673 v4 (Broadwell)

DS-series - memory optimized

Newer size recommendation: [Dsv3-series](#)

ACU: 160-250^{1,2}

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPUs	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUGHPUT: IOPS/MBPS (CACHE SIZE IN GIB)	MAX UNCACHED DISK THROUGHPUT: IOPS/MBPS	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_DS11	2	14	28	8	8000/64 (72)	6400/64	2/1000
Standard_DS12	4	28	56	16	16000/128 (144)	12800/128	4/2000
Standard_DS13	8	56	112	32	32000/256 (288)	25600/256	8/4000
Standard_DS14	16	112	224	64	64000/512 (576)	51200/512	8/8000

¹ The maximum disk throughput (IOPS or MBps) possible with a DS series VM may be limited by the number, size and striping of the attached disk(s). For details, see designing for high performance for [Windows](#) or [Linux](#). ² VM Family can run on one of the following CPU's: 2.2 GHz Intel Xeon® E5-2660 v2, 2.4 GHz Intel Xeon® E5-2673 v3 (Haswell) or 2.3 GHz Intel XEON® E5-2673 v4 (Broadwell)

Ls-series

The Ls-series offers up to 32 vCPUs, using the [Intel® Xeon® processor E5 v3 family](#). The Ls-series gets the same CPU performance as the G/GS-Series and comes with 8 GiB of memory per vCPU.

The Ls-series does not support the creation of a local cache to increase the IOPS achievable by durable data disks. The high throughput and IOPS of the local disk makes Ls-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB which replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

ACU: 180-240

Premium Storage: Supported

Premium Storage caching: Not Supported

SIZE	VCPUs	MEMORY (GIB)	TEMP STORAGE (GIB)	MAX DATA DISKS	MAX TEMP STORAGE THROUGHPUT (IOPS/MBPS)	MAX UNCACHED DISK THROUGHPUT (IOPS/MBPS)	MAX NICS/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_L4s	4	32	678	16	20000/200	5000/125	2/4000

SIZE	VCPUs	MEMORY (GiB)	TEMP STORAGE (GiB)	MAX DATA DISKS	MAX TEMP STORAGE THROUHPUT (IOPS/MBPS)	MAX UNCACHED DISK THROUHPUT (IOPS/MBPS)	MAX NICs/EXPECTED NETWORK BANDWIDTH (Mbps)
Standard_L8s	8	64	1388	32	40000/400	10000/250	4/8000
Standard_L16s	16	128	2807	64	80000/800	20000/500	8/16000
Standard_L32s ¹	32	256	5630	64	160000/1600	40000/1000	8/20000

The maximum disk throughput possible with Ls-series VMs may be limited by the number, size, and striping of any attached disks. For details, see designing for high performance for [Windows](#) or [Linux](#).

¹ Instance is isolated to hardware dedicated to a single customer.

GS-series

ACU: 180 - 240¹

Premium Storage: Supported

Premium Storage caching: Supported

SIZE	VCPUs	MEMORY: GiB	TEMP STORAGE (SSD) GiB	MAX DATA DISKS	MAX CACHED AND TEMP STORAGE THROUHPUT: IOPS / MBPS (CACHE SIZE IN GiB)	MAX UNCACHED DISK THROUHPUT: IOPS/MBPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (Mbps)
Standard_GS1	2	28	56	8	10000/100 (264)	5000/125	2/2000
Standard_GS2	4	56	112	16	20000/200 (528)	10000/250	2/4000
Standard_GS3	8	112	224	32	40000/400 (1056)	20000/500	4/8000
Standard_GS4 ³	16	224	448	64	80000/800 (2112)	40000/1000	8/16000
Standard_GS5 ^{2,3}	32	448	896	64	160000/1600 (4224)	80000/2000	8/20000

¹ The maximum disk throughput (IOPS or MBps) possible with a GS series VM may be limited by the number, size and striping of the attached disk(s). For details, see designing for high performance for [Windows](#) or [Linux](#).

² Instance is isolated to hardware dedicated to a single customer.

³ Constrained core sizes available.

G-series

ACU: 180 - 240

Premium Storage: Not Supported

Premium Storage caching: Not Supported

SIZE	VCPU	MEMORY: GIB	TEMP STORAGE (SSD) GIB	MAX TEMP STORAGE THROUGHPUT: IOPS/READ MBPS/WRITE MBPS	MAX DATA DISKS/THROUGHPUT: IOPS	MAX NICs/EXPECTED NETWORK BANDWIDTH (MBPS)
Standard_G1	2	28	384	6000/93/46	8/8x500	2/2000
Standard_G2	4	56	768	12000/187/93	16/16x500	2/4000
Standard_G3	8	112	1536	24000/375/187	32/32x500	4/8000
Standard_G4	16	224	3072	48000/750/375	64/64x500	8/16000
Standard_G5 ¹	32	448	6144	96000/1500/750	64/64x500	8/20000

¹ Instance is isolated to hardware dedicated to a single customer.

Other sizes

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- GPU
- High performance compute

Next steps

Learn more about how [Azure compute units \(ACU\)](#) can help you compare compute performance across Azure SKUs.

Virtual machine isolation in Azure

11/13/2019 • 4 minutes to read • [Edit Online](#)

Azure Compute offers virtual machine sizes that are Isolated to a specific hardware type and dedicated to a single customer. These virtual machine sizes are best suited for workloads that require a high degree of isolation from other customers for workloads involving elements like compliance and regulatory requirements. Customers can also choose to further subdivide the resources of these Isolated virtual machines by using [Azure support for nested virtual machines](#).

Utilizing an isolated size guarantees that your virtual machine will be the only one running on that specific server instance. The current Isolated virtual machine offerings include:

- Standard_E64is_v3
- Standard_E64i_v3
- Standard_M128ms
- Standard_GS5
- Standard_G5
- Standard_DS15_v2
- Standard_D15_v2
- Standard_F72s_v2

You can learn more about each available isolated size [here](#).

Retiring D15_v2/DS15_v2 isolation on May 15, 2020

Update on February 10, 2020: The "isolation" retirement timeline has been extended to May 15, 2020"

Azure Dedicated Host is now GA, which allows you to run your organization's Linux and Windows virtual machines on single-tenant physical servers. We plan to fully replace isolated Azure VMs with Azure Dedicated Host. After **May 15, 2020** the D15_v2/DS15_v2 Azure VMs will no longer be hardware isolated.

How does this affect me?

After May 15, 2020, we will no longer provide an isolation guarantee for your D15_v2/DS15_v2 Azure virtual machines.

What actions should I take?

If hardware isolation is not required for you, there is no action you need to take.

If isolation is required to you, before May 15, 2020, you would need to either:

- [Migrate](#) your workload to Azure Dedicated Host.
- [Request access](#) to a D15i_v2 and DS15i_v2 Azure VM, to get the same price performance. This option is only available for pay-as-you-go and one-year reserved instance scenarios.
- [Migrate](#) your workload to another Azure isolated virtual machine.

For details see below:

Timeline

DATE	ACTION
November 18, 2019	Availability of D/DS15i_v2 (PAYG, 1-year RI)
May 14, 2020	Last day to buy D/DS15i_v2 1-year RI
May 15, 2020	D/DS15_v2 isolation guarantee removed
May 15, 2021	Retire D/DS15i_v2 (all customers except who bought 3-year RI of D/DS15_v2 before November 18, 2019)
November 17, 2022	Retire D/DS15i_v2 when 3-year RIs done (for customers who bought 3-year RI of D/DS15_v2 before November 18, 2019)

FAQ

Q: Is the size D/DS15_v2 going to get retired?

A: No, only "isolation" feature is going to get retired. If you do not need isolation, you do not need to take any action.

Q: Is the size D/DS15i_v2 going to get retired?

A: Yes, the size is only available until May 15, 2021. For customers who have bought 3-year RIs on D/DS15_v2 before November 18, 2019 will have access to D/DS15i_v2 until November 17, 2022.

Q: Why am I not seeing the new D/DS15i_v2 sizes in the portal?

A: If you are a current D/DS15_v2 customer and want to use the new D/DS15i_v2 sizes, fill this [form](#)

Q: Why I am not seeing any quota for the new D/DS15i_v2 sizes?

A: If you are a current D/DS15_v2 customer and want to use the new D/DS15i_v2 sizes, fill this [form](#)

Q: When are the other isolated sizes going to retire?

A: We will provide reminders 12 months in advance of the official decommissioning of the sizes.

Q: Is there a downtime when my vm lands on a non-isolated hardware?

A: If you do not need isolation, you do not need to take any action and you would not see any downtime.

Q: Are there any cost changes for moving to a non-isolated virtual machine?

A: No

Q: I already purchased 1- or 3-year Reserved Instance for D15_v2 or Ds15_v2. How will the discount be applied to my VM usage?

A: RIs purchased before November 18, 2019 will automatically extend coverage to the new isolated VM series.

RI	INSTANCE SIZE FLEXIBILITY	BENEFIT ELIGIBILITY
D15_v2	Off	D15_v2 and D15i_v2
D15_v2	On	D15_v2 series and D15i_v2 will all receive the RI benefit.
D14_v2	On	D15_v2 series and D15i_v2 will all receive the RI benefit.

Likewise for Dsv2 series.

Q: I want to purchase additional Reserved Instances for Dv2. Which one should I choose?

A: All RIs purchased after Nov 18, 2019, have the following behavior.

RI	INSTANCE SIZE FLEXIBILITY	BENEFIT ELIGIBILITY
D15_v2	Off	D15_v2 only
D15_v2	On	D15_v2 series will receive the RI benefit. The new D15i_v2 will not be eligible for RI benefit from this RI type.
D15i_v2	Off	D15i_v2 only
D15i_v2	On	D15i_v2 only

Instance Size Flexibility cannot be used to apply to any other sizes such as D2_v2, D4_v2, or D15_v2. Likewise, for Dsv2 series.

Q: Can I buy a new 3-year RI for D15i_v2 and DS15i_v2?

A: Unfortunately no, only 1-year RI is available for new purchase.

Q: Can I move my existing D15_v2/DS15_v2 Reserve Instance to an isolated size Reserved Instance?

A: This action is not necessary since the benefit will apply to both isolated and non-isolated sizes. But Azure will support changing existing D15_v2/DS15_v2 Reserved Instances to D15i_v2/DS15i_v2. For all other Dv2/Dsv2 Reserved Instances, use the existing Reserved Instance or buy new Reserved Instances for the isolated sizes.

Q: I'm an Azure Service Fabric Customer relying on the Silver or Gold Durability Tiers. Does this change impact me?

A: No. The guarantees provided by Service Fabric's [Durability Tiers](#) will continue to function even after this change. If you require physical hardware isolation for other reasons, you may still need to take one of the actions described above.

Next steps

- You can deploy a dedicated host using [Azure PowerShell](#), the [portal](#), and [Azure CLI](#). For more information, see the [Dedicated hosts](#) overview.

Azure compute unit (ACU)

2/28/2020 • 2 minutes to read • [Edit Online](#)

The concept of the Azure Compute Unit (ACU) provides a way of comparing compute (CPU) performance across Azure SKUs. This will help you easily identify which SKU is most likely to satisfy your performance needs. ACU is currently standardized on a Small (Standard_A1) VM being 100 and all other SKUs then represent approximately how much faster that SKU can run a standard benchmark.

IMPORTANT

The ACU is only a guideline. The results for your workload may vary.

SKU FAMILY	ACU \ VCPU	VCPU: CORE
A0	50	1:1
A1 - A4	100	1:1
A5 - A7	100	1:1
A1_v2 - A8_v2	100	1:1
A2m_v2 - A8m_v2	100	1:1
A8 - A11	225*	1:1
D1 - D14	160 - 250	1:1
D1_v2 - D15_v2	210 - 250*	1:1
DS1 - DS14	160 - 250	1:1
DS1_v2 - DS15_v2	210 - 250*	1:1
D_v3	160 - 190*	2:1***
Ds_v3	160 - 190*	2:1***
E_v3	160 - 190*	2:1***
Es_v3	160 - 190*	2:1***
F2s_v2 - F72s_v2	195 - 210*	2:1***
F1 - F16	210 - 250*	1:1
F1s - F16s	210 - 250*	1:1

SKU FAMILY	ACU \ VCPU	VCPU: CORE
G1 - G5	180 - 240*	1:1
GS1 - GS5	180 - 240*	1:1
H	290 - 300*	1:1
HB	199 - 216**	1:1
HC	297 - 315*	1:1
L4s - L32s	180 - 240*	1:1
L8s_v2 - L80s_v2	150 - 175**	2:1
M	160 - 180	2:1***

*ACUs use Intel® Turbo technology to increase CPU frequency and provide a performance increase. The amount of the performance increase can vary based on the VM size, workload, and other workloads running on the same host. **ACUs use AMD® Boost technology to increase CPU frequency and provide a performance increase. The amount of the performance increase can vary based on the VM size, workload, and other workloads running on the same host. ***Hyper-threaded and capable of running nested virtualization

Here are links to more information about the different sizes:

- [General-purpose](#)
- [Memory optimized](#)
- [Compute optimized](#)
- [GPU optimized](#)
- [High performance compute](#)
- [Storage optimized](#)

Compute benchmark scores for Windows VMs

2/26/2020 • 17 minutes to read • [Edit Online](#)

The following SPECInt benchmark scores show compute performance for select Azure VMs running Windows Server. Compute benchmark scores are also available for [Linux VMs](#).

Av2 - General Compute

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_A1_v2	1	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	12	14.2	0.3
Standard_A1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	13.2	0.6
Standard_A1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	10	14.1	0.7
Standard_A2_v2	2	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	14	28.9	0.6
Standard_A2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	10	27.4	1.6
Standard_A2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	17	28.9	1.8
Standard_A2_m_v2	2	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	14	29.0	0.5
Standard_A2_m_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	11	26.3	0.8
Standard_A2_m_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	21	28.4	1.0

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	STDDEV
Standard_A4_v2	4	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	27	56.6	1.0
Standard_A4_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	13	52.8	2.0
Standard_A4_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	15	52.1	4.5
Standard_A4_m_v2	4	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	17	56.4	1.8
Standard_A4_m_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	6	53.4	1.9
Standard_A4_m_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	23	57.1	3.6
Standard_A8_v2	8	1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	14	109.1	1.6
Standard_A8_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	6	101.5	2.8
Standard_A8_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	11	101.9	2.7
Standard_A8_m_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	11	101.4	1.2

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_A8m_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	10	104.5	5.1
Standard_A8m_v2	8	2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	13	111.6	2.3

Note: Av2-series VMs can be deployed on a variety of hardware types and processors (as seen above). Av2-series VMs have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled to offer relatively consistent processor performance for the running instance, regardless of the hardware it is deployed on; however, software that takes advantage of specific newer processor optimizations may see more significant variation across processor types.

B - Burstable

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_B1ms	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	6.3	0.2
Standard_B1ms	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	47	6.4	0.2
Standard_B2ms	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	36	19.8	0.8
Standard_B2s	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	13.0	0.0
Standard_B2s	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	29	13.0	0.5
Standard_B4ms	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	6	27.1	1.0
Standard_B4ms	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	43	28.3	0.7

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_B8ms	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	3	42.0	0.0
Standard_B8ms	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	25	41.4	0.9

Note: B-Series VMs are for workloads with burstable performance requirements. VM instances accumulate credits when using less than its baseline. When the VM has accumulated credit, the VM can burst above the baseline using up to 100% to meet short CPU burst requirements. Burst time depends on available credits which is a function of VM size and time.

SPEC Int is a fairly long running test that typically exhausts available burst credits. Therefore the numbers above are closer to the baseline performance of the VM (although they may reflect some burst time accumulated between runs). For short, bursty, workloads (typical on B-Series) performance will typically be closer to that of the Ds v3 Series..

DSv3 - General Compute + Premium Storage

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D2s_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	10	40.8	2.3
Standard_D2s_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	52	43.3	2.1
Standard_D4s_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	21	77.9	2.6
Standard_D4s_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	29	82.3	2.5
Standard_D8s_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	7	148.3	1.9
Standard_D8s_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	155.4	5.6

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D16_s_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	3	275.7	5.1
Standard_D16_s_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	38	298.2	4.4
Standard_D32_s_v3	32	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	24	545.8	10.5
Standard_D32_s_v3	32	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	535.6	12.6
Standard_D64_s_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	35	1070.6	2.4

Dv3 - General Compute

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D2_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	10	38.6	1.8
Standard_D2_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	24	41.8	3.3
Standard_D4_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	77.8	1.3
Standard_D4_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	45	82.7	4.5
Standard_D8_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	146.7	10.4

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D8_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	159.9	8.3
Standard_D16_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	10	274.1	3.8
Standard_D16_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	300.7	8.8
Standard_D32_v3	32	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	24	549.3	11.1
Standard_D32_v3	32	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	7	538.6	9.4
Standard_D64_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	1070.6	12.4

DSv2 - Storage Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_DS1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	12	33.0	1.1
Standard_DS1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	37	33.8	2.5
Standard_DS2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	33	63.9	1.7
Standard_DS2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	66.6	4.8

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_DS3_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	15	125.5	3.2
Standard_DS3_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	47	130.1	4.3
Standard_DS4_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	23	235.7	6.6
Standard_DS4_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	34	249.4	2.8
Standard_DS5_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	11	414.9	5.1
Standard_DS5_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	470.6	5.7
Standard_DS1_1_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	22	66.3	2.8
Standard_DS1_1_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	34	64.8	2.8
Standard_DS1_1-1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	33.6	1.8
Standard_DS1_1-1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	41	36.0	1.7
Standard_DS1_2_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	10	126.8	2.7

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_DS1_2_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	30	127.5	3.3
Standard_DS1_2-1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	33.5	1.4
Standard_DS1_2-1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	30	34.8	2.4
Standard_DS1_2-2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	65.5	2.3
Standard_DS1_2-2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	33	67.7	5.1
Standard_DS1_3_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	234.1	7.1
Standard_DS1_3_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	23	248.0	2.2
Standard_DS1_3-2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	65.2	3.1
Standard_DS1_3-2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	15	72.8	3.8
Standard_DS1_3-4_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	24	126.1	4.3
Standard_DS1_3-4_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	133.3	2.8

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_DS1_4_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	22	469.5	6.9
Standard_DS1_4_v2	16	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	456.6	7.3
Standard_DS1_4-4_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	132.8	6.6
Standard_DS1_4-4_v2	4	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	125.1	4.8
Standard_DS1_4-8_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	251.3	2.4
Standard_DS1_4-8_v2	8	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	14	247.4	10.2
Standard_DS1_5_v2	20	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	45	546.1	10.5

Dv2 - General Compute

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	30	33.5	1.7
Standard_D1_v2	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	34.7	2.5
Standard_D2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	18	66.0	1.8

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D2_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	69.9	5.0
Standard_D3_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	27	127.7	3.0
Standard_D3_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	133.4	9.1
Standard_D4_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	15	238.7	4.4
Standard_D4_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	36	248.9	4.8
Standard_D5_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	413.9	14.1
Standard_D5_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	470.2	8.1
Standard_D5_v2	16	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	5	466.0	0.0
Standard_D11_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	22	66.4	2.9
Standard_D11_v2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	69.0	6.7
Standard_D12_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	24	127.7	4.6

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_D12_v2	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	135.9	9.3
Standard_D13_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	237.4	6.6
Standard_D13_v2	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	250.2	3.8
Standard_D14_v2	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	23	473.0	9.4
Standard_D14_v2	16	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	443.9	18.8
Standard_D15_v2	20	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	37	558.8	8.4

Esv3 - Memory Optimized + Premium Storage

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_E2s_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	39	42.5	2.2
Standard_E4s_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	81.4	3.3
Standard_E8s_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	29	156.3	5.1
Standard_E8-2s_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	57	41.8	2.6

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_E8-4s_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	45	82.9	3.0
Standard_E16s_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	295.7	4.5
Standard_E16-4s_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	45	82.7	3.8
Standard_E16-8s_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	39	158.3	4.5
Standard_E20s_v3	20	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	27	369.7	3.2
Standard_E32s_v3	32	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	577.9	9.4
Standard_E32-8s_v3	8	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	163.4	6.8
Standard_E32-16s_v3	16	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	41	307.1	8.7
Standard_E4-2s_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	65	41.9	2.4
Standard_E64s_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1080.0	0.0
Standard_E64-16s_v3	16	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	3	334.3	1.5

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_E64-32s_v3	32	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	592.5	4.4

Eisv3 - Memory Opt + Premium Storage (isolated)

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_E64i_s_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	1073.9	5.7

Ev3 - Memory Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_E2_v3	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	41	41.2	2.4
Standard_E4_v3	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	43	81.4	5.3
Standard_E8_v3	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	39	157.4	8.1
Standard_E16_v3	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	49	301.6	8.9
Standard_E20_v3	20	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	35	371.0	6.9
Standard_E32_v3	32	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	35	579.9	16.1
Standard_E64_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	31	1080.0	11.3

Eiv3 - Memory Optimized (isolated)

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_E64i_v3	64	2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	28	1081.4	11.1

Fsv2 - Compute + Storage Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_F2s_v2	2	1	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	46	56.5	2.4
Standard_F4s_v2	4	1	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	60	110.2	4.7
Standard_F8s_v2	8	1	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	36	215.2	5.3
Standard_F16s_v2	16	1	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	36	409.3	15.5
Standard_F32s_v2	32	1	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	31	760.9	16.9
Standard_F64s_v2	64	2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	33	1440.9	26.0
Standard_F72s_v2	72	2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	29	1372.1	8.2

Fs - Compute and Storage Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_F1s	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	31	33.2	1.0
Standard_F1s	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	41	35.1	2.0
Standard_F2s	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	18	63.7	1.8
Standard_F2s	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	21	66.6	3.8
Standard_F4s	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	14	128.4	2.9
Standard_F4s	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	25	127.7	4.5
Standard_F8s	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	11	234.9	3.7
Standard_F8s	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	19	251.2	4.5
Standard_F16s	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	9	413.9	3.6
Standard_F16s	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	36	471.8	7.5

F - Compute Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
------	-------	------------	-----	------	---------------	--------

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_F1	1	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	15	32.8	1.8
Standard_F1	1	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	13	33.3	2.0
Standard_F2	2	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	27	64.9	6.0
Standard_F2	2	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	21	67.8	4.9
Standard_F4	4	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	18	128.4	3.3
Standard_F4	4	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	132.1	7.8
Standard_F8	8	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	17	239.4	2.3
Standard_F8	8	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	25	251.2	7.0
Standard_F16	16	1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	19	424.1	8.2
Standard_F16	16	1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	467.8	11.1
Standard_F16	16	2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	6	472.3	13.2

GS - Storage Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_GS1	2	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	29	63.6	4.7
Standard_GS2	4	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	29	122.3	6.9
Standard_GS3	8	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	222.4	8.1
Standard_GS4	16	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	391.4	28.6
Standard_GS4-4	4	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	28	127.5	5.3
Standard_GS4-8	8	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	226.7	5.8
Standard_GS5	32	2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	760.9	6.2
Standard_GS5-8	8	2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	259.5	2.7
Standard_GS5-16	16	2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	447.9	4.0

G - Compute Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_G1	2	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	29	64.7	9.2

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_G2	4	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	30	127.9	12.2
Standard_G3	8	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	30	231.7	12.6
Standard_G4	16	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	400.2	3.9
Standard_G5	32	2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	774.1	4.1

H - High Performance Compute (HPC)

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_H8	8	1	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	31	296.1	1.4
Standard_H8m	8	1	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	34	295.1	1.5
Standard_H16	16	2	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	19	563.5	4.3
Standard_H16m	16	2	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	19	562.9	3.3
Standard_H16mr	16	2	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	18	563.6	3.7
Standard_H16r	16	2	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz	17	562.2	4.2

Ls - Storage Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_L4s	4	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	29	122.7	6.6
Standard_L8s	8	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	30	223.3	7.5
Standard_L16s	16	1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	397.3	2.5
Standard_L32s	32	2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	31	766.1	3.5

M - Memory Optimized

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_M8-2ms	2	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	15	42.1	2.1
Standard_M8-4ms	4	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	81.6	2.9
Standard_M16-4ms	4	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	14	82.5	2.5
Standard_M16-8ms	8	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	20	157.2	6.0
Standard_M32-8ms	8	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	18	162.5	2.1
Standard_M32-16ms	16	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	306.5	0.5

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_M6_4	64	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	11	1010.9	5.4
Standard_M6_4-16ms	16	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	316.0	2.4
Standard_M6_4-32ms	32	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	586.8	5.4
Standard_M6_4m	64	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1005.5	12.3
Standard_M6_4ms	64	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1012.9	12.5
Standard_M6_4s	64	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1012.5	4.5
Standard_M1_28	128	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	11	1777.3	15.6
Standard_M1_28-32ms	32	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	620.5	2.5
Standard_M1_28-64ms	64	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1140.8	2.9
Standard_M1_28m	128	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1778.3	10.3
Standard_M1_28ms	128	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	15	1780.7	18.3

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_M1 28s	128	4	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	1775.8	11.6
Standard_M1 6ms	16	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	20	293.1	11.8
Standard_M3 2ls	32	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	535.2	4.8
Standard_M3 2ms	32	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	11	534.1	4.6
Standard_M3 2ms	32	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	1	589.0	0.0
Standard_M3 2ts	32	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	12	538.6	3.2
Standard_M6 4ls	64	2	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	1015.2	10.0
Standard_M8 ms	8	1	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	13	158.2	5.5

NCSv3 - GPU Enabled

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_NC6 s_v3	6	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	6	230.2	1.6
Standard_NC1 2s_v3	12	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	7	425.0	3.6

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_NC2_4rs_v3	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	2	811.0	4.2
Standard_NC2_4s_v3	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	3	809.3	2.3

NCSv2 - GPU Enabled

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_NC6_s_v2	6	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	11	227.0	6.2
Standard_NC1_2s_v2	12	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	9	427.3	1.3
Standard_NC2_4rs_v2	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	12	811.0	5.4
Standard_NC2_4s_v2	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	11	811.5	4.4

NC - GPU Enabled

SIZE	VCPUS	NUMA NODES	CPU	RUNS	AVG BASE RATE	STDDEV
Standard_NC6	6	1	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	27	209.6	4.4
Standard_NC1_2	12	1	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	28	394.4	3.8
Standard_NC2_4	24	2	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	28	751.7	3.5

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_NC2_4r	24	2	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	27	752.9	3.4

NDs- GPU Enabled

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_ND6_s	6	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	8	230.1	1.2
Standard_ND1_2s	12	1	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	11	426.5	1.4
Standard_ND2_4rs	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	10	811.4	3.5
Standard_ND2_4s	24	2	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	11	812.6	4.4

NV - GPU Enabled

SIZE	VCPUS	NUMA NODES	CPU	RUNS	Avg Base Rate	StdDev
Standard_NV6	6	1	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	28	210.5	6.1
Standard_NV1_2	12	1	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	28	394.5	2.3
Standard_NV2_4	24	2	Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz	26	752.2	4.4

About SPECint

Windows numbers were computed by running [SPECint 2006](#) on Windows Server. SPECint was run using the base rate option (SPECint_rate2006), with one copy per vCPU. SPECint consists of 12 separate tests, each run three times, taking the median value from each test and weighting them to form a composite score. Those tests

were then run across multiple VMs to provide the average scores shown.

Next steps

- For storage capacities, disk details, and additional considerations for choosing among VM sizes, see [Sizes for virtual machines](#).

Virtual machine vCPU quotas

1/10/2020 • 2 minutes to read • [Edit Online](#)

The vCPU quotas for virtual machines and virtual machine scale sets are arranged in two tiers for each subscription, in each region. The first tier is the Total Regional vCPUs, and the second tier is the various VM size family cores such as the D-series vCPUs. Any time a new VM is deployed the vCPUs for the VM must not exceed the vCPU quota for the VM size family or the total regional vCPU quota. If either of those quotas are exceeded, the VM deployment will not be allowed. There is also a quota for the overall number of virtual machines in the region. The details on each of these quotas can be seen in the **Usage + quotas** section of the **Subscription** page in the [Azure portal](#), or you can query for the values using PowerShell.

Check usage

You can use the [Get-AzVMUsage](#) cmdlet to check on your quota usage.

```
Get-AzVMUsage -Location "East US"
```

The output will look similar to this:

Name	Current	Value	Limit	Unit
Availability Sets	0	2000	Count	
Total Regional vCPUs	4	260	Count	
Virtual Machines	4	10000	Count	
Virtual Machine Scale Sets	1	2000	Count	
Standard B Family vCPUs	1	10	Count	
Standard DSV2 Family vCPUs	1	100	Count	
Standard Dv2 Family vCPUs	2	100	Count	
Basic A Family vCPUs	0	100	Count	
Standard A0-A7 Family vCPUs	0	250	Count	
Standard A8-A11 Family vCPUs	0	100	Count	
Standard D Family vCPUs	0	100	Count	
Standard G Family vCPUs	0	100	Count	
Standard DS Family vCPUs	0	100	Count	
Standard GS Family vCPUs	0	100	Count	
Standard F Family vCPUs	0	100	Count	
Standard FS Family vCPUs	0	100	Count	
Standard NV Family vCPUs	0	24	Count	
Standard NC Family vCPUs	0	48	Count	
Standard H Family vCPUs	0	8	Count	
Standard Av2 Family vCPUs	0	100	Count	
Standard LS Family vCPUs	0	100	Count	
Standard Dv2 Promo Family vCPUs	0	100	Count	
Standard DSV2 Promo Family vCPUs	0	100	Count	
Standard MS Family vCPUs	0	0	Count	
Standard Dv3 Family vCPUs	0	100	Count	
Standard DSV3 Family vCPUs	0	100	Count	
Standard Ev3 Family vCPUs	0	100	Count	
Standard ESv3 Family vCPUs	0	100	Count	
Standard FSv2 Family vCPUs	0	100	Count	
Standard ND Family vCPUs	0	0	Count	
Standard NCv2 Family vCPUs	0	0	Count	
Standard NCv3 Family vCPUs	0	0	Count	
Standard LSv2 Family vCPUs	0	0	Count	
Standard Storage Managed Disks	2	10000	Count	
Premium Storage Managed Disks	1	10000	Count	

Reserved VM Instances

Reserved VM Instances, which are scoped to a single subscription without VM size flexibility, will add a new aspect to the vCPU quotas. These values describe the number of instances of the stated size that must be deployable in the subscription. They work as a placeholder in the quota system to ensure that quota is reserved to ensure reserved VM instances are deployable in the subscription. For example, if a specific subscription has 10 Standard_D1 reserved VM instances the usages limit for Standard_D1 reserved VM instances will be 10. This will cause Azure to ensure that there are always at least 10 vCPUs available in the Total Regional vCPUs quota to be used for Standard_D1 instances and there are at least 10 vCPUs available in the Standard D Family vCPU quota to be used for Standard_D1 instances.

If a quota increase is required to purchase a Single Subscription RI, you can [request a quota increase](#) on your subscription.

Next steps

For more information about billing and quotas, see [Azure subscription and service limits, quotas, and constraints](#).

Save costs with Azure Reserved VM Instances

11/12/2019 • 7 minutes to read • [Edit Online](#)

When you commit to an Azure reserved VM instance you can save money. The reservation discount is applied automatically to the number of running virtual machines that match the reservation scope and attributes. You don't need to assign a reservation to a virtual machine to get the discounts. A reserved instance purchase covers only the compute part of your VM usage. For Windows VMs, the usage meter is split into two separate meters. There's a compute meter, which is same as the Linux meter, and a Windows IP meter. The charges that you see when you make the purchase are only for the compute costs. Charges don't include Windows software costs. For more information about software costs, see [Software costs not included with Azure Reserved VM Instances](#).

Determine the right VM size before you buy

Before you buy a reservation, you should determine the size of the VM that you need. The following sections will help you determine the right VM size.

Use reservation recommendations

You can use reservation recommendations to help determine the reservations you should purchase.

- Purchase recommendations and recommended quantity are show when you purchase a VM reserved instance in the Azure portal.
- Azure Advisor provides purchase recommendations for individual subscriptions.
- You can use the APIs to get purchase recommendations for both shared scope and single subscription scope. For more information, see [Reserved instance purchase recommendation APIs for enterprise customers](#).
- For Enterprise Agreement (EA) and Microsoft Customer Agreement (MCA) customers, purchase recommendations for shared and single subscription scopes are available with the [Azure Consumption Insights Power BI content pack](#).

Services that get VM reservation discounts

Your VM reservations can apply to VM usage emitted from multiple services - not just for your VM deployments. Resources that get reservation discounts change depending on the instance size flexibility setting.

Instance size flexibility setting

The instance size flexibility setting determines which services get the reserved instance discounts.

Whether the setting is on or off, reservation discounts automatically apply to any matching VM usage when the *ConsumedService* is `Microsoft.Compute`. So, check your usage data for the *ConsumedService* value. Some examples include:

- Virtual machines
- Virtual machine scale sets
- Container service
- Azure Batch deployments (in user subscriptions mode)
- Azure Kubernetes Service (AKS)
- Service Fabric

When the setting is on, reservation discounts automatically apply to matching VM usage when the *ConsumedService* is any of the following items:

- `Microsoft.Compute`
- `Microsoft.ClassicCompute`

- Microsoft.Batch
- Microsoft.MachineLearningServices
- Microsoft.Kusto

Check the *ConsumedService* value in your usage data to determine if the usage is eligible for reservation discounts.

For more information about instance size flexibility, see [Virtual machine size flexibility with Reserved VM Instances](#).

Analyze your usage information

Analyze your usage information to help determine which reservations you should purchase.

Usage data is available in the usage file and APIs. Use them together to determine which reservation to purchase. Check for VM instances that have high usage on daily basis to determine the quantity of reservations to purchase.

Avoid the `Meter` subcategory and `Product` fields in usage data. They don't distinguish between VM sizes that use premium storage. If you use these fields to determine the VM size for reservation purchase, you may buy the wrong size. Then you won't get the reservation discount you expect. Instead, refer to the `AdditionalInfo` field in your usage file or usage API to determine the correct VM size.

Purchase restriction considerations

Reserved VM Instances are available for most VM sizes with some exceptions. Reservation discounts don't apply for the following VMs:

- **VM series** - A-series, Av2-series, or G-series.
- **Preview or Promo VMs** - Any VM-series or size that is in preview or uses promotional meter.
- **Clouds** - Reservations aren't available for purchase in Germany or China regions.
- **Insufficient quota** - A reservation that is scoped to a single subscription must have vCPU quota available in the subscription for the new RI. For example, if the target subscription has a quota limit of 10 vCPUs for D-Series, then you can't buy a reservation for 11 Standard_D1 instances. The quota check for reservations includes the VMs already deployed in the subscription. For example, if the subscription has a quota of 10 vCPUs for D-Series and has two standard_D1 instances deployed, then you can buy a reservation for 10 standard_D1 instances in this subscription. You can [create quote increase request](#) to resolve this issue.
- **Capacity restrictions** - In rare circumstances, Azure limits the purchase of new reservations for subset of VM sizes, because of low capacity in a region.

Buy a Reserved VM Instance

You can buy a reserved VM instance in the [Azure portal](#). Pay for the reservation [up front or with monthly payments](#). These requirements apply to buying a reserved VM instance:

- You must be in an Owner role for at least one EA subscription or a subscription with a pay-as-you-go rate.
- For EA subscriptions, the **Add Reserved Instances** option must be enabled in the [EA portal](#). Or, if that setting is disabled, you must be an EA Admin for the subscription.
- For the Cloud Solution Provider (CSP) program, only the admin agents or sales agents can buy reservations.

To buy an instance:

1. Sign in to the [Azure portal](#).
2. Select **All services > Reservations**.
3. Select **Add** to purchase a new reservation and then click **Virtual machine**.
4. Enter required fields. Running VM instances that match the attributes you select qualify to get the reservation discount. The actual number of your VM instances that get the discount depend on the scope and quantity

selected.

If you have an EA agreement, you can use the **Add more option** to quickly add additional instances. The option isn't available for other subscription types.

FIELD	DESCRIPTION
Subscription	The subscription used to pay for the reservation. The payment method on the subscription is charged the costs for the reservation. The subscription type must be an enterprise agreement (offer numbers: MS-AZR-0017P or MS-AZR-0148P) or Microsoft Customer Agreement or an individual subscription with pay-as-you-go rates (offer numbers: MS-AZR-0003P or MS-AZR-0023P). The charges are deducted from the monetary commitment balance, if available, or charged as overage. For a subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.
Scope	The reservation's scope can cover one subscription or multiple subscriptions (shared scope). If you select: <ul style="list-style-type: none">• Single resource group scope — Applies the reservation discount to the matching resources in the selected resource group only.• Single subscription scope — Applies the reservation discount to the matching resources in the selected subscription.• Shared scope — Applies the reservation discount to matching resources in eligible subscriptions that are in the billing context. For EA customers, the billing context is the enrollment. For individual subscriptions with pay-as-you-go rates, the billing scope is all eligible subscriptions created by the account administrator.
Region	The Azure region that's covered by the reservation.
VM Size	The size of the VM instances.
Optimize for	VM instance size flexibility is selected by default. Click Advanced settings to change the instance size flexibility value to apply the reservation discount to other VMs in the same VM size group . Capacity priority prioritizes data center capacity for your deployments. It offers additional confidence in your ability to launch the VM instances when you need them. Capacity priority is only available when the reservation scope is single subscription.
Term	One year or three years.
Quantity	The number of instances being purchased within the reservation. The quantity is the number of running VM instances that can get the billing discount. For example, if you are running 10 Standard_D2 VMs in the East US, then you would specify quantity as 10 to maximize the benefit for all running VMs.

Usage data and reservation utilization

Your usage data has an effective price of zero for the usage that gets a reservation discount. You can see which VM instance received the reservation discount for each reservation.

For more information about how reservation discounts appear in usage data, see [Understand Azure reservation usage for your Enterprise enrollment](#) if you are an EA customer. If you have an individual subscription, see [Understand Azure reservation usage for your Pay-As-You-Go subscription](#).

Change a reservation after purchase

You can make the following types of changes to a reservation after purchase:

- Update reservation scope
- Instance size flexibility (if applicable)
- Ownership

You can also split a reservation into smaller chunks and merge already split reservations. None of the changes cause a new commercial transaction or change the end date of the reservation.

You can't make the following types of changes after purchase, directly:

- An existing reservation's region
- SKU
- Quantity
- Duration

However, you can *exchange* a reservation if you want to make changes.

Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations with certain limitations. For more information, see [Self-service exchanges and refunds for Azure Reservations](#).

Need help? Contact us.

If you have questions or need help, [create a support request](#).

Next steps

- To learn how to manage a reservation, see [Manage Azure Reservations](#).
- To learn more about Azure Reservations, see the following articles:
 - [What are Azure Reservations?](#)
 - [Manage Reservations in Azure](#)
 - [Understand how the reservation discount is applied](#)
 - [Understand reservation usage for a subscription with pay-as-you-go rates](#)
 - [Understand reservation usage for your Enterprise enrollment](#)
 - [Windows software costs not included with reservations](#)
 - [Azure Reservations in Partner Center Cloud Solution Provider \(CSP\) program](#)

2 minutes to read

Virtual machine size flexibility with Reserved VM Instances

2/19/2020 • 2 minutes to read • [Edit Online](#)

When you buy a Reserved VM Instance, you can choose to optimize for instance size flexibility or capacity priority. For more information about setting or changing the optimize setting for reserved VM instances, see [Change the optimize setting for reserved VM instances](#).

With a reserved virtual machine instance that's optimized for instance size flexibility, the reservation you buy can apply to the virtual machines (VMs) sizes in the same instance size flexibility group. For example, if you buy a reservation for a VM size that's listed in the DSv2 Series, like Standard_DS5_v2, the reservation discount can apply to the other four sizes that are listed in that same instance size flexibility group:

- Standard_DS1_v2
- Standard_DS2_v2
- Standard_DS3_v2
- Standard_DS4_v2

But that reservation discount doesn't apply to VMs sizes that are listed in different instance size flexibility groups, like SKUs in DSv2 Series High Memory: Standard_DS11_v2, Standard_DS12_v2, and so on.

Within the instance size flexibility group, the number of VMs the reservation discount applies to depends on the VM size you pick when you buy a reservation. It also depends on the sizes of the VMs that you have running. The ratio column compares the relative footprint for each VM size in that instance size flexibility group. Use the ratio value to calculate how the reservation discount applies to the VMs you have running.

Examples

The following examples use the sizes and ratios in the DSv2-series table.

You buy a reserved VM instance with the size Standard_DS4_v2 where the ratio or relative footprint compared to the other sizes in that series is 8.

- Scenario 1: Run eight Standard_DS1_v2 sized VMs with a ratio of 1. Your reservation discount applies to all eight of those VMs.
- Scenario 2: Run two Standard_DS2_v2 sized VMs with a ratio of 2 each. Also run a Standard_DS3_v2 sized VM with a ratio of 4. The total footprint is $2+2+4=8$. So your reservation discount applies to all three of those VMs.
- Scenario 3: Run one Standard_DS5_v2 with a ratio of 16. Your reservation discount applies to half that VM's compute cost.

The following sections show what sizes are in the same size series group when you buy a reserved VM instance optimized for instance size flexibility.

Instance size flexibility ratio for VMs

CSV below has the instance size flexibility groups, ArmSkuName and the ratios.

Instance size flexibility ratios

We will keep the file URL and the schema fixed so you can consume this file programmatically. The data will also be available through API soon.

Azure Dedicated Hosts

1/9/2020 • 7 minutes to read • [Edit Online](#)

Azure Dedicated Host is a service that provides physical servers - able to host one or more virtual machines - dedicated to one Azure subscription. Dedicated hosts are the same physical servers used in our data centers, provided as a resource. You can provision dedicated hosts within a region, availability zone, and fault domain. Then, you can place VMs directly into your provisioned hosts, in whatever configuration best meets your needs.

Limitations

- Virtual machine scale sets are not currently supported on dedicated hosts.
- The following VM series are supported: DSv3, ESv3 and Fsv2.

Benefits

Reserving the entire host provides the following benefits:

- Hardware isolation at the physical server level. No other VMs will be placed on your hosts. Dedicated hosts are deployed in the same data centers and share the same network and underlying storage infrastructure as other, non-isolated hosts.
- Control over maintenance events initiated by the Azure platform. While the majority of maintenance events have little to no impact on your virtual machines, there are some sensitive workloads where each second of pause can have an impact. With dedicated hosts, you can opt-in to a maintenance window to reduce the impact to your service.
- With the Azure hybrid benefit, you can bring your own licenses for Windows and SQL to Azure. Using the hybrid benefits provides you with additional benefits. For more information, see [Azure Hybrid Benefit](#).

Groups, hosts, and VMs



A **host group** is a resource that represents a collection of dedicated hosts. You create a host group in a region and an availability zone, and add hosts to it.

A **host** is a resource, mapped to a physical server in an Azure data center. The physical server is allocated when the host is created. A host is created within a host group. A host has a SKU describing which VM sizes can be created. Each host can host multiple VMs, of different sizes, as long as they are from the same size series.

When creating a VM in Azure, you can select which dedicated host to use for your VM. You have full control as to which VMs are placed on your hosts.

High Availability considerations

For high availability, you should deploy multiple VMs, spread across multiple hosts (minimum of 2). With Azure Dedicated Hosts, you have several options to provision your infrastructure to shape your fault isolation

boundaries.

Use Availability Zones for fault isolation

Availability zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. A host group is created in a single availability zone. Once created, all hosts will be placed within that zone. To achieve high availability across zones, you need to create multiple host groups (one per zone) and spread your hosts accordingly.

If you assign a host group to an availability zone, all VMs created on that host must be created in the same zone.

Use Fault Domains for fault isolation

A host can be created in a specific fault domain. Just like VM in a scale set or availability set, hosts in different fault domains will be placed on different physical racks in the data center. When you create a host group, you are required to specify the fault domain count. When creating hosts within the host group, you assign fault domain for each host. The VMs do not require any fault domain assignment.

Fault domains are not the same as collocation. Having the same fault domain for two hosts does not mean they are in proximity with each other.

Fault domains are scoped to the host group. You should not make any assumption on anti-affinity between two host groups (unless they are in different availability zones).

VMs deployed to hosts with different fault domains, will have their underlying managed disks services on multiple storage stamps, to increase the fault isolation protection.

Using Availability Zones and Fault Domains

You can use both capabilities together to achieve even more fault isolation. In this case, you will specify the availability zone and fault domain count in for each host group, assign a fault domain to each of your hosts in the group, and assign an availability zone to each of your VMs

The Resource Manager sample template found [here](#) uses zones and fault domains to spread hosts for maximum resiliency in a region.

Maintenance control

The infrastructure supporting your virtual machines may occasionally be updated to improve reliability, performance, security, and to launch new features. The Azure platform tries to minimize the impact of platform maintenance whenever possible, but customers with *maintenance sensitive* workloads can't tolerate even few seconds that the VM needs to be frozen or disconnected for maintenance.

Maintenance Control provides customers with an option to skip regular platform updates scheduled on their dedicated hosts, then apply it at the time of their choice within a 35-day rolling window.

NOTE

Maintenance control is currently in public preview. For more information, see [Control updates with Maintenance Control using CLI or PowerShell](#).

Capacity considerations

Once a dedicated host is provisioned, Azure assigns it to physical server. This guarantees the availability of the capacity when you need to provision your VM. Azure uses the entire capacity in the region (or zone) to pick a physical server for your host. It also means that customers can expect to be able to grow their dedicated host footprint without the concern of running out of space in the cluster.

Quotas

There is a default quota limit of 3000 vCPUs for dedicated hosts, per region. But, the number of hosts you can deploy is also limited by the quota for the VM size family used for the host. For example, a **Pay-as-you-go** subscription may only have a quota of 10 vCPUs available for the Dsv3 size series, in the East US region. In this case, you need to request a quota increase to at least 64 vCPUs before you can deploy a dedicated host. Select the **Request increase** button in the upper right corner to file a request if needed.

QUOTA	PROVIDER	LOCATION	USAGE
Standard DSv3 Family vCPUs	Microsoft.Compute	East US	0 % 0 of 10

For more information, see [Virtual machine vCPU quotas](#).

Free trial and MSDN subscriptions do not have quota for Azure Dedicated Hosts.

Pricing

Users are charged per dedicated host, regardless how many VMs are deployed. In your monthly statement you will see a new billable resource type of hosts. The VMs on a dedicated host will still be shown in your statement, but will carry a price of 0.

The host price is set based on VM family, type (hardware size), and region. A host price is relative to the largest VM size supported on the host.

Software licensing, storage and network usage are billed separately from the host and VMs. There is no change to those billable items.

For more information, see [Azure Dedicated Host pricing](#).

VM families and Hardware generations

A SKU is defined for a host and it represents the VM size series and type. You can mix multiple VMs of different sizes within a single host as long as they are of the same size series. The type is the hardware generation currently available in the region.

Different `types` for the same VM series will be from different CPU vendors and have different CPU generations and number of cores.

Refer to the host [pricing page](#) to learn more.

Dedicated hosts support the following host SKU\types: DSv3_Type1 and ESv3_Type1

Host life cycle

Azure monitors and manages the health status of your hosts. The following states will be returned when you query your host:

Health State	Description
Host Available	There are no known issues with your host.
Host Under Investigation	We're having some issues with the host which we're looking into. This is a transitional state required for Azure to try and identify the scope and root cause for the issue identified. Virtual machines running on the host may be impacted.
Host Pending Deallocate	Azure can't restore the host back to a healthy state and ask you to redeploy your virtual machines out of this host. If <code>autoReplaceOnFailure</code> is enabled, your virtual machines are <i>service healed</i> to healthy hardware. Otherwise, your virtual machine may be running on a host that is about to fail.
Host deallocated	All virtual machines have been removed from the host. You are no longer being charged for this host since the hardware was taken out of rotation.

Next steps

- You can deploy a dedicated host using [Azure PowerShell](#), the [portal](#), and [Azure CLI](#).
- There is sample template, found [here](#), that uses both zones and fault domains for maximum resiliency in a region.

Maintenance for virtual machines in Azure

2/10/2020 • 6 minutes to read • [Edit Online](#)

Azure periodically updates its platform to improve the reliability, performance, and security of the host infrastructure for virtual machines. The purpose of these updates ranges from patching software components in the hosting environment to upgrading networking components or decommissioning hardware.

Updates rarely affect the hosted VMs. When updates do have an effect, Azure chooses the least impactful method for updates:

- If the update doesn't require a reboot, the VM is paused while the host is updated, or the VM is live-migrated to an already updated host.
- If maintenance requires a reboot, you're notified of the planned maintenance. Azure also provides a time window in which you can start the maintenance yourself, at a time that works for you. The self-maintenance window is typically 30 days unless the maintenance is urgent. Azure is investing in technologies to reduce the number of cases in which planned platform maintenance requires the VMs to be rebooted. For instructions on managing planned maintenance, see [Handling planned maintenance notifications using the Azure CLI](#), [PowerShell](#) or [portal](#).

This page describes how Azure performs both types of maintenance. For more information about unplanned events (outages), see [Manage the availability of VMs for Windows](#) or the corresponding article for [Linux](#).

Within a VM, you can get notifications about upcoming maintenance by [using Scheduled Events for Windows](#) or for [Linux](#).

Maintenance that doesn't require a reboot

Most platform updates don't affect customer VMs. When a no-impact update isn't possible, Azure chooses the update mechanism that's least impactful to customer VMs.

Most nonzero-impact maintenance pauses the VM for less than 10 seconds. In certain cases, Azure uses memory-preserving maintenance mechanisms. These mechanisms pause the VM for up to 30 seconds and preserve the memory in RAM. The VM is then resumed, and its clock is automatically synchronized.

Memory-preserving maintenance works for more than 90 percent of Azure VMs. It doesn't work for G, M, N, and H series. Azure increasingly uses live-migration technologies and improves memory-preserving maintenance mechanisms to reduce the pause durations.

These maintenance operations that don't require a reboot are applied one fault domain at a time. They stop if they receive any warning health signals.

These types of updates can affect some applications. When the VM is live-migrated to a different host, some sensitive workloads might show a slight performance degradation in the few minutes leading up to the VM pause. To prepare for VM maintenance and reduce impact during Azure maintenance, try [using Scheduled Events for Windows](#) or [Linux](#) for such applications.

There is also a feature, maintenance control, in public preview that can help manage maintenance that doesn't require a reboot. You must be using either [Azure Dedicated Hosts](#) or an [isolated VM](#). Maintenance control gives you the option to skip platform updates and apply the updates at your choice of time within a 35-day rolling window. For more information, see [Control updates with Maintenance Control and the Azure CLI](#).

Live migration

Live migration is an operation that doesn't require a reboot and that preserves memory for the VM. It causes a

pause or freeze, typically lasting no more than 5 seconds. Except for G, M, N, and H series, all infrastructure as a service (IaaS) VMs, are eligible for live migration. Eligible VMs represent more than 90 percent of the IaaS VMs that are deployed to the Azure fleet.

The Azure platform starts live migration in the following scenarios:

- Planned maintenance
- Hardware failure
- Allocation optimizations

Some planned-maintenance scenarios use live migration, and you can use Scheduled Events to know in advance when live migration operations will start.

Live migration can also be used to move VMs when Azure Machine Learning algorithms predict an impending hardware failure or when you want to optimize VM allocations. For more information about predictive modeling that detects instances of degraded hardware, see [Improving Azure VM resiliency with predictive machine learning and live migration](#). Live-migration notifications appear in the Azure portal in the Monitor and Service Health logs as well as in Scheduled Events if you use these services.

Maintenance that requires a reboot

In the rare case where VMs need to be rebooted for planned maintenance, you'll be notified in advance. Planned maintenance has two phases: the self-service phase and a scheduled maintenance phase.

During the *self-service phase*, which typically lasts four weeks, you start the maintenance on your VMs. As part of the self-service, you can query each VM to see its status and the result of your last maintenance request.

When you start self-service maintenance, your VM is redeployed to an already updated node. Because the VM reboots, the temporary disk is lost and dynamic IP addresses associated with the virtual network interface are updated.

If an error arises during self-service maintenance, the operation stops, the VM isn't updated, and you get the option to retry the self-service maintenance.

When the self-service phase ends, the *scheduled maintenance phase* begins. During this phase, you can still query for the maintenance phase, but you can't start the maintenance yourself.

For more information on managing maintenance that requires a reboot, see **Handling planned maintenance notifications** using the Azure [CLI](#), [PowerShell](#) or [portal](#).

Availability considerations during scheduled maintenance

If you decide to wait until the scheduled maintenance phase, there are a few things you should consider to maintain the highest availability of your VMs.

Paired regions

Each Azure region is paired with another region within the same geographical vicinity. Together, they make a region pair. During the scheduled maintenance phase, Azure updates only the VMs in a single region of a region pair. For example, while updating the VM in North Central US, Azure doesn't update any VM in South Central US at the same time. However, other regions such as North Europe can be under maintenance at the same time as East US. Understanding how region pairs work can help you better distribute your VMs across regions. For more information, see [Azure region pairs](#).

Availability sets and scale sets

When deploying a workload on Azure VMs, you can create the VMs within an *availability set* to provide high availability to your application. Using availability sets, you can ensure that during either an outage or maintenance events that require a reboot, at least one VM is available.

Within an availability set, individual VMs are spread across up to 20 update domains. During scheduled

maintenance, only one update domain is updated at any given time. Update domains aren't necessarily updated sequentially.

Virtual machine *scale sets* are an Azure compute resource that you can use to deploy and manage a set of identical VMs as a single resource. The scale set is automatically deployed across UD, like VMs in an availability set. As with availability sets, when you use scale sets, only one UD is updated at any given time during scheduled maintenance.

For more information about setting up your VMs for high availability, see [Manage the availability of your VMs for Windows](#) or the corresponding article for [Linux](#).

Availability zones

Availability zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.

An availability zone is a combination of a fault domain and an update domain. If you create three or more VMs across three zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains. The Azure platform recognizes this distribution across update domains to make sure that VMs in different zones are not updated at the same time.

Each infrastructure update rolls out zone by zone, within a single region. But, you can have deployment going on in Zone 1, and different deployment going in Zone 2, at the same time. Deployments are not all serialized. But, a single deployment only rolls out one zone at a time to reduce risk.

Next steps

You can use the [Azure CLI](#), [Azure PowerShell](#) or the [portal](#) to manage planned maintenance.

Introduction to Azure managed disks

12/16/2019 • 9 minutes to read • [Edit Online](#)

Azure managed disks are block-level storage volumes that are managed by Azure and used with Azure Virtual Machines. Managed disks are like a physical disk in an on-premises server but virtualized. With managed disks, all you have to do is specify the disk size, the disk type, and provision the disk. Once you provision the disk, Azure handles the rest.

The available types of disks are ultra disks, premium solid-state drives (SSD), standard SSDs, and standard hard disk drives (HDD). For information about each individual disk type, see [Select a disk type for IaaS VMs](#).

Benefits of managed disks

Let's go over some of the benefits you gain by using managed disks.

Highly durable and available

Managed disks are designed for 99.999% availability. Managed disks achieve this by providing you with three replicas of your data, allowing for high durability. If one or even two replicas experience issues, the remaining replicas help ensure persistence of your data and high tolerance against failures. This architecture has helped Azure consistently deliver enterprise-grade durability for infrastructure as a service (IaaS) disks, with an industry-leading ZERO% annualized failure rate.

Simple and scalable VM deployment

Using managed disks, you can create up to 50,000 VM **disks** of a type in a subscription per region, allowing you to create thousands of **VMs** in a single subscription. This feature also further increases the scalability of [virtual machine scale sets](#) by allowing you to create up to 1,000 VMs in a virtual machine scale set using a Marketplace image.

Integration with availability sets

Managed disks are integrated with availability sets to ensure that the disks of [VMs in an availability set](#) are sufficiently isolated from each other to avoid a single point of failure. Disks are automatically placed in different storage scale units (stamps). If a stamp fails due to hardware or software failure, only the VM instances with disks on those stamps fail. For example, let's say you have an application running on five VMs, and the VMs are in an Availability Set. The disks for those VMs won't all be stored in the same stamp, so if one stamp goes down, the other instances of the application continue to run.

Integration with Availability Zones

Managed disks support [Availability Zones](#), which is a high-availability offering that protects your applications from datacenter failures. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions. With Availability Zones, Azure offers industry best 99.99% VM uptime SLA.

Azure Backup support

To protect against regional disasters, [Azure Backup](#) can be used to create a backup job with time-based backups and backup retention policies. This allows you to perform easy VM restorations at will. Currently Azure Backup supports disk sizes up to four tebibyte (TiB) disks. Azure Backup supports backup and restore of managed disks. [Learn more](#) about Azure VM backup support.

Granular access control

You can use [Azure role-based access control \(RBAC\)](#) to assign specific permissions for a managed disk to one or more users. Managed disks expose a variety of operations, including read, write (create/update), delete, and retrieving a [shared access signature \(SAS\) URI](#) for the disk. You can grant access to only the operations a person needs to perform their job. For example, if you don't want a person to copy a managed disk to a storage account, you can choose not to grant access to the export action for that managed disk. Similarly, if you don't want a person to use an SAS URI to copy a managed disk, you can choose not to grant that permission to the managed disk.

Upload your vhd

Direct upload makes it easy to transfer your vhd to an Azure managed disk. Previously, you had to follow a more involved process that included staging your data in a storage account. Now, there are fewer steps. It is easier to upload on premises VMs to Azure, upload to large managed disks, and the backup and restore process is simplified. It also reduces cost by allowing you to upload data to managed disks directly without attaching them to VMs. You can use direct upload to upload vhds up to 32 TiB in size.

To learn how to transfer your vhd to Azure, see the [CLI](#) or [PowerShell](#) articles.

Encryption

Managed disks offer two different kinds of encryption. The first is Server Side Encryption (SSE), which is performed by the storage service. The second one is Azure Disk Encryption (ADE), which you can enable on the OS and data disks for your VMs.

Server-side encryption

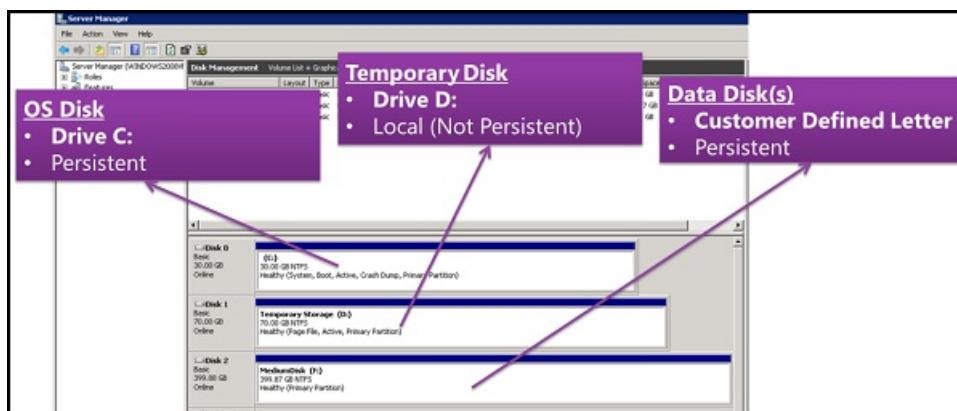
[Azure Server-side Encryption](#) provides encryption-at-rest and safeguards your data to meet your organizational security and compliance commitments. Server-side encryption is enabled by default for all managed disks, snapshots, and images in all the regions where managed disks are available. You can either allow Azure to manage your keys for you, these are platform-managed keys, or you can manage the keys yourself, these are customer-managed keys. Visit the [Managed Disks FAQ page](#) for more details.

Azure Disk Encryption

Azure Disk Encryption allows you to encrypt the OS and Data disks used by an IaaS Virtual Machine. This encryption includes managed disks. For Windows, the drives are encrypted using industry-standard BitLocker encryption technology. For Linux, the disks are encrypted using the DM-Crypt technology. The encryption process is integrated with Azure Key Vault to allow you to control and manage the disk encryption keys. For more information, see [Azure Disk Encryption for IaaS VMs](#).

Disk roles

There are three main disk roles in Azure: the data disk, the OS disk, and the temporary disk. These roles map to disks that are attached to your virtual machine.



Data disk

A data disk is a managed disk that's attached to a virtual machine to store application data, or other data you need to keep. Data disks are registered as SCSI drives and are labeled with a letter that you choose. Each data disk has a maximum capacity of 32,767 gibibytes (GiB). The size of the virtual machine determines how many data disks you can attach to it and the type of storage you can use to host the disks.

OS disk

Every virtual machine has one attached operating system disk. That OS disk has a pre-installed OS, which was selected when the VM was created. This disk contains the boot volume.

This disk has a maximum capacity of 2,048 GiB.

Temporary disk

Every VM contains a temporary disk, which is not a managed disk. The temporary disk provides short-term storage for applications and processes and is intended to only store data such as page or swap files. Data on the temporary disk may be lost during a [maintenance event](#) event or when you [redeploy a VM](#). On Azure Linux VMs, the temporary disk is /dev/sdb by default and on Windows VMs the temporary disk is D: by default. During a successful standard reboot of the VM, the data on the temporary disk will persist.

Managed disk snapshots

A managed disk snapshot is a read-only crash-consistent full copy of a managed disk that is stored as a standard managed disk by default. With snapshots, you can back up your managed disks at any point in time. These snapshots exist independent of the source disk and can be used to create new managed disks.

Snapshots are billed based on the used size. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB, that snapshot is billed only for the used data size of 10 GiB. You can see the used size of your snapshots by looking at the [Azure usage report](#). For example, if the used data size of a snapshot is 10 GiB, the **daily** usage report will show $10 \text{ GiB} / (31 \text{ days}) = 0.3226$ as the consumed quantity.

To learn more about how to create snapshots for managed disks, see the following resources:

- [Create a snapshot of a managed disk in Windows](#)
- [Create a snapshot of a managed disk in Linux](#)

Images

Managed disks also support creating a managed custom image. You can create an image from your custom VHD in a storage account or directly from a generalized (sysprepped) VM. This process captures a single image. This image contains all managed disks associated with a VM, including both the OS and data disks. This managed custom image enables creating hundreds of VMs using your custom image without the need to copy or manage any storage accounts.

For information on creating images, see the following articles:

- [How to capture a managed image of a generalized VM in Azure](#)
- [How to generalize and capture a Linux virtual machine using the Azure CLI](#)

Images versus snapshots

It's important to understand the difference between images and snapshots. With managed disks, you can take an image of a generalized VM that has been deallocated. This image includes all of the disks attached to the VM. You can use this image to create a VM, and it includes all of the disks.

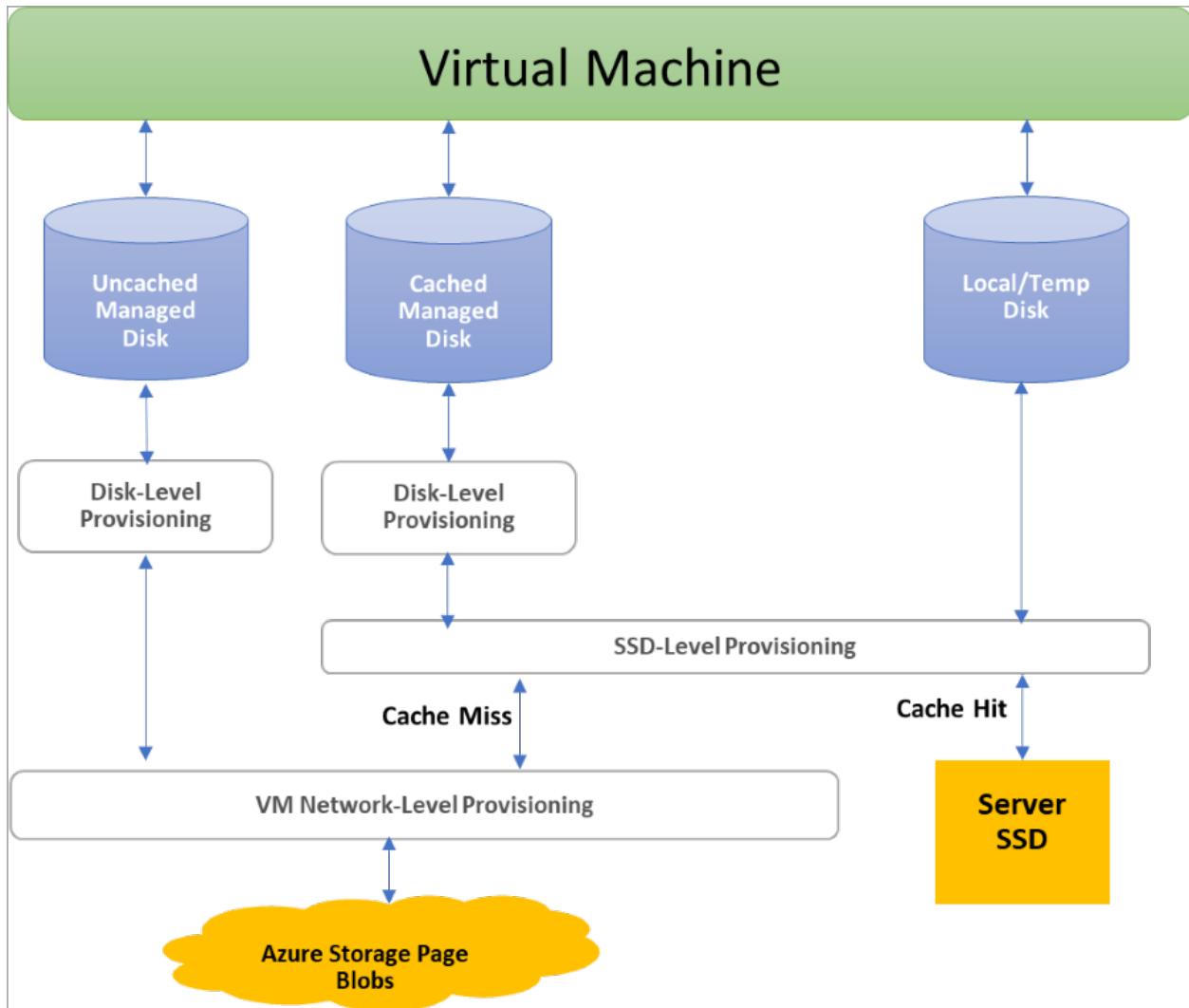
A snapshot is a copy of a disk at the point in time the snapshot is taken. It applies only to one disk. If you have a VM that has one disk (the OS disk), you can take a snapshot or an image of it and create a VM from either the snapshot or the image.

A snapshot doesn't have awareness of any disk except the one it contains. This makes it problematic to use in

scenarios that require the coordination of multiple disks, such as striping. Snapshots would need to be able to coordinate with each other and this is currently not supported.

Disk allocation and performance

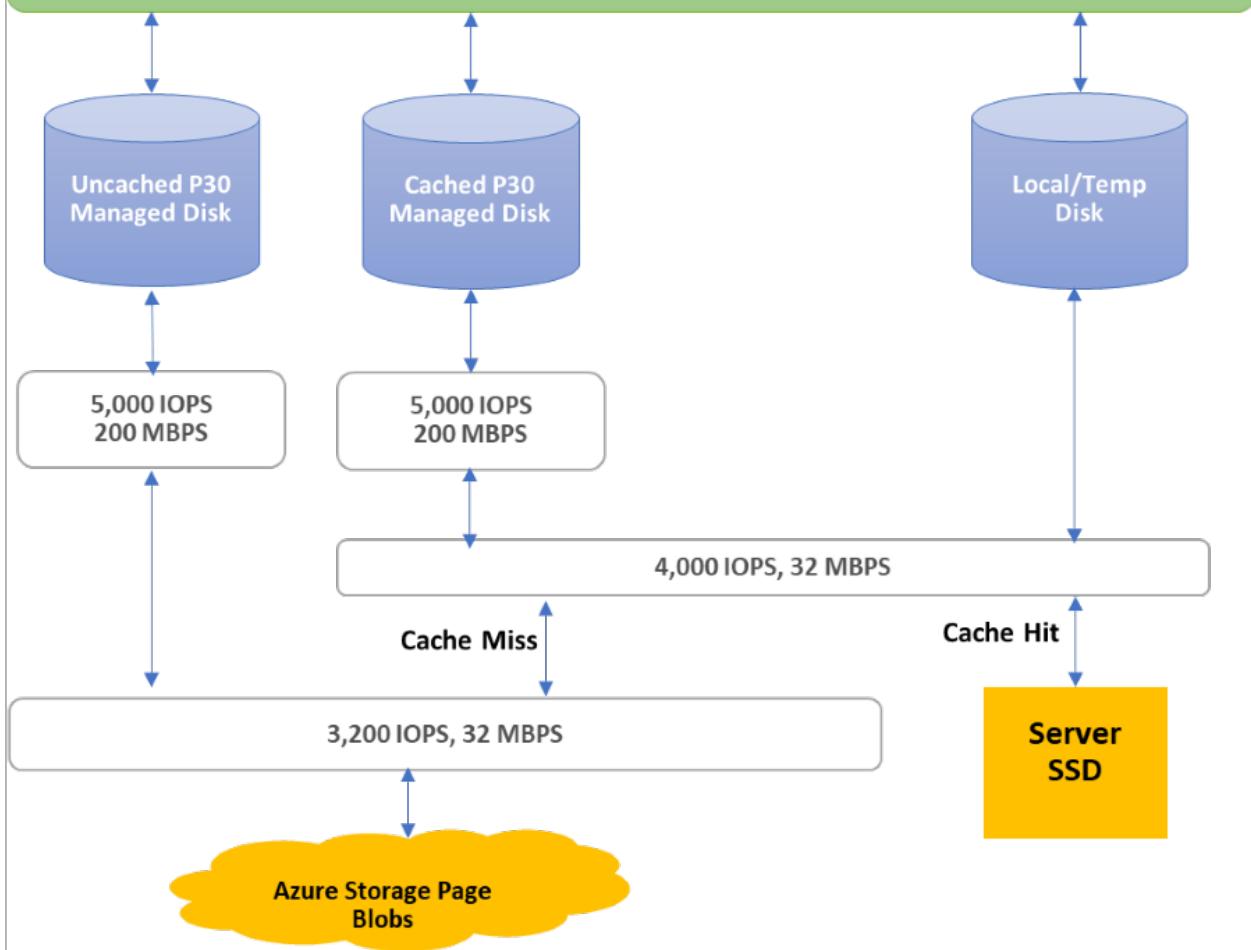
The following diagram depicts real-time allocation of bandwidth and IOPS for disks, using a three-level provisioning system:



The first level provisioning sets the per-disk IOPS and bandwidth assignment. At the second level, compute server host implements SSD provisioning, applying it only to data that is stored on the server's SSD, which includes disks with caching (ReadWrite and ReadOnly) as well as local and temp disks. Finally, VM network provisioning takes place at the third level for any I/O that the compute host sends to Azure Storage's backend. With this scheme, the performance of a VM depends on a variety of factors, from how the VM uses the local SSD, to the number of disks attached, as well as the performance and caching type of the disks it has attached.

As an example of these limitations, a Standard_DS1v1 VM is prevented from achieving the 5,000 IOPS potential of a P30 disk, whether it is cached or not, because of limits at the SSD and network levels:

Virtual Machine: Standard_DS1v1



Azure uses prioritized network channel for disk traffic, which gets the precedence over other low priority of network traffic. This helps disks maintain their expected performance in case of network contentions. Similarly, Azure Storage handles resource contentions and other issues in the background with automatic load balancing. Azure Storage allocates required resources when you create a disk, and applies proactive and reactive balancing of resources to handle the traffic level. This further ensures disks can sustain their expected IOPS and throughput targets. You can use the VM-level and Disk-level metrics to track the performance and setup alerts as needed.

Refer to our [design for high performance](#) article, to learn the best practices for optimizing VM + Disk configurations so that you can achieve your desired performance

Next steps

If you'd like a video going into more detail on managed disks, check out: [Better Azure VM Resiliency with Managed Disks](#).

Learn more about the individual disk types Azure offers, which type is a good fit for your needs, and learn about their performance targets in our article on disk types.

[Select a disk type for IaaS VMs](#)

What disk types are available in Azure?

11/12/2019 • 13 minutes to read • [Edit Online](#)

Azure managed disks currently offers four disk types, each type is aimed towards specific customer scenarios.

Disk comparison

The following table provides a comparison of ultra disks, premium solid-state drives (SSD), standard SSD, and standard hard disk drives (HDD) for managed disks to help you decide what to use.

	ULTRA DISK	PREMIUM SSD	STANDARD SSD	STANDARD HDD
Disk type	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA , top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 gibibyte (GiB)	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	2,000 MiB/s	900 MiB/s	750 MiB/s	500 MiB/s
Max IOPS	160,000	20,000	6,000	2,000

Ultra disk

Azure ultra disks deliver high throughput, high IOPS, and consistent low latency disk storage for Azure IaaS VMs. Some additional benefits of ultra disks include the ability to dynamically change the performance of the disk, along with your workloads, without the need to restart your virtual machines (VM). Ultra disks are suited for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads. Ultra disks can only be used as data disks. We recommend using premium SSDs as OS disks.

Performance

When you provision an ultra disk, you can independently configure the capacity and the performance of the disk. Ultra disks come in several fixed sizes, ranging from 4 GiB up to 64 TiB, and feature a flexible performance configuration model that allows you to independently configure IOPS and throughput.

Some key capabilities of ultra disks are:

- Disk capacity: Ultra disks capacity ranges from 4 GiB up to 64 TiB.
- Disk IOPS: Ultra disks support IOPS limits of 300 IOPS/GiB, up to a maximum of 160 K IOPS per disk. To achieve the IOPS that you provisioned, ensure that the selected Disk IOPS are less than the VM IOPS limit. The minimum IOPS per disk is 2 IOPS/GiB, with an overall baseline minimum of 100 IOPS. For example, if you had a 4 GiB ultra disk, you will have a minimum of 100 IOPS, instead of eight IOPS.
- Disk throughput: With ultra disks, the throughput limit of a single disk is 256 KiB/s for each provisioned IOPS, up to a maximum of 2000 MBps per disk (where MBps = 10^6 Bytes per second). The minimum throughput

per disk is 4KiB/s for each provisioned IOPS, with an overall baseline minimum of 1 MBps.

- Ultra disks support adjusting the disk performance attributes (IOPS and throughput) at runtime without detaching the disk from the virtual machine. Once a disk performance resize operation has been issued on a disk, it can take up to an hour for the change to actually take effect. There is a limit of four performance resize operations during a 24 hour window. It is possible for a performance resize operation to fail due to a lack of performance bandwidth capacity.

Disk size

DISK SIZE (GiB)	IOPS CAP	THROUGHPUT CAP (MBPS)
4	1,200	300
8	2,400	600
16	4,800	1,200
32	9,600	2,000
64	19,200	2,000
128	38,400	2,000
256	76,800	2,000
512	80,000	2,000
1,024-65,536 (sizes in this range increasing in increments of 1 TiB)	160,000	2,000

GA scope and limitations

For now, ultra disks have additional limitations, they are as follows:

- Are supported in the following regions, with a varying number of availability zones per region:
 - East US 2
 - East US
 - West US 2
 - SouthEast Asia
 - North Europe
 - West Europe
 - UK South
- Can only be used with availability zones (availability sets and single VM deployments outside of zones will not have the ability to attach an ultra disk)
- Are only supported on the following VM series:
 - [ESv3](#)
 - [DSv3](#)
 - [FSv2](#)
 - [M](#)
 - [Mv2](#)
- Not every VM size is available in every supported region with ultra disks
- Are only available as data disks and only support 4k physical sector size. Due to the 4K native sector size of Ultra Disk, there are some applications that won't be compatible with ultra disks. One example would be Oracle

Database, which requires release 12.2 or later in order to support ultra disks.

- Can only be created as empty disks
- Do not yet support disk snapshots, VM images, availability sets, and Azure disk encryption
- Do not yet support integration with Azure Backup or Azure Site Recovery
- The current maximum limit for IOPS on GA VMs is 80,000.
- If you would like to participate in a limited preview of a VM that can accomplish 160,000 IOPS with ultra disks, please email UltraDiskFeedback@microsoft.com

If you would like to start using ultra disks, see our article on the subject: [Using Azure ultra disks](#).

Premium SSD

Azure premium SSDs deliver high-performance and low-latency disk support for virtual machines (VMs) with input/output (IO)-intensive workloads. To take advantage of the speed and performance of premium storage disks, you can migrate existing VM disks to Premium SSDs. Premium SSDs are suitable for mission-critical production applications. Premium SSDs can only be used with VM series that are premium storage-compatible.

To learn more about individual VM types and sizes in Azure for Windows, including which sizes are premium storage-compatible, see [Windows VM sizes](#). To learn more about individual VM types and sizes in Azure for Linux, including which sizes are premium storage-compatible, see [Linux VM sizes](#).

Disk size

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,02 4	2,04 8	4,09 6	8,19 2	16,3 84	32,7 67
IOP S per disk	120	120	120	120	240	500	1,1 00	2,30 0	5,00 0	7,50 0	7,50 0	16,0 00	18,0 00	20,0 00
Thr oug hpu t per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max bur st IOP S per disk **	3,5 00	3,5 00	3,5 00	3,5 00	3,5 00	3,5 00	3,5 00	3,5 0	3,50 0					

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Max burst throughput per disk **	170 MiB /sec													
Max burst duration**	30 min													
Eligible for reservation	No	Yes, up to one year												

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

When you provision a premium storage disk, unlike standard storage, you are guaranteed the capacity, IOPS, and throughput of that disk. For example, if you create a P50 disk, Azure provisions 4,095-GB storage capacity, 7,500 IOPS, and 250-MB/s throughput for that disk. Your application can use all or part of the capacity and performance. Premium SSD disks are designed to provide low single-digit millisecond latencies and target IOPS and throughput described in the preceding table 99.9% of the time.

Bursting (preview)

Premium SSD sizes smaller than P30 now offer disk bursting (preview) and can burst their IOPS per disk up to 3,500 and their bandwidth up to 170 Mbps. Bursting is automated and operates based on a credit system. Credits are automatically accumulated in a burst bucket when disk traffic is below the provisioned performance target and credits are automatically consumed when traffic bursts beyond the target, up to the max burst limit. The max burst limit defines the ceiling of disk IOPS & Bandwidth even if you have burst credits to consume from. Disk bursting provides better tolerance on unpredictable changes of IO patterns. You can best leverage it for OS disk boot and applications with spiky traffic.

Disk bursting support will be enabled on new deployments of applicable disk sizes in the [preview regions](#) by default, with no user action required. For existing disks of the applicable sizes, you can enable bursting with either of two options: detach and reattach the disk or stop and restart the attached VM. All burst applicable disk sizes will start with a full burst credit bucket when the disk is attached to a Virtual Machine that supports a max duration at peak burst limit of 30 mins. To learn more about how bursting work on Azure Disks, see [Premium SSD bursting](#).

Transactions

For premium SSDs, each I/O operation less than or equal to 256 KiB of throughput is considered a single I/O operation. I/O operations larger than 256 KiB of throughput are considered multiple I/Os of size 256 KiB.

Standard SSD

Azure standard SSDs are a cost-effective storage option optimized for workloads that need consistent performance at lower IOPS levels. Standard SSD offers a good entry level experience for those who wish to move to the cloud, especially if you experience issues with the variance of workloads running on your HDD solutions on premises. Compared to standard HDDs, standard SSDs deliver better availability, consistency, reliability, and latency. Standard SSDs are suitable for Web servers, low IOPS application servers, lightly used enterprise applications, and Dev/Test workloads. Like standard HDDs, standard SSDs are available on all Azure VMs.

Disk size

STANDBY RD SSD SIZE S	E1*	E2*	E3*	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 120	Up to 120	Up to 120	Up to 120	Up to 240	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000					
Throughput per disk	Up to 25 MiB /sec	Up to 50 MiB /sec	Up to 60 MiB /sec	Up to 400 MiB /sec	Up to 600 MiB /sec	Up to 750 MiB /sec								

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

Standard SSDs are designed to provide single-digit millisecond latencies and the IOPS and throughput up to the limits described in the preceding table 99% of the time. Actual IOPS and throughput may vary sometimes depending on the traffic patterns. Standard SSDs will provide more consistent performance than the HDD disks with the lower latency.

Transactions

For standard SSDs, each I/O operation less than or equal to 256 KiB of throughput is considered a single I/O operation. I/O operations larger than 256 KiB of throughput are considered multiple I/Os of size 256 KiB. These transactions have a billing impact.

Standard HDD

Azure standard HDDs deliver reliable, low-cost disk support for VMs running latency-insensitive workloads. With standard storage, the data is stored on hard disk drives (HDDs). Latency, IOPS, and Throughput of Standard HDD disks may vary more widely as compared to SSD-based disks. Standard HDD Disks are designed to deliver write latencies under 10ms and read latencies under 20ms for most IO operations, however the actual performance may vary depending on the IO size and workload pattern. When working with VMs, you can use standard HDD

disks for dev/test scenarios and less critical workloads. Standard HDDs are available in all Azure regions and can be used with all Azure VMs.

Disk size

STANDARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000							
Throughput per disk	Up to 60 MiB/s ec	Up to 300 MiB/s ec	Up to 500 MiB/s ec	Up to 500 MiB/s ec							

Transactions

For Standard HDDs, each IO operation is considered as a single transaction, regardless of the I/O size. These transactions have a billing impact.

Billing

When using managed disks, the following billing considerations apply:

- Disk type
- managed disk Size
- Snapshots
- Outbound data transfers
- Number of transactions

Managed disk size: managed disks are billed on the provisioned size. Azure maps the provisioned size (rounded up) to the nearest offered disk size. For details of the disk sizes offered, see the previous tables. Each disk maps to a supported provisioned disk size offering and is billed accordingly. For example, if you provisioned a 200 GiB Standard SSD, it maps to the disk size offer of E15 (256 GiB). Billing for any provisioned disk is prorated hourly by using the monthly price for the Premium Storage offer. For example, if you provisioned an E10 disk and deleted it after 20 hours, you're billed for the E10 offering prorated to 20 hours. This is regardless of the amount of actual data written to the disk.

Snapshots: Snapshots are billed based on the size used. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB, the snapshot is billed only for the used data size of 10 GiB.

For more information on snapshots, see the section on snapshots in the [managed disk overview](#).

Outbound data transfers: [Outbound data transfers](#) (data going out of Azure data centers) incur billing for bandwidth usage.

Transactions: You're billed for the number of transactions that you perform on a standard managed disk. For standard SSDs, each I/O operation less than or equal to 256 KiB of throughput is considered a single I/O operation. I/O operations larger than 256 KiB of throughput are considered multiple I/Os of size 256 KiB. For

Standard HDDs, each IO operation is considered as a single transaction, regardless of the I/O size.

For detailed information on pricing for Managed Disks, including transaction costs, see [Managed Disks Pricing](#).

Ultra disk VM reservation fee

Azure VMs have the capability to indicate if they are compatible with ultra disks. An ultra disk compatible VM allocates dedicated bandwidth capacity between the compute VM instance and the block storage scale unit to optimize the performance and reduce latency. Adding this capability on the VM results in a reservation charge that is only imposed if you enabled ultra disk capability on the VM without attaching an ultra disk to it. When an ultra disk is attached to the ultra disk compatible VM, this charge would not be applied. This charge is per vCPU provisioned on the VM.

NOTE

For [constrained core VM sizes](#), the reservation fee is based on the actual number of vCPUs and not the constrained cores.

For Standard_E32-8s_v3, the reservation fee will be based on 32 cores.

Refer to the [Azure Disks pricing page](#) for ultra disk pricing details.

Azure disk reservation

Disk reservation is the option to purchase one year of disk storage in advance at a discount, reducing your total cost. When purchasing a disk reservation, you select a specific Disk SKU in a target region, for example, 10 P30 (1TiB) premium SSDs in East US 2 region for a one year term. The reservation experience is similar to reserved virtual machine (VM) instances. You can bundle VM and Disk reservations to maximize your savings. For now, Azure Disks Reservation offers one year commitment plan for premium SSD SKUs from P30 (1TiB) to P80 (32 TiB) in all production regions. For more details on the Reserved Disks pricing, see [Azure Disks pricing page](#).

Server side encryption of Azure managed disks

2/18/2020 • 12 minutes to read • [Edit Online](#)

Azure managed disks automatically encrypt your data by default when persisting it to the cloud. Server-side encryption protects your data and helps you meet your organizational security and compliance commitments. Data in Azure managed disks is encrypted transparently using 256-bit [AES encryption](#), one of the strongest block ciphers available, and is FIPS 140-2 compliant.

Encryption does not impact the performance of managed disks. There is no additional cost for the encryption.

For more information about the cryptographic modules underlying Azure managed disks, see [Cryptography API: Next Generation](#)

About encryption key management

You can rely on platform-managed keys for the encryption of your managed disk, or you can manage encryption using your own keys. If you choose to manage encryption with your own keys, you can specify a *customer-managed key* to use for encrypting and decrypting all data in managed disks.

The following sections describe each of the options for key management in greater detail.

Platform-managed keys

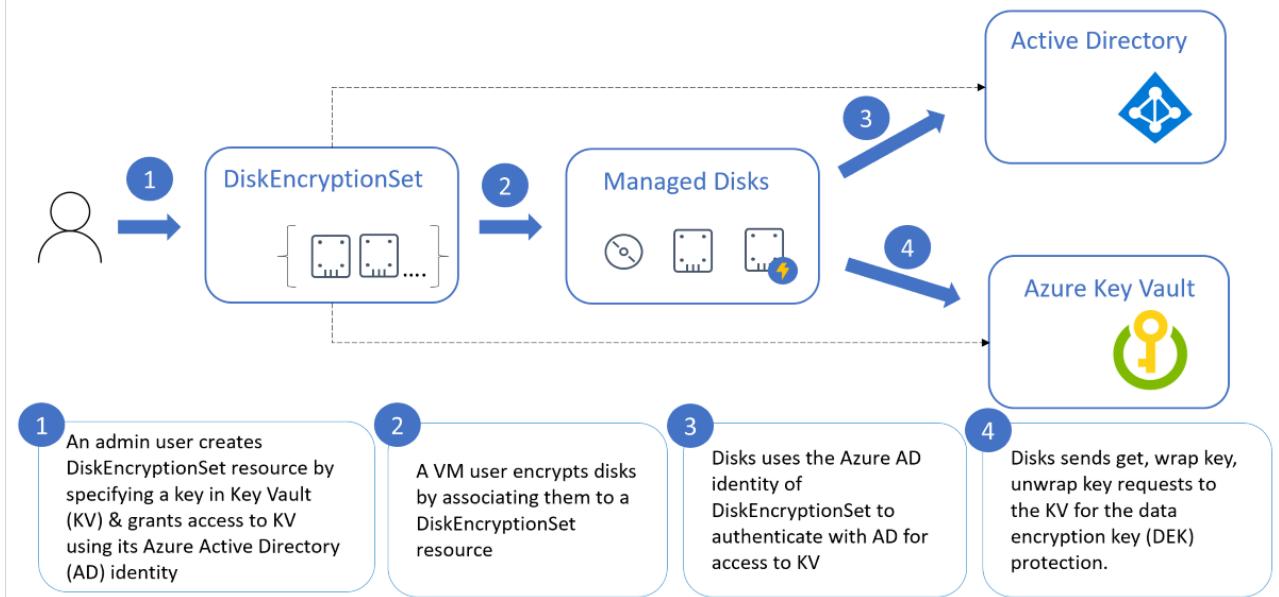
By default, managed disks use platform-managed encryption keys. As of June 10, 2017, all new managed disks, snapshots, images, and new data written to existing managed disks are automatically encrypted-at-rest with platform-managed keys.

Customer-managed keys

You can choose to manage encryption at the level of each managed disk, with your own keys. Server-side encryption for managed disks with customer-managed keys offers an integrated experience with Azure Key Vault. You can either import [your RSA keys](#) to your Key Vault or generate new RSA keys in Azure Key Vault. Azure managed disks handles the encryption and decryption in a fully transparent fashion using [envelope encryption](#). It encrypts data using an [AES 256](#) based data encryption key (DEK), which is, in turn, protected using your keys. You have to grant access to managed disks in your Key Vault to use your keys for encrypting and decrypting the DEK. This allows you full control of your data and keys. You can disable your keys or revoke access to managed disks at any time. You can also audit the encryption key usage with Azure Key Vault monitoring to ensure that only managed disks or other trusted Azure services are accessing your keys.

The following diagram shows how managed disks use Azure Active Directory and Azure Key Vault to make requests using the customer-managed key:

SSE+CMK Workflow



The following list explains the diagram in even more detail:

- 1 An Azure Key Vault administrator creates key vault resources.
- 2 The key vault admin either imports their RSA keys to Key Vault or generate new RSA keys in Key Vault.
- 3 That administrator creates an instance of Disk Encryption Set resource, specifying an Azure Key Vault ID and a key URL. Disk Encryption Set is a new resource introduced for simplifying the key management for managed disks.
- 4 When a disk encryption set is created, a [system-assigned managed identity](#) is created in Azure Active Directory (AD) and associated with the disk encryption set.
- 5 The Azure key vault administrator then grants the managed identity permission to perform operations in the key vault.
- 6 A VM user creates disks by associating them with the disk encryption set. The VM user can also enable server-side encryption with customer-managed keys for existing resources by associating them with the disk encryption set.
- 7 Managed disks use the managed identity to send requests to the Azure Key Vault.
- 8 For reading or writing data, managed disks sends requests to Azure Key Vault to encrypt (wrap) and decrypt (unwrap) the data encryption key in order to perform encryption and decryption of the data.

To revoke access to customer-managed keys, see [Azure Key Vault PowerShell](#) and [Azure Key Vault CLI](#). Revoking access effectively blocks access to all data in the storage account, as the encryption key is inaccessible by Azure Storage.

Supported regions

Only the following regions are currently supported:

- Available as a GA offering in the East US, West US 2, South Central US, UK South regions.
- Available as a public preview in the West Central US, East US 2, Canada Central, and North Europe regions.

Restrictions

For now, customer-managed keys have the following restrictions:

- Only "soft" and "hard" RSA keys of size 2048 are supported, no other keys or sizes.
- Disks created from custom images that are encrypted using server-side encryption and customer-managed keys must be encrypted using the same customer-managed keys and must be in the same subscription.

- Snapshots created from disks that are encrypted with server-side encryption and customer-managed keys must be encrypted with the same customer-managed keys.
- Custom images encrypted using server-side encryption and customer-managed keys cannot be used in the shared image gallery.
- All resources related to your customer-managed keys (Azure Key Vaults, disk encryption sets, VMs, disks, and snapshots) must be in the same subscription and region.
- Disks, snapshots, and images encrypted with customer-managed keys cannot move to another subscription.
- If you use the Azure portal to create your disk encryption set, you cannot use snapshots for now.

PowerShell

Setting up your Azure Key Vault and DiskEncryptionSet

1. Make sure that you have installed latest [Azure PowerShell version](#), and you are signed in to an Azure account in with Connect-AzAccount
2. Create an instance of Azure Key Vault and encryption key.

When creating the Key Vault instance, you must enable soft delete and purge protection. Soft delete ensures that the Key Vault holds a deleted key for a given retention period (90 day default). Purge protection ensures that a deleted key cannot be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

```
$ResourceGroupName="yourResourceGroupName"
$LocationName="westcentralus"
$keyVaultName="yourKeyVaultName"
$keyName="yourKeyName"
$keyDestination="Software"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$keyVault = New-AzKeyVault -Name $keyVaultName -ResourceGroupName $ResourceGroupName -Location
$LocationName -EnableSoftDelete -EnablePurgeProtection

$key = Add-AzKeyVaultKey -VaultName $keyVaultName -Name $keyName -Destination $keyDestination
```

3. Create an instance of a DiskEncryptionSet.

```
$desConfig=New-AzDiskEncryptionSetConfig -Location $LocationName -SourceVaultId $keyVault.ResourceId -
KeyUrl $key.Key.Kid -IdentityType SystemAssigned

$des=New-AzDiskEncryptionSet -Name $diskEncryptionSetName -ResourceGroupName $ResourceGroupName -
InputObject $desConfig
```

4. Grant the DiskEncryptionSet resource access to the key vault.

NOTE

It may take few minutes for Azure to create the identity of your DiskEncryptionSet in your Azure Active Directory. If you get an error like "Cannot find the Active Directory object" when running the following command, wait a few minutes and try again.

```

$identity = Get-AzADServicePrincipal -DisplayName myDiskEncryptionSet1

Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ObjectId $des.Identity.PrincipalId -
PermissionsToKeys wrapkey,unwrapkey,get

New-AzRoleAssignment -ResourceName $keyVaultName -ResourceGroupName $ResourceGroupName -ResourceType
"Microsoft.KeyVault/vaults" -ObjectId $des.Identity.PrincipalId -RoleDefinitionName "Reader"

```

Create a VM using a Marketplace image, encrypting the OS and data disks with customer-managed keys

```

$VMLocalAdminUser = "yourVMLocalAdminUserName"
$VMLocalAdminSecurePassword = ConvertTo-SecureString <password> -AsPlainText -Force
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$ComputerName = "yourComputerName"
$VMName = "yourVMName"
$VMSize = "Standard_DS3_v2"
$diskEncryptionSetName="yourdiskEncryptionSetName"

$NetworkName = "yourNetworkName"
$NICName = "yourNICName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -AddressPrefix $SubnetAddressPrefix
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName $ResourceGroupName -Location $LocationName -
AddressPrefix $VnetAddressPrefix -Subnet $SingleSubnet
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName $ResourceGroupName -Location $LocationName -
SubnetId $Vnet.Subnets[0].Id

$Credential = New-Object System.Management.Automation.PSCredential ($VMLocalAdminUser,
$VMLocalAdminSecurePassword);

$VirtualMachine = New-AzVMConfig -VMName $VMName -VMSize $VMSize
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -ComputerName $ComputerName -Credential
$Credential -ProvisionVMAgent -EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName 'MicrosoftWindowsServer' -Offer
'WindowsServer' -Skus '2012-R2-Datacenter' -Version latest

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName $ResourceGroupName -Name $diskEncryptionSetName

$VirtualMachine = Set-AzVMDisk -VM $VirtualMachine -Name $($VMName + "_OSDisk") -DiskEncryptionSetId
$diskEncryptionSet.Id -CreateOption FromImage

$VirtualMachine = Add-AzVMDataDisk -VM $VirtualMachine -Name $($VMName + "DataDisk1") -DiskSizeInGB 128 -
StorageAccountType Premium_LRS -CreateOption Empty -Lun 0 -DiskEncryptionSetId $diskEncryptionSet.Id

New-AzVM -ResourceGroupName $ResourceGroupName -Location $LocationName -VM $VirtualMachine -Verbose

```

Create an empty disk encrypted using server-side encryption with customer-managed keys and attach it to a VM

```

$vmName = "yourVMName"
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$diskName = "yourDiskName"
$diskSKU = "Premium_LRS"
$diskSizeinGiB = 30
$diskLUN = 1
$diskEncryptionSetName="yourDiskEncryptionSetName"

$vm = Get-AzVM -Name $vmName -ResourceGroupName $ResourceGroupName

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName $ResourceGroupName -Name $diskEncryptionSetName

$vm = Add-AzVMDATADisk -VM $vm -Name $diskName -CreateOption Empty -DiskSizeInGB $diskSizeinGiB -
StorageAccountType $diskSKU -Lun $diskLUN -DiskEncryptionSetId $diskEncryptionSet.Id

Update-AzVM -ResourceGroupName $ResourceGroupName -VM $vm

```

Encrypt existing unattached managed disks

Your existing disks must not be attached to a running VM in order for you to encrypt them using the following script:

```

$rgName = "yourResourceGroupName"
$diskName = "yourDiskName"
$diskEncryptionSetName = "yourDiskEncryptionSetName"

$diskEncryptionSet = Get-AzDiskEncryptionSet -ResourceGroupName $rgName -Name $diskEncryptionSetName

New-AzDiskUpdateConfig -EncryptionType "EncryptionAtRestWithCustomerKey" -DiskEncryptionSetId
$diskEncryptionSet.Id | Update-AzDisk -ResourceGroupName $rgName -DiskName $diskName

```

Create a virtual machine scale set using a Marketplace image, encrypting the OS and data disks with customer-managed keys

```

$VMLocalAdminUser = "yourLocalAdminUser"
$VMLocalAdminSecurePassword = ConvertTo-SecureString Password@123 -AsPlainText -Force
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$ComputerNamePrefix = "yourComputerNamePrefix"
$VMScaleSetName = "yourVMSSName"
$VMSize = "Standard_DS3_v2"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$NetworkName = "yourVNETName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -AddressPrefix $SubnetAddressPrefix

$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName $ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix -Subnet $SingleSubnet

$ipConfig = New-AzVmssIpConfig -Name "myIPConfig" -SubnetId $Vnet.Subnets[0].Id

$VMSS = New-AzVmssConfig -Location $LocationName -SkuCapacity 2 -SkuName $VMSize -UpgradePolicyMode 'Automatic'

$VMSS = Add-AzVmssNetworkInterfaceConfiguration -Name "myVMSSNetworkConfig" -VirtualMachineScaleSet $VMSS -Primary $true -IpConfiguration $ipConfig

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName $ResourceGroupName -Name $diskEncryptionSetName

# Enable encryption at rest with customer managed keys for OS disk by setting DiskEncryptionSetId property

$VMSS = Set-AzVmssStorageProfile $VMSS -OsDiskCreateOption "FromImage" -DiskEncryptionSetId
$diskEncryptionSet.Id -ImageReferenceOffer 'WindowsServer' -ImageReferenceSku '2012-R2-Datacenter' -ImageReferenceVersion latest -ImageReferencePublisher 'MicrosoftWindowsServer'

$VMSS = Set-AzVmssOsProfile $VMSS -ComputerNamePrefix $ComputerNamePrefix -AdminUsername $VMLocalAdminUser -AdminPassword $VMLocalAdminSecurePassword

# Add a data disk encrypted at rest with customer managed keys by setting DiskEncryptionSetId property

$VMSS = Add-AzVmssDataDisk -VirtualMachineScaleSet $VMSS -CreateOption Empty -Lun 1 -DiskSizeGB 128 -StorageAccountType Premium_LRS -DiskEncryptionSetId $diskEncryptionSet.Id

$Credential = New-Object System.Management.Automation.PSCredential ($VMLocalAdminUser, $VMLocalAdminSecurePassword);

New-AzVmss -VirtualMachineScaleSet $VMSS -ResourceGroupName $ResourceGroupName -VMScaleSetName $VMScaleSetName

```

IMPORTANT

Customer-managed keys rely on managed identities for Azure resources, a feature of Azure Active Directory (Azure AD). When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Azure AD directory to another, the managed identity associated with managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see [Transferring a subscription between Azure AD directories](#).

Portal

Setting up customer-managed keys for your disks will require you to create resources in a particular order, if you're doing it for the first time. First, you will need to create and set up an Azure Key Vault.

Setting up your Azure Key Vault

1. Sign into the [Azure portal](#) and search for Key Vault

2. Search for and select **Key Vaults**.

IMPORTANT

Your Azure key vault, disk encryption set, VM, disks, and snapshots must all be in the same region and subscription for deployment to succeed.

3. Select **+Add** to create a new Key Vault.
4. Create a new resource group
5. Enter a key vault name, select a region, and select a pricing tier.
6. Select **Review + Create**, verify your choices, then select **Create**.

Dashboard > Key vaults > Create key vault

Create key vault

Basics Access policy Networking Tags Review + create

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance. [Learn more](#)

Project details

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

Subscription * My Example Subscription

Resource group * Select existing... [Create new](#)

Instance details

Key vault name * ⓘ

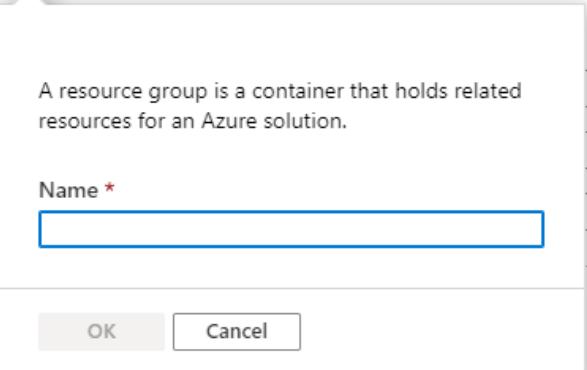
Region *

Pricing tier * ⓘ

[Review + create](#) < Prev OK Cancel

A resource group is a container that holds related resources for an Azure solution.

Name *



7. Once your key vault finishes deploying, select it.

8. Select **Keys** under **Settings**.

9. Select **Generate/Import**

my-example-vault - Keys

Key vault

Search (Ctrl+ /) < + Generate/Import Refresh Restore Backup

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Events (preview)

Settings

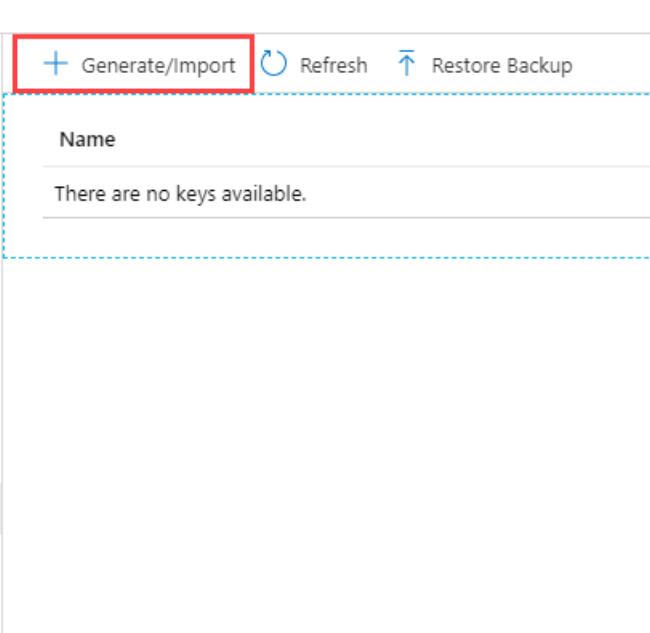
Keys (highlighted with a red box)

Secrets Certificates

+ Generate/Import

Name

There are no keys available.



10. Leave both **Key Type** set to **RSA** and **RSA Key Size** set to **2048**.

11. Fill in the remaining selections as you like and then select **Create**.

The screenshot shows the 'Create a key' dialog box. At the top, it says 'Create a key'. Below that is a section titled 'Options' with a dropdown menu showing 'Generate'. The 'Name *' field is empty. Under 'Key Type', 'RSA' is selected over 'EC'. Under 'RSA Key Size', '2048' is selected over '3072' and '4096'. There are checkboxes for 'Set activation date?' and 'Set expiration date?', both of which are unchecked. Below that is a 'Enabled?' section with 'Yes' selected over 'No'. At the bottom is a large red-bordered 'Create' button.

Setting up your disk encryption set

To create and configure disk encryption sets, you must use the following link: <https://aka.ms/diskencryptionsets>. Disk encryption set creation is not yet available in the global Azure portal.

1. Open the [disk encryption sets link](#).

2. Select **+Add**.

The screenshot shows the 'Disk Encryption Sets' blade in the Azure portal. At the top, it says 'Dashboard > Disk Encryption Sets'. Below that is a table header 'Disk Encryption Sets' with a 'Microsoft' filter. The '+ Add' button is highlighted with a red box. The table shows one record: 'Showing 1 to 1 of 1 records.' The filters at the bottom include 'Subscription == My Example Subscription', 'Resource group == all', 'Location == all', and a 'Add filter' button.

3. Select your resource group, name your encryption set, and select the same region as your key vault.

4. Select **Key vault and key**.

5. Select the key vault and key you created previously, as well as the version.

6. Press **Select**.

7. Select **Review + Create** and then **Create**.

Dashboard > Disk Encryption Sets > Create a disk encryption set

Create a disk encryption set

Basics Tags Review + create

Disk encryption sets allow you to manage encryption keys using server-side encryption for Standard HDD, Standard SSD, and Premium SSD managed disks. It will give you control of the encryption keys to meet your security and compliance needs in a few clicks. [Learn more about disk encryption 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 * ⓘ My Example Subscription

Resource group * ⓘ southcmksseregroup [Create new](#)

Instance details

Disk encryption set name * ⓘ myexamplesetname

Region * ⓘ (US) South Central US

Key vault and key * ⓘ Click to select a key

Review + create < Previous Next : Tags >

Select

- Open the disk encryption set once it finishes creating and select the alert that pops up.

exampleSetName
Disk Encryption Set - PREVIEW

Search (Ctrl+ /) Delete

Overview Activity log Access control (IAM)

To associate a disk, image, or snapshot with this disk encryption set, you must grant permissions to the key vault my-example-vault.

Resource group (change) : southcmksseregroup
Status : ---
Location : South Central US
Key vault : my-example-vault
Key : my-key

Two notifications should pop up and succeed. Doing this will allow you to use the disk encryption set with your key vault.

Successfully assigned role 1:03 PM
Successfully assigned role to the key vault 'my-example-vault'.

Successfully granted permissions 1:03 PM
Successfully granted permissions to the key vault 'my-example-vault'.

Deploy a VM

Now that you've created and set up your key vault and the disk encryption set, you can deploy a VM using the encryption. The VM deployment process is similar to the standard deployment process, the only differences are that you need to deploy the VM in the same region as your other resources and you opt to use a customer managed key.

- Open the [disk encryption sets link](#).
- Search for **Virtual Machines** and select **+ Add** to create a VM.
- On the **Basic** tab, select the same region as your disk encryption set and Azure Key Vault.
- Fill in the other values on the **Basic** tab as you like.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.

Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

Project details

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

Subscription * ⓘ

My Example Subscription

Resource group * ⓘ

exampleresourcegroup

[Create new](#)

Instance details

Virtual machine name * ⓘ

examplevmname

Region * ⓘ

(US) South Central US

Availability options ⓘ

No infrastructure redundancy required

Image * ⓘ

Windows Server 2019 Datacenter

[Browse all public and private images](#)

Azure Spot instance ⓘ

Yes No

Size * ⓘ

Standard DS1 v2

1 vcpu, 3.5 GiB memory (\$87.05/month)

[Change size](#)

[Review + create](#)

< Previous

Next : Disks >

5. On the **Disks** tab, select **Encryption at rest with a customer-managed key**.

6. Select your disk encryption set in the **Disk encryption set** drop-down.

7. Make the remaining selections as you like.

Dashboard > Virtual machines > Create a virtual machine

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

OS disk type * ⓘ Premium SSD

Encryption options Encryption at rest with a platform-managed key Encryption at rest with a customer-managed key

⚠️ Once a customer-managed key is used, you can't change the selection back to a platform-managed key.
[Learn more about disk encryption.](#)

Disk encryption set * exampleSetName

Enable Ultra Disk compatibility ⓘ Yes No
Ultra Disk compatibility is not available for this VM size and location.

Data disks

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	Name	Size (GiB)	Disk type	Host caching
Create and attach a new disk Attach an existing disk				

✓ Advanced

Review + create < Previous Next : Networking >

Enable on an existing disk

To manage and configure disk encryption on your existing disks, you must use the following link:

<https://aka.ms/diskencryptionsets>. Enabling customer-managed keys on existing disks is not yet available in the global Azure portal.

Caution

Enabling disk encryption on any disks attached to a VM will require that you stop the VM.

1. Open the [disk encryption sets link](#).
2. Navigate to a VM which is in the same region as one of your disk encryption sets.
3. Open the VM and select **Stop**.

my-example-vm
Virtual machine

Search (Ctrl+/
Resource group (change) : southcentralusregion
Status : Running
Location : South Central US
Subscription (change) : My Example Subscription

Connect Start Restart Stop Capture Delete Refresh

- After the VM has finished stopping, select **Disks** and then select the disk you want to encrypt.

my-example-vm - Disks
Virtual machine

Search (Ctrl+/
Edit Refresh Encryption Swap OS Disk

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Networking Disks Size Security Extensions Continuous delivery (Preview) Availability + scaling

Managed disks created since June 10, 2017 are encrypted at rest with Storage Service Encry

Ultra Disk compatibility is not available for this location.

Disk settings

Enable Ultra Disk compatibility ⓘ
 Yes No

OS disk

Name	Size
my-example-vm_OsDisk_1_c6be1c817df34ea8bc60e4ef70404870...	127 GiB

Data disks

None

+ Add data disk

- Select **Encryption** and select **Encryption at rest with a customer-managed key** and then select your disk encryption set in the drop-down list.

- Select **Save**.

my-example-vm_OsDisk_1_c6be1c817df34ea8bc60e4ef70404870 - Encryption
Disk

Search (Ctrl+/
Save Discard

Encryption

Encryption at rest with a platform-managed key
 Encryption at rest with a customer-managed key

⚠ Once a customer-managed key is used, you can't change the selection back to a platform-managed key.
[Learn more about disk encryption.](#)

Disk encryption set * exampleSetName

- Repeat this process for any other disks attached to the VM you'd like to encrypt.

- When your disks finish switching over to customer-managed keys, if there are no other attached disks you'd like to encrypt, you may start your VM.

IMPORTANT

Customer-managed keys rely on managed identities for Azure resources, a feature of Azure Active Directory (Azure AD). When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Azure AD directory to another, the managed identity associated with managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see [Transferring a subscription between Azure AD directories](#).

Server-side encryption versus Azure disk encryption

[Azure Disk Encryption](#) leverages the [BitLocker](#) feature of Windows and the [DM-Crypt](#) feature of Linux to encrypt managed disks with customer-managed keys within the guest VM. Server-side encryption with customer-managed keys improves on ADE by enabling you to use any OS types and images for your VMs by encrypting data in the Storage service.

Next steps

- [Explore the Azure Resource Manager templates for creating encrypted disks with customer-managed keys](#)
- [What is Azure Key Vault?](#)
- [Replicate machines with customer-managed keys enabled disks](#)
- [Set up disaster recovery of VMware VMs to Azure with PowerShell](#)
- [Set up disaster recovery to Azure for Hyper-V VMs using PowerShell and Azure Resource Manager](#)

Understand how your reservation discount is applied to Azure disk storage

2/24/2020 • 2 minutes to read • [Edit Online](#)

After you purchase Azure disk reserved capacity, a reservation discount is automatically applied to disk resources that match the terms of your reservation. The reservation discount applies to disk SKUs only. Disk snapshots are charged at pay-as-you-go rates.

For more information about Azure disk reservation, see [Save costs with Azure disk reservation](#). For information about pricing for Azure disk reservation, see [Azure Managed Disks pricing](#).

How the reservation discount is applied

The Azure disk reservation discount is a use-it-or-lose-it discount. It's applied to managed disk resources hourly. For a given hour, if you have no managed disk resources that meet the reservation terms, you lose a reservation quantity for that hour. Unused hours don't carry forward.

When you delete a resource, the reservation discount automatically applies to another matching resource in the specified scope. If no matching resource is found, the reserved hours are lost.

Discount examples

The following examples show how the Azure disk reservation discount applies depending on your deployment.

Suppose you purchase and reserve 100 P30 disks in the US West 2 region for a one-year term. Each disk has approximately 1 TiB of storage. Assume the cost of this sample reservation is \$140,100. You can choose to pay either the full amount up front or fixed monthly installments of \$11,675 for the next 12 months.

The following scenarios describe what happens if you underuse, overuse, or tier your reserved capacity. For these examples, assume you've signed up for a monthly reservation-payment plan.

Underusing your capacity

Suppose you deploy only 99 of your 100 reserved Azure premium solid-state drive (SSD) P30 disks for an hour within the reservation period. The remaining P30 disk isn't applied during that hour. It also doesn't carry over.

Overusing your capacity

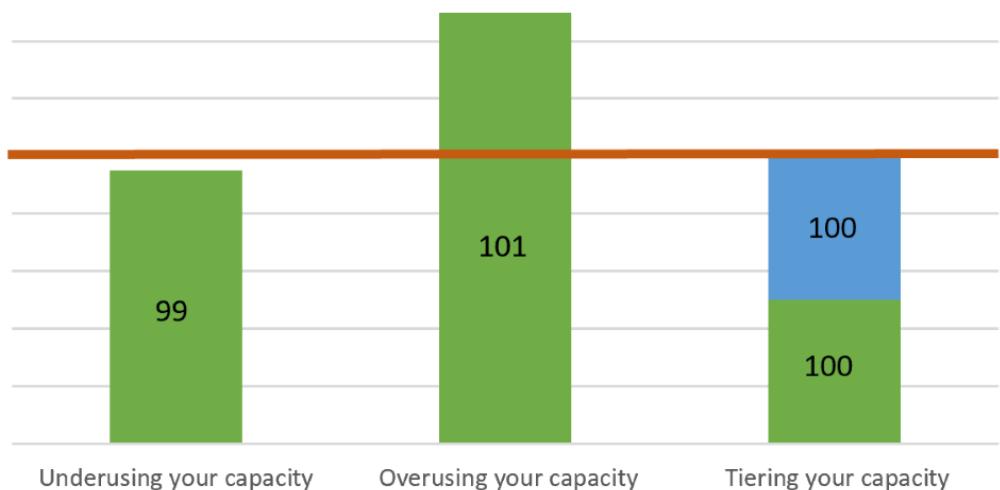
Suppose that for an hour within the reservation period, you use 101 premium SSD P30 disks. The reservation discount applies only to 100 P30 disks. The remaining P30 disk is charged at pay-as-you-go rates for that hour. For the next hour, if your usage goes down to 100 P30 disks, all usage is covered by the reservation.

Tiering your capacity

Suppose that in a given hour within your reservation period, you want to use a total of 200 P30 premium SSDs. Also suppose you use only 100 for the first 30 minutes. During this period, your use is fully covered because you made a reservation for 100 P30 disks. If you then discontinue the use of the first 100 (so that you're using zero) and then begin to use the other 100 for the remaining 30 minutes, that usage is also covered under your reservation.

Reserved disks example scenarios

One hour



Need help? Contact us

If you have questions or need help, [create a support request](#).

Next steps

- [Reduce costs with Azure Disks Reservation \(Linux\)](#)
- [Reduce costs with Azure Disks Reservation \(Windows\)](#)
- [What are Azure Reservations?](#)

Azure premium storage: design for high performance

12/23/2019 • 37 minutes to read • [Edit Online](#)

This article provides guidelines for building high performance applications using Azure Premium Storage. You can use the instructions provided in this document combined with performance best practices applicable to technologies used by your application. To illustrate the guidelines, we have used SQL Server running on Premium Storage as an example throughout this document.

While we address performance scenarios for the Storage layer in this article, you will need to optimize the application layer. For example, if you are hosting a SharePoint Farm on Azure Premium Storage, you can use the SQL Server examples from this article to optimize the database server. Additionally, optimize the SharePoint Farm's Web server and Application server to get the most performance.

This article will help answer following common questions about optimizing application performance on Azure Premium Storage,

- How to measure your application performance?
- Why are you not seeing expected high performance?
- Which factors influence your application performance on Premium Storage?
- How do these factors influence performance of your application on Premium Storage?
- How can you optimize for IOPS, Bandwidth and Latency?

We have provided these guidelines specifically for Premium Storage because workloads running on Premium Storage are highly performance sensitive. We have provided examples where appropriate. You can also apply some of these guidelines to applications running on IaaS VMs with Standard Storage disks.

NOTE

Sometimes, what appears to be a disk performance issue is actually a network bottleneck. In these situations, you should optimize your [network performance](#).

If you are looking to benchmark your disk, see our article on [Benchmarking a disk](#).

If your VM supports accelerated networking, you should make sure it is enabled. If it is not enabled, you can enable it on already deployed VMs on both [Windows](#) and [Linux](#).

Before you begin, if you are new to Premium Storage, first read the [Select an Azure disk type for IaaS VMs](#) and [Scalability targets for premium page blob storage accounts](#).

Application performance indicators

We assess whether an application is performing well or not using performance indicators like, how fast an application is processing a user request, how much data an application is processing per request, how many requests an application processing in a specific period of time, how long a user has to wait to get a response after submitting their request. The technical terms for these performance indicators are, IOPS, Throughput or Bandwidth, and Latency.

In this section, we will discuss the common performance indicators in the context of Premium Storage. In the following section, [Gathering Application Requirements](#), you will learn how to measure these performance indicators for your application. Later in [Optimizing Application Performance](#), you will learn about the factors affecting these performance indicators and recommendations to optimize them.

IOPS

IOPS, or Input/output Operations Per Second, is the number of requests that your application is sending to the storage disks in one second. An input/output operation could be read or write, sequential, or random. Online Transaction Processing (OLTP) applications like an online retail website need to process many concurrent user requests immediately. The user requests are insert and update intensive database transactions, which the application must process quickly. Therefore, OLTP applications require very high IOPS. Such applications handle millions of small and random IO requests. If you have such an application, you must design the application infrastructure to optimize for IOPS. In the later section, *Optimizing Application Performance*, we discuss in detail all the factors that you must consider to get high IOPS.

When you attach a premium storage disk to your high scale VM, Azure provisions for you a guaranteed number of IOPS as per the disk specification. For example, a P50 disk provisions 7500 IOPS. Each high scale VM size also has a specific IOPS limit that it can sustain. For example, a Standard GS5 VM has 80,000 IOPS limit.

Throughput

Throughput, or bandwidth is the amount of data that your application is sending to the storage disks in a specified interval. If your application is performing input/output operations with large IO unit sizes, it requires high throughput. Data warehouse applications tend to issue scan intensive operations that access large portions of data at a time and commonly perform bulk operations. In other words, such applications require higher throughput. If you have such an application, you must design its infrastructure to optimize for throughput. In the next section, we discuss in detail the factors you must tune to achieve this.

When you attach a premium storage disk to a high scale VM, Azure provisions throughput as per that disk specification. For example, a P50 disk provisions 250 MB per second disk throughput. Each high scale VM size also has as specific throughput limit that it can sustain. For example, Standard GS5 VM has a maximum throughput of 2,000 MB per second.

There is a relation between throughput and IOPS as shown in the formula below.

$$\text{IOPS} \times \text{IO Size} = \text{Throughput}$$

Therefore, it is important to determine the optimal throughput and IOPS values that your application requires. As you try to optimize one, the other also gets affected. In a later section, *Optimizing Application Performance*, we will discuss in more details about optimizing IOPS and Throughput.

Latency

Latency is the time it takes an application to receive a single request, send it to the storage disks and send the response to the client. This is a critical measure of an application's performance in addition to IOPS and Throughput. The Latency of a premium storage disk is the time it takes to retrieve the information for a request and communicate it back to your application. Premium Storage provides consistent low latencies. Premium Disks are designed to provide single-digit millisecond latencies for most IO operations. If you enable ReadOnly host caching on premium storage disks, you can get much lower read latency. We will discuss Disk Caching in more detail in later section on *Optimizing Application Performance*.

When you are optimizing your application to get higher IOPS and Throughput, it will affect the latency of your application. After tuning the application performance, always evaluate the latency of the application to avoid unexpected high latency behavior.

The following control plane operations on Managed Disks may involve movement of the Disk from one Storage location to another. This is orchestrated via background copy of data that can take several hours to complete,

typically less than 24 hours depending on the amount of data in the disks. During that time your application can experience higher than usual read latency as some reads can get redirected to the original location and can take longer to complete. There is no impact on write latency during this period.

- Update the storage type.
- Detach and attach a disk from one VM to another.
- Create a managed disk from a VHD.
- Create a managed disk from a snapshot.
- Convert unmanaged disks to managed disks.

Performance Application Checklist for disks

The first step in designing high-performance applications running on Azure Premium Storage is understanding the performance requirements of your application. After you have gathered performance requirements, you can optimize your application to achieve the most optimal performance.

In the previous section, we explained the common performance indicators, IOPS, Throughput, and Latency. You must identify which of these performance indicators are critical to your application to deliver the desired user experience. For example, high IOPS matters most to OLTP applications processing millions of transactions in a second. Whereas, high Throughput is critical for Data Warehouse applications processing large amounts of data in a second. Extremely low Latency is crucial for real-time applications like live video streaming websites.

Next, measure the maximum performance requirements of your application throughout its lifetime. Use the sample checklist below as a start. Record the maximum performance requirements during normal, peak, and off-hours workload periods. By identifying requirements for all workloads levels, you will be able to determine the overall performance requirement of your application. For example, the normal workload of an e-commerce website will be the transactions it serves during most days in a year. The peak workload of the website will be the transactions it serves during holiday season or special sale events. The peak workload is typically experienced for a limited period, but can require your application to scale two or more times its normal operation. Find out the 50 percentile, 90 percentile, and 99 percentile requirements. This helps filter out any outliers in the performance requirements and you can focus your efforts on optimizing for the right values.

Application performance requirements checklist

PERFORMANCE REQUIREMENTS	50 PERCENTILE	90 PERCENTILE	99 PERCENTILE
Max. Transactions per second			
% Read operations			
% Write operations			
% Random operations			
% Sequential operations			
IO request size			
Average Throughput			
Max. Throughput			

PERFORMANCE REQUIREMENTS	50 PERCENTILE	90 PERCENTILE	99 PERCENTILE
Min. Latency			
Average Latency			
Max. CPU			
Average CPU			
Max. Memory			
Average Memory			
Queue Depth			

NOTE

You should consider scaling these numbers based on expected future growth of your application. It is a good idea to plan for growth ahead of time, because it could be harder to change the infrastructure for improving performance later.

If you have an existing application and want to move to Premium Storage, first build the checklist above for the existing application. Then, build a prototype of your application on Premium Storage and design the application based on guidelines described in *Optimizing Application Performance* in a later section of this document. The next article describes the tools you can use to gather the performance measurements.

Counters to measure application performance requirements

The best way to measure performance requirements of your application, is to use performance-monitoring tools provided by the operating system of the server. You can use PerfMon for Windows and iostat for Linux. These tools capture counters corresponding to each measure explained in the above section. You must capture the values of these counters when your application is running its normal, peak, and off-hours workloads.

The PerfMon counters are available for processor, memory and, each logical disk and physical disk of your server. When you use premium storage disks with a VM, the physical disk counters are for each premium storage disk, and logical disk counters are for each volume created on the premium storage disks. You must capture the values for the disks that host your application workload. If there is a one to one mapping between logical and physical disks, you can refer to physical disk counters; otherwise refer to the logical disk counters. On Linux, the iostat command generates a CPU and disk utilization report. The disk utilization report provides statistics per physical device or partition. If you have a database server with its data and logs on separate disks, collect this data for both disks. Below table describes counters for disks, processors, and memory:

COUNTER	DESCRIPTION	PERFMON	IOSTAT
IOPS or Transactions per second	Number of I/O requests issued to the storage disk per second.	Disk Reads/sec Disk Writes/sec	tps r/s w/s
Disk Reads and Writes	% of Reads and Write operations performed on the disk.	% Disk Read Time % Disk Write Time	r/s w/s

COUNTER	DESCRIPTION	PERFMON	IOSTAT
Throughput	Amount of data read from or written to the disk per second.	Disk Read Bytes/sec Disk Write Bytes/sec	kB_read/s kB_wrtn/s
Latency	Total time to complete a disk IO request.	Average Disk sec/Read Average disk sec/Write	await svctm
IO size	The size of I/O requests issued to the storage disks.	Average Disk Bytes/Read Average Disk Bytes/Write	avgrq-sz
Queue Depth	Number of outstanding I/O requests waiting to be read from or written to the storage disk.	Current Disk Queue Length	avgqu-sz
Max. Memory	Amount of memory required to run application smoothly	% Committed Bytes in Use	Use vmstat
Max. CPU	Amount CPU required to run application smoothly	% Processor time	%util

Learn more about [iostat](#) and [PerfMon](#).

Optimize application performance

The main factors that influence performance of an application running on Premium Storage are Nature of IO requests, VM size, Disk size, Number of disks, disk caching, multithreading, and queue depth. You can control some of these factors with knobs provided by the system. Most applications may not give you an option to alter the IO size and Queue Depth directly. For example, if you are using SQL Server, you cannot choose the IO size and queue depth. SQL Server chooses the optimal IO size and queue depth values to get the most performance. It is important to understand the effects of both types of factors on your application performance, so that you can provision appropriate resources to meet performance needs.

Throughout this section, refer to the application requirements checklist that you created, to identify how much you need to optimize your application performance. Based on that, you will be able to determine which factors from this section you will need to tune. To witness the effects of each factor on your application performance, run benchmarking tools on your application setup. Refer to the Benchmarking article, linked at the end, for steps to run common benchmarking tools on Windows and Linux VMs.

Optimize IOPS, throughput, and latency at a glance

The table below summarizes performance factors and the steps necessary to optimize IOPS, throughput, and latency. The sections following this summary will describe each factor in much more depth.

For more information on VM sizes and on the IOPS, throughput, and latency available for each type of VM, see [Linux VM sizes](#) or [Windows VM sizes](#).

	IOPS	THROUGHPUT	LATENCY
Example Scenario	Enterprise OLTP application requiring very high transactions per second rate.	Enterprise Data warehousing application processing large amounts of data.	Near real-time applications requiring instant responses to user requests, like online gaming.

	IOPS	THROUGHPUT	LATENCY
Performance factors			
IO size	Smaller IO size yields higher IOPS.	Larger IO size to yields higher Throughput.	
VM size	Use a VM size that offers IOPS greater than your application requirement.	Use a VM size with throughput limit greater than your application requirement.	Use a VM size that offers scale limits greater than your application requirement.
Disk size	Use a disk size that offers IOPS greater than your application requirement.	Use a disk size with Throughput limit greater than your application requirement.	Use a disk size that offers scale limits greater than your application requirement.
VM and Disk Scale Limits	IOPS limit of the VM size chosen should be greater than total IOPS driven by storage disks attached to it.	Throughput limit of the VM size chosen should be greater than total Throughput driven by premium storage disks attached to it.	Scale limits of the VM size chosen must be greater than total scale limits of attached premium storage disks.
Disk Caching	Enable ReadOnly Cache on premium storage disks with Read heavy operations to get higher Read IOPS.		Enable ReadOnly Cache on premium storage disks with Ready heavy operations to get very low Read latencies.
Disk Striping	Use multiple disks and stripe them together to get a combined higher IOPS and Throughput limit. The combined limit per VM should be higher than the combined limits of attached premium disks.		
Stripe Size	Smaller stripe size for random small IO pattern seen in OLTP applications. For example, use stripe size of 64 KB for SQL Server OLTP application.	Larger stripe size for sequential large IO pattern seen in Data Warehouse applications. For example, use 256 KB stripe size for SQL Server Data warehouse application.	
Multithreading	Use multithreading to push higher number of requests to Premium Storage that will lead to higher IOPS and Throughput. For example, on SQL Server set a high MAXDOP value to allocate more CPUs to SQL Server.		
Queue Depth	Larger Queue Depth yields higher IOPS.	Larger Queue Depth yields higher Throughput.	Smaller Queue Depth yields lower latencies.

Nature of IO requests

An IO request is a unit of input/output operation that your application will be performing. Identifying the nature of IO requests, random or sequential, read or write, small or large, will help you determine the performance requirements of your application. It is important to understand the nature of IO requests, to make the right decisions when designing your application infrastructure. IOs must be distributed evenly to achieve the best performance possible.

IO size is one of the more important factors. The IO size is the size of the input/output operation request generated by your application. The IO size has a significant impact on performance especially on the IOPS and Bandwidth that the application is able to achieve. The following formula shows the relationship between IOPS, IO size, and Bandwidth/Throughput.



Some applications allow you to alter their IO size, while some applications do not. For example, SQL Server determines the optimal IO size itself, and does not provide users with any knobs to change it. On the other hand, Oracle provides a parameter called [DB_BLOCK_SIZE](#) using which you can configure the I/O request size of the database.

If you are using an application, which does not allow you to change the IO size, use the guidelines in this article to optimize the performance KPI that is most relevant to your application. For example,

- An OLTP application generates millions of small and random IO requests. To handle these types of IO requests, you must design your application infrastructure to get higher IOPS.
- A data warehousing application generates large and sequential IO requests. To handle these types of IO requests, you must design your application infrastructure to get higher Bandwidth or Throughput.

If you are using an application, which allows you to change the IO size, use this rule of thumb for the IO size in addition to other performance guidelines,

- Smaller IO size to get higher IOPS. For example, 8 KB for an OLTP application.
- Larger IO size to get higher Bandwidth/Throughput. For example, 1024 KB for a data warehouse application.

Here is an example on how you can calculate the IOPS and Throughput/Bandwidth for your application. Consider an application using a P30 disk. The maximum IOPS and Throughput/Bandwidth a P30 disk can achieve is 5000 IOPS and 200 MB per second respectively. Now, if your application requires the maximum IOPS from the P30 disk and you use a smaller IO size like 8 KB, the resulting Bandwidth you will be able to get is 40 MB per second. However, if your application requires the maximum Throughput/Bandwidth from P30 disk, and you use a larger IO size like 1024 KB, the resulting IOPS will be less, 200 IOPS. Therefore, tune the IO size such that it meets both your application's IOPS and Throughput/Bandwidth requirement. The following table summarizes the different IO sizes and their corresponding IOPS and Throughput for a P30 disk.

APPLICATION REQUIREMENT	I/O SIZE	IOPS	THROUGHPUT/BANDWIDTH
Max IOPS	8 KB	5,000	40 MB per second
Max Throughput	1024 KB	200	200 MB per second
Max Throughput + high IOPS	64 KB	3,200	200 MB per second
Max IOPS + high Throughput	32 KB	5,000	160 MB per second

To get IOPS and Bandwidth higher than the maximum value of a single premium storage disk, use multiple premium disks striped together. For example, stripe two P30 disks to get a combined IOPS of 10,000 IOPS or a combined Throughput of 400 MB per second. As explained in the next section, you must use a VM size that supports the combined disk IOPS and Throughput.

NOTE

As you increase either IOPS or Throughput the other also increases, make sure you do not hit throughput or IOPS limits of the disk or VM when increasing either one.

To witness the effects of IO size on application performance, you can run benchmarking tools on your VM and disks. Create multiple test runs and use different IO size for each run to see the impact. Refer to the Benchmarking article, linked at the end, for more details.

High scale VM sizes

When you start designing an application, one of the first things to do is, choose a VM to host your application. Premium Storage comes with High Scale VM sizes that can run applications requiring higher compute power and a high local disk I/O performance. These VMs provide faster processors, a higher memory-to-core ratio, and a Solid-State Drive (SSD) for the local disk. Examples of High Scale VMs supporting Premium Storage are the DS and GS series VMs.

High Scale VMs are available in different sizes with a different number of CPU cores, memory, OS, and temporary disk size. Each VM size also has maximum number of data disks that you can attach to the VM. Therefore, the chosen VM size will affect how much processing, memory, and storage capacity is available for your application. It also affects the Compute and Storage cost. For example, below are the specifications of the largest VM size in a DS series and a GS series:

VM SIZE	CPU CORES	MEMORY	VM DISK SIZES	MAX. DATA DISKS	CACHE SIZE	IOPS	BANDWIDTH CACHE IO LIMITS
Standard_D S14	16	112 GB	OS = 1023 GB Local SSD = 224 GB	32	576 GB	50,000 IOPS 512 MB per second	4,000 IOPS and 33 MB per second
Standard_G S5	32	448 GB	OS = 1023 GB Local SSD = 896 GB	64	4224 GB	80,000 IOPS 2,000 MB per second	5,000 IOPS and 50 MB per second

To view a complete list of all available Azure VM sizes, refer to [Windows VM sizes](#) or [Linux VM sizes](#). Choose a VM size that can meet and scale to your desired application performance requirements. In addition to this, take into account following important considerations when choosing VM sizes.

Scale Limits

The maximum IOPS limits per VM and per disk are different and independent of each other. Make sure that the application is driving IOPS within the limits of the VM as well as the premium disks attached to it. Otherwise, application performance will experience throttling.

As an example, suppose an application requirement is a maximum of 4,000 IOPS. To achieve this, you provision a P30 disk on a DS1 VM. The P30 disk can deliver up to 5,000 IOPS. However, the DS1 VM is limited to 3,200 IOPS. Consequently, the application performance will be constrained by the VM limit at 3,200 IOPS and there will be degraded performance. To prevent this situation, choose a VM and disk size that will both meet application requirements.

Cost of Operation

In many cases, it is possible that your overall cost of operation using Premium Storage is lower than using Standard Storage.

For example, consider an application requiring 16,000 IOPS. To achieve this performance, you will need a Standard_D14 Azure IaaS VM, which can give a maximum IOPS of 16,000 using 32 standard storage 1 TB disks. Each 1-TB standard storage disk can achieve a maximum of 500 IOPS. The estimated cost of this VM per month will be \$1,570. The monthly cost of 32 standard storage disks will be \$1,638. The estimated total monthly cost will be \$3,208.

However, if you hosted the same application on Premium Storage, you will need a smaller VM size and fewer premium storage disks, thus reducing the overall cost. A Standard_DS13 VM can meet the 16,000 IOPS requirement using four P30 disks. The DS13 VM has a maximum IOPS of 25,600 and each P30 disk has a maximum IOPS of 5,000. Overall, this configuration can achieve $5,000 \times 4 = 20,000$ IOPS. The estimated cost of this VM per month will be \$1,003. The monthly cost of four P30 premium storage disks will be \$544.34. The estimated total monthly cost will be \$1,544.

Table below summarizes the cost breakdown of this scenario for Standard and Premium Storage.

	STANDARD	PREMIUM
Cost of VM per month	\$1,570.58 (Standard_D14)	\$1,003.66 (Standard_DS13)
Cost of Disks per month	\$1,638.40 (32 x 1-TB disks)	\$544.34 (4 x P30 disks)
Overall Cost per month	\$3,208.98	\$1,544.34

Linux Distros

With Azure Premium Storage, you get the same level of Performance for VMs running Windows and Linux. We support many flavors of Linux distros, and you can see the complete list [here](#). It is important to note that different distros are better suited for different types of workloads. You will see different levels of performance depending on the distro your workload is running on. Test the Linux distros with your application and choose the one that works best.

When running Linux with Premium Storage, check the latest updates about required drivers to ensure high performance.

Premium storage disk sizes

Azure Premium Storage offers a variety of sizes so you can choose one that best suits your needs. Each disk size has a different scale limit for IOPS, bandwidth, and storage. Choose the right Premium Storage Disk size depending on the application requirements and the high scale VM size. The table below shows the disks sizes and their capabilities. P4, P6, P15, P60, P70, and P80 sizes are currently only supported for Managed Disks.

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
IOP S per disk	120	120	120	120	240	500	1,10 0	2,30 0	5,00 0	7,50 0	7,50 0	16,0 00	18,0 00	20,0 00
Thr oug hpu t per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max bur st IOP S per disk **	3,5 00	3,5 00	3,5 00	3,5 00	3,5 00	3,50 0	3,50 0	3,50 0						
Max bur st thro ugh put per disk **	170 MiB /sec													
Max bur st dur at ion**	30 min													
Eligi ble for rese rvat ion	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year								

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

How many disks you choose depends on the disk size chosen. You could use a single P50 disk or multiple P10 disks to meet your application requirement. Take into account considerations listed below when making the choice.

Scale Limits (IOPS and Throughput)

The IOPS and Throughput limits of each Premium disk size is different and independent from the VM scale limits.

Make sure that the total IOPS and Throughput from the disks are within scale limits of the chosen VM size.

For example, if an application requirement is a maximum of 250 MB/sec Throughput and you are using a DS4 VM with a single P30 disk. The DS4 VM can give up to 256 MB/sec Throughput. However, a single P30 disk has Throughput limit of 200 MB/sec. Consequently, the application will be constrained at 200 MB/sec due to the disk limit. To overcome this limit, provision more than one data disks to the VM or resize your disks to P40 or P50.

NOTE

Reads served by the cache are not included in the disk IOPS and Throughput, hence not subject to disk limits. Cache has its separate IOPS and Throughput limit per VM.

For example, initially your reads and writes are 60MB/sec and 40MB/sec respectively. Over time, the cache warms up and serves more and more of the reads from the cache. Then, you can get higher write Throughput from the disk.

Number of Disks

Determine the number of disks you will need by assessing application requirements. Each VM size also has a limit on the number of disks that you can attach to the VM. Typically, this is twice the number of cores. Ensure that the VM size you choose can support the number of disks needed.

Remember, the Premium Storage disks have higher performance capabilities compared to Standard Storage disks. Therefore, if you are migrating your application from Azure IaaS VM using Standard Storage to Premium Storage, you will likely need fewer premium disks to achieve the same or higher performance for your application.

Disk caching

High Scale VMs that leverage Azure Premium Storage have a multi-tier caching technology called BlobCache. BlobCache uses a combination of the Virtual Machine RAM and local SSD for caching. This cache is available for the Premium Storage persistent disks and the VM local disks. By default, this cache setting is set to Read/Write for OS disks and ReadOnly for data disks hosted on Premium Storage. With disk caching enabled on the Premium Storage disks, the high scale VMs can achieve extremely high levels of performance that exceed the underlying disk performance.

WARNING

Disk Caching is not supported for disks 4 TiB and larger. If multiple disks are attached to your VM, each disk that is smaller than 4 TiB will support caching.

Changing the cache setting of an Azure disk detaches and re-attaches the target disk. If it is the operating system disk, the VM is restarted. Stop all applications/services that might be affected by this disruption before changing the disk cache setting.

To learn more about how BlobCache works, refer to the Inside [Azure Premium Storage](#) blog post.

It is important to enable cache on the right set of disks. Whether you should enable disk caching on a premium disk or not will depend on the workload pattern that disk will be handling. Table below shows the default cache settings for OS and Data disks.

DISK TYPE	DEFAULT CACHE SETTING
OS disk	ReadWrite
Data disk	ReadOnly

Following are the recommended disk cache settings for data disks,

DISK CACHING SETTING	RECOMMENDATION ON WHEN TO USE THIS SETTING
None	Configure host-cache as None for write-only and write-heavy disks.
ReadOnly	Configure host-cache as ReadOnly for read-only and read-write disks.
ReadWrite	Configure host-cache as ReadWrite only if your application properly handles writing cached data to persistent disks when needed.

ReadOnly

By configuring ReadOnly caching on Premium Storage data disks, you can achieve low Read latency and get very high Read IOPS and Throughput for your application. This is due two reasons,

1. Reads performed from cache, which is on the VM memory and local SSD, are much faster than reads from the data disk, which is on the Azure blob storage.
 2. Premium Storage does not count the Reads served from cache, towards the disk IOPS and Throughput.
- Therefore, your application is able to achieve higher total IOPS and Throughput.

ReadWrite

By default, the OS disks have ReadWrite caching enabled. We have recently added support for ReadWrite caching on data disks as well. If you are using ReadWrite caching, you must have a proper way to write the data from cache to persistent disks. For example, SQL Server handles writing cached data to the persistent storage disks on its own. Using ReadWrite cache with an application that does not handle persisting the required data can lead to data loss, if the VM crashes.

None

Currently, **None** is only supported on data disks. It is not supported on OS disks. If you set **None** on an OS disk it will override this internally and set it to **ReadOnly**.

As an example, you can apply these guidelines to SQL Server running on Premium Storage by doing the following,

1. Configure "ReadOnly" cache on premium storage disks hosting data files.
 - a. The fast reads from cache lower the SQL Server query time since data pages are retrieved much faster from the cache compared to directly from the data disks.
 - b. Serving reads from cache, means there is additional Throughput available from premium data disks. SQL Server can use this additional Throughput towards retrieving more data pages and other operations like backup/restore, batch loads, and index rebuilds.
2. Configure "None" cache on premium storage disks hosting the log files.
 - a. Log files have primarily write-heavy operations. Therefore, they do not benefit from the ReadOnly cache.

Optimize performance on Linux VMs

For all premium SSDs or ultra disks with cache set to **ReadOnly** or **None**, you must disable "barriers" when you mount the file system. You don't need barriers in this scenario because the writes to premium storage disks are durable for these cache settings. When the write request successfully finishes, data has been written to the persistent store. To disable "barriers," use one of the following methods. Choose the one for your file system:

- For **reiserFS**, to disable barriers, use the `barrier=none` mount option. (To enable barriers, use `barrier=flush`.)
- For **ext3/ext4**, to disable barriers, use the `barrier=0` mount option. (To enable barriers, use `barrier=1`.)
- For **XFS**, to disable barriers, use the `nobarrier` mount option. (To enable barriers, use `barrier`.)
- For premium storage disks with cache set to **ReadWrite**, enable barriers for write durability.

- For volume labels to persist after you restart the VM, you must update /etc/fstab with the universally unique identifier (UUID) references to the disks. For more information, see [Add a managed disk to a Linux VM](#).

The following Linux distributions have been validated for premium SSDs. For better performance and stability with premium SSDs, we recommend that you upgrade your VMs to one of these versions or newer.

Some of the versions require the latest Linux Integration Services (LIS), v4.0, for Azure. To download and install a distribution, follow the link listed in the following table. We add images to the list as we complete validation. Our validations show that performance varies for each image. Performance depends on workload characteristics and your image settings. Different images are tuned for different kinds of workloads.

DISTRIBUTION	VERSION	SUPPORTED KERNEL	DETAILS
Ubuntu	12.04 or newer	3.2.0-75.110+	
Ubuntu	14.04 or newer	3.13.0-44.73+	
Debian	7.x, 8.x or newer	3.16.7-ckt4-1+	
SUSE	SLES 12 or newer	3.12.36-38.1+	
SUSE	SLES 11 SP4 or newer	3.0.101-0.63.1+	
CoreOS	584.0.0+ or newer	3.18.4+	
CentOS	6.5, 6.6, 6.7, 7.0, or newer		LIS4 required <i>See note in the next section</i>
CentOS	7.1+ or newer	3.10.0-229.1.2.el7+	LIS4 recommended <i>See note in the next section</i>
Red Hat Enterprise Linux (RHEL)	6.8+, 7.2+, or newer		
Oracle	6.0+, 7.2+, or newer		UEK4 or RHCK
Oracle	7.0-7.1 or newer		UEK4 or RHCK w/ LIS4
Oracle	6.4-6.7 or newer		UEK4 or RHCK w/ LIS4

LIS drivers for OpenLogic CentOS

If you're running OpenLogic CentOS VMs, run the following command to install the latest drivers:

```
sudo yum remove hypervkvpd ## (Might return an error if not installed. That's OK.)
sudo yum install microsoft-hyper-v
sudo reboot
```

In some cases the command above will upgrade the kernel as well. If a kernel update is required then you may need to run the above commands again after rebooting to fully install the microsoft-hyper-v package.

Disk striping

When a high scale VM is attached with several premium storage persistent disks, the disks can be striped together to aggregate their IOPs, bandwidth, and storage capacity.

On Windows, you can use Storage Spaces to stripe disks together. You must configure one column for each disk in a pool. Otherwise, the overall performance of striped volume can be lower than expected, due to uneven distribution of traffic across the disks.

Important: Using Server Manager UI, you can set the total number of columns up to 8 for a striped volume. When attaching more than eight disks, use PowerShell to create the volume. Using PowerShell, you can set the number of columns equal to the number of disks. For example, if there are 16 disks in a single stripe set; specify 16 columns in the *NumberOfColumns* parameter of the *New-VirtualDisk* PowerShell cmdlet.

On Linux, use the MDADM utility to stripe disks together. For detailed steps on striping disks on Linux refer to [Configure Software RAID on Linux](#).

Stripe Size

An important configuration in disk striping is the stripe size. The stripe size or block size is the smallest chunk of data that application can address on a striped volume. The stripe size you configure depends on the type of application and its request pattern. If you choose the wrong stripe size, it could lead to IO misalignment, which leads to degraded performance of your application.

For example, if an IO request generated by your application is bigger than the disk stripe size, the storage system writes it across stripe unit boundaries on more than one disk. When it is time to access that data, it will have to seek across more than one stripe units to complete the request. The cumulative effect of such behavior can lead to substantial performance degradation. On the other hand, if the IO request size is smaller than stripe size, and if it is random in nature, the IO requests may add up on the same disk causing a bottleneck and ultimately degrading the IO performance.

Depending on the type of workload your application is running, choose an appropriate stripe size. For random small IO requests, use a smaller stripe size. Whereas for large sequential IO requests use a larger stripe size. Find out the stripe size recommendations for the application you will be running on Premium Storage. For SQL Server, configure stripe size of 64 KB for OLTP workloads and 256 KB for data warehousing workloads. See [Performance best practices for SQL Server on Azure VMs](#) to learn more.

NOTE

You can stripe together a maximum of 32 premium storage disks on a DS series VM and 64 premium storage disks on a GS series VM.

Multi-threading

Azure has designed Premium Storage platform to be massively parallel. Therefore, a multi-threaded application achieves much higher performance than a single-threaded application. A multi-threaded application splits up its tasks across multiple threads and increases efficiency of its execution by utilizing the VM and disk resources to the maximum.

For example, if your application is running on a single core VM using two threads, the CPU can switch between the two threads to achieve efficiency. While one thread is waiting on a disk IO to complete, the CPU can switch to the other thread. In this way, two threads can accomplish more than a single thread would. If the VM has more than one core, it further decreases running time since each core can execute tasks in parallel.

You may not be able to change the way an off-the-shelf application implements single threading or multi-threading. For example, SQL Server is capable of handling multi-CPU and multi-core. However, SQL Server decides under what conditions it will leverage one or more threads to process a query. It can run queries and build indexes using multi-threading. For a query that involves joining large tables and sorting data before returning to the user, SQL Server will likely use multiple threads. However, a user cannot control whether SQL Server executes a query using a single thread or multiple threads.

There are configuration settings that you can alter to influence this multi-threading or parallel processing of an application. For example, in case of SQL Server it is the maximum Degree of Parallelism configuration. This setting called MAXDOP, allows you to configure the maximum number of processors SQL Server can use when parallel processing. You can configure MAXDOP for individual queries or index operations. This is beneficial when you want to balance resources of your system for a performance critical application.

For example, say your application using SQL Server is executing a large query and an index operation at the same time. Let us assume that you wanted the index operation to be more performant compared to the large query. In such a case, you can set MAXDOP value of the index operation to be higher than the MAXDOP value for the query. This way, SQL Server has more number of processors that it can leverage for the index operation compared to the number of processors it can dedicate to the large query. Remember, you do not control the number of threads SQL Server will use for each operation. You can control the maximum number of processors being dedicated for multi-threading.

Learn more about [Degrees of Parallelism](#) in SQL Server. Find out such settings that influence multi-threading in your application and their configurations to optimize performance.

Queue depth

The queue depth or queue length or queue size is the number of pending IO requests in the system. The value of queue depth determines how many IO operations your application can line up, which the storage disks will be processing. It affects all the three application performance indicators that we discussed in this article viz., IOPS, throughput, and latency.

Queue Depth and multi-threading are closely related. The Queue Depth value indicates how much multi-threading can be achieved by the application. If the Queue Depth is large, application can execute more operations concurrently, in other words, more multi-threading. If the Queue Depth is small, even though application is multi-threaded, it will not have enough requests lined up for concurrent execution.

Typically, off the shelf applications do not allow you to change the queue depth, because if set incorrectly it will do more harm than good. Applications will set the right value of queue depth to get the optimal performance. However, it is important to understand this concept so that you can troubleshoot performance issues with your application. You can also observe the effects of queue depth by running benchmarking tools on your system.

Some applications provide settings to influence the Queue Depth. For example, the MAXDOP (maximum degree of parallelism) setting in SQL Server explained in previous section. MAXDOP is a way to influence Queue Depth and multi-threading, although it does not directly change the Queue Depth value of SQL Server.

High queue depth

A high queue depth lines up more operations on the disk. The disk knows the next request in its queue ahead of time. Consequently, the disk can schedule operations ahead of time and process them in an optimal sequence. Since the application is sending more requests to the disk, the disk can process more parallel IOs. Ultimately, the application will be able to achieve higher IOPS. Since application is processing more requests, the total Throughput of the application also increases.

Typically, an application can achieve maximum Throughput with 8-16+ outstanding IOs per attached disk. If a queue depth is one, application is not pushing enough IOs to the system, and it will process less amount of in a given period. In other words, less Throughput.

For example, in SQL Server, setting the MAXDOP value for a query to "4" informs SQL Server that it can use up to four cores to execute the query. SQL Server will determine what is best queue depth value and the number of cores for the query execution.

Optimal queue depth

Very high queue depth value also has its drawbacks. If queue depth value is too high, the application will try to drive very high IOPS. Unless application has persistent disks with sufficient provisioned IOPS, this can negatively

affect application latencies. Following formula shows the relationship between IOPS, latency, and queue depth.

$$\text{IOPS} \times \text{Latency} = \text{Queue Depth}$$

You should not configure Queue Depth to any high value, but to an optimal value, which can deliver enough IOPS for the application without affecting latencies. For example, if the application latency needs to be 1 millisecond, the Queue Depth required to achieve 5,000 IOPS is, $\text{QD} = 5000 \times 0.001 = 5$.

Queue Depth for Striped Volume

For a striped volume, maintain a high enough queue depth such that, every disk has a peak queue depth individually. For example, consider an application that pushes a queue depth of 2 and there are four disks in the stripe. The two IO requests will go to two disks and remaining two disks will be idle. Therefore, configure the queue depth such that all the disks can be busy. Formula below shows how to determine the queue depth of striped volumes.

$$\text{QD per Disk} \times \text{No. of Columns per Volume} = \text{QD of Striped Volume}$$

Throttling

Azure Premium Storage provisions specified number of IOPS and Throughput depending on the VM sizes and disk sizes you choose. Anytime your application tries to drive IOPS or Throughput above these limits of what the VM or disk can handle, Premium Storage will throttle it. This manifests in the form of degraded performance in your application. This can mean higher latency, lower Throughput, or lower IOPS. If Premium Storage does not throttle, your application could completely fail by exceeding what its resources are capable of achieving. So, to avoid performance issues due to throttling, always provision sufficient resources for your application. Take into consideration what we discussed in the VM sizes and Disk sizes sections above. Benchmarking is the best way to figure out what resources you will need to host your application.

Next steps

If you are looking to benchmark your disk, see our article on [Benchmarking a disk](#).

Learn more about the available disk types: [Select a disk type](#)

For SQL Server users, read articles on Performance Best Practices for SQL Server:

- [Performance Best Practices for SQL Server in Azure Virtual Machines](#)
- [Azure Premium Storage provides highest performance for SQL Server in Azure VM](#)

Premium SSD bursting (preview)

12/2/2019 • 4 minutes to read • [Edit Online](#)

Disk bursting is currently a preview feature for premium SSDs. Bursting is supported on any premium SSD disk sizes <= 512 GiB (P20 or below). These disk sizes support bursting on a best effort basis and utilize a credit system to manage bursting. Credits accumulate in a burst bucket whenever disk traffic is below the provisioned performance target for their disk size, and consume credits when traffic bursts beyond the target. Disk traffic is tracked against both IOPS and bandwidth in the provisioned target.

Disk bursting is enabled by default on new deployments of the disk sizes that support it. Existing disk sizes, if they support disk bursting, can enable bursting through either of the following methods:

- Detach and reattach the disk.
- Stop and start the VM.

Burst states

All burst applicable disk sizes will start with a full burst credit bucket when the disk is attached to a Virtual Machine. The max duration of bursting is determined by the size of the burst credit bucket. You can only accumulate unused credits up to the size of the credit bucket. At any point of time, your disk burst credit bucket can be in one of the following three states:

- Accruing, when the disk traffic is using less than the provisioned performance target. You can accumulate credit if disk traffic is beyond IOPS or bandwidth targets or both. You can still accumulate IO credits when you are consuming full disk bandwidth, vice versa.
- Declining, when the disk traffic is using more than the provisioned performance target. The burst traffic will independently consume credits from IOPS or bandwidth.
- Remaining constant, when the disk traffic is exactly at the provisioned performance target.

The disk sizes that provide bursting support along with the burst specifications are summarized in the table below.

Regional availability

Currently, disk bursting is only available in the West Central US region.

Disk sizes

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
IOP S per disk	120	120	120	120	240	500	1,10 0	2,30 0	5,00 0	7,50 0	7,50 0	16,0 00	18,0 00	20,0 00
Thr oug hpu t per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max bur st IOP S per disk **	3,5 00	3,5 00	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0						
Max bur st thro ugh put per disk **	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec								
Max bur st dur at io n**	30 min	30 min	30 min	30 min	30 min	30 min								
Eligi ble for rese rvat ion	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year							

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

Example scenarios

To give you a better idea of how this works, here's a few example scenarios:

- One common scenario that can benefit from disk bursting is faster VM boot and application launch on OS

disks. Take a Linux VM with an 8 GiB OS image as an example. If we use a P2 disk as the OS disk, the provisioned target is 120 IOPS and 25 MBps. When VM starts, there will be a read spike to the OS disk loading the boot files. With the introduction of bursting, you can read at the max burst speed of 3500 IOPS and 170 MBps, accelerating the load time by at least 6x. After VM boot, the traffic level on the OS disk is usually low, since most data operations by the application will be against the attached data disks. If the traffic is below the provisioned target, you will accumulate credits.

- If you are hosting a Remote Virtual Desktop environment, whenever an active user launches an application like AutoCAD, read traffic to the OS disk significantly increases. In this case, burst traffic will consume accumulated credits, allowing you to go beyond the provisioned target, and launching the application much faster.
- A P1 disk has a provisioned target of 120 IOPS and 25 MBps. If the actual traffic on the disk was 100 IOPS and 20 MBps in the past 1 second interval, then the unused 20 IOs and 5 MB are credited to the burst bucket of the disk. Credits in the burst bucket can later be used when the traffic exceeds the provisioned target, up to the max burst limit. The max burst limit defines the ceiling of disk traffic even if you have burst credits to consume from. In this case, even if you have 10,000 IOs in the credit bucket, a P1 disk cannot issue more than the max burst of 3,500 IO per sec.

Next steps

[Attach a managed data disk to a Windows VM by using the Azure portal](#)

Scalability and performance targets for VM disks on Windows

1/3/2020 • 5 minutes to read • [Edit Online](#)

You can attach a number of data disks to an Azure virtual machine. Based on the scalability and performance targets for a VM's data disks, you can determine the number and type of disk that you need to meet your performance and capacity requirements.

IMPORTANT

For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks aren't highly utilized at the same time, the virtual machine can support a larger number of disks.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. There is no limit for the number of Managed Disks, snapshots and images per resource group.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Standard managed disks	50,000	50,000
Standard SSD managed disks	50,000	50,000
Premium managed disks	50,000	50,000
Standard_LRS snapshots	50,000	50,000
Standard_ZRS snapshots	50,000	50,000
Managed image	50,000	50,000

- **For Standard storage accounts:** A Standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a Standard storage account should not exceed this limit.

You can roughly calculate the number of highly utilized disks supported by a single Standard storage account based on the request rate limit. For example, for a Basic tier VM, the maximum number of highly utilized disks is about 66, which is $20,000/300$ IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is $20,000/500$ IOPS per disk.

- **For Premium storage accounts:** A Premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

See [Windows VM sizes](#) for additional details.

Managed virtual machine disks

Sizes denoted with an asterisk are currently in preview. See our [FAQ](#) to learn what regions they are available in.

Standard HDD managed disks

STANDARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000							
Throughput per disk	Up to 60 MiB/sec	Up to 300 MiB/sec	Up to 500 MiB/sec	Up to 500 MiB/sec							

Standard SSD managed disks

STANDARD SSD SIZE S	E1*	E2*	E3*	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 120	Up to 120	Up to 120	Up to 120	Up to 240	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000					
Throughput per disk	Up to 25 MiB/sec	Up to 50 MiB/sec	Up to 60 MiB/sec	Up to 400 MiB/sec	Up to 600 MiB/sec	Up to 750 MiB/sec								

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

Premium SSD managed disks: Per-disk limits

PREMIUM SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767

PRE MIU M SSD SIZE S	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
-------------------------------------	-----	-----	-----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----

IOP S per disk	120	120	120	120	240	500	1,10 0	2,30 0	5,00 0	7,50 0	7,50 0	16,0 00	18,0 00	20,0 00
Throughput per disk	25 MiB /sec	25 MiB /sec	25 MiB /sec	25 MiB /sec	50 MiB /sec	100 MiB /sec	125 MiB /sec	150 MiB /sec	200 MiB /sec	250 MiB /sec	250 MiB /sec	500 MiB /sec	750 MiB /sec	900 MiB /sec
Max burst IOP S per disk **	3,5 00	3,5 00	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0	3,50 0
Max burst throughput per disk **	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec	170 MiB /sec								
Max burst duration**	30 min	30 min	30 min	30 min	30 min	30 min								
Eligible for reservation	No	Yes, up to one year												

*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

**Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

Premium SSD managed disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: Per-disk limits

VM TIER	BASIC TIER VM	STANDARD TIER VM
Disk size	4,095 GB	4,095 GB
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum IOPS	66	40

Premium unmanaged virtual machine disks: Per-account limits

RESOURCE	DEFAULT LIMIT
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB
Maximum bandwidth per account (ingress + egress) ¹	<=50 Gbps

¹*Ingress* refers to all data from requests that are sent to a storage account. *Egress* refers to all data from responses that are received from a storage account.

Premium unmanaged virtual machine disks: Per-disk limits

PREMIUM STORAGE DISK TYPE	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

See also

[Azure subscription and service limits, quotas, and constraints](#)

Backup and disaster recovery for Azure IaaS disks

12/10/2019 • 21 minutes to read • [Edit Online](#)

This article explains how to plan for backup and disaster recovery (DR) of IaaS virtual machines (VMs) and disks in Azure. This document covers both managed and unmanaged disks.

First, we cover the built-in fault tolerance capabilities in the Azure platform that helps guard against local failures. We then discuss the disaster scenarios not fully covered by the built-in capabilities. We also show several examples of workload scenarios where different backup and DR considerations can apply. We then review possible solutions for the DR of IaaS disks.

Introduction

The Azure platform uses various methods for redundancy and fault tolerance to help protect customers from localized hardware failures. Local failures can include problems with an Azure Storage server machine that stores part of the data for a virtual disk or failures of an SSD or HDD on that server. Such isolated hardware component failures can happen during normal operations.

The Azure platform is designed to be resilient to these failures. Major disasters can result in failures or the inaccessibility of many storage servers or even a whole datacenter. Although your VMs and disks are normally protected from localized failures, additional steps are necessary to protect your workload from region-wide catastrophic failures, such as a major disaster, that can affect your VM and disks.

In addition to the possibility of platform failures, problems with a customer application or data can occur. For example, a new version of your application might inadvertently make a change to the data that causes it to break. In that case, you might want to revert the application and the data to a prior version that contains the last known good state. This requires maintaining regular backups.

For regional disaster recovery, you must back up your IaaS VM disks to a different region.

Before we look at backup and DR options, let's recap a few methods available for handling localized failures.

Azure IaaS resiliency

Resiliency refers to the tolerance for normal failures that occur in hardware components. Resiliency is the ability to recover from failures and continue to function. It's not about avoiding failures, but responding to failures in a way that avoids downtime or data loss. The goal of resiliency is to return the application to a fully functioning state following a failure. Azure virtual machines and disks are designed to be resilient to common hardware faults. Let's look at how the Azure IaaS platform provides this resiliency.

A virtual machine consists mainly of two parts: a compute server and the persistent disks. Both affect the fault tolerance of a virtual machine.

If the Azure compute host server that houses your VM experiences a hardware failure, which is rare, Azure is designed to automatically restore the VM on another server. If this scenario, your computer reboots, and the VM comes back up after some time. Azure automatically detects such hardware failures and executes recoveries to help ensure the customer VM is available as soon as possible.

Regarding IaaS disks, the durability of data is critical for a persistent storage platform. Azure customers have important business applications running on IaaS, and they depend on the persistence of the data. Azure designs protection for these IaaS disks, with three redundant copies of the data that is stored locally. These copies provide for high durability against local failures. If one of the hardware components that holds your disk fails, your VM is not affected, because there are two additional copies to support disk requests. It works fine, even if two different

hardware components that support a disk fail at the same time (which is rare).

To ensure that you always maintain three replicas, Azure Storage automatically spawns a new copy of the data in the background if one of the three copies becomes unavailable. Therefore, it should not be necessary to use RAID with Azure disks for fault tolerance. A simple RAID 0 configuration should be sufficient for striping the disks, if necessary, to create larger volumes.

Because of this architecture, Azure has consistently delivered enterprise-grade durability for IaaS disks, with an industry-leading zero percent [annualized failure rate](#).

Localized hardware faults on the compute host or in the Storage platform can sometimes result in the temporary unavailability of the VM that is covered by the [Azure SLA](#) for VM availability. Azure also provides an industry-leading SLA for single VM instances that use Azure premium SSDs.

To safeguard application workloads from downtime due to the temporary unavailability of a disk or VM, customers can use [availability sets](#). Two or more virtual machines in an availability set provide redundancy for the application. Azure then creates these VMs and disks in separate fault domains with different power, network, and server components.

Because of these separate fault domains, localized hardware failures typically do not affect multiple VMs in the set at the same time. Having separate fault domains provides high availability for your application. It's considered a good practice to use availability sets when high availability is required. The next section covers the disaster recovery aspect.

Backup and disaster recovery

Disaster recovery is the ability to recover from rare, but major, incidents. These incidents include non-transient, wide-scale failures, such as service disruption that affects an entire region. Disaster recovery includes data backup and archiving, and might include manual intervention, such as restoring a database from a backup.

The Azure platform's built-in protection against localized failures might not fully protect the VMs/disks if a major disaster causes large-scale outages. These large-scale outages include catastrophic events, such as if a datacenter is hit by a hurricane, earthquake, fire, or if there is a large-scale hardware unit failure. In addition, you might encounter failures due to application or data issues.

To help protect your IaaS workloads from outages, you should plan for redundancy and have backups to enable recovery. For disaster recovery, you should back up in a different geographic location away from the primary site. This approach helps ensure your backup is not affected by the same event that originally affected the VM or disks. For more information, see [Disaster recovery for Azure applications](#).

Your DR considerations might include the following aspects:

- High availability: The ability of the application to continue running in a healthy state, without significant downtime. By *healthy state*, this state means that the application is responsive, and users can connect to the application and interact with it. Certain mission-critical applications and databases might be required to always be available, even when there are failures in the platform. For these workloads, you might need to plan redundancy for the application, as well as the data.
- Data durability: In some cases, the main consideration is ensuring that the data is preserved if a disaster happens. Therefore, you might need a backup of your data in a different site. For such workloads, you might not need full redundancy for the application, but only a regular backup of the disks.

Backup and DR scenarios

Let's look at a few typical examples of application workload scenarios and the considerations for planning for disaster recovery.

Scenario 1: Major database solutions

Consider a production database server, like SQL Server or Oracle, that can support high availability. Critical production applications and users depend on this database. The disaster recovery plan for this system might need to support the following requirements:

- The data must be protected and recoverable.
- The server must be available for use.

The disaster recovery plan might require maintaining a replica of the database in a different region as a backup. Depending on the requirements for server availability and data recovery, the solution might range from an active-active or active-passive replica site to periodic offline backups of the data. Relational databases, such as SQL Server and Oracle, provide various options for replication. For SQL Server, use [SQL Server AlwaysOn Availability Groups](#) for high availability.

NoSQL databases, like MongoDB, also support [replicas](#) for redundancy. The replicas for high availability are used.

Scenario 2: A cluster of redundant VMs

Consider a workload handled by a cluster of VMs that provide redundancy and load balancing. One example is a Cassandra cluster deployed in a region. This type of architecture already provides a high level of redundancy within that region. However, to protect the workload from a regional-level failure, you should consider spreading the cluster across two regions or making periodic backups to another region.

Scenario 3: IaaS application workload

Let's look at the IaaS application workload. For example, this application might be a typical production workload running on an Azure VM. It might be a web server or file server holding the content and other resources of a site. It might also be a custom-built business application running on a VM that stored its data, resources, and application state on the VM disks. In this case, it's important to make sure you take backups on a regular basis. Backup frequency should be based on the nature of the VM workload. For example, if the application runs every day and modifies data, then the backup should be taken every hour.

Another example is a reporting server that pulls data from other sources and generates aggregated reports. The loss of this VM or disks might lead to the loss of the reports. However, it might be possible to rerun the reporting process and regenerate the output. In that case, you don't really have a loss of data, even if the reporting server is hit with a disaster. As a result, you might have a higher level of tolerance for losing part of the data on the reporting server. In that case, less frequent backups are an option to reduce costs.

Scenario 4: IaaS application data issues

IaaS application data issues are another possibility. Consider an application that computes, maintains, and serves critical commercial data, such as pricing information. A new version of your application had a software bug that incorrectly computed the pricing and corrupted the existing commerce data served by the platform. Here, the best course of action is to revert to the earlier version of the application and the data. To enable this, take periodic backups of your system.

Disaster recovery solution: Azure Backup

[Azure Backup](#) is used for backups and DR, and it works with [managed disks](#) as well as unmanaged disks. You can create a backup job with time-based backups, easy VM restoration, and backup retention policies.

If you use [premium SSDs](#), [managed disks](#), or other disk types with the [locally redundant storage](#) option, it's especially important to make periodic DR backups. Azure Backup stores the data in your recovery services vault for long-term retention. Choose the [geo-redundant storage](#) option for the backup recovery services vault. That option ensures that backups are replicated to a different Azure region for safeguarding from regional disasters.

For unmanaged disks, you can use the locally redundant storage type for IaaS disks, but ensure that Azure Backup is enabled with the geo-redundant storage option for the recovery services vault.

NOTE

If you use the [geo-redundant storage](#) or [read-access geo-redundant storage](#) option for your unmanaged disks, you still need consistent snapshots for backup and DR. Use either [Azure Backup](#) or [consistent snapshots](#).

The following table is a summary of the solutions available for DR.

SCENARIO	AUTOMATIC REPLICATION	DR SOLUTION
Premium SSD disks	Local (locally redundant storage)	Azure Backup
Managed disks	Local (locally redundant storage)	Azure Backup
Unmanaged locally redundant storage disks	Local (locally redundant storage)	Azure Backup
Unmanaged geo-redundant storage disks	Cross region (geo-redundant storage)	Azure Backup Consistent snapshots
Unmanaged read-access geo-redundant storage disks	Cross region (read-access geo-redundant storage)	Azure Backup Consistent snapshots

High availability is best met by using managed disks in an availability set along with Azure Backup. If you use unmanaged disks, you can still use Azure Backup for DR. If you are unable to use Azure Backup, then taking [consistent snapshots](#), as described in a later section, is an alternative solution for backup and DR.

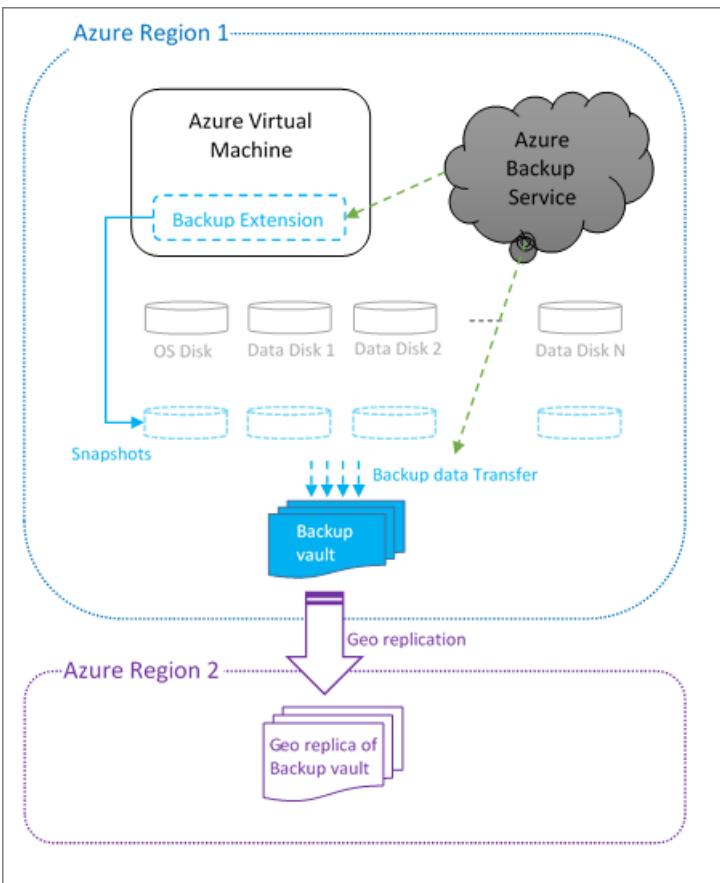
Your choices for high availability, backup, and DR at application or infrastructure levels can be represented as follows:

LEVEL	HIGH AVAILABILITY	BACKUP OR DR
Application	SQL Server AlwaysOn	Azure Backup
Infrastructure	Availability set	Geo-redundant storage with consistent snapshots

Using Azure Backup

[Azure Backup](#) can back up your VMs running Windows or Linux to the Azure recovery services vault. Backing up and restoring business-critical data is complicated by the fact that business-critical data must be backed up while the applications that produce the data are running.

To address this issue, Azure Backup provides application-consistent backups for Microsoft workloads. It uses the volume shadow service to ensure that data is written correctly to storage. For Linux VMs, the default backup consistency mode is file-consistent backups, because Linux does not have functionality equivalent to the volume shadow service as in the case of Windows. For Linux machines, see [Application-consistent backup of Azure Linux VMs](#).



When Azure Backup initiates a backup job at the scheduled time, it triggers the backup extension installed in the VM to take a point-in-time snapshot. A snapshot is taken in coordination with the volume shadow service to get a consistent snapshot of the disks in the virtual machine without having to shut it down. The backup extension in the VM flushes all writes before taking a consistent snapshot of all of the disks. After taking the snapshot, the data is transferred by Azure Backup to the backup vault. To make the backup process more efficient, the service identifies and transfers only the blocks of data that have changed after the last backup.

To restore, you can view the available backups through Azure Backup and then initiate a restore. You can create and restore Azure backups through the [Azure portal](#), by [using PowerShell](#), or by using the [Azure CLI](#).

Steps to enable a backup

Use the following steps to enable backups of your VMs by using the [Azure portal](#). There is some variation depending on your exact scenario. Refer to the [Azure Backup](#) documentation for full details. Azure Backup also [supports VMs with managed disks](#).

1. Create a recovery services vault for a VM:
 - a. In the [Azure portal](#), browse **All resources** and find **Recovery Services vaults**.
 - b. On the **Recovery Services vaults** menu, click **Add** and follow the steps to create a new vault in the same region as the VM. For example, if your VM is in the West US region, pick West US for the vault.
2. Verify the storage replication for the newly created vault. Access the vault under **Recovery Services vaults** and go to **Properties > Backup Configuration > Update**. Ensure the **geo-redundant storage** option is selected by default. This option ensures that your vault is automatically replicated to a secondary datacenter. For example, your vault in West US is automatically replicated to East US.
3. Configure the backup policy and select the VM from the same UI.
4. Make sure the Backup Agent is installed on the VM. If your VM is created by using an Azure gallery image, then the Backup Agent is already installed. Otherwise (that is, if you use a custom image), use the instructions to [install the VM agent on a virtual machine](#).

5. After the previous steps are completed, the backup runs at regular intervals as specified in the backup policy. If necessary, you can trigger the first backup manually from the vault dashboard on the Azure portal.

For automating Azure Backup by using scripts, refer to [PowerShell cmdlets for VM backup](#).

Steps for recovery

If you need to repair or rebuild a VM, you can restore the VM from any of the backup recovery points in the vault. There are a couple of different options for performing the recovery:

- You can create a new VM as a point-in-time representation of your backed-up VM.
- You can restore the disks, and then use the template for the VM to customize and rebuild the restored VM.

For more information, see the instructions to [use the Azure portal to restore virtual machines](#). This document also explains the specific steps for restoring backed-up VMs to a paired datacenter by using your geo-redundant backup vault if there is a disaster at the primary datacenter. In that case, Azure Backup uses the Compute service from the secondary region to create the restored virtual machine.

You can also use PowerShell for [creating a new VM from restored disks](#).

Alternative solution: Consistent snapshots

If you are unable to use Azure Backup, you can implement your own backup mechanism by using snapshots. Creating consistent snapshots for all the disks used by a VM and then replicating those snapshots to another region is complicated. For this reason, Azure considers using the Backup service as a better option than building a custom solution.

If you use read-access geo-redundant storage/geo-redundant storage for disks, snapshots are automatically replicated to a secondary datacenter. If you use locally redundant storage for disks, you need to replicate the data yourself. For more information, see [Back up Azure-unmanaged VM disks with incremental snapshots](#).

A snapshot is a representation of an object at a specific point in time. A snapshot incurs billing for the incremental size of the data it holds. For more information, see [Create a blob snapshot](#).

Create snapshots while the VM is running

Although you can take a snapshot at any time, if the VM is running, there is still data being streamed to the disks. The snapshots might contain partial operations that were in flight. Also, if there are several disks involved, the snapshots of different disks might have occurred at different times. These scenarios may cause the snapshots to be uncoordinated. This lack of co-ordination is especially problematic for striped volumes whose files might be corrupted if changes were being made during backup.

To avoid this situation, the backup process must implement the following steps:

1. Freeze all the disks.
2. Flush all the pending writes.
3. [Create a blob snapshot](#) for all the disks.

Some Windows applications, like SQL Server, provide a coordinated backup mechanism via a volume shadow service to create application-consistent backups. On Linux, you can use a tool like `fsfreeze` for coordinating the disks. This tool provides file-consistent backups, but not application-consistent snapshots. This process is complex, so you should consider using [Azure Backup](#) or a third-party backup solution that already implements this procedure.

The previous process results in a collection of coordinated snapshots for all of the VM disks, representing a specific point-in-time view of the VM. This is a backup restore point for the VM. You can repeat the process at scheduled intervals to create periodic backups. See [Copy the backups to another region](#) for steps to copy the snapshots to

another region for DR.

Create snapshots while the VM is offline

Another option to create consistent backups is to shut down the VM and take blob snapshots of each disk. Taking blob snapshots is easier than coordinating snapshots of a running VM, but it requires a few minutes of downtime.

1. Shut down the VM.
2. Create a snapshot of each virtual hard drive blob, which only takes a few seconds.

To create a snapshot, you can use [PowerShell](#), the [Azure Storage REST API](#), [Azure CLI](#), or one of the Azure Storage client libraries, such as [the Storage client library for .NET](#).

3. Start the VM, which ends the downtime. Typically, the entire process finishes within a few minutes.

This process yields a collection of consistent snapshots for all the disks, providing a backup restore point for the VM.

Copy the snapshots to another region

Creation of the snapshots alone might not be sufficient for DR. You must also replicate the snapshot backups to another region.

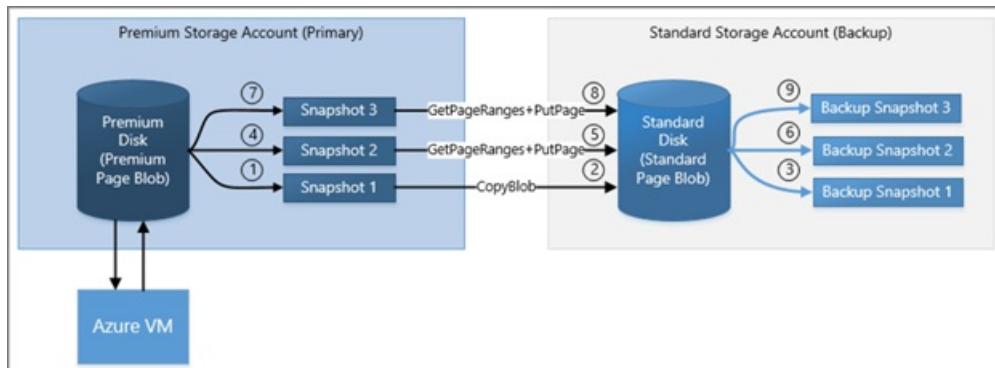
If you use geo-redundant storage or read-access geo-redundant storage for your disks, then the snapshots are replicated to the secondary region automatically. There can be a few minutes of lag before the replication. If the primary datacenter goes down before the snapshots finish replicating, you cannot access the snapshots from the secondary datacenter. The likelihood of this is small.

NOTE

Only having the disks in a geo-redundant storage or read-access geo-redundant storage account does not protect the VM from disasters. You must also create coordinated snapshots or use Azure Backup. This is required to recover a VM to a consistent state.

If you use locally redundant storage, you must copy the snapshots to a different storage account immediately after creating the snapshot. The copy target might be a locally redundant storage account in a different region, resulting in the copy being in a remote region. You can also copy the snapshot to a read-access geo-redundant storage account in the same region. In this case, the snapshot is lazily replicated to the remote secondary region. Your backup is protected from disasters at the primary site after the copying and replication is complete.

To copy your incremental snapshots for DR efficiently, review the instructions in [Back up Azure unmanaged VM disks with incremental snapshots](#).



Recovery from snapshots

To retrieve a snapshot, copy it to make a new blob. If you are copying the snapshot from the primary account, you can copy the snapshot over to the base blob of the snapshot. This process reverts the disk to the snapshot. This process is known as promoting the snapshot. If you are copying the snapshot backup from a secondary account, in

the case of a read-access geo-redundant storage account, you must copy it to a primary account. You can copy a snapshot by [using PowerShell](#) or by using the AzCopy utility. For more information, see [Transfer data with the AzCopy command-line utility](#).

For VMs with multiple disks, you must copy all the snapshots that are part of the same coordinated restore point. After you copy the snapshots to writable VHD blobs, you can use the blobs to recreate your VM by using the template for the VM.

Other options

SQL Server

SQL Server running in a VM has its own built-in capabilities to back up your SQL Server database to Azure Blob storage or a file share. If the storage account is geo-redundant storage or read-access geo-redundant storage, you can access those backups in the storage account's secondary datacenter in the event of a disaster, with the same restrictions as previously discussed. For more information, see [Back up and restore for SQL Server in Azure virtual machines](#). In addition to back up and restore, [SQL Server AlwaysOn availability groups](#) can maintain secondary replicas of databases. This ability greatly reduces the disaster recovery time.

Other considerations

This article has discussed how to back up or take snapshots of your VMs and their disks to support disaster recovery and how to use those backups or snapshots to recover your data. With the Azure Resource Manager model, many people use templates to create their VMs and other infrastructures in Azure. You can use a template to create a VM that has the same configuration every time. If you use custom images for creating your VMs, you must also make sure that your images are protected by using a read-access geo-redundant storage account to store them.

Consequently, your backup process can be a combination of two things:

- Back up the data (disks).
- Back up the configuration (templates and custom images).

Depending on the backup option you choose, you might have to handle the backup of both the data and the configuration, or the backup service might handle all of that for you.

Appendix: Understanding the impact of data redundancy

For storage accounts in Azure, there are three types of data redundancy that you should consider regarding disaster recovery: locally redundant, geo-redundant, or geo-redundant with read access.

Locally redundant storage retains three copies of the data in the same datacenter. When the VM writes the data, all three copies are updated before success is returned to the caller, so you know they are identical. Your disk is protected from local failures, because it's unlikely that all three copies are affected at the same time. In the case of locally redundant storage, there is no geo-redundancy, so the disk is not protected from catastrophic failures that can affect an entire datacenter or storage unit.

With geo-redundant storage and read-access geo-redundant storage, three copies of your data are retained in the primary region that is selected by you. Three more copies of your data are retained in a corresponding secondary region that is set by Azure. For example, if you store data in West US, the data is replicated to East US. Copy retention is done asynchronously, and there is a small delay between updates to the primary and secondary sites. Replicas of the disks on the secondary site are consistent on a per-disk basis (with the delay), but replicas of multiple active disks might not be in sync with each other. To have consistent replicas across multiple disks, consistent snapshots are needed.

The main difference between geo-redundant storage and read-access geo-redundant storage is that with read-

access geo-redundant storage, you can read the secondary copy at any time. If there is a problem that renders the data in the primary region inaccessible, the Azure team makes every effort to restore access. While the primary is down, if you have read-access geo-redundant storage enabled, you can access the data in the secondary datacenter. Therefore, if you plan to read from the replica while the primary is inaccessible, then read-access geo-redundant storage should be considered.

If it turns out to be a significant outage, the Azure team might trigger a geo-failover and change the primary DNS entries to point to secondary storage. At this point, if you have either geo-redundant storage or read-access geo-redundant storage enabled, you can access the data in the region that used to be the secondary. In other words, if your storage account is geo-redundant storage and there is a problem, you can access the secondary storage only if there is a geo-failover.

For more information, see [What to do if an Azure Storage outage occurs](#).

NOTE

Microsoft controls whether a failover occurs. Failover is not controlled per storage account, so it's not decided by individual customers. To implement disaster recovery for specific storage accounts or virtual machine disks, you must use the techniques described previously in this article.

Azure shared disks

2/19/2020 • 4 minutes to read • [Edit Online](#)

Azure shared disks (preview) is a new feature for Azure managed disks that enables attaching an Azure managed disk to multiple virtual machines (VMs) simultaneously. Attaching a managed disk to multiple VMs allows you to either deploy new or migrate existing clustered applications to Azure.

How it works

VMs in the cluster can read or write to your attached disk based on the reservation chosen by the clustered application using [SCSI Persistent Reservations](#) (SCSI PR). SCSI PR is a well-known industry standard leveraged by applications running on Storage Area Network (SAN) on-premises. Enabling SCSI PR on a managed disk allows you to migrate these applications to Azure as-is.

Managed disks with shared disks enabled offer shared block storage that can be accessed by multiple VMs, this is exposed as logical unit numbers (LUNs). LUNs are then presented to an initiator (VM) from a target (disk). These LUNs look like direct-attached-storage (DAS) or a local drive to the VM.

Managed disks with shared disks enabled do not natively offer a fully-managed file system that can be accessed using SMB/NFS. You will need to use a cluster manager, like Windows Server Failover Cluster (WSFC) or Pacemaker, that handles cluster node communication as well as write locking.

Limitations

While in preview, managed disks that have shared disks enabled are subject to the following limitations:

- Currently only available with premium SSDs.
- Currently only supported in the West Central US region.
- All virtual machines sharing a disk must be deployed in the same [proximity placement groups](#).
- Can only be enabled on data disks, not OS disks.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see [Failover clustering hardware requirements and storage options](#).
- ReadOnly host caching is not available for premium SSDs with `maxShares>1`.
- Availability sets and virtual machine scale sets can only be used with `FaultDomainCount` set to 1.
- Azure Backup and Azure Site Recovery support is not yet available.

If you're interested in trying shared disks then [sign up for our preview](#).

Disk sizes

For now, only premium SSDs can enable shared disks. The disk sizes that support this feature are P15 and greater. Different disk sizes may have a different `maxShares` limit, which you cannot exceed when setting the `maxshares` value.

For each disk, you can define a `maxShares` value that represents the maximum number of nodes that can simultaneously share the disk. For example, if you plan to set up a 2-node failover cluster, you would set `maxShares=2`. The maximum value is an upper bound. Nodes can join or leave the cluster (mount or unmount the disk) as long as the number of nodes is lower than the specified `maxShares` value.

NOTE

The `maxShares` value can only be set or edited when the disk is detached from all nodes.

The following table illustrates the allowed maximum values for `maxShares` by disk size:

DISK SIZES	MAXSHARES LIMIT
P15, P20	2
P30, P40, P50	5
P60, P70, P80	10

The IOPS and bandwidth limits for a disk are not affected by the `maxShares` value. For example, the max IOPS of a P15 disk are 1100 whether `maxShares = 1` or `maxShares > 1`.

Sample workloads

Windows

Most Windows-based clustering build on WSFC, which handles all core infrastructure for cluster node communication, allowing your applications to take advantage of parallel access patterns. WSFC enables both CSV and non-CVS-based options depending on your version of Windows Server. For details, refer to [Create a failover cluster](#).

Some popular applications running on WSFC include:

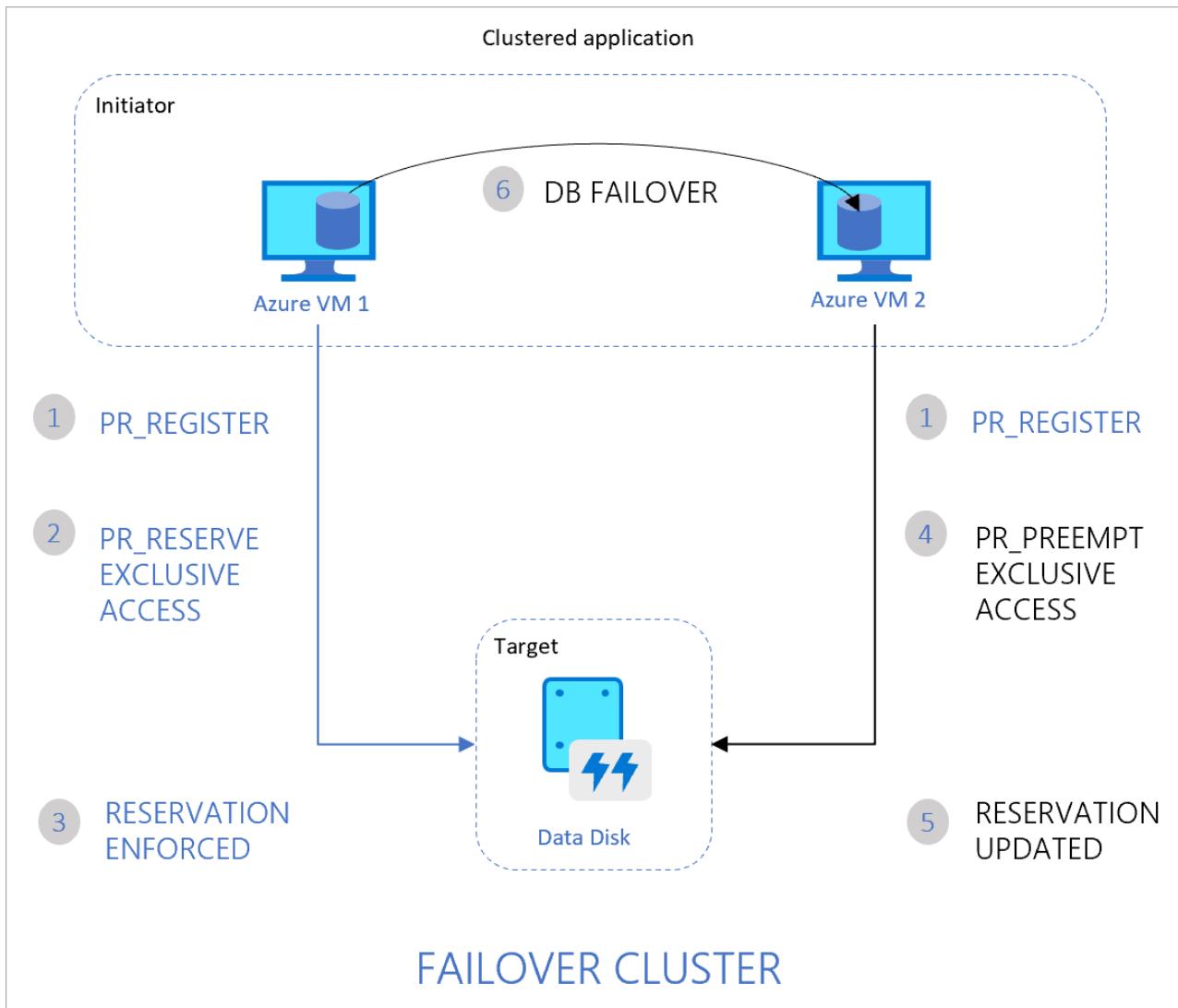
- SQL Server Failover Cluster Instances (FCI)
- Scale-out File Server (SoFS)
- File Server for General Use (IW workload)
- Remote Desktop Server User Profile Disk (RDS UPD)
- SAP ASCS/SCS

Linux

Linux clusters can leverage cluster managers such as [Pacemaker](#). Pacemaker builds on [Corosync](#), enabling cluster communications for applications deployed in highly available environments. Some common clustered filesystems include [ocfs2](#) and [gfs2](#). You can manipulate reservations and registrations using utilities such as [fence_scsi](#) and [sg_persist](#).

Persistent Reservation flow

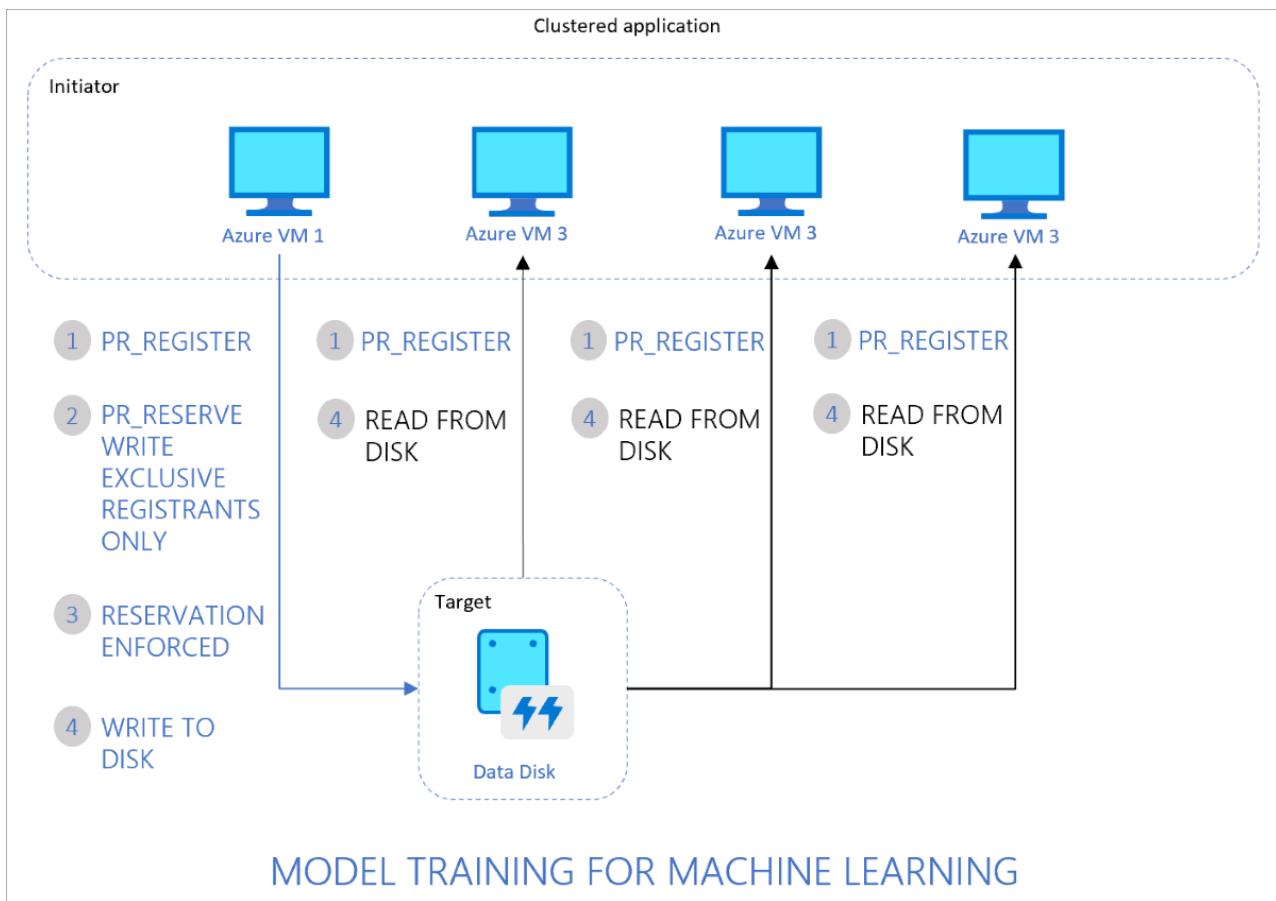
The following diagram illustrates a sample 2-node clustered database application that leverages SCSI PR to enable failover from one node to the other.



The flow is as follows:

1. The clustered application running on both Azure VM1 and VM2 registers its intent to read or write to the disk.
2. The application instance on VM1 then takes exclusive reservation to write to the disk.
3. This reservation is enforced on your Azure disk and the database can now exclusively write to the disk. Any writes from the application instance on VM2 will not succeed.
4. If the application instance on VM1 goes down, the instance on VM2 can now initiate a database failover and take-over of the disk.
5. This reservation is now enforced on the Azure disk and the disk will no longer accept writes from VM1. It will only accept writes from VM2.
6. The clustered application can complete the database failover and serve requests from VM2.

The following diagram illustrates another common clustered workload consisting of multiple nodes reading data from the disk for running parallel processes, such as training of machine learning models.



The flow is as follows:

1. The clustered application running on all VMs registers the intent to read or write to the disk.
2. The application instance on VM1 takes an exclusive reservation to write to the disk while opening up reads to the disk from other VMs.
3. This reservation is enforced on your Azure disk.
4. All nodes in the cluster can now read from the disk. Only one node writes back results to the disk, on behalf of all nodes in the cluster.

Next steps

If you're interested in enabling and using shared disks for your managed disks, proceed to our article [Enable shared disk](#).

Ephemeral OS disks for Azure VMs

11/13/2019 • 6 minutes to read • [Edit Online](#)

Ephemeral OS disks are created on the local virtual machine (VM) storage and not saved to the remote Azure Storage. Ephemeral OS disks work well for stateless workloads, where applications are tolerant of individual VM failures, but are more affected by VM deployment time or reimaging the individual VM instances. With Ephemeral OS disk, you get lower read/write latency to the OS disk and faster VM reimage.

The key features of ephemeral disks are:

- Ideal for stateless applications.
- They can be used with both Marketplace and custom images.
- Ability to fast reset or reimagine VMs and scale set instances to the original boot state.
- Lower latency, similar to a temporary disk.
- Ephemeral OS disks are free, you incur no storage cost for OS disk.
- They are available in all Azure regions.
- Ephemeral OS Disk is supported by [Shared Image Gallery](#).

Key differences between persistent and ephemeral OS disks:

	PERSISTENT OS DISK	EPHEMERAL OS DISK
Size limit for OS disk	2 TiB	Cache size for the VM size or 2 TiB, whichever is smaller. For the cache size in GiB , see DS , ES , M , FS , and GS
VM sizes supported	All	DSv1, DSv2, DSv3, Esv3, Fs, FsV2, GS, M
Disk type support	Managed and unmanaged OS disk	Managed OS disk only
Region support	All regions	All regions
Data persistence	OS disk data written to OS disk are stored in Azure Storage	Data written to OS disk is stored to the local VM storage and is not persisted to Azure Storage.
Stop-deallocated state	VMs and scale set instances can be stop-deallocated and restarted from the stop-deallocated state	VMs and scale set instances cannot be stop-deallocated
Specialized OS disk support	Yes	No
OS disk resize	Supported during VM creation and after VM is stop-deallocated	Supported during VM creation only

	PERSISTENT OS DISK	EPHEMERAL OS DISK
Resizing to a new VM size	OS disk data is preserved	Data on the OS disk is deleted, OS is re-provisioned

Size requirements

You can deploy VM and instance images up to the size of the VM cache. For example, Standard Windows Server images from the marketplace are about 127 GiB, which means that you need a VM size that has a cache larger than 127 GiB. In this case, the [Standard_DS2_v2](#) has a cache size of 86 GiB, which is not large enough. The Standard_DS3_v2 has a cache size of 172 GiB, which is large enough. In this case, the Standard_DS3_v2 is the smallest size in the DSv2 series that you can use with this image. Basic Linux images in the Marketplace and Windows Server images that are denoted by `[smallsize]` tend to be around 30 GiB and can use most of the available VM sizes.

Ephemeral disks also require that the VM size supports Premium storage. The sizes usually (but not always) have an `s` in the name, like DSv2 and EsV3. For more information, see [Azure VM sizes](#) for details around which sizes support Premium storage.

PowerShell

To use an ephemeral disk for a PowerShell VM deployment, use [Set-AzVMOSDisk](#) in your VM configuration. Set the `-DiffDiskSetting` to `Local` and `-Caching` to `ReadOnly`.

```
Set-AzVMOSDisk -DiffDiskSetting Local -Caching ReadOnly
```

For scale set deployments, use the [Set-AzVmssStorageProfile](#) cmdlet in your configuration. Set the `-DiffDiskSetting` to `Local` and `-Caching` to `ReadOnly`.

```
Set-AzVmssStorageProfile -DiffDiskSetting Local -OsDiskCaching ReadOnly
```

CLI

To use an ephemeral disk for a CLI VM deployment, set the `--ephemeral-os-disk` parameter in [az vm create](#) to `true` and the `--os-disk-caching` parameter to `ReadOnly`.

```
az vm create \
--resource-group myResourceGroup \
--name myVM \
--image UbuntuLTS \
--ephemeral-os-disk true \
--os-disk-caching ReadOnly \
--admin-username azureuser \
--generate-ssh-keys
```

For scale sets, you use the same `--ephemeral-os-disk true` parameter for [az-vmss-create](#) and set the `--os-disk-caching` parameter to `ReadOnly`.

Portal

In the Azure portal, you can choose to use ephemeral disks when deploying a VM by opening the **Advanced** section of the **Disks** tab. For **Use ephemeral OS disk** select **Yes**.

Home > New > Create a virtual machine

Create a virtual machine

Basics **Disks** **Networking** **Management** **Advanced** **Tags** **Review + create**

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

* OS disk type Ephemeral OS Disks only support the Standard HDD disk type.

Enable Ultra SSD compatibility (Preview) Yes No
Ultra SSD disks are not available when using ephemeral disks.

Data disks

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	NAME	SIZE (GiB)	DISK TYPE	HOST CACHING

[Create and attach a new disk](#) [Attach an existing disk](#)

Advanced

Use managed disks No Yes

Use ephemeral OS disk No Yes

If the option for using an ephemeral disk is greyed out, you might have selected a VM size that does not have a cache size larger than the OS image or that doesn't support Premium storage. Go back to the **Basics** page and try choosing another VM size.

You can also create scale-sets with ephemeral OS disks using the portal. Just make sure you select a VM size with a large enough cache size and then in **Use ephemeral OS disk** select **Yes**.

INSTANCES

* Instance count

* Instance size **Standard DS3 v2**
4 vcpus, 14 GiB memory [Change size](#)

Deploy as low priority (preview) No Yes

Use managed disks No Yes

Use ephemeral OS disk No Yes

Scale set template deployment

The process to create a scale set that uses an ephemeral OS disk is to add the `diffDiskSettings` property to the `Microsoft.Compute/virtualMachineScaleSets/virtualMachineProfile` resource type in the template. Also, the caching policy must be set to `ReadOnly` for the ephemeral OS disk.

```
{
  "type": "Microsoft.Compute/virtualMachineScaleSets",
  "name": "myScaleSet",
  "location": "East US 2",
  "apiVersion": "2018-06-01",
  "sku": {
    "name": "Standard_DS2_v2",
    "capacity": "2"
  },
  "properties": {
    "upgradePolicy": {
      "mode": "Automatic"
    },
    "virtualMachineProfile": {
      "storageProfile": {
        "osDisk": {
          "diffDiskSettings": {
            "option": "Local"
          },
          "caching": "ReadOnly",
          "createOption": "FromImage"
        },
        "imageReference": {
          "publisher": "Canonical",
          "offer": "UbuntuServer",
          "sku": "16.04-LTS",
          "version": "latest"
        }
      },
      "osProfile": {
        "computerNamePrefix": "myvmss",
        "adminUsername": "azureuser",
        "adminPassword": "P@ssw0rd!"
      }
    }
  }
}
```

VM template deployment

You can deploy a VM with an ephemeral OS disk using a template. The process to create a VM that uses ephemeral OS disks is to add the `diffDiskSettings` property to the `Microsoft.Compute/virtualMachines` resource type in the template. Also, the caching policy must be set to `ReadOnly` for the ephemeral OS disk.

```
{
  "type": "Microsoft.Compute/virtualMachines",
  "name": "myVirtualMachine",
  "location": "East US 2",
  "apiVersion": "2018-06-01",
  "properties": {
    "storageProfile": {
      "osDisk": {
        "diffDiskSettings": {
          "option": "Local"
        },
        "caching": "ReadOnly",
        "createOption": "FromImage"
      },
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2016-Datacenter-smalldisk",
        "version": "latest"
      },
      "hardwareProfile": {
        "vmSize": "Standard_DS2_v2"
      }
    },
    "osProfile": {
      "computerNamePrefix": "myvirtualmachine",
      "adminUsername": "azureuser",
      "adminPassword": "P@ssw0rd!"
    }
  }
}
```

Reimage a VM using REST

You can reimage a Virtual Machine instance with ephemeral OS disk using REST API as described below and via Azure Portal by going to Overview pane of the VM. For scale sets, reimaging is already available through Powershell, CLI, and the portal.

```
POST https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rgName}/providers/Microsoft.Compute/VirtualMachines/{vmName}/reimage?api-version=2018-06-01"
```

Frequently asked questions

Q: What is the size of the local OS Disks?

A: We support platform and custom images, up to the VM cache size, where all read/writes to the OS disk will be local on the same node as the Virtual Machine.

Q: Can the ephemeral OS disk be resized?

A: No, once the ephemeral OS disk is provisioned, the OS disk cannot be resized.

Q: Can I attach a Managed Disks to an Ephemeral VM?

A: Yes, you can attach a managed data disk to a VM that uses an ephemeral OS disk.

Q: Will all VM sizes be supported for ephemeral OS disks?

A: No, all Premium Storage VM sizes are supported (DS, ES, FS, GS and M) except the B-series, N-series, and H-series sizes.

Q: Can the ephemeral OS disk be applied to existing VMs and scale sets?

A: No, ephemeral OS disk can only be used during VM and scale set creation.

Q: Can you mix ephemeral and normal OS disks in a scale set?

A: No, you can't have a mix of ephemeral and persistent OS disk instances within the same scale set.

Q: Can the ephemeral OS disk be created using Powershell or CLI?

A: Yes, you can create VMs with Ephemeral OS Disk using REST, Templates, PowerShell and CLI.

Q: What features are not supported with ephemeral OS disk?

A: Ephemeral disks do not support:

- Capturing VM images
- Disk snapshots
- Azure Disk Encryption
- Azure Backup
- Azure Site Recovery
- OS Disk Swap

Next steps

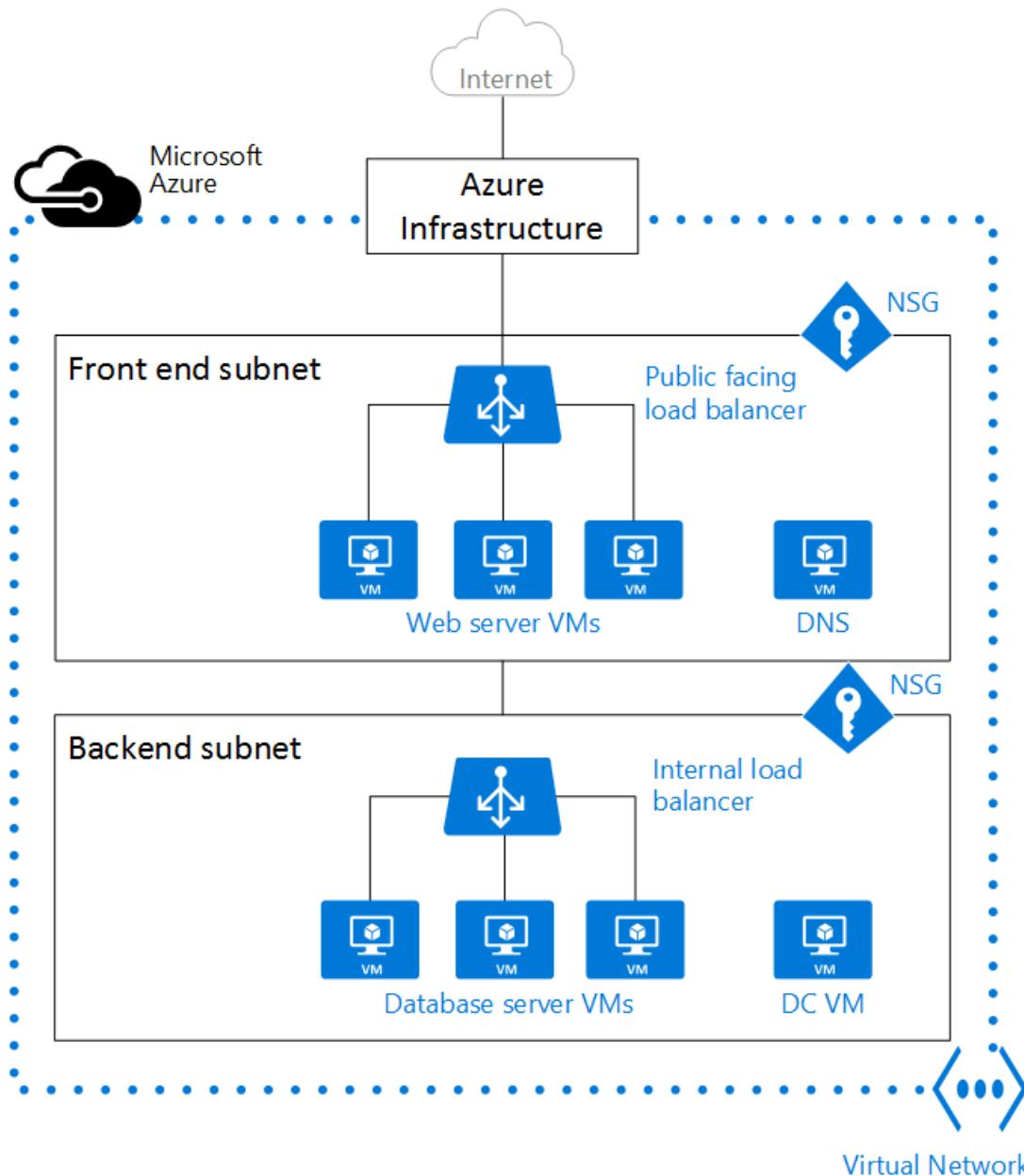
You can create a VM with an ephemeral OS disk using [Azure PowerShell](#).

Virtual networks and virtual machines in Azure

11/13/2019 • 13 minutes to read • [Edit Online](#)

When you create an Azure virtual machine (VM), you must create a [virtual network](#) (VNet) or use an existing VNet. You also need to decide how your VMs are intended to be accessed on the VNet. It is important to [plan before creating resources](#) and make sure that you understand the [limits of networking resources](#).

In the following figure, VMs are represented as web servers and database servers. Each set of VMs are assigned to separate subnets in the VNet.



You can create a VNet before you create a VM or you can do so as you create a VM. You create these resources to support communication with a VM:

- Network interfaces
- IP addresses
- Virtual network and subnets

In addition to those basic resources, you should also consider these optional resources:

- Network security groups
- Load balancers

Network interfaces

A [network interface \(NIC\)](#) is the interconnection between a VM and a virtual network (VNet). A VM must have at least one NIC, but can have more than one, depending on the size of the VM you create. Learn about how many NICs each VM size supports for [Windows](#) or [Linux](#).

You can create a VM with multiple NICs, and add or remove NICs through the lifecycle of a VM. Multiple NICs allow a VM to connect to different subnets and send or receive traffic over the most appropriate interface. VMs with any number of network interfaces can exist in the same availability set, up to the number supported by the VM size.

Each NIC attached to a VM must exist in the same location and subscription as the VM. Each NIC must be connected to a VNet that exists in the same Azure location and subscription as the NIC. You can change the subnet a VM is connected to after it's created, but you cannot change the VNet. Each NIC attached to a VM is assigned a MAC address that doesn't change until the VM is deleted.

This table lists the methods that you can use to create a network interface.

METHOD	DESCRIPTION
Azure portal	When you create a VM in the Azure portal, a network interface is automatically created for you (you cannot use a NIC you create separately). The portal creates a VM with only one NIC. If you want to create a VM with more than one NIC, you must create it with a different method.
Azure PowerShell	Use New-AzNetworkInterface with the -PublicIpAddressId parameter to provide the identifier of the public IP address that you previously created.
Azure CLI	To provide the identifier of the public IP address that you previously created, use az network nic create with the --public-ip-address parameter.
Template	Use Network Interface in a Virtual Network with Public IP Address as a guide for deploying a network interface using a template.

IP addresses

You can assign these types of [IP addresses](#) to a NIC in Azure:

- **Public IP addresses** - Used to communicate inbound and outbound (without network address translation (NAT)) with the Internet and other Azure resources not connected to a VNet. Assigning a public IP address to a NIC is optional. Public IP addresses have a nominal charge, and there's a maximum number that can be used per subscription.
- **Private IP addresses** - Used for communication within a VNet, your on-premises network, and the Internet (with NAT). You must assign at least one private IP address to a VM. To learn more about NAT in Azure, read [Understanding outbound connections in Azure](#).

You can assign public IP addresses to VMs or internet-facing load balancers. You can assign private IP addresses to VMs and internal load balancers. You assign IP addresses to a VM using a network interface.

There are two methods in which an IP address is allocated to a resource - dynamic or static. The default allocation method is dynamic, where an IP address is not allocated when it's created. Instead, the IP address is allocated when you create a VM or start a stopped VM. The IP address is released when you stop or delete the VM.

To ensure the IP address for the VM remains the same, you can set the allocation method explicitly to static. In this case, an IP address is assigned immediately. It is released only when you delete the VM or change its allocation method to dynamic.

This table lists the methods that you can use to create an IP address.

METHOD	DESCRIPTION
Azure portal	By default, public IP addresses are dynamic and the address associated to them may change when the VM is stopped or deleted. To guarantee that the VM always uses the same public IP address, create a static public IP address. By default, the portal assigns a dynamic private IP address to a NIC when creating a VM. You can change this IP address to static after the VM is created.
Azure PowerShell	You use New-AzPublicIpAddress with the -AllocationMethod parameter as Dynamic or Static.
Azure CLI	You use az network public-ip create with the --allocation-method parameter as Dynamic or Static.
Template	Use Network Interface in a Virtual Network with Public IP Address as a guide for deploying a public IP address using a template.

After you create a public IP address, you can associate it with a VM by assigning it to a NIC.

Virtual network and subnets

A subnet is a range of IP addresses in the VNet. You can divide a VNet into multiple subnets for organization and security. Each NIC in a VM is connected to one subnet in one VNet. NICs connected to subnets (same or different) within a VNet can communicate with each other without any extra configuration.

When you set up a VNet, you specify the topology, including the available address spaces and subnets. If the VNet is to be connected to other VNets or on-premises networks, you must select address ranges that don't overlap. The IP addresses are private and can't be accessed from the Internet, which was true only for the non-routable IP addresses such as 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16. Now, Azure treats any address range as part of the private VNet IP address space that is only reachable within the VNet, within interconnected VNets, and from your on-premises location.

If you work within an organization in which someone else is responsible for the internal networks, you should talk to that person before selecting your address space. Make sure there is no overlap and let them know the space you want to use so they don't try to use the same range of IP addresses.

By default, there is no security boundary between subnets, so VMs in each of these subnets can talk to one another. However, you can set up Network Security Groups (NSGs), which allow you to control the traffic flow to and from subnets and to and from VMs.

This table lists the methods that you can use to create a VNet and subnets.

METHOD	DESCRIPTION
Azure portal	If you let Azure create a VNet when you create a VM, the name is a combination of the resource group name that contains the VNet and -vnet . The address space is 10.0.0.0/24, the required subnet name is default , and the subnet address range is 10.0.0.0/24.
Azure PowerShell	You use New-AzVirtualNetworkSubnetConfig and New-AzVirtualNetwork to create a subnet and a VNet. You can also use Add-AzVirtualNetworkSubnetConfig to add a subnet to an existing VNet.
Azure CLI	The subnet and the VNet are created at the same time. Provide a --subnet-name parameter to az network vnet create with the subnet name.
Template	The easiest way to create a VNet and subnets is to download an existing template, such as Virtual Network with two subnets , and modify it for your needs.

Network security groups

A [network security group \(NSG\)](#) contains a list of Access Control List (ACL) rules that allow or deny network traffic to subnets, NICs, or both. NSGs can be associated with either subnets or individual NICs connected to a subnet. When an NSG is associated with a subnet, the ACL rules apply to all the VMs in that subnet. In addition, traffic to an individual NIC can be restricted by associating an NSG directly to a NIC.

NSGs contain two sets of rules: inbound and outbound. The priority for a rule must be unique within each set. Each rule has properties of protocol, source and destination port ranges, address prefixes, direction of traffic, priority, and access type.

All NSGs contain a set of default rules. The default rules cannot be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

When you associate an NSG to a NIC, the network access rules in the NSG are applied only to that NIC. If an NSG is applied to a single NIC on a multi-NIC VM, it does not affect traffic to the other NICs. You can associate different NSGs to a NIC (or VM, depending on the deployment model) and the subnet that a NIC or VM is bound to. Priority is given based on the direction of traffic.

Be sure to [plan](#) your NSGs when you plan your VMs and VNet.

This table lists the methods that you can use to create a network security group.

METHOD	DESCRIPTION
Azure portal	When you create a VM in the Azure portal, an NSG is automatically created and associated to the NIC the portal creates. The name of the NSG is a combination of the name of the VM and -nsg . This NSG contains one inbound rule with a priority of 1000, service set to RDP, the protocol set to TCP, port set to 3389, and action set to Allow. If you want to allow any other inbound traffic to the VM, you must add additional rules to the NSG.

METHOD	DESCRIPTION
Azure PowerShell	Use New-AzNetworkSecurityRuleConfig and provide the required rule information. Use New-AzNetworkSecurityGroup to create the NSG. Use Set-AzVirtualNetworkSubnetConfig to configure the NSG for the subnet. Use Set-AzVirtualNetwork to add the NSG to the VNet.
Azure CLI	Use az network nsg create to initially create the NSG. Use az network nsg rule create to add rules to the NSG. Use az network vnet subnet update to add the NSG to the subnet.
Template	Use Create a Network Security Group as a guide for deploying a network security group using a template.

Load balancers

[Azure Load Balancer](#) delivers high availability and network performance to your applications. A load balancer can be configured to [balance incoming Internet traffic](#) to VMs or [balance traffic between VMs in a VNet](#). A load balancer can also balance traffic between on-premises computers and VMs in a cross-premises network, or forward external traffic to a specific VM.

The load balancer maps incoming and outgoing traffic between the public IP address and port on the load balancer and the private IP address and port of the VM.

When you create a load balancer, you must also consider these configuration elements:

- **Front-end IP configuration** – A load balancer can include one or more front-end IP addresses, otherwise known as virtual IPs (VIPs). These IP addresses serve as ingress for the traffic.
- **Back-end address pool** – IP addresses that are associated with the NIC to which load is distributed.
- **NAT rules** - Defines how inbound traffic flows through the front-end IP and distributed to the back-end IP.
- **Load balancer rules** - Maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. A single load balancer can have multiple load balancing rules. Each rule is a combination of a front-end IP and port and back-end IP and port associated with VMs.
- **Probes** - Monitors the health of VMs. When a probe fails to respond, the load balancer stops sending new connections to the unhealthy VM. The existing connections are not affected, and new connections are sent to healthy VMs.

This table lists the methods that you can use to create an internet-facing load balancer.

METHOD	DESCRIPTION
Azure portal	You can load balance internet traffic to VMs using the Azure portal .
Azure PowerShell	To provide the identifier of the public IP address that you previously created, use New-AzLoadBalancerFrontendIpConfig with the -PublicIpAddress parameter. Use New-AzLoadBalancerBackendAddressPoolConfig to create the configuration of the back-end address pool. Use New-AzLoadBalancerInboundNatRuleConfig to create inbound NAT rules associated with the front-end IP configuration that you created. Use New-AzLoadBalancerProbeConfig to create the probes that you need. Use New-AzLoadBalancerRuleConfig to create the load balancer configuration. Use New-AzLoadBalancer to create the load balancer.

METHOD	DESCRIPTION
Azure CLI	Use az network lb create to create the initial load balancer configuration. Use az network lb frontend-ip create to add the public IP address that you previously created. Use az network lb address-pool create to add the configuration of the back-end address pool. Use az network lb inbound-nat-rule create to add NAT rules. Use az network lb rule create to add the load balancer rules. Use az network lb probe create to add the probes.
Template	Use 2 VMs in a Load Balancer and configure NAT rules on the LB as a guide for deploying a load balancer using a template.

This table lists the methods that you can use to create an internal load balancer.

METHOD	DESCRIPTION
Azure portal	You can balance internal traffic load with a Basic load balancer in the Azure portal .
Azure PowerShell	To provide a private IP address in the network subnet, use New-AzLoadBalancerFrontendIpConfig with the -PrivateIpAddress parameter. Use New-AzLoadBalancerBackendAddressPoolConfig to create the configuration of the back-end address pool. Use New-AzLoadBalancerInboundNatRuleConfig to create inbound NAT rules associated with the front-end IP configuration that you created. Use New-AzLoadBalancerProbeConfig to create the probes that you need. Use New-AzLoadBalancerRuleConfig to create the load balancer configuration. Use New-AzLoadBalancer to create the load balancer.
Azure CLI	Use the az network lb create command to create the initial load balancer configuration. To define the private IP address, use az network lb frontend-ip create with the --private-ip-address parameter. Use az network lb address-pool create to add the configuration of the back-end address pool. Use az network lb inbound-nat-rule create to add NAT rules. Use az network lb rule create to add the load balancer rules. Use az network lb probe create to add the probes.
Template	Use 2 VMs in a Load Balancer and configure NAT rules on the LB as a guide for deploying a load balancer using a template.

VMs

VMs can be created in the same VNet and they can connect to each other using private IP addresses. They can connect even if they are in different subnets without the need to configure a gateway or use public IP addresses. To put VMs into a VNet, you create the VNet and then as you create each VM, you assign it to the VNet and subnet. VMs acquire their network settings during deployment or startup.

VMs are assigned an IP address when they are deployed. If you deploy multiple VMs into a VNet or subnet, they are assigned IP addresses as they boot up. You can also allocate a static IP to a VM. If you allocate a static IP, you should consider using a specific subnet to avoid accidentally reusing a static IP for another VM.

If you create a VM and later want to migrate it into a VNet, it is not a simple configuration change. You must redeploy the VM into the VNet. The easiest way to redeploy is to delete the VM, but not any disks attached to it,

and then re-create the VM using the original disks in the VNet.

This table lists the methods that you can use to create a VM in a VNet.

METHOD	DESCRIPTION
Azure portal	Uses the default network settings that were previously mentioned to create a VM with a single NIC. To create a VM with multiple NICs, you must use a different method.
Azure PowerShell	Includes the use of <code>Add-AzVMNetworkInterface</code> to add the NIC that you previously created to the VM configuration.
Azure CLI	Create and connect a VM to a Vnet, subnet, and NIC that build as individual steps.
Template	Use Very simple deployment of a Windows VM as a guide for deploying a VM using a template.

Next steps

For VM-specific steps on how to manage Azure virtual networks for VMs, see the [Windows](#) or [Linux](#) tutorials.

There are also tutorials on how to load balance VMs and create highly available applications for [Windows](#) or [Linux](#).

- Learn how to configure [user-defined routes and IP forwarding](#).
- Learn how to configure [VNet to VNet connections](#).
- Learn how to [Troubleshoot routes](#).
- Learn more about [Virtual machine network bandwidth](#).

What are virtual machine scale sets?

1/19/2020 • 4 minutes to read • [Edit Online](#)

Azure virtual machine scale sets let you create and manage a group of identical, load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update a large number of VMs. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.

Why use virtual machine scale sets?

To provide redundancy and improved performance, applications are typically distributed across multiple instances. Customers may access your application through a load balancer that distributes requests to one of the application instances. If you need to perform maintenance or update an application instance, your customers must be distributed to another available application instance. To keep up with additional customer demand, you may need to increase the number of application instances that run your application.

Azure virtual machine scale sets provide the management capabilities for applications that run across many VMs, [automatic scaling of resources](#), and load balancing of traffic. Scale sets provide the following key benefits:

- **Easy to create and manage multiple VMs**

- When you have many VMs that run your application, it's important to maintain a consistent configuration across your environment. For reliable performance of your application, the VM size, disk configuration, and application installs should match across all VMs.
- With scale sets, all VM instances are created from the same base OS image and configuration. This approach lets you easily manage hundreds of VMs without additional configuration tasks or network management.
- Scale sets support the use of the [Azure load balancer](#) for basic layer-4 traffic distribution, and [Azure Application Gateway](#) for more advanced layer-7 traffic distribution and SSL termination.

- **Provides high availability and application resiliency**

- Scale sets are used to run multiple instances of your application. If one of these VM instances has a problem, customers continue to access your application through one of the other VM instances with minimal interruption.
- For additional availability, you can use [Availability Zones](#) to automatically distribute VM instances in a scale set within a single datacenter or across multiple datacenters.

- **Allows your application to automatically scale as resource demand changes**

- Customer demand for your application may change throughout the day or week. To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases.
- Autoscale also minimizes the number of unnecessary VM instances that run your application when demand is low, while customers continue to receive an acceptable level of performance as demand grows and additional VM instances are automatically added. This ability helps reduce costs and efficiently create Azure resources as required.

- **Works at large-scale**

- Scale sets support up to 1,000 VM instances. If you create and upload your own custom VM images, the limit is 600 VM instances.

- For the best performance with production workloads, use [Azure Managed Disks](#).

Differences between virtual machines and scale sets

Scale sets are built from virtual machines. With scale sets, the management and automation layers are provided to run and scale your applications. You could instead manually create and manage individual VMs, or integrate existing tools to build a similar level of automation. The following table outlines the benefits of scale sets compared to manually managing multiple VM instances.

SCENARIO	MANUAL GROUP OF VMs	VIRTUAL MACHINE SCALE SET
Add additional VM instances	Manual process to create, configure, and ensure compliance	Automatically create from central configuration
Traffic balancing and distribution	Manual process to create and configure Azure load balancer or Application Gateway	Can automatically create and integrate with Azure load balancer or Application Gateway
High availability and redundancy	Manually create Availability Set or distribute and track VMs across Availability Zones	Automatic distribution of VM instances across Availability Zones or Availability Sets
Scaling of VMs	Manual monitoring and Azure Automation	Autoscale based on host metrics, in-guest metrics, Application Insights, or schedule

There is no additional cost to scale sets. You only pay for the underlying compute resources such as the VM instances, load balancer, or Managed Disk storage. The management and automation features, such as autoscale and redundancy, incur no additional charges over the use of VMs.

How to monitor your scale sets

Use [Azure Monitor for VMs](#), which has a simple onboarding process and will automate the collection of important CPU, memory, disk, and network performance counters from the VMs in your scale set. It also includes additional monitoring capabilities and pre-defined visualizations that help you focus on the availability and performance of your scale sets.

Enable monitoring for your [virtual machine scale set application](#) with Application Insights to collect detailed information about your application including page views, application requests, and exceptions. Further verify the availability of your application by configuring an [availability test](#) to simulate user traffic.

Next steps

To get started, create your first virtual machine scale set in the Azure portal.

[Create a scale set in the Azure portal](#)

Use infrastructure automation tools with virtual machines in Azure

11/13/2019 • 6 minutes to read • [Edit Online](#)

To create and manage Azure virtual machines (VMs) in a consistent manner at scale, some form of automation is typically desired. There are many tools and solutions that allow you to automate the complete Azure infrastructure deployment and management lifecycle. This article introduces some of the infrastructure automation tools that you can use in Azure. These tools commonly fit in to one of the following approaches:

- Automate the configuration of VMs
 - Tools include [Ansible](#), [Chef](#), and [Puppet](#).
 - Tools specific to VM customization include [cloud-init](#) for Linux VMs, [PowerShell Desired State Configuration \(DSC\)](#), and the [Azure Custom Script Extension](#) for all Azure VMs.
- Automate infrastructure management
 - Tools include [Packer](#) to automate custom VM image builds, and [Terraform](#) to automate the infrastructure build process.
 - [Azure Automation](#) can perform actions across your Azure and on-premises infrastructure.
- Automate application deployment and delivery
 - Examples include [Azure DevOps Services](#) and [Jenkins](#).

Ansible

[Ansible](#) is an automation engine for configuration management, VM creation, or application deployment. Ansible uses an agent-less model, typically with SSH keys, to authenticate and manage target machines. Configuration tasks are defined in playbooks, with a number of Ansible modules available to carry out specific tasks. For more information, see [How Ansible works](#).

Learn how to:

- [Install and configure Ansible on Linux for use with Azure](#).
- [Create a Linux virtual machine](#).
- [Manage a Linux virtual machine](#).

Chef

[Chef](#) is an automation platform that helps define how your infrastructure is configured, deployed, and managed. Additional components included Chef Habitat for application lifecycle automation rather than the infrastructure, and Chef InSpec that helps automate compliance with security and policy requirements. Chef Clients are installed on target machines, with one or more central Chef Servers that store and manage the configurations. For more information, see [An Overview of Chef](#).

Learn how to:

- [Deploy Chef Automate from the Azure Marketplace](#).
- [Install Chef on Windows and create Azure VMs](#).

Puppet

Puppet is an enterprise-ready automation platform that handles the application delivery and deployment process. Agents are installed on target machines to allow Puppet Master to run manifests that define the desired configuration of the Azure infrastructure and VMs. Puppet can integrate with other solutions such as Jenkins and GitHub for an improved devops workflow. For more information, see [How Puppet works](#).

Learn how to:

- [Deploy Puppet from the Azure Marketplace](#).

Cloud-init

[Cloud-init](#) is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. Because cloud-init is called during the initial boot process, there are no additional steps or required agents to apply your configuration. For more information on how to properly format your `#cloud-config` files, see the [cloud-init documentation site](#). `#cloud-config` files are text files encoded in base64.

Cloud-init also works across distributions. For example, you don't use **apt-get install** or **yum install** to install a package. Instead you can define a list of packages to install. Cloud-init automatically uses the native package management tool for the distro you select.

We are actively working with our endorsed Linux distro partners in order to have cloud-init enabled images available in the Azure marketplace. These images make your cloud-init deployments and configurations work seamlessly with VMs and virtual machine scale sets. Learn more details about cloud-init on Azure:

- [Cloud-init support for Linux virtual machines in Azure](#)
- [Try a tutorial on automated VM configuration using cloud-init](#).

PowerShell DSC

[PowerShell Desired State Configuration \(DSC\)](#) is a management platform to define the configuration of target machines. DSC can also be used on Linux through the [Open Management Infrastructure \(OMI\) server](#).

DSC configurations define what to install on a machine and how to configure the host. A Local Configuration Manager (LCM) engine runs on each target node that processes requested actions based on pushed configurations. A pull server is a web service that runs on a central host to store the DSC configurations and associated resources. The pull server communicates with the LCM engine on each target host to provide the required configurations and report on compliance.

Learn how to:

- [Create a basic DSC configuration](#).
- [Configure a DSC pull server](#).
- [Use DSC for Linux](#).

Azure Custom Script Extension

The Azure Custom Script Extension for [Linux](#) or [Windows](#) downloads and executes scripts on Azure VMs. You can use the extension when you create a VM, or any time after the VM is in use.

Scripts can be downloaded from Azure storage or any public location such as a GitHub repository. With the Custom Script Extension, you can write scripts in any language that runs on the source VM. These scripts can be used to install applications or configure the VM as desired. To secure credentials, sensitive information such as passwords can be stored in a protected configuration. These credentials are only decrypted inside the VM.

Learn how to:

- [Create a Linux VM with the Azure CLI and use the Custom Script Extension.](#)
- [Create a Windows VM with Azure PowerShell and use the Custom Script Extension.](#)

Packer

[Packer](#) automates the build process when you create a custom VM image in Azure. You use Packer to define the OS and run post-configuration scripts that customize the VM for your specific needs. Once configured, the VM is then captured as a Managed Disk image. Packer automates the process to create the source VM, network and storage resources, run configuration scripts, and then create the VM image.

Learn how to:

- [Use Packer to create a Linux VM image in Azure.](#)
- [Use Packer to create a Windows VM image in Azure.](#)

Terraform

[Terraform](#) is an automation tool that allows you to define and create an entire Azure infrastructure with a single template format language - the HashiCorp Configuration Language (HCL). With Terraform, you define templates that automate the process to create network, storage, and VM resources for a given application solution. You can use your existing Terraform templates for other platforms with Azure to ensure consistency and simplify the infrastructure deployment without needing to convert to an Azure Resource Manager template.

Learn how to:

- [Install and configure Terraform with Azure.](#)
- [Create an Azure infrastructure with Terraform.](#)

Azure Automation

[Azure Automation](#) uses runbooks to process a set of tasks on the VMs you target. Azure Automation is used to manage existing VMs rather than to create an infrastructure. Azure Automation can run across both Linux and Windows VMs, as well as on-premises virtual or physical machines with a hybrid runbook worker. Runbooks can be stored in a source control repository, such as GitHub. These runbooks can then run manually or on a defined schedule.

Azure Automation also provides a Desired State Configuration (DSC) service that allows you to create definitions for how a given set of VMs should be configured. DSC then ensures that the required configuration is applied and the VM stays consistent. Azure Automation DSC runs on both Windows and Linux machines.

Learn how to:

- [Create a PowerShell runbook.](#)
- [Use Hybrid Runbook Worker to manage on-premises resources.](#)
- [Use Azure Automation DSC.](#)

Azure DevOps Services

[Azure DevOps Services](#) is a suite of tools that help you share and track code, use automated builds, and create a complete continuous integration and development (CI/CD) pipeline. Azure DevOps Services integrates with Visual Studio and other editors to simplify usage. Azure DevOps Services can also create and configure Azure VMs and then deploy code to them.

Learn more about:

- [Azure DevOps Services.](#)

Jenkins

Jenkins is a continuous integration server that helps deploy and test applications, and create automated pipelines for code delivery. There are hundreds of plugins to extend the core Jenkins platform, and you can also integrate with many other products and solutions through webhooks. You can manually install Jenkins on an Azure VM, run Jenkins from within a Docker container, or use a pre-built Azure Marketplace image.

Learn how to:

- [Create a development infrastructure on a Linux VM in Azure with Jenkins, GitHub, and Docker.](#)

Next steps

There are many different options to use infrastructure automation tools in Azure. You have the freedom to use the solution that best fits your needs and environment. To get started and try some of the tools built-in to Azure, see how to automate the customization of a [Linux](#) or [Windows](#) VM.

Secure and use policies on virtual machines in Azure

11/13/2019 • 4 minutes to read • [Edit Online](#)

It's important to keep your virtual machine (VM) secure for the applications that you run. Securing your VMs can include one or more Azure services and features that cover secure access to your VMs and secure storage of your data. This article provides information that enables you to keep your VM and applications secure.

Antimalware

The modern threat landscape for cloud environments is dynamic, increasing the pressure to maintain effective protection in order to meet compliance and security requirements. [Microsoft Antimalware for Azure](#) is a free real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Alerts can be configured to notify you when known malicious or unwanted software attempts to install itself or run on your VM. It is not supported on VMs running Linux or Windows Server 2008.

Azure Security Center

[Azure Security Center](#) helps you prevent, detect, and respond to threats to your VMs. Security Center provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Security Center's just-in-time access can be applied across your VM deployment to lock down inbound traffic to your Azure VMs, reducing exposure to attacks while providing easy access to connect to VMs when needed. When just-in-time is enabled and a user requests access to a VM, Security Center checks what permissions the user has for the VM. If they have the correct permissions, the request is approved and Security Center automatically configures the Network Security Groups (NSGs) to allow inbound traffic to the selected ports for a limited amount of time. After the time has expired, Security Center restores the NSGs to their previous states.

Encryption

For enhanced [Windows VM](#) and [Linux VM](#) security and compliance, virtual disks in Azure can be encrypted. Virtual disks on Windows VMs are encrypted at rest using BitLocker. Virtual disks on Linux VMs are encrypted at rest using dm-crypt.

There is no charge for encrypting virtual disks in Azure. Cryptographic keys are stored in Azure Key Vault using software-protection, or you can import or generate your keys in Hardware Security Modules (HSMs) certified to FIPS 140-2 level 2 standards. These cryptographic keys are used to encrypt and decrypt virtual disks attached to your VM. You retain control of these cryptographic keys and can audit their use. An Azure Active Directory service principal provides a secure mechanism for issuing these cryptographic keys as VMs are powered on and off.

Key Vault and SSH Keys

Secrets and certificates can be modeled as resources and provided by [Key Vault](#). You can use Azure PowerShell to create key vaults for [Windows VMs](#) and the Azure CLI for [Linux VMs](#). You can also create keys for encryption.

Key vault access policies grant permissions to keys, secrets, and certificates separately. For example, you can give a user access to only keys, but no permissions for secrets. However, permissions to access keys or secrets or certificates are at the vault level. In other words, [key vault access policy](#) does not support object level permissions.

When you connect to VMs, you should use public-key cryptography to provide a more secure way to sign in to them. This process involves a public and private key exchange using the secure shell (SSH) command to

authenticate yourself rather than a username and password. Passwords are vulnerable to brute-force attacks, especially on Internet-facing VMs such as web servers. With a secure shell (SSH) key pair, you can create a [Linux VM](#) that uses SSH keys for authentication, eliminating the need for passwords to sign-in. You can also use SSH keys to connect from a [Windows VM](#) to a Linux VM.

Managed identities for Azure resources

A common challenge when building cloud applications is how to manage the credentials in your code for authenticating to cloud services. Keeping the credentials secure is an important task. Ideally, the credentials never appear on developer workstations and aren't checked into source control. Azure Key Vault provides a way to securely store credentials, secrets, and other keys, but your code has to authenticate to Key Vault to retrieve them.

The managed identities for Azure resources feature in Azure Active Directory (Azure AD) solves this problem. The feature provides Azure services with an automatically managed identity in Azure AD. You can use the identity to authenticate to any service that supports Azure AD authentication, including Key Vault, without any credentials in your code. Your code that's running on a VM can request a token from two endpoints that are accessible only from within the VM. For more detailed information about this service, review the [managed identities for Azure resources](#) overview page.

Policies

[Azure policies](#) can be used to define the desired behavior for your organization's [Windows VMs](#) and [Linux VMs](#). By using policies, an organization can enforce various conventions and rules throughout the enterprise. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization.

Role-based access control

Using [role-based access control \(RBAC\)](#), you can segregate duties within your team and grant only the amount of access to users on your VM that they need to perform their jobs. Instead of giving everybody unrestricted permissions on the VM, you can allow only certain actions. You can configure access control for the VM in the [Azure portal](#), using the [Azure CLI](#), or [Azure PowerShell](#).

Next steps

- Walk through the steps to monitor virtual machine security by using Azure Security Center for [Linux](#) or [Windows](#).

Azure Disk Encryption for Windows VMs

10/16/2019 • 4 minutes to read • [Edit Online](#)

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the [BitLocker](#) feature of Windows to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with [Azure Key Vault](#) to help you control and manage the disk encryption keys and secrets.

If you use [Azure Security Center](#), you're alerted if you have VMs that aren't encrypted. The alerts show as High Severity and the recommendation is to encrypt these VMs.

VIRTUAL MACHINES RECOMMENDATIONS		TOTAL					
Missing disk encryption		2 of 2 VMs					
Virtual machines							
NAME	ONBOARDING	SYSTEM UPDATES	ANTIMALWARE	BASELINE	DISK ENCRYPTION		
ASC-VM1	✓	✓	✓	✓	✓	!	
ASC-VM2	✓	✓	✓	✓	✓	!	

WARNING

- If you have previously used Azure Disk Encryption with Azure AD to encrypt a VM, you must continue use this option to encrypt your VM. See [Azure Disk Encryption with Azure AD \(previous release\)](#) for details.
- Certain recommendations might increase data, network, or compute resource usage, resulting in additional license or subscription costs. You must have a valid active Azure subscription to create resources in Azure in the supported regions.

You can learn the fundamentals of Azure Disk Encryption for Windows in just a few minutes with the [Create and encrypt a Windows VM with Azure CLI quickstart](#) or the [Create and encrypt a Windows VM with Azure Powershell quickstart](#).

Supported VMs and operating systems

Supported VM sizes

Windows VMs are available in a [range of sizes](#). Azure Disk Encryption is not available on [Basic, A-series VMs](#), or on virtual machines with a less than 2 GB of memory.

Azure Disk Encryption is also available for VMs with premium storage.

Supported operating systems

- Windows client: Windows 8 and later.
- Windows Server: Windows Server 2008 R2 and later.

NOTE

Windows Server 2008 R2 requires the .NET Framework 4.5 to be installed for encryption; install it from Windows Update with the optional update Microsoft .NET Framework 4.5.2 for Windows Server 2008 R2 x64-based systems ([KB2901983](#)).

Windows Server 2012 R2 Core and Windows Server 2016 Core requires the bdehdcfg component to be installed on the VM for encryption.

Networking requirements

To enable Azure Disk Encryption, the VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the Windows VM must be able to connect to an Azure Active Directory endpoint, [login.microsoftonline.com].
- To write the encryption keys to your key vault, the Windows VM must be able to connect to the key vault endpoint.
- The Windows VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see [Azure Key Vault behind a firewall](#).

Group Policy requirements

Azure Disk Encryption uses the BitLocker external key protector for Windows VMs. For domain joined VMs, don't push any group policies that enforce TPM protectors. For information about the group policy for "Allow BitLocker without a compatible TPM," see [BitLocker Group Policy Reference](#).

BitLocker policy on domain joined virtual machines with custom group policy must include the following setting: [Configure user storage of BitLocker recovery information -> Allow 256-bit recovery key](#). Azure Disk Encryption will fail when custom group policy settings for BitLocker are incompatible. On machines that didn't have the correct policy setting, apply the new policy, force the new policy to update (gpupdate.exe /force), and then restarting may be required.

Azure Disk Encryption will fail if domain level group policy blocks the AES-CBC algorithm, which is used by BitLocker.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see [Creating and configuring a key vault for Azure Disk Encryption](#).

Terminology

The following table defines some of the common terms used in Azure disk encryption documentation:

TERMINOLOGY	DEFINITION
-------------	------------

TERMINOLOGY	DEFINITION
Azure Key Vault	Key Vault is a cryptographic, key management service that's based on Federal Information Processing Standards (FIPS) validated hardware security modules. These standards help to safeguard your cryptographic keys and sensitive secrets. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
Azure CLI	The Azure CLI is optimized for managing and administering Azure resources from the command line.
BitLocker	BitLocker is an industry-recognized Windows volume encryption technology that's used to enable disk encryption on Windows VMs.
Key encryption key (KEK)	The asymmetric key (RSA 2048) that you can use to protect or wrap the secret. You can provide a hardware security module (HSM)-protected key or software-protected key. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets .

Next steps

- [Quickstart - Create and encrypt a Windows VM with Azure CLI](#)
- [Quickstart - Create and encrypt a Windows VM with Azure Powershell](#)
- [Azure Disk Encryption scenarios on Windows VMs](#)
- [Azure Disk Encryption prerequisites CLI script](#)
- [Azure Disk Encryption prerequisites PowerShell script](#)
- [Creating and configuring a key vault for Azure Disk Encryption](#)

Security controls for Windows Virtual Machines

2/12/2020 • 2 minutes to read • [Edit Online](#)

This article documents the security controls built into Windows Virtual Machines.

A security control is a quality or feature of an Azure service that contributes to the service's ability to prevent, detect, and respond to security vulnerabilities.

For each control, we use "Yes" or "No" to indicate whether it is currently in place for the service, "N/A" for a control that is not applicable to the service. We might also provide a note or links to more information about an attribute.

Network

SECURITY CONTROL	YES/NO	NOTES
Service endpoint support	Yes	
VNet injection support	Yes	
Network Isolation and Firewalling support	Yes	
Forced tunneling support	Yes	See Configure forced tunneling using the Azure Resource Manager deployment model .

Monitoring & logging

SECURITY CONTROL	YES/NO	NOTES
Azure monitoring support (Log analytics, App insights, etc.)	Yes	Monitor and update a Windows virtual machine in Azure .
Control and management plane logging and audit	Yes	
Data plane logging and audit	No	

Identity

SECURITY CONTROL	YES/NO	NOTES
Authentication	Yes	
Authorization	Yes	

Data protection

SECURITY CONTROL	YES/NO	NOTES
Server-side encryption at rest: Microsoft-managed keys	Yes	See Encrypt virtual disks on a Windows VM .
Encryption in transit (such as ExpressRoute encryption, in VNet encryption, and VNet-VNet encryption)	Yes	Azure Virtual Machines supports ExpressRoute and VNet encryption. See In-transit encryption in VMs .
Server-side encryption at rest: customer-managed keys (BYOK)	Yes	Customer-managed keys is a supported Azure encryption scenario; see Azure encryption overview .
Column level encryption (Azure Data Services)	N/A	
API calls encrypted	Yes	Via HTTPS and TLS.

Configuration management

SECURITY CONTROL	YES/NO	NOTES
Configuration management support (versioning of configuration, etc.)	Yes	

Next steps

- Learn more about the [built-in security controls across Azure services](#).

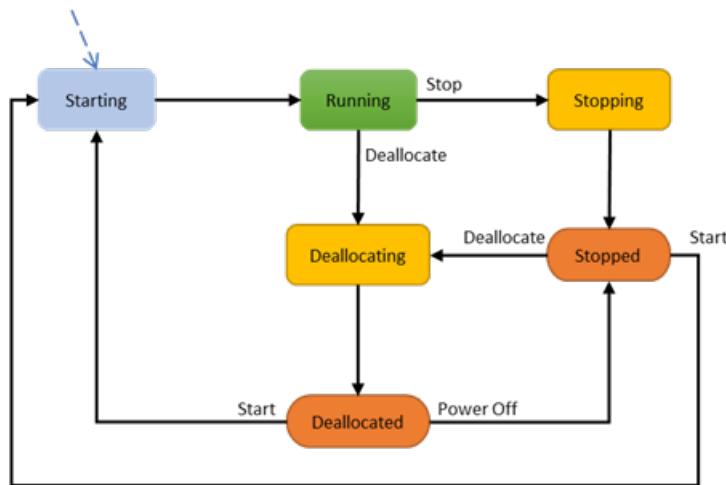
Virtual machines lifecycle and states

11/13/2019 • 3 minutes to read • [Edit Online](#)

Azure Virtual Machines (VMs) go through different states that can be categorized into *provisioning* and *power* states. The purpose of this article is to describe these states and specifically highlight when customers are billed for instance usage.

Power states

The power state represents the last known state of the VM.



The following table provides a description of each instance state and indicates whether it is billed for instance usage or not.

STATE	DESCRIPTION	INSTANCE USAGE BILLING
Starting	VM is starting up. <pre>"statuses": [{ "code": "PowerState/starting", "level": "Info", "displayStatus": "VM starting" }]</pre>	Not billed
Running	Normal working state for a VM <pre>"statuses": [{ "code": "PowerState/running", "level": "Info", "displayStatus": "VM running" }]</pre>	Billed

Stopping	This is a transitional state. When completed, it will show as **Stopped**. <pre>"statuses": [{ "code": "PowerState/stopping", "level": "Info", "displayStatus": "VM stopping" }]</pre>	Billed
Stopped	The VM has been shut down from within the guest OS or using the PowerOff APIs. Hardware is still allocated to the VM and it remains on the host. <pre>"statuses": [{ "code": "PowerState/stopped", "level": "Info", "displayStatus": "VM stopped" }]</pre>	Billed*
Deallocating	Transitional state. When completed, the VM will show as **Deallocated**. <pre>"statuses": [{ "code": "PowerState/deallocating", "level": "Info", "displayStatus": "VM deallocating" }]</pre>	Not billed*
Deallocated	The VM has been stopped successfully and removed from the host. <pre>"statuses": [{ "code": "PowerState/deallocated", "level": "Info", "displayStatus": "VM deallocated" }]</pre>	Not billed

*Some Azure resources, such as Disks and Networking, incur charges. Software licenses on the instance do not incur charges.

Provisioning states

A provisioning state is the status of a user-initiated, control-plane operation on the VM. These states are separate from the power state of a VM.

- **Create** – VM creation.
- **Update** – updates the model for an existing VM. Some non-model changes to VM such as Start/Restart also fall under update.
- **Delete** – VM deletion.

- **Deallocate** – is where a VM is stopped and removed from the host. Deallocating a VM is considered an update, so it will display provisioning states related to updating.

Here are the transitional operation states after the platform has accepted a user-initiated action:

States	Description
Creating	<pre>"statuses": [{ "code": "ProvisioningState/creating", "level": "Info", "displayStatus": "Creating" }]</pre>
Updating	<pre>"statuses": [{ "code": "ProvisioningState/updating", "level": "Info", "displayStatus": "Updating" }]</pre>
Deleting	<pre>"statuses": [{ "code": "ProvisioningState/deleting", "level": "Info", "displayStatus": "Deleting" }]</pre>
OS provisioning states	<p>If a VM is created with an OS image and not with a specialized image, then following substates can be observed:</p> <ol style="list-style-type: none"> 1. OSProvisioningInProgress – The VM is running, and installation of guest OS is in progress. <pre>"statuses": [{ "code": "ProvisioningState/creating/OSProvisioningInProgress", "level": "Info", "displayStatus": "OS Provisioning In progress" }]</pre> <ol style="list-style-type: none"> 2. OSProvisioningComplete – Short-lived state. The VM quickly transitions to **Success** unless any extensions need to be installed. Installing extensions can take time. <pre>"statuses": [{ "code": "ProvisioningState/creating/OSProvisioningComplete", "level": "Info", "displayStatus": "OS Provisioning Complete" }]</pre> <p>Note: OS Provisioning can transition to **Failed** if there is an OS failure or the OS doesn't install in time. Customers will be billed for the deployed VM on the infrastructure.</p>

Once the operation is complete, the VM will transition into one of the following states:

- **Succeeded** – the user-initiated actions have completed.

```
"statuses": [
{
  "code": "ProvisioningState/succeeded",
  "level": "Info",
  "displayStatus": "Provisioning succeeded",
  "time": "time"
}
]
```

- **Failed** – represents a failed operation. Refer to the error codes to get more information and possible solutions.

```
"statuses": [
{
  "code": "ProvisioningState/failed/InternalOperationError",
  "level": "Error",
  "displayStatus": "Provisioning failed",
  "message": "Operation abandoned due to internal error. Please try again later.",
  "time": "time"
}
]
```

VM instance view

The instance view API provides VM running-state information. For more information, see the [Virtual Machines - Instance View](#) API documentation.

Azure Resources explorer provides a simple UI for viewing the VM running state: [Resource Explorer](#).

Provisioning states are visible on VM properties and instance view. Power states are available in instance view of VM.

Next steps

To learn more about monitoring your VM, see [How to monitor virtual machines in Azure](#).

How to monitor virtual machines in Azure

11/13/2019 • 5 minutes to read • [Edit Online](#)

With the significant growth of VMs hosted in Azure, it's important to identify performance and health issues that impact applications and infrastructure services they support. Basic monitoring is delivered by default with Azure by the metric types CPU usage, disk utilization, memory utilization, and network traffic collected by the host hypervisor. Additional metric and log data can be collected using [extensions](#) to configure diagnostics on your VMs from the guest operating system.

To detect and help diagnose performance and health issues with the guest operating system, .NET based or Java web application components running inside the VM, Azure Monitor delivers centralized monitoring with comprehensive features such as Azure Monitor for VMs and Application Insights.

Diagnostics and metrics

You can set up and monitor the collection of [diagnostics data](#) using [metrics](#) in the Azure portal, the Azure CLI, Azure PowerShell, and programming Applications Programming Interfaces (APIs). For example, you can:

- **Observe basic metrics for the VM.** On the Overview screen of the Azure portal, the basic metrics shown include CPU usage, network usage, total of disk bytes, and disk operations per second.
- **Enable the collection of boot diagnostics and view it using the Azure portal.** When bringing your own image to Azure or even booting one of the platform images, there can be many reasons why a VM gets into a non-bootable state. You can easily enable boot diagnostics when you create a VM by clicking **Enabled** for Boot Diagnostics under the Monitoring section of the Settings screen.

As VMs boot, the boot diagnostic agent captures boot output and stores it in Azure storage. This data can be used to troubleshoot VM boot issues. Boot diagnostics are not automatically enabled when you create a VM from command-line tools. Before enabling boot diagnostics, a storage account needs to be created for storing boot logs. If you enable boot diagnostics in the Azure portal, a storage account is automatically created for you.

If you didn't enable boot diagnostics when the VM was created, you can always enable it later by using [Azure CLI](#), [Azure PowerShell](#), or an [Azure Resource Manager template](#).

- **Enable the collection of guest OS diagnostics data.** When you create a VM, you have the opportunity on the settings screen to enable guest OS diagnostics. When you do enable the collection of diagnostics data, the [IaaS Diagnostics extension for Linux](#) or the [IaaS Diagnostics extension for Windows](#) is added to the VM, which enables you to collect additional disk, CPU, and memory data.

Using the collected diagnostics data, you can configure autoscaling for your VMs. You can also configure [Azure Monitor Logs](#) to store the data and set up alerts to let you know when performance isn't right.

Alerts

You can create [alerts](#) based on specific performance metrics. Examples of the issues you can be alerted about include when average CPU usage exceeds a certain threshold, or available free disk space drops below a certain amount. Alerts can be configured in the [Azure portal](#), using [Azure Resource Manager templates](#), or [Azure CLI](#).

Azure Service Health

[Azure Service Health](#) provides personalized guidance and support when issues in Azure services affect you, and

helps you prepare for upcoming planned maintenance. Azure Service Health alerts you and your teams using targeted and flexible notifications.

Azure Resource Health

[Azure Resource health](#) helps you diagnose and get support when an Azure issue impacts your resources. It informs you about the current and past health of your resources and helps you mitigate issues. Resource health provides technical support when you need help with Azure service issues.

Azure Activity Log

The [Azure Activity Log](#) is a subscription log that provides insight into subscription-level events that have occurred in Azure. The log includes a range of data, from Azure Resource Manager operational data to updates on Service Health events. You can click Activity Log in the Azure portal to view the log for your VM.

Some of the things you can do with the activity log include:

- Create an [alert on an Activity Log event](#).
- [Stream it to an Event Hub](#) for ingestion by a third-party service or custom analytics solution such as Power BI.
- Analyze it in Power BI using the [Power BI content pack](#).
- [Save it to a storage account](#) for archival or manual inspection. You can specify the retention time (in days) using the Log Profile.

You can also access activity log data by using [Azure PowerShell](#), the [Azure CLI](#), or [Monitor REST APIs](#).

[Azure Resource Logs](#) are logs emitted by your VM that provide rich, frequent data about its operation. Resource logs differ from the activity log by providing insight about operations that were performed within the VM.

Some of the things you can do with diagnostics logs include:

- [Save them to a storage account](#) for auditing or manual inspection. You can specify the retention time (in days) using Resource Diagnostic Settings.
- [Stream them to Event Hubs](#) for ingestion by a third-party service or custom analytics solution such as Power BI.
- Analyze them with [Log Analytics](#).

Advanced monitoring

For visibility of the application or service supported by the Azure VM and virtual machine scale sets, identification of issues with the guest OS or workload running in the VM to understand if it is impacting availability or performance of the application, or is an issue with the application, enable both [Azure Monitor for VMs](#) and [Application Insights](#).

Azure Monitor for VMs monitors your Azure virtual machines (VM) at scale by analyzing the performance and health of your Windows and Linux VMs, including the different processes and interconnected dependencies on other resources and external processes it discovers. It includes several trend performance charts to help during investigation of problems and assess capacity of your VMs. The dependency map shows monitored and unmonitored machines, failed and active network connections between processes and these machines, and shows trend charts with standard network connection metrics. Combined with Application Insights, you monitor your application and capture telemetry such as HTTP requests, exceptions, etc. so you can correlate issues between the VMs and your application. Configure [Azure Monitor alerts](#) to alert you on important conditions detected from monitoring data collected by Azure Monitor for VMs.

Next steps

- Walk through the steps in [Monitor a Windows Virtual Machine with Azure PowerShell](#) or [Monitor a Linux](#)

[Virtual Machine with the Azure CLI.](#)

- Learn more about the best practices around [Monitoring and diagnostics](#).

Backup and restore options for virtual machines in Azure

11/13/2019 • 2 minutes to read • [Edit Online](#)

You can protect your data by taking backups at regular intervals. There are several backup options available for VMs, depending on your use-case.

Azure Backup

For backing up Azure VMs running production workloads, use Azure Backup. Azure Backup supports application-consistent backups for both Windows and Linux VMs. Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore the whole VM or just specific files.

For a simple, hands-on introduction to Azure Backup for Azure VMs, see the "Back up Azure virtual machines" tutorial for [Linux](#) or [Windows](#).

For more information on how Azure Backup works, see [Plan your VM backup infrastructure in Azure](#)

Azure Site Recovery

Azure Site Recovery protects your VMs from a major disaster scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. You can configure Azure Site Recovery for your VMs so that you can recover your application with a single click in matter of minutes. You can replicate to an Azure region of your choice, it is not restricted to paired regions.

You can run disaster-recovery drills with on-demand test failovers, without affecting your production workloads or ongoing replication. Create recovery plans to orchestrate failover and failback of the entire application running on multiple VMs. The recovery plan feature is integrated with Azure automation runbooks.

You can get started by [replicating your virtual machines](#).

Managed snapshots

In development and test environments, snapshots provide a quick and simple option for backing up VMs that use Managed Disks. A managed snapshot is a read-only full copy of a managed disk. Snapshots exist independent of the source disk and can be used to create new managed disks for rebuilding a VM. They are billed based on the used portion of the disk. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GB and actual used data size of 10 GB, snapshot will be billed only for the used data size of 10 GB.

For more information on creating snapshots, see:

- [Create copy of VHD stored as a Managed Disk using Snapshots in Windows](#)
- [Create copy of VHD stored as a Managed Disk using Snapshots in Linux](#)

Next steps

You can try out Azure Backup by following the "Back up Windows virtual machines tutorial" for [Linux](#) or [Windows](#).

Example Azure infrastructure walkthrough for Windows VMs

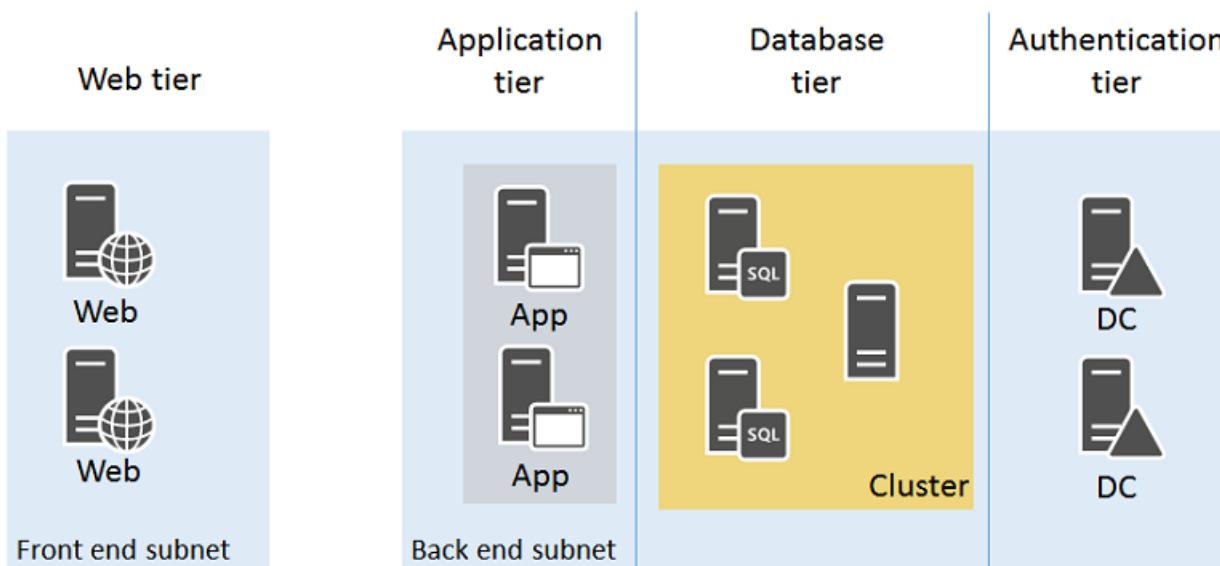
11/13/2019 • 3 minutes to read • [Edit Online](#)

This article walks through building out an example application infrastructure. We detail designing an infrastructure for a simple online store that brings together all the guidelines and decisions around naming conventions, availability sets, virtual networks and load balancers, and actually deploying your virtual machines (VMs).

Example workload

Adventure Works Cycles wants to build an online store application in Azure that consists of:

- Two IIS servers running the client front-end in a web tier
- Two IIS servers processing data and orders in an application tier
- Two Microsoft SQL Server instances with AlwaysOn availability groups (two SQL Servers and a majority node witness) for storing product data and orders in a database tier
- Two Active Directory domain controllers for customer accounts and suppliers in an authentication tier
- All the servers are located in two subnets:
 - a front-end subnet for the web servers
 - a back-end subnet for the application servers, SQL cluster, and domain controllers



Incoming secure web traffic must be load-balanced among the web servers as customers browse the online store. Order processing traffic in the form of HTTP requests from the web servers must be balanced among the application servers. Additionally, the infrastructure must be designed for high availability.

The resulting design must incorporate:

- An Azure subscription and account
- A single resource group
- Azure Managed Disks
- A virtual network with two subnets
- Availability sets for the VMs with a similar role
- Virtual machines

All the above follow these naming conventions:

- Adventure Works Cycles uses **[IT workload]-[location]-[Azure resource]** as a prefix
 - For this example, "azos" (Azure Online Store) is the IT workload name and "use" (East US 2) is the location
- Virtual networks use AZOS-USE-VN**[number]**
- Availability sets use azos-use-as-[**role**]
- Virtual machine names use azos-use-vm-[**vmname**]

Azure subscriptions and accounts

Adventure Works Cycles is using their Enterprise subscription, named Adventure Works Enterprise Subscription, to provide billing for this IT workload.

Storage

Adventure Works Cycles determined that they should use Azure Managed Disks. When creating VMs, both available storage tiers are used:

- **Standard storage** for the web servers, application servers, and domain controllers and their data disks.
- **Premium storage** for the SQL Server VMs and their data disks.

Virtual network and subnets

Because the virtual network does not need ongoing connectivity to the Adventure Work Cycles on-premises network, they decided on a cloud-only virtual network.

They created a cloud-only virtual network with the following settings using the Azure portal:

- Name: AZOS-USE-VN01
- Location: East US 2
- Virtual network address space: 10.0.0.0/8
- First subnet:
 - Name: FrontEnd
 - Address space: 10.0.1.0/24
- Second subnet:
 - Name: BackEnd
 - Address space: 10.0.2.0/24

Availability sets

To maintain high availability of all four tiers of their online store, Adventure Works Cycles decided on four availability sets:

- **azos-use-as-web** for the web servers
- **azos-use-as-app** for the application servers
- **azos-use-as-sql** for the SQL Servers
- **azos-use-as-dc** for the domain controllers

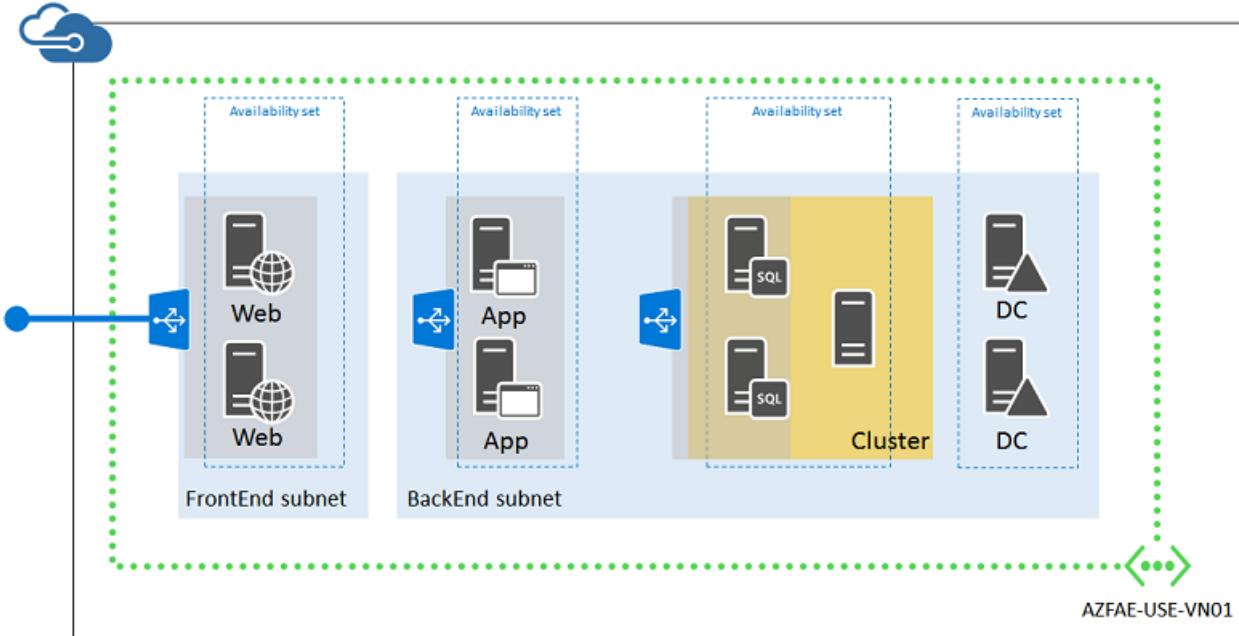
Virtual machines

Adventure Works Cycles decided on the following names for their Azure VMs:

- **azos-use-vm-web01** for the first web server

- **azos-use-vm-web02** for the second web server
- **azos-use-vm-app01** for the first application server
- **azos-use-vm-app02** for the second application server
- **azos-use-vm-sql01** for the first SQL Server server in the cluster
- **azos-use-vm-sql02** for the second SQL Server server in the cluster
- **azos-use-vm-dc01** for the first domain controller
- **azos-use-vm-dc02** for the second domain controller

Here is the resulting configuration.



This configuration incorporates:

- A cloud-only virtual network with two subnets (FrontEnd and BackEnd)
- Azure Managed Disks with both Standard and Premium disks
- Four availability sets, one for each tier of the online store
- The virtual machines for the four tiers
- An external load balanced set for HTTPS-based web traffic from the Internet to the web servers
- An internal load balanced set for unencrypted web traffic from the web servers to the application servers
- A single resource group

Deploy VMs to dedicated hosts using the Azure PowerShell

2/12/2020 • 3 minutes to read • [Edit Online](#)

This article guides you through how to create an Azure [dedicated host](#) to host your virtual machines (VMs).

Make sure that you have installed Azure PowerShell version 2.8.0 or later, and you are signed in to an Azure account in with `Connect-AzAccount`.

Limitations

- Virtual machine scale sets are not currently supported on dedicated hosts.
- The following VM series are supported: DSv3, ESv3, and Fsv2.

Create a host group

A **host group** is a resource that represents a collection of dedicated hosts. You create a host group in a region and an availability zone, and add hosts to it. When planning for high availability, there are additional options. You can use one or both of the following options with your dedicated hosts:

- Span across multiple availability zones. In this case, you are required to have a host group in each of the zones you wish to use.
- Span across multiple fault domains which are mapped to physical racks.

In either case, you are need to provide the fault domain count for your host group. If you do not want to span fault domains in your group, use a fault domain count of 1.

You can also decide to use both availability zones and fault domains. This example creates a host group in zone 1, with 2 fault domains.

```
$rgName = "myDHResourceGroup"
$location = "East US"

New-AzResourceGroup -Location $location -Name $rgName
$hostGroup = New-AzHostGroup ` 
    -Location $location ` 
    -Name myHostGroup ` 
    -PlatformFaultDomain 2 ` 
    -ResourceGroupName $rgName ` 
    -Zone 1
```

Create a host

Now let's create a dedicated host in the host group. In addition to a name for the host, you are required to provide the SKU for the host. Host SKU captures the supported VM series as well as the hardware generation for your dedicated host.

For more information about the host SKUs and pricing, see [Azure Dedicated Host pricing](#).

If you set a fault domain count for your host group, you will be asked to specify the fault domain for your host. In this example, we set the fault domain for the host to 1.

```
$dHost = New-AzHost `  
-HostGroupName $hostGroup.Name `  
-Location $location -Name myHost `  
-ResourceGroupName $rgName `  
-Sku DSv3-Type1 `  
-AutoReplaceOnFailure 1 `  
-PlatformFaultDomain 1
```

Create a VM

Create a virtual machine on the dedicated host.

If you specified an availability zone when creating your host group, you are required to use the same zone when creating the virtual machine. For this example, because our host group is in zone 1, we need to create the VM in zone 1.

```
$cred = Get-Credential  
New-AzVM `  
-Credential $cred `  
-ResourceGroupName $rgName `  
-Location $location `  
-Name myVM `  
-HostId $dhost.Id `  
-Image Win2016Datacenter `  
-Zone 1 `  
-Size Standard_D4s_v3
```

WARNING

If you create a virtual machine on a host which does not have enough resources, the virtual machine will be created in a FAILED state.

Check the status of the host

You can check the host health status and how many virtual machines you can still deploy to the host using [GetAzHost](#) with the `-InstanceView` parameter.

```
Get-AzHost `  
-ResourceGroupName $rgName `  
-Name myHost `  
-HostGroupName $hostGroup.Name `  
-InstanceView
```

The output will look similar to this:

```

ResourceGroupName      : myDHRG
PlatformFaultDomain   : 1
AutoReplaceOnFailure  : True
HostId                : 12345678-1234-1234-abcd-abc123456789
ProvisioningTime       : 7/28/2019 5:31:01 PM
ProvisioningState      : Succeeded
InstanceView           :
  AssetId              : abc45678-abcd-1234-abcd-123456789abc
  AvailableCapacity     :
    AllocatableVMs[0]   :
      VmSize             : Standard_D2s_v3
      Count               : 32
    AllocatableVMs[1]   :
      VmSize             : Standard_D4s_v3
      Count               : 16
    AllocatableVMs[2]   :
      VmSize             : Standard_D8s_v3
      Count               : 8
    AllocatableVMs[3]   :
      VmSize             : Standard_D16s_v3
      Count               : 4
    AllocatableVMs[4]   :
      VmSize             : Standard_D32-8s_v3
      Count               : 2
    AllocatableVMs[5]   :
      VmSize             : Standard_D32-16s_v3
      Count               : 2
    AllocatableVMs[6]   :
      VmSize             : Standard_D32s_v3
      Count               : 2
    AllocatableVMs[7]   :
      VmSize             : Standard_D64-16s_v3
      Count               : 1
    AllocatableVMs[8]   :
      VmSize             : Standard_D64-32s_v3
      Count               : 1
    AllocatableVMs[9]   :
      VmSize             : Standard_D64s_v3
      Count               : 1
  Statuses[0]           :
    Code                : ProvisioningState/succeeded
    Level               : Info
    DisplayStatus       : Provisioning succeeded
    Time                : 7/28/2019 5:31:01 PM
  Statuses[1]           :
    Code                : HealthState/available
    Level               : Info
    DisplayStatus       : Host available
Sku                   :
  Name                : DSV3-Type1
Id                    : /subscriptions/10101010-1010-1010-1010-1010101010/re
sourceGroups/myDHRG/providers/Microsoft.Compute/hostGroups/myHostGroup/hosts
/myHost
  Name                : myHost
  Location            : eastus
  Tags                : {}

```

Clean up

You are being charged for your dedicated hosts even when no virtual machines are deployed. You should delete any hosts you are currently not using to save costs.

You can only delete a host when there are no any longer virtual machines using it. Delete the VMs using [Remove-AzVM](#).

```
Remove-AzVM -ResourceGroupName $rgName -Name myVM
```

After deleting the VMs, you can delete the host using [Remove-AzHost](#).

```
Remove-AzHost -ResourceGroupName $rgName -Name myHost
```

Once you have deleted all of your hosts, you may delete the host group using [Remove-AzHostGroup](#).

```
Remove-AzHost -ResourceGroupName $rgName -Name myHost
```

You can also delete the entire resource group in a single command using [Remove-AzResourceGroup](#). This will delete all resources created in the group, including all of the VMs, hosts and host groups.

```
Remove-AzResourceGroup -Name $rgName
```

Next steps

- There is sample template, found [here](#), that uses both zones and fault domains for maximum resiliency in a region.
- You can also deploy dedicated hosts using the [Azure portal](#).

Deploy VMs to dedicated hosts using the portal

1/9/2020 • 3 minutes to read • [Edit Online](#)

This article guides you through how to create an Azure [dedicated host](#) to host your virtual machines (VMs).

Limitations

- Virtual machine scale sets are not currently supported on dedicated hosts.
- The initial release supports the following VM series: DSv3, ESv3, FSv2, LSv2, and MSv2.

Create a host group

A **host group** is a new resource that represents a collection of dedicated hosts. You create a host group in a region and an availability zone, and add hosts to it. When planning for high availability, there are additional options. You can use one or both of the following options with your dedicated hosts:

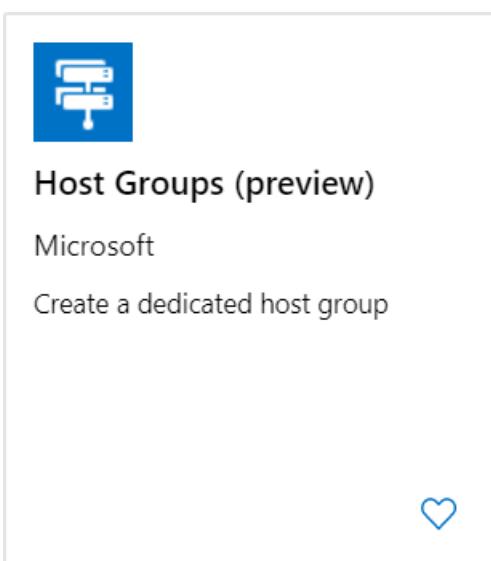
- Span across multiple availability zones. In this case, you are required to have a host group in each of the zones you wish to use.
- Span across multiple fault domains which are mapped to physical racks.

In either case, you are need to provide the fault domain count for your host group. If you do not want to span fault domains in your group, use a fault domain count of 1.

You can also decide to use both availability zones and fault domains.

In this example, we will create a host group using 1 availability zone and 2 fault domains.

1. Open the Azure [portal](#).
2. Select **Create a resource** in the upper left corner.
3. Search for **Host group** and then select **Host Groups** from the results.



4. In the **Host Groups** page, select **Create**.
5. Select the subscription you would like to use, and then select **Create new** to create a new resource group.
6. Type *myDedicatedHostsRG* as the **Name** and then select **OK**.

7. For **Host group name**, type *myHostGroup*.
8. For **Location**, select **East US**.
9. For **Availability Zone**, select **1**.
10. For **Fault domain count**, select **2**.
11. Select **Review + create** and then wait for validation.

Home > New > Marketplace > Host Groups (preview) > Create host group

Create host group

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

* Host group name ✓

* Location ▼

Availability zone ▼

* Fault domain count ▼

Review + create < Previous Next : Tags >

12. Once you see the **Validation passed** message, select **Create** to create the host group.

It should only take a few moments to create the host group.

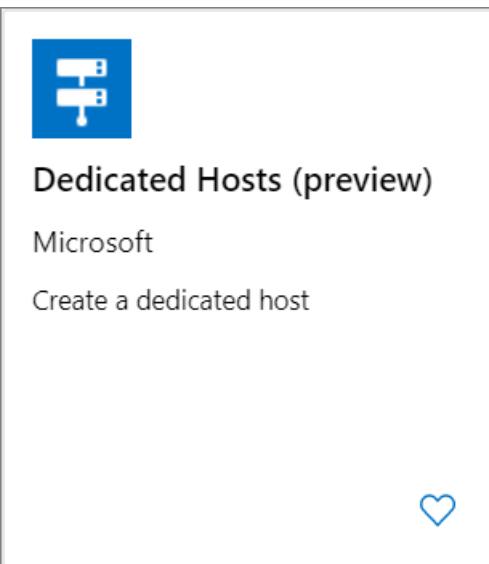
Create a dedicated host

Now create a dedicated host in the host group. In addition to a name for the host, you are required to provide the SKU for the host. Host SKU captures the supported VM series as well as the hardware generation for your dedicated host.

For more information about the host SKUs and pricing, see [Azure Dedicated Host pricing](#).

If you set a fault domain count for your host group, you will be asked to specify the fault domain for your host.

1. Select **Create a resource** in the upper left corner.
2. Search for **Dedicated host** and then select **Dedicated hosts** from the results.



3. In the **Dedicated Hosts** page, select **Create**.
4. Select the subscription you would like to use.
5. Select *myDedicatedHostsRG* as the **Resource group**.
6. In **Instance details**, type *myHost* for the **Name** and select *East US* for the location.
7. In **Hardware profile**, select *Standard Es3 family - Type 1* for the **Size family**, select *myHostGrup* for the **Host group** and then select *1* for the **Fault domain**. Leave the defaults for the rest of the fields.
8. When you are done, select **Review + create** and wait for validation.

Project details

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

* Subscription <small>?</small>	VMTesting1
└─ * Resource group <small>?</small>	myDedicatedHostRG
	Create new

Instance details

* Name <small>?</small>	myHost
* Location <small>?</small>	(US) East US

Hardware profile

* Size family <small>?</small>	Standard Es3 Family - Type 1
* Host group <small>?</small>	myHostGroup
* Fault domain	1
* Automatically replace host on failure <small>?</small>	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled

9. Once you see the **Validation passed** message, select **Create** to create the host.

Create a VM

1. Choose **Create a resource** in the upper left-hand corner of the Azure portal.
2. In the **New** page, under **Popular**, select **Windows Server 2016 Datacenter**.
3. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then select *myDedicatedHostsRG* as the **Resource group**.

4. Under **Instance details**, type *myVM* for the **Virtual machine name** and choose *East US* for your **Location**.
5. In **Availability options** select **Availability zone**, select *1* from the drop-down.
6. For the size, select **Change size**. In the list of available sizes, choose one from the Esv3 series, like **Standard E2s v3**. You may need to clear the filter in order to see all of the available sizes.
7. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the [defined complexity requirements](#).
8. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** from the drop-down.
9. At the top of the page, select the **Advanced** tab and in the **Host** section, select *myHostGroup* for **Host group** and *myHost* for the **Host**.

Host

Optionally placing your virtual machine in a host [Learn more](#)

Host group  myHostGroup | Zone 1 | eastus

Host  myHost

10. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.
11. When you see the message that validation has passed, select **Create**.

Next steps

- For more information, see the [Dedicated hosts](#) overview.
- There is sample template, found [here](#), that uses both zones and fault domains for maximum resiliency in a region.
- You can also deploy a dedicated host using [Azure PowerShell](#).

Preview: Deploy Spot VMs using the Azure portal

2/12/2020 • 2 minutes to read • [Edit Online](#)

Using [Spot VMs](#) allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Spot VMs. Therefore, Spot VMs are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Spot VMs is variable, based on region and SKU. For more information, see VM pricing for [Linux](#) and [Windows](#). For more information about setting the max price, see [Spot VMs - Pricing](#).

You have option to set a max price you are willing to pay, per hour, for the VM. The max price for a Spot VM can be set in US dollars (USD), using up to 5 decimal places. For example, the value would be a max price of \$0.05701 USD per hour. If you set the max price to be , the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, whichever is less, as long as there is capacity and quota available.

IMPORTANT

Spot instances are currently in public preview. This preview version is not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Create the VM

The process to create a VM that uses Spot VMs is the same as detailed in the [quickstart](#). When you are deploying a VM, you can choose to use an Azure spot instance.

On the **Basics** tab, in the **Instance details** section, **No** is the default for using an Azure spot instance.

The screenshot shows the 'Instance details' section of the Azure portal's VM creation wizard. It includes fields for Virtual machine name, Region, Availability options, Image, and Azure Spot instance selection. The 'Azure Spot instance' section is highlighted with a red border around the 'Yes' radio button, indicating it is the selected option.

Instance details	
Virtual machine name *	<input type="text"/>
Region *	(US) West US
Availability options	No infrastructure redundancy required
Image *	Ubuntu Server 18.04 LTS
Browse all public and private images	
Azure Spot instance	<input type="radio"/> Yes <input checked="" type="radio"/> No
Size *	Select size

If you select **Yes**, the section expands and you can choose your [eviction type and eviction policy](#).

Azure Spot instance [\(i\)](#)

Yes No

Eviction type [\(i\)](#)

Capacity only: evict virtual machine when Azure needs the capacity for pay as you go workloads. Your max price is set to the pay as you go rate.

Price or capacity: choose a max price and Azure will evict your virtual machine when the cost of the instance is greater than your max price or when Azure needs the capacity for pay as you go workloads.

Eviction policy [\(i\)](#)

Stop / Deallocate Delete (currently not supported)

Size * [\(i\)](#)

Standard D2s v3
2 vcpus, 8 GiB memory (\$0.01900/hour)
[Change size](#)

Maximum price you want to pay per hour
(USD) [\(i\)](#)

0.05701 

[Compare prices in nearby regions](#)

Next steps

You can also create Spot VMs using [PowerShell](#).

Preview: Deploy Spot VMs using Azure PowerShell

2/12/2020 • 2 minutes to read • [Edit Online](#)

Using [Spot VMs](#) allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Spot VMs. Therefore, Spot VMs are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Spot VMs is variable, based on region and SKU. For more information, see VM pricing for [Linux](#) and [Windows](#). For more information about setting the max price, see [Spot VMs - Pricing](#).

You have option to set a max price you are willing to pay, per hour, for the VM. The max price for a Spot VM can be set in US dollars (USD), using up to 5 decimal places. For example, the value `0.98765` would be a max price of \$0.98765 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, whichever is less, as long as there is capacity and quota available.

IMPORTANT

Spot instances are currently in public preview. This preview version is not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Create the VM

Create a spotVM using [New-AzVmConfig](#) to create the configuration. Include `-Priority Spot` and set `-MaxPrice` to either:

- `-1` so the VM is not evicted based on price.
- a dollar amount, up to 5 digits. For example, `-MaxPrice .98765` means that the VM will be deallocated once the price for a spotVM goes about \$.98765 per hour.

This example creates a spotVM that will not be deallocated based on pricing (only when Azure needs the capacity back).

```

$resourceGroup = "mySpotRG"
.setLocation = "eastus"
$vmName = "mySpotVM"
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."
New-AzResourceGroup -Name $resourceGroup -Location $location
$subnetConfig = New-AzVirtualNetworkSubnetConfig ` 
    -Name mySubnet -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup ` 
    -Location $location -Name MYvNET -AddressPrefix 192.168.0.0/16 ` 
    -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location $location ` 
    -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp ` 
    -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * ` 
    -DestinationPortRange 3389 -Access Allow
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location ` 
    -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location $location ` 
    -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration and set this to be a Spot VM

$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1 -Priority "Spot" -MaxPrice -1| ` 
Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | ` 
Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2016-Datacenter -Version latest | ` 
Add-AzVMNetworkInterface -Id $nic.Id

New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

```

After the VM is created, you can query to see the max price for all of the VMs in the resource group.

```

Get-AzVM -ResourceGroupName $resourceGroup | ` 
Select-Object Name,@{Name="maxPrice"; Expression={$_._BillingProfile.MaxPrice}}

```

Next steps

You can also create a Spot VM using the [Azure CLI](#) or a [template](#).

If you encounter an error, see [Error codes](#).

Deploy Spot VMs using a Resource Manager template

2/12/2020 • 2 minutes to read • [Edit Online](#)

Using [Spot VMs](#) allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Spot VMs. Therefore, Spot VMs are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Spot VMs is variable, based on region and SKU. For more information, see VM pricing for [Linux](#) and [Windows](#).

You have option to set a max price you are willing to pay, per hour, for the VM. The max price for a Spot VM can be set in US dollars (USD), using up to 5 decimal places. For example, the value `0.98765` would be a max price of \$0.98765 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for Spot or the price for a standard VM, whichever is less, as long as there is capacity and quota available. For more information about setting the max price, see [Spot VMs - Pricing](#).

IMPORTANT

Spot instances are currently in public preview. This preview version is not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Use a template

For Spot template deployments, use `"apiVersion": "2019-03-01"` or later. Add the `priority`, `evictionPolicy` and `billingProfile` properties to in your template:

```
    "priority": "Spot",
    "evictionPolicy": "Deallocate",
    "billingProfile": {
        "maxPrice": -1
    }
```

Here is a sample template with the added properties for a Spot VM. Replace the resource names with your own and `<password>` with a password for the local administrator account on the VM.

```
{
    "$schema": "http://schema.management.azure.com/schemas/2019-03-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {
        "vnetId": "/subscriptions/ec9fc04-e188-48b9-abfc-
abcd515f1836/resourceGroups/spotVM/providers/Microsoft.Network/virtualNetworks/spotVM",
        "subnetName": "default",
        "networkInterfaceName": "spotVMNIC",
        "publicIpAddressName": "spotVM-ip",
        "publicIpAddressType": "Dynamic",
        "publicIpAddressSku": "Basic",
        "virtualMachineName": "spotVM",
```

```

    "osDiskType": "Premium_LRS",
    "virtualMachineSize": "Standard_D2s_v3",
    "adminUsername": "azureuser",
    "adminPassword": "<password>",
    "diagnosticsStorageAccountName": "diagstoragespot2019",
    "diagnosticsStorageAccountId": "Microsoft.Storage/storageAccounts/diagstoragespot2019",
    "diagnosticsStorageAccountType": "Standard_LRS",
    "diagnosticsStorageAccountKind": "Storage",
    "subnetRef": "[concat(variables('vnetId'), '/subnets/', variables('subnetName'))]"
},
"resources": [
{
    "name": "spotVM",
    "type": "Microsoft.Network/networkInterfaces",
    "apiVersion": "2019-03-01",
    "location": "eastus",
    "dependsOn": [
        "[concat('Microsoft.Network/publicIpAddresses/', variables('publicIpAddressName'))]"
    ],
    "properties": {
        "ipConfigurations": [
            {
                "name": "ipconfig1",
                "properties": {
                    "subnet": {
                        "id": "[variables('subnetRef')]"
                    },
                    "privateIPAllocationMethod": "Dynamic",
                    "publicIpAddress": {
                        "id": "[resourceId(resourceGroup().name,
'Microsoft.Network/publicIpAddresses', variables('publicIpAddressName'))]"
                    }
                }
            }
        ]
    }
},
{
    "name": "[variables('publicIpAddressName')]",
    "type": "Microsoft.Network/publicIpAddresses",
    "apiVersion": "2019-02-01",
    "location": "eastus",
    "properties": {
        "publicIpAllocationMethod": "[variables('publicIpAddressType')]"
    },
    "sku": {
        "name": "[variables('publicIpAddressSku')]"
    }
},
{
    "name": "[variables('virtualMachineName')]",
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2019-03-01",
    "location": "eastus",
    "dependsOn": [
        "[concat('Microsoft.Network/networkInterfaces/', variables('networkInterfaceName'))]",
        "[concat('Microsoft.Storage/storageAccounts/', variables('diagnosticsStorageAccountName'))]"
    ],
    "properties": {
        "hardwareProfile": {
            "vmSize": "[variables('virtualMachineSize')]"
        },
        "storageProfile": {
            "osDisk": {
                "createOption": "fromImage",
                "managedDisk": {
                    "storageAccountType": "[variables('osDiskType')]"
                }
            }
        }
    }
}
]
}

```

```

        "imageReference": {
            "publisher": "Canonical",
            "offer": "UbuntuServer",
            "sku": "18.04-LTS",
            "version": "latest"
        }
    },
    "networkProfile": {
        "networkInterfaces": [
            {
                "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
            }
        ]
    },
    "osProfile": {
        "computerName": "[variables('virtualMachineName')]",
        "adminUsername": "[variables('adminUsername')]",
        "adminPassword": "[variables('adminPassword')]"
    },
    "diagnosticsProfile": {
        "bootDiagnostics": {
            "enabled": true,
            "storageUri": "[concat('https://', variables('diagnosticsStorageAccountName'),
'.blob.core.windows.net/')]"
        }
    },
    "priority": "Spot",
    "evictionPolicy": "Deallocate",
    "billingProfile": {
        "maxPrice": -1
    }
},
{
    "name": "[variables('diagnosticsStorageAccountName')]",
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2019-04-01",
    "location": "eastus",
    "properties": {},
    "kind": "[variables('diagnosticsStorageAccountKind')]",
    "sku": {
        "name": "[variables('diagnosticsStorageAccountType')]"
    }
},
],
"outputs": {
    "adminUsername": {
        "type": "string",
        "value": "[variables('adminUsername')]"
    }
}
}

```

Next steps

You can also create a Spot VM using [Azure PowerShell](#) or the [Azure CLI](#).

If you encounter an error, see [Error codes](#).

Preview: Error messages for Spot VMs and scale sets

2/10/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Spot VMs and virtual machine scale sets are currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Here are some possible error codes you could receive when using Spot VMs and scale sets.

KEY	MESSAGE	DESCRIPTION
SkuNotAvailable	The requested tier for resource '<resource>' is currently not available in location '<location>' for subscription '<subscriptionID>'. Please try another tier or deploy to a different location.	There is not enough Azure Spot capacity in this location to create your VM or scale set instance.
EvictionPolicyCanBeSetOnlyOnAzureSpotVirtualMachines	Eviction policy can be set only on Azure Spot Virtual Machines.	This VM is not a Spot VM, so you can't set the eviction policy.
AzureSpotVMNotSupportedInAvailabilitySet	Azure Spot Virtual Machine is not supported in Availability Set.	You need to choose to either use a Spot VM or use a VM in an availability set, you can't choose both.
AzureSpotFeatureNotEnabledForSubscription	Subscription not enabled with Azure Spot feature.	You need to have a subscription that supports Spot VMs.
VMPriorityCannotBeApplied	The specified priority value '{0}' cannot be applied to the Virtual Machine '{1}' since no priority was specified when the Virtual Machine was created.	You need to specify the priority when the VM is created.
SpotPriceGreater ThanProvidedMaxPrice	Unable to perform operation '{0}' since the provided max price '{1} USD' is lower than the current spot price '{2} USD' for Azure Spot VM size '{3}'.	Select a higher max price. For more information, see pricing information for Linux or Windows .
MaxPriceValueInvalid	Invalid max price value. The only supported values for max price are -1 or a decimal greater than zero. Max price value of -1 indicates the Azure Spot virtual machine will not be evicted for price reasons.	Enter a valid max price. For more information, see pricing for Linux or Windows .
MaxPriceChangeNotAllowedForAllocatedVMs	Max price change is not allowed when the VM '{0}' is currently allocated. Please deallocate and try again.	Stop\Deallocate the VM so that you can change the max price.
MaxPriceChangeNotAllowed	Max price change is not allowed.	You cannot change the max price for this VM.

KEY	MESSAGE	DESCRIPTION
AzureSpotIsNotSupportedForThisAPIVersion	Azure Spot is not supported for this API version.	The API version needs to be 2019-03-01.
AzureSpotIsNotSupportedForThisVMSize	Azure Spot is not supported for this VM size {0}.	Select another VM size. For more information, see Spot Virtual Machines .
MaxPriceIsSupportedOnlyForAzureSpotVirtualMachines	Max price is supported only for Azure Spot Virtual Machines.	For more information, see Spot Virtual Machines .
MoveResourcesWithAzureSpotVMNotSupported	The Move resources request contains a Azure Spot virtual machine. This is currently not supported. Please check the error details for virtual machine Ids.	You cannot move Spot VMs.
MoveResourcesWithAzureSpotVmssNotSupported	The Move resources request contains a Azure Spot virtual machine scale set. This is currently not supported. Please check the error details for virtual machine scale set Ids.	You cannot move Spot scale set instances.
EphemeralOSDisksNotSupportedForSpotVMs	Ephemeral OS disks are not supported for Spot VMs.	You need to be using a regular OS disk for your Spot VM.
AzureSpotVMNotSupportedInVmssWithVMOrchestrationMode	Azure Spot Virtual Machine is not supported in Virtual Machine Scale Set with VM Orchestration mode.	Set the orchestration mode to virtual machine scale set in order to use Spot instances.

Next steps For more information, see [spot Virtual Machines](#).

Create and manage Windows VMs in Azure using C#

12/23/2019 • 7 minutes to read • [Edit Online](#)

An [Azure Virtual Machine](#) (VM) needs several supporting Azure resources. This article covers creating, managing, and deleting VM resources using C#. You learn how to:

- Create a Visual Studio project
- Install the package
- Create credentials
- Create resources
- Perform management tasks
- Delete resources
- Run the application

It takes about 20 minutes to do these steps.

Create a Visual Studio project

1. If you haven't already, install [Visual Studio](#). Select **.NET desktop development** on the Workloads page, and then click **Install**. In the summary, you can see that **.NET Framework 4 - 4.6 development tools** is automatically selected for you. If you have already installed Visual Studio, you can add the .NET workload using the Visual Studio Launcher.
2. In Visual Studio, click **File > New > Project**.
3. In **Templates > Visual C#**, select **Console App (.NET Framework)**, enter *myDotnetProject* for the name of the project, select the location of the project, and then click **OK**.

Install the package

NuGet packages are the easiest way to install the libraries that you need to finish these steps. To get the libraries that you need in Visual Studio, do these steps:

1. Click **Tools > Nuget Package Manager**, and then click **Package Manager Console**.
2. Type this command in the console:

```
Install-Package Microsoft.Azure.Management.Fluent
```

Create credentials

Before you start this step, make sure that you have access to an [Active Directory service principal](#). You should also record the application ID, the authentication key, and the tenant ID that you need in a later step.

Create the authorization file

1. In Solution Explorer, right-click *myDotnetProject* > **Add > New Item**, and then select **Text File** in **Visual C# Items**. Name the file *azureauth.properties*, and then click **Add**.
2. Add these authorization properties:

```
subscription=<subscription-id>
client=<application-id>
key=<authentication-key>
tenant=<tenant-id>
managementURI=https://management.core.windows.net/
baseURL=https://management.azure.com/
authURL=https://login.windows.net/
graphURL=https://graph.windows.net/
```

Replace **<subscription-id>** with your subscription identifier, **<application-id>** with the Active Directory application identifier, **<authentication-key>** with the application key, and **<tenant-id>** with the tenant identifier.

3. Save the `azureauth.properties` file.
4. Set an environment variable in Windows named `AZURE_AUTH_LOCATION` with the full path to authorization file that you created. For example, the following PowerShell command can be used:

```
[Environment]::SetEnvironmentVariable("AZURE_AUTH_LOCATION", "C:\Visual Studio 2019\Projects\myDotnetProject\myDotnetProject\azureauth.properties", "User")
```

Create the management client

1. Open the `Program.cs` file for the project that you created. Then, add these using statements to the existing statements at top of the file:

```
using Microsoft.Azure.Management.Compute.Fluent;
using Microsoft.Azure.Management.Compute.Fluent.Models;
using Microsoft.Azure.Management.Fluent;
using Microsoft.Azure.Management.ResourceManager.Fluent;
using Microsoft.Azure.Management.ResourceManager.Fluent.Core;
```

2. To create the management client, add this code to the `Main` method:

```
var credentials = SdkContext.AzureCredentialsFactory
    .FromFile(Environment.GetEnvironmentVariable("AZURE_AUTH_LOCATION"));

var azure = Azure
    .Configure()
    .WithLogLevel(HttpLoggingDelegatingHandler.Level.Basic)
    .Authenticate(credentials)
    .WithDefaultSubscription();
```

Create resources

Create the resource group

All resources must be contained in a [Resource group](#).

To specify values for the application and create the resource group, add this code to the `Main` method:

```
var groupName = "myResourceGroup";
var vmName = "myVM";
var location = Region.USWest;

Console.WriteLine("Creating resource group...");
var resourceGroup = azure.ResourceGroups.Define(groupName)
    .WithRegion(location)
    .Create();
```

Create the availability set

[Availability sets](#) make it easier for you to maintain the virtual machines used by your application.

To create the availability set, add this code to the Main method:

```
Console.WriteLine("Creating availability set...");
var availabilitySet = azure.AvailabilitySets.Define("myAVSet")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithSku(AvailabilitySetSkuTypes.Managed)
    .Create();
```

Create the public IP address

A [Public IP address](#) is needed to communicate with the virtual machine.

To create the public IP address for the virtual machine, add this code to the Main method:

```
Console.WriteLine("Creating public IP address...");
var publicIPAddress = azure.PublicIPAddresses.Define("myPublicIP")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithDynamicIP()
    .Create();
```

Create the virtual network

A virtual machine must be in a subnet of a [Virtual network](#).

To create a subnet and a virtual network, add this code to the Main method:

```
Console.WriteLine("Creating virtual network...");
var network = azure.Networks.Define("myVNet")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithAddressSpace("10.0.0.0/16")
    .WithSubnet("mySubnet", "10.0.0.0/24")
    .Create();
```

Create the network interface

A virtual machine needs a network interface to communicate on the virtual network.

To create a network interface, add this code to the Main method:

```
Console.WriteLine("Creating network interface...");
var networkInterface = azure.NetworkInterfaces.Define("myNIC")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithExistingPrimaryNetwork(network)
    .WithSubnet("mySubnet")
    .WithPrimaryPrivateIPAddressDynamic()
    .WithExistingPrimaryPublicIPAddress(publicIPAddress)
    .Create();
```

Create the virtual machine

Now that you created all the supporting resources, you can create a virtual machine.

To create the virtual machine, add this code to the Main method:

```
Console.WriteLine("Creating virtual machine...");
azure.VirtualMachines.Define(vmName)
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithExistingPrimaryNetworkInterface(networkInterface)
    .WithLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer", "2012-R2-Datacenter")
    .WithAdminUsername("azureuser")
    .WithAdminPassword("Azure12345678")
    .WithComputerName(vmName)
    .WithExistingAvailabilitySet(availabilitySet)
    .WithSize(VirtualMachineSizeTypes.StandardDS1)
    .Create();
```

NOTE

This tutorial creates a virtual machine running a version of the Windows Server operating system. To learn more about selecting other images, see [Navigate and select Azure virtual machine images with Windows PowerShell and the Azure CLI](#).

If you want to use an existing disk instead of a marketplace image, use this code:

```
var managedDisk = azure.Disks.Define("myosdisk")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithWindowsFromVhd("https://mystorage.blob.core.windows.net/vhds/myosdisk.vhd")
    .WithSizeInGB(128)
    .WithSku(DiskSkuTypes.PremiumLRS)
    .Create();

azure.VirtualMachines.Define("myVM")
    .WithRegion(location)
    .WithExistingResourceGroup(groupName)
    .WithExistingPrimaryNetworkInterface(networkInterface)
    .WithSpecializedOSDisk(managedDisk, OperatingSystemTypes.Windows)
    .WithExistingAvailabilitySet(availabilitySet)
    .WithSize(VirtualMachineSizeTypes.StandardDS1)
    .Create();
```

Perform management tasks

During the lifecycle of a virtual machine, you may want to run management tasks such as starting, stopping, or deleting a virtual machine. Additionally, you may want to create code to automate repetitive or complex tasks.

When you need to do anything with the VM, you need to get an instance of it:

```
var vm = azure.VirtualMachines.GetByResourceGroup(groupName, vmName);
```

Get information about the VM

To get information about the virtual machine, add this code to the Main method:

```

Console.WriteLine("Getting information about the virtual machine...");
Console.WriteLine("hardwareProfile");
Console.WriteLine("  vmSize: " + vm.Size);
Console.WriteLine("storageProfile");
Console.WriteLine("  imageReference");
Console.WriteLine("    publisher: " + vm.StorageProfile.ImageReference.Publisher);
Console.WriteLine("    offer: " + vm.StorageProfile.ImageReference.Offer);
Console.WriteLine("    sku: " + vm.StorageProfile.ImageReference.Sku);
Console.WriteLine("    version: " + vm.StorageProfile.ImageReference.Version);
Console.WriteLine("osDisk");
Console.WriteLine("  osType: " + vm.StorageProfile.OsDisk.OsType);
Console.WriteLine("  name: " + vm.StorageProfile.OsDisk.Name);
Console.WriteLine("  createOption: " + vm.StorageProfile.OsDisk.CreateOption);
Console.WriteLine("  caching: " + vm.StorageProfile.OsDisk.Caching);
Console.WriteLine("osProfile");
Console.WriteLine("  computerName: " + vm.OSProfile.ComputerName);
Console.WriteLine("  adminUsername: " + vm.OSProfile.AdminUsername);
Console.WriteLine("  provisionVMAgent: " + vm.OSProfile.WindowsConfiguration.ProvisionVMAgent.Value);
Console.WriteLine("  enableAutomaticUpdates: " +
vm.OSProfile.WindowsConfiguration.EnableAutomaticUpdates.Value);
Console.WriteLine("networkProfile");
foreach (string nicId in vm.NetworkInterfaceIds)
{
    Console.WriteLine("  networkInterface id: " + nicId);
}
Console.WriteLine("vmAgent");
Console.WriteLine("  vmAgentVersion" + vm.InstanceView.VmAgent.VmAgentVersion);
Console.WriteLine("  statuses");
foreach (InstanceViewStatus stat in vm.InstanceView.VmAgent.Statuses)
{
    Console.WriteLine("    code: " + stat.Code);
    Console.WriteLine("    level: " + stat.Level);
    Console.WriteLine("    displayStatus: " + stat.DisplayStatus);
    Console.WriteLine("    message: " + stat.Message);
    Console.WriteLine("    time: " + stat.Time);
}
Console.WriteLine("disks");
foreach (DiskInstanceView disk in vm.InstanceView.Disks)
{
    Console.WriteLine("  name: " + disk.Name);
    Console.WriteLine("  statuses");
    foreach (InstanceViewStatus stat in disk.Statuses)
    {
        Console.WriteLine("    code: " + stat.Code);
        Console.WriteLine("    level: " + stat.Level);
        Console.WriteLine("    displayStatus: " + stat.DisplayStatus);
        Console.WriteLine("    time: " + stat.Time);
    }
}
Console.WriteLine("VM general status");
Console.WriteLine("  provisioningStatus: " + vm.ProvisioningState);
Console.WriteLine("  id: " + vm.Id);
Console.WriteLine("  name: " + vm.Name);
Console.WriteLine("  type: " + vm.Type);
Console.WriteLine("  location: " + vm.Region);
Console.WriteLine("VM instance status");
foreach (InstanceViewStatus stat in vm.InstanceView.Statuses)
{
    Console.WriteLine("  code: " + stat.Code);
    Console.WriteLine("  level: " + stat.Level);
    Console.WriteLine("  displayStatus: " + stat.DisplayStatus);
}
Console.WriteLine("Press enter to continue...");
Console.ReadLine();

```

Stop the VM

You can stop a virtual machine and keep all its settings, but continue to be charged for it, or you can stop a virtual machine and deallocate it. When a virtual machine is deallocated, all resources associated with it are also deallocated and billing ends for it.

To stop the virtual machine without deallocating it, add this code to the Main method:

```
Console.WriteLine("Stopping vm...");
vm.PowerOff();
Console.WriteLine("Press enter to continue...");
Console.ReadLine();
```

If you want to deallocate the virtual machine, change the PowerOff call to this code:

```
vm.Deallocate();
```

Start the VM

To start the virtual machine, add this code to the Main method:

```
Console.WriteLine("Starting vm...");
vm.Start();
Console.WriteLine("Press enter to continue...");
Console.ReadLine();
```

Resize the VM

Many aspects of deployment should be considered when deciding on a size for your virtual machine. For more information, see [VM sizes](#).

To change size of the virtual machine, add this code to the Main method:

```
Console.WriteLine("Resizing vm...");
vm.Update()
    .WithSize(VirtualMachineSizeTypes.StandardDS2)
    .Apply();
Console.WriteLine("Press enter to continue...");
Console.ReadLine();
```

Add a data disk to the VM

To add a data disk to the virtual machine, add this code to the Main method. This example adds a data disk that is 2 GB in size, has a LUN of 0 and a caching type of ReadWrite:

```
Console.WriteLine("Adding data disk to vm...");
vm.Update()
    .WithNewDataDisk(2, 0, CachingTypes.ReadWrite)
    .Apply();
Console.WriteLine("Press enter to delete resources...");
Console.ReadLine();
```

Delete resources

Because you are charged for resources used in Azure, it is always good practice to delete resources that are no longer needed. If you want to delete the virtual machines and all the supporting resources, all you have to do is delete the resource group.

To delete the resource group, add this code to the Main method:

```
azure.ResourceGroups.DeleteByName(groupName);
```

Run the application

It should take about five minutes for this console application to run completely from start to finish.

1. To run the console application, click **Start**.
2. Before you press **Enter** to start deleting resources, you could take a few minutes to verify the creation of the resources in the Azure portal. Click the deployment status to see information about the deployment.

Next steps

- Take advantage of using a template to create a virtual machine by using the information in [Deploy an Azure Virtual Machine using C# and a Resource Manager template](#).
- Learn more about using the [Azure libraries for .NET](#).

Create a VM from a VHD by using the Azure portal

11/13/2019 • 3 minutes to read • [Edit Online](#)

There are several ways to create a virtual machine (VM) in Azure:

- If you already have a virtual hard disk (VHD) to use or you want to copy the VHD from an existing VM to use, you can create a new VM by *attaching* the VHD to the new VM as an OS disk.
- You can create a new VM from the VHD of a VM that has been deleted. For example, if you have an Azure VM that isn't working correctly, you can delete the VM and use its VHD to create a new VM. You can either reuse the same VHD or create a copy of the VHD by creating a snapshot and then creating a new managed disk from the snapshot. Although creating a snapshot takes a few more steps, it preserves the original VHD and provides you with a fallback.
- Take a classic VM and use the VHD to create a new VM that uses the Resource Manager deployment model and managed disks. For the best results, **Stop** the classic VM in the Azure portal before creating the snapshot.
- You can create an Azure VM from an on-premises VHD by uploading the on-premises VHD and attaching it to a new VM. You use PowerShell or another tool to upload the VHD to a storage account, and then you create a managed disk from the VHD. For more information, see [Upload a specialized VHD](#).

Don't use a specialized disk if you want to create multiple VMs. Instead, for larger deployments, [create an image](#) and then [use that image to create multiple VMs](#).

We recommend that you limit the number of concurrent deployments to 20 VMs from a single snapshot or VHD.

Copy a disk

Create a snapshot and then create a disk from the snapshot. This strategy allows you to keep the original VHD as a fallback:

1. From the [Azure portal](#), on the left menu, select **All services**.
2. In the **All services** search box, enter **disks** and then select **Disk** to display the list of available disks.
3. Select the disk that you would like to use. The **Disk** page for that disk appears.
4. From the menu at the top, select **Create snapshot**.
5. Enter a **Name** for the snapshot.
6. Choose a **Resource group** for the snapshot. You can use either an existing resource group or create a new one.
7. For **Account type**, choose either **Standard (HDD)** or **Premium (SSD)** storage.
8. When you're done, select **Create** to create the snapshot.
9. After the snapshot has been created, select **Create a resource** in the left menu.
10. In the search box, enter **managed disk** and then select **Managed Disks** from the list.
11. On the **Managed Disks** page, select **Create**.
12. Enter a **Name** for the disk.
13. Choose a **Resource group** for the disk. You can use either an existing resource group or create a new one. This selection will also be used as the resource group where you create the VM from the disk.
14. For **Account type**, choose either **Standard (HDD)** or **Premium (SSD)** storage.
15. In **Source type**, ensure **Snapshot** is selected.
16. In the **Source snapshot** drop-down, select the snapshot you want to use.
17. Make any other adjustments as needed and then select **Create** to create the disk.

Create a VM from a disk

After you have the managed disk VHD that you want to use, you can create the VM in the portal:

1. From the [Azure portal](#), on the left menu, select **All services**.
2. In the **All services** search box, enter **disks** and then select **Disks** to display the list of available disks.
3. Select the disk that you would like to use. The **Disk** page for that disk opens.
4. In the **Overview** page, ensure that **DISK STATE** is listed as **Unattached**. If it isn't, you might need to either detach the disk from the VM or delete the VM to free up the disk.
5. In the menu at the top of the page, select **Create VM**.
6. On the **Basics** page for the new VM, enter a **Virtual machine name** and either select an existing **Resource group** or create a new one.
7. For **Size**, select **Change size** to access the **Size** page.
8. Select a VM size row and then choose **Select**.
9. On the **Networking** page, you can either let the portal create all new resources or you can select an existing **Virtual network** and **Network security group**. The portal always creates a new network interface and public IP address for the new VM.
10. On the **Management** page, make any changes to the monitoring options.
11. On the **Guest config** page, add any extensions as needed.
12. When you're done, select **Review + create**.
13. If the VM configuration passes validation, select **Create** to start the deployment.

Next steps

You can also use PowerShell to [upload a VHD to Azure and create a specialized VM](#).

Create a Windows VM from a specialized disk by using PowerShell

12/18/2019 • 6 minutes to read • [Edit Online](#)

Create a new VM by attaching a specialized managed disk as the OS disk. A specialized disk is a copy of a virtual hard disk (VHD) from an existing VM that contains the user accounts, applications, and other state data from your original VM.

When you use a specialized VHD to create a new VM, the new VM retains the computer name of the original VM. Other computer-specific information is also kept and, in some cases, this duplicate information could cause issues. When copying a VM, be aware of what types of computer-specific information your applications rely on.

You have several options:

- [Use an existing managed disk](#). This option is useful if you have a VM that isn't working correctly. You can delete the VM and then reuse the managed disk to create a new VM.
- [Upload a VHD](#)
- [Copy an existing Azure VM by using snapshots](#)

You can also use the Azure portal to [create a new VM from a specialized VHD](#).

This article shows you how to use managed disks. If you have a legacy deployment that requires using a storage account, see [Create a VM from a specialized VHD in a storage account](#).

We recommend that you limit the number of concurrent deployments to 20 VMs from a single VHD or snapshot.

Option 1: Use an existing disk

If you had a VM that you deleted and you want to reuse the OS disk to create a new VM, use [Get-AzDisk](#).

```
$resourceGroupName = 'myResourceGroup'  
$osDiskName = 'myOsDisk'  
$osDisk = Get-AzDisk `  
-ResourceGroupName $resourceGroupName `  
-DiskName $osDiskName
```

You can now attach this disk as the OS disk to a [new VM](#).

Option 2: Upload a specialized VHD

You can upload the VHD from a specialized VM created with an on-premises virtualization tool, like Hyper-V, or a VM exported from another cloud.

Prepare the VM

Use the VHD as-is to create a new VM.

- [Prepare a Windows VHD to upload to Azure](#). **Do not** generalize the VM by using Sysprep.
- Remove any guest virtualization tools and agents that are installed on the VM (such as VMware tools).
- Make sure the VM is configured to get the IP address and DNS settings from DHCP. This ensures that the server obtains an IP address within the virtual network when it starts up.

Upload the VHD

You can now upload a VHD straight into a managed disk. For instructions, see [Upload a VHD to Azure using Azure PowerShell](#).

Option 3: Copy an existing Azure VM

You can create a copy of a VM that uses managed disks by taking a snapshot of the VM, and then by using that snapshot to create a new managed disk and a new VM.

If you want to copy an existing VM to another region, you might want to use azcopy to [create a copy of a disk in another region](#).

Take a snapshot of the OS disk

You can take a snapshot of an entire VM (including all disks) or of just a single disk. The following steps show you how to take a snapshot of just the OS disk of your VM with the [New-AzSnapshot](#) cmdlet.

First, set some parameters.

```
$resourceGroupName = 'myResourceGroup'  
$vmName = 'myVM'  
$location = 'westus'  
$snapshotName = 'mySnapshot'
```

Get the VM object.

```
$vm = Get-AzVM -Name $vmName  
-ResourceGroupName $resourceGroupName
```

Get the OS disk name.

```
$disk = Get-AzDisk -ResourceGroupName $resourceGroupName  
-DiskName $vm.StorageProfile.OsDisk.Name
```

Create the snapshot configuration.

```
$snapshotConfig = New-AzSnapshotConfig `  
-SourceUri $disk.Id `  
-OsType Windows `  
-CreateOption Copy `  
-Location $location
```

Take the snapshot.

```
$snapshot = New-AzSnapshot `  
-Snapshot $snapshotConfig `  
-SnapshotName $snapshotName `  
-ResourceGroupName $resourceGroupName
```

To use this snapshot to create a VM that needs to be high-performing, add the parameter

`-AccountType Premium_LRS` to the `New-AzSnapshotConfig` command. This parameter creates the snapshot so that it's stored as a Premium Managed Disk. Premium Managed Disks are more expensive than Standard, so be sure you'll need Premium before using this parameter.

Create a new disk from the snapshot

Create a managed disk from the snapshot by using `New-AzDisk`. This example uses *myOSDisk* for the disk name.

Create a new resource group for the new VM.

```
$destinationResourceGroup = 'myDestinationResourceGroup'  
New-AzResourceGroup -Location $location `  
-Name $destinationResourceGroup
```

Set the OS disk name.

```
$osDiskName = 'myOsDisk'
```

Create the managed disk.

```
$osDisk = New-AzDisk -DiskName $osDiskName -Disk `  
    (New-AzDiskConfig -Location $location -CreateOption Copy `  
    -SourceResourceId $snapshot.Id) `  
    -ResourceGroupName $destinationResourceGroup
```

Create the new VM

Create networking and other VM resources to be used by the new VM.

Create the subnet and virtual network

Create the [virtual network](#) and subnet for the VM.

1. Create the subnet. This example creates a subnet named *mySubNet*, in the resource group *myDestinationResourceGroup*, and sets the subnet address prefix to *10.0.0.0/24*.

```
$subnetName = 'mySubNet'  
$singleSubnet = New-AzVirtualNetworkSubnetConfig `  
    -Name $subnetName `  
    -AddressPrefix 10.0.0.0/24
```

2. Create the virtual network. This example sets the virtual network name to *myVnetName*, the location to *West US*, and the address prefix for the virtual network to *10.0.0.0/16*.

```
$vnetName = "myVnetName"  
$vnet = New-AzVirtualNetwork `  
    -Name $vnetName -ResourceGroupName $destinationResourceGroup `  
    -Location $location `  
    -AddressPrefix 10.0.0.0/16 `  
    -Subnet $singleSubnet
```

Create the network security group and an RDP rule

To be able to sign in to your VM with remote desktop protocol (RDP), you'll need to have a security rule that allows RDP access on port 3389. In our example, the VHD for the new VM was created from an existing specialized VM, so you can use an account that existed on the source virtual machine for RDP.

This example sets the network security group (NSG) name to *myNsg* and the RDP rule name to *myRdpRule*.

```

$nsgName = "myNsg"

$rdpRule = New-AzNetworkSecurityRuleConfig -Name myRdpRule -Description "Allow RDP" ` 
    -Access Allow -Protocol Tcp -Direction Inbound -Priority 110 ` 
    -SourceAddressPrefix Internet -SourcePortRange * ` 
    -DestinationAddressPrefix * -DestinationPortRange 3389
$nsg = New-AzNetworkSecurityGroup ` 
    -ResourceGroupName $destinationResourceGroup ` 
    -Location $location ` 
    -Name $nsgName -SecurityRules $rdpRule

```

For more information about endpoints and NSG rules, see [Opening ports to a VM in Azure by using PowerShell](#).

Create a public IP address and NIC

To enable communication with the virtual machine in the virtual network, you'll need a [public IP address](#) and a network interface.

1. Create the public IP. In this example, the public IP address name is set to *myIP*.

```

$ipName = "myIP"
$pip = New-AzPublicIpAddress ` 
    -Name $ipName -ResourceGroupName $destinationResourceGroup ` 
    -Location $location ` 
    -AllocationMethod Dynamic

```

2. Create the NIC. In this example, the NIC name is set to *myNicName*.

```

$nicName = "myNicName"
$nic = New-AzNetworkInterface -Name $nicName ` 
    -ResourceGroupName $destinationResourceGroup ` 
    -Location $location -SubnetId $vnet.Subnets[0].Id ` 
    -PublicIpAddressId $pip.Id ` 
    -NetworkSecurityGroupId $nsg.Id

```

Set the VM name and size

This example sets the VM name to *myVM* and the VM size to *Standard_A2*.

```

$vmName = "myVM"
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize "Standard_A2"

```

Add the NIC

```
$vm = Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id
```

Add the OS disk

Add the OS disk to the configuration by using [Set-AzVMOSDisk](#). This example sets the size of the disk to *128 GB* and attaches the managed disk as a *Windows* OS disk.

```
$vm = Set-AzVMOSDisk -VM $vm -ManagedDiskId $osDisk.Id -StorageAccountType Standard_LRS ` 
    -DiskSizeInGB 128 -CreateOption Attach -Windows
```

Complete the VM

Create the VM by using [New-AzVM](#) with the configurations that we just created.

```
New-AzVM -ResourceGroupName $destinationResourceGroup -Location $location -VM $vm
```

If this command is successful, you'll see output like this:

RequestId	IsSuccessStatusCode	StatusCode	ReasonPhrase
-----	-----	-----	-----
True		OK	OK

Verify that the VM was created

You should see the newly created VM either in the [Azure portal](#) under **Browse > Virtual machines**, or by using the following PowerShell commands.

```
$vmList = Get-AzVM -ResourceGroupName $destinationResourceGroup  
$vmList.Name
```

Next steps

Sign in to your new virtual machine. For more information, see [How to connect and log on to an Azure virtual machine running Windows](#).

Deploy an Azure Virtual Machine using C# and a Resource Manager template

11/13/2019 • 5 minutes to read • [Edit Online](#)

This article shows you how to deploy an Azure Resource Manager template using C#. The template that you create deploys a single virtual machine running Windows Server in a new virtual network with a single subnet.

For a detailed description of the virtual machine resource, see [Virtual machines in an Azure Resource Manager template](#). For more information about all the resources in a template, see [Azure Resource Manager template walkthrough](#).

It takes about 10 minutes to do these steps.

Create a Visual Studio project

In this step, you make sure that Visual Studio is installed and you create a console application used to deploy the template.

1. If you haven't already, install [Visual Studio](#). Select **.NET desktop development** on the Workloads page, and then click **Install**. In the summary, you can see that **.NET Framework 4 - 4.6 development tools** is automatically selected for you. If you have already installed Visual Studio, you can add the .NET workload using the Visual Studio Launcher.
2. In Visual Studio, click **File > New > Project**.
3. In **Templates > Visual C#**, select **Console App (.NET Framework)**, enter *myDotnetProject* for the name of the project, select the location of the project, and then click **OK**.

Install the packages

NuGet packages are the easiest way to install the libraries that you need to finish these steps. To get the libraries that you need in Visual Studio, do these steps:

1. Click **Tools > Nuget Package Manager**, and then click **Package Manager Console**.
2. Type these commands in the console:

```
Install-Package Microsoft.Azure.Management.Fluent  
Install-Package WindowsAzure.Storage
```

Create the files

In this step, you create a template file that deploys the resources and a parameters file that supplies parameter values to the template. You also create an authorization file that is used to perform Azure Resource Manager operations.

Create the template file

1. In Solution Explorer, right-click *myDotnetProject* > **Add > New Item**, and then select **Text File** in *Visual C# Items*. Name the file *CreateVMTemplate.json*, and then click **Add**.
2. Add this JSON code to the file that you created:

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "adminUsername": { "type": "string" },
    "adminPassword": { "type": "securestring" }
  },
  "variables": {
    "vnetID": "[resourceId('Microsoft.Network/virtualNetworks', 'myVNet')]",
    "subnetRef": "[concat(variables('vnetID'), '/subnets/mySubnet')]"
  },
  "resources": [
    {
      "apiVersion": "2016-03-30",
      "type": "Microsoft.Network/publicIPAddresses",
      "name": "myPublicIPAddress",
      "location": "[resourceGroup().location]",
      "properties": {
        "publicIPAllocationMethod": "Dynamic",
        "dnsSettings": {
          "domainNameLabel": "myresourcegroupdns1"
        }
      }
    },
    {
      "apiVersion": "2016-03-30",
      "type": "Microsoft.Network/virtualNetworks",
      "name": "myVNet",
      "location": "[resourceGroup().location]",
      "properties": {
        "addressSpace": { "addressPrefixes": [ "10.0.0.0/16" ] },
        "subnets": [
          {
            "name": "mySubnet",
            "properties": { "addressPrefix": "10.0.0.0/24" }
          }
        ]
      }
    },
    {
      "apiVersion": "2016-03-30",
      "type": "Microsoft.Network/networkInterfaces",
      "name": "myNic",
      "location": "[resourceGroup().location]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses/', 'myPublicIPAddress')]",
        "[resourceId('Microsoft.Network/virtualNetworks/', 'myVNet')]"
      ],
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": { "id": "[resourceId('Microsoft.Network/publicIPAddresses', 'myPublicIPAddress')]",
                "subnet": { "id": "[variables('subnetRef')]" }
              }
            }
          }
        ]
      }
    },
    {
      "apiVersion": "2016-04-30-preview",
      "type": "Microsoft.Compute/virtualMachines",
      "name": "myVM",
      "location": "[resourceGroup().location]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces/', 'myNic')]"
      ]
    }
  ]
}
```

```

],
"properties": {
    "hardwareProfile": { "vmSize": "Standard_DS1" },
    "osProfile": {
        "computerName": "myVM",
        "adminUsername": "[parameters('adminUsername')]",
        "adminPassword": "[parameters('adminPassword')]"
    },
    "storageProfile": {
        "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "2012-R2-Datacenter",
            "version": "latest"
        },
        "osDisk": {
            "name": "myManagedOSDisk",
            "caching": "ReadWrite",
            "createOption": "FromImage"
        }
    },
    "networkProfile": {
        "networkInterfaces": [
            {
                "id": "[resourceId('Microsoft.Network/networkInterfaces','myNic')]"
            }
        ]
    }
}
]
}

```

- Save the CreateVMTemplate.json file.

Create the parameters file

To specify values for the resource parameters in the template, you create a parameters file that contains the values.

- In Solution Explorer, right-click *myDotnetProject* > **Add** > **New Item**, and then select **Text File** in *Visual C# Items*. Name the file *Parameters.json*, and then click **Add**.
- Add this JSON code to the file that you created:

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "adminUserName": { "value": "azureuser" },
        "adminPassword": { "value": "Azure12345678" }
    }
}
```

- Save the Parameters.json file.

Create the authorization file

Before you can deploy a template, make sure that you have access to an [Active Directory service principal](#). From the service principal, you acquire a token for authenticating requests to Azure Resource Manager. You should also record the application ID, the authentication key, and the tenant ID that you need in the authorization file.

- In Solution Explorer, right-click *myDotnetProject* > **Add** > **New Item**, and then select **Text File** in *Visual C# Items*. Name the file *azureauth.properties*, and then click **Add**.
- Add these authorization properties:

```
subscription=<subscription-id>
client=<application-id>
key=<authentication-key>
tenant=<tenant-id>
managementURI=https://management.core.windows.net/
baseURL=https://management.azure.com/
authURL=https://login.windows.net/
graphURL=https://graph.windows.net/
```

Replace **<subscription-id>** with your subscription identifier, **<application-id>** with the Active Directory application identifier, **<authentication-key>** with the application key, and **<tenant-id>** with the tenant identifier.

3. Save the azureauth.properties file.
4. Set an environment variable in Windows named AZURE_AUTH_LOCATION with the full path to authorization file that you created, for example you can use the following PowerShell command:

```
[Environment]::SetEnvironmentVariable("AZURE_AUTH_LOCATION", "C:\Visual Studio 2019\Projects\myDotnetProject\myDotnetProject\azureauth.properties", "User")
```

Create the management client

1. Open the Program.cs file for the project that you created. Then, add these using statements to the existing statements at top of the file:

```
using Microsoft.Azure.Management.Compute.Fluent;
using Microsoft.Azure.Management.Compute.Fluent.Models;
using Microsoft.Azure.Management.Fluent;
using Microsoft.Azure.Management.ResourceManager.Fluent;
using Microsoft.Azure.Management.ResourceManager.Fluent.Core;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Blob;
```

2. To create the management client, add this code to the Main method:

```
var credentials = SdkContext.AzureCredentialsFactory
    .FromFile(Environment.GetEnvironmentVariable("AZURE_AUTH_LOCATION"));

var azure = Azure
    .Configure()
    .WithLogLevel(HttpLoggingDelegatingHandler.Level.Basic)
    .Authenticate(credentials)
    .WithDefaultSubscription();
```

Create a resource group

To specify values for the application, add code to the Main method:

```
var groupName = "myResourceGroup";
var location = Region.USWest;

var resourceGroup = azure.ResourceGroups.Define(groupName)
    .WithRegion(location)
    .Create();
```

Create a storage account

The template and parameters are deployed from a storage account in Azure. In this step, you create the account and upload the files.

To create the account, add this code to the Main method:

```
string storageAccountName = SdkContext.RandomResourceName("st", 10);

Console.WriteLine("Creating storage account...");
var storage = azure.StorageAccounts.Define(storageAccountName)
    .WithRegion(Region.USWest)
    .WithExistingResourceGroup(resourceGroup)
    .Create();

var storageKeys = storage.GetKeys();
string storageConnectionString = "DefaultEndpointsProtocol=https;" +
    + "AccountName=" + storage.Name
    + ";AccountKey=" + storageKeys[0].Value
    + ";EndpointSuffix=core.windows.net";

var account = CloudStorageAccount.Parse(storageConnectionString);
var serviceClient = account.CreateCloudBlobClient();

Console.WriteLine("Creating container...");
var container = serviceClient.GetContainerReference("templates");
container.CreateIfNotExistsAsync().Wait();
var containerPermissions = new BlobContainerPermissions()
    { PublicAccess = BlobContainerPublicAccessType.Container };
container.SetPermissionsAsync(containerPermissions).Wait();

Console.WriteLine("Uploading template file...");
var templateblob = container.GetBlockBlobReference("CreateVMTemplate.json");
templateblob.UploadFromFileAsync("../..\\CreateVMTemplate.json").Result();

Console.WriteLine("Uploading parameters file...");
var paramblob = container.GetBlockBlobReference("Parameters.json");
paramblob.UploadFromFileAsync("../..\\Parameters.json").Result();
```

Deploy the template

Deploy the template and parameters from the storage account that was created.

To deploy the template, add this code to the Main method:

```
var templatePath = "https://" + storageAccountName + ".blob.core.windows.net/templates/CreateVMTemplate.json";
var paramPath = "https://" + storageAccountName + ".blob.core.windows.net/templates/Parameters.json";
var deployment = azure.Deployments.Define("myDeployment")
    .WithExistingResourceGroup(groupName)
    .WithTemplateLink(templatePath, "1.0.0.0")
    .WithParametersLink(paramPath, "1.0.0.0")
    .WithMode(Microsoft.Azure.Management.ResourceManager.Fluent.Models.DeploymentMode.Incremental)
    .Create();
Console.WriteLine("Press enter to delete the resource group...");
Console.ReadLine();
```

Delete the resources

Because you are charged for resources used in Azure, it is always good practice to delete resources that are no longer needed. You don't need to delete each resource separately from a resource group. Delete the resource group and all its resources are automatically deleted.

To delete the resource group, add this code to the Main method:

```
azure.ResourceGroups.DeleteByName(groupName);
```

Run the application

It should take about five minutes for this console application to run completely from start to finish.

1. To run the console application, click **Start**.
2. Before you press **Enter** to start deleting resources, you could take a few minutes to verify the creation of the resources in the Azure portal. Click the deployment status to see information about the deployment.

Next steps

- If there were issues with the deployment, a next step would be to look at [Troubleshoot common Azure deployment errors with Azure Resource Manager](#).
- Learn how to deploy a virtual machine and its supporting resources by reviewing [Deploy an Azure Virtual Machine Using C#](#).

Quickstart - Configure a Windows virtual machine in Azure using Chef

2/24/2020 • 7 minutes to read • [Edit Online](#)

Chef enables you to deliver automation and desired state configurations.

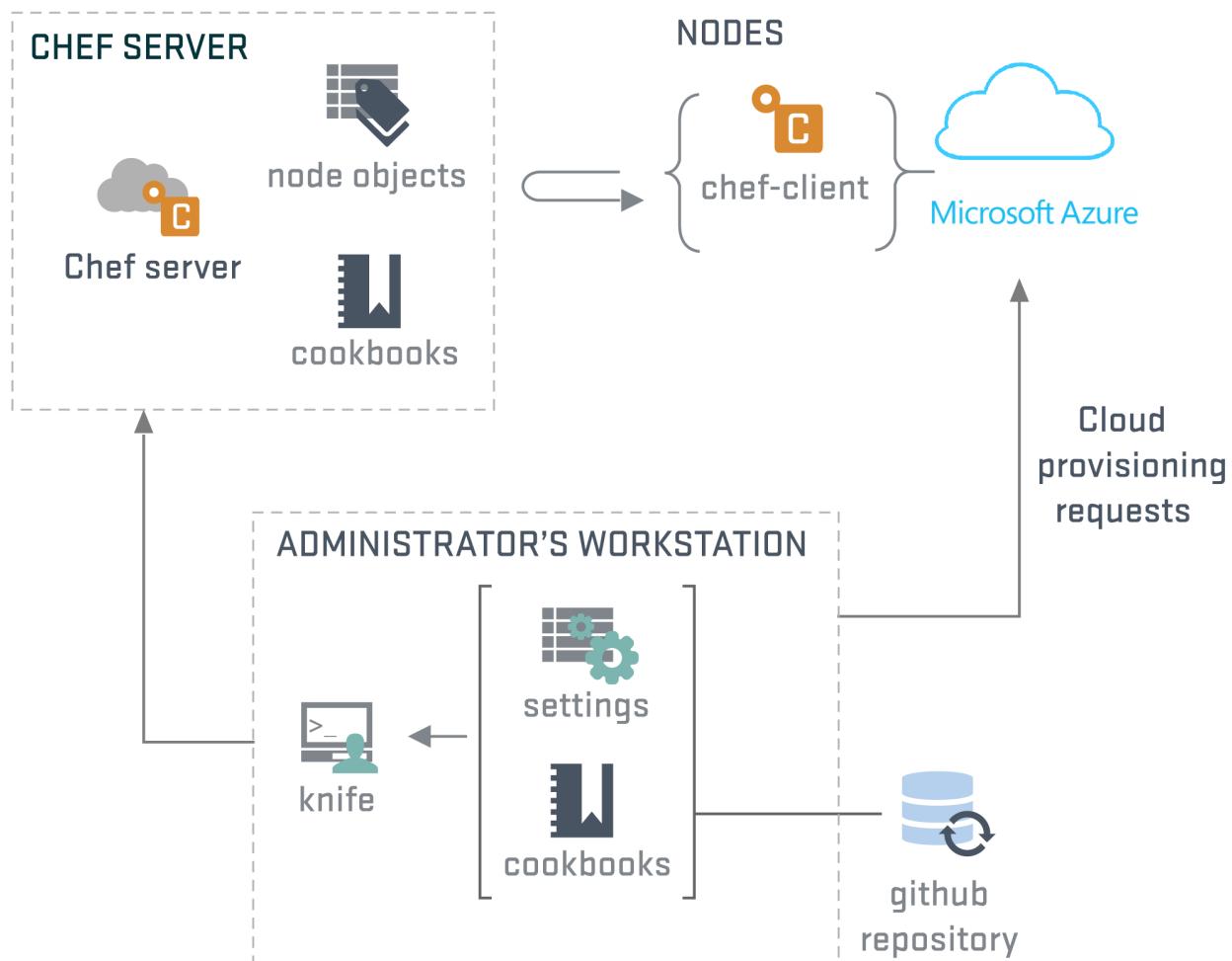
With the latest cloud API release, Chef provides seamless integration with Azure, giving you the ability to provision and deploy configuration states through a single command.

In this article, you set up your Chef environment to provision Azure virtual machines and walk through creating a policy or cookbook and then deploying this cookbook to an Azure virtual machine.

Chef basics

Before you begin with this article, [review the basic concepts of Chef](#).

The following diagram shows the high-level Chef architecture.



Chef has three main architectural components:

- Chef Server - The management point and there are two options for the Chef Server: a hosted solution or an on-premises solution.

- Chef Client (node) - The agent that sits on the servers you are managing.
- Chef Workstation - The name for both the admin workstation (where you create policies and run management commands) and the software package of Chef tools.

Generally, you see **your workstation** as the location where you run commands and **Chef Workstation** for the software package.

For example, you download the knife command as part of the **Chef Workstation**, but you run knife commands from **your workstation** to manage infrastructure.

Chef also uses the concepts of *cookbooks* and *recipes*. These terms are the policies that are defined and applied to the servers, respectively.

Preparing your workstation

First, prep your workstation by creating a directory to store Chef configuration files and cookbooks.

Create a directory named `c:\Chef`.

Download and install the latest [Azure CLI](#) version on to your workstation.

Configure Azure Service Principal

We'll be using a Service Principal to help us create Azure resources from our Chef Workstation. To create the relevant Service Principal with the required permissions, run the following commands within PowerShell:

```
Login-AzureRmAccount  
Get-AzureRmSubscription  
Select-AzureRmSubscription -SubscriptionName "<yourSubscriptionName>"  
$myApplication = New-AzureRmADApplication -DisplayName "automation-app" -HomePage "https://chef-automation-test.com" -IdentifierUris "https://chef-automation-test.com" -Password "#1234p$wdchef19"  
New-AzureRmADServicePrincipal -ApplicationId $myApplication.ApplicationId  
New-AzureRmRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName $myApplication.ApplicationId
```

Take note of your SubscriptionID, TenantID, ClientID, and Client Secret (the password you set previously in this tutorial) as you will need these values.

Setup Chef Server

This guide assumes that you'll sign up for Hosted Chef.

If you're not already using a Chef Server, you can:

- Sign up for [Hosted Chef](#), which is the fastest way to get started with Chef.
- Install a standalone Chef Server on linux-based machine, following the [installation instructions](#) from [Chef Docs](#).

Creating a Hosted Chef account

Sign up for a Hosted Chef account [here](#).

During the sign-up process, you will be asked to create a new organization.

Create Organization



Full Name (example: Chef, Inc.)

Short Name (example: chef)

Short name is required

[Cancel](#) [Create Organization](#)

Once your organization is created, download the starter kit.

The screenshot shows the 'Create Organization' step in the Chef Manage interface. On the left, there's a sidebar with navigation links: 'Organizations' (selected), 'Create', 'Reset Validation Key', 'Generate Knife Config', 'Invite User', 'Leave Organization', and 'Starter Kit'. Below that are 'Users', 'Groups', and 'Global Permissions'. The main content area has a heading 'Thank you for choosing hosted Chef!' and sub-headings 'Follow these steps to be on your way to using hosted Chef', 'What's next?', 'Chef Documentation', and 'Browse Community Cookbooks'. It also includes a 'Download Starter Kit' button and a 'Learn Chef' button.

NOTE

If you receive a prompt warning you that your keys will be reset, it's okay to proceed as we have no existing infrastructure configured as yet.

This starter kit zip file contains your organization configuration files and user key in the `.chef` directory.

The `organization-validator.pem` must be downloaded separately, because it's a private key and private keys should not be stored on the Chef Server. From [Chef Manage](#), go into the Administration section, and select "Reset Validation Key", which provides a file for you to download separately. Save the file to `c:\chef`.

Configuring your Chef workstation

Extract the content of the `chef-starter.zip` to `c:\chef`.

Copy all files under `chef-starter\chef-repo\chef` to your `c:\chef` directory.

Copy the `organization-validator.pem` file to `c:\chef`, if it's saved in `c:\Downloads`.

Your directory should now look something like the following example.

```
Directory: C:\Users\username\chef

Mode           LastWriteTime    Length Name
----          -----          ---- 
d----  12/6/2018  6:41 PM        .chef
d----  12/6/2018  5:40 PM        chef-repo
d----  12/6/2018  5:38 PM        cookbooks
d----  12/6/2018  5:38 PM        roles
-a---- 12/6/2018  5:38 PM       495 .gitignore
-a---- 12/6/2018  5:37 PM      1678 azuredocs-validator.pem
-a---- 12/6/2018  5:38 PM      1674 user.pem
-a---- 12/6/2018  5:53 PM       414 knife.rb
-a---- 12/6/2018  5:38 PM     2341 README.md
```

You should now have five files and four directories (including the empty chef-repo directory) in the root of `c:\chef`.

Edit knife.rb

The PEM files contain your organization and administrative private keys for communication and the knife.rb file contains your knife configuration. We will need to edit the knife.rb file.

Open the knife.rb file in the editor of your choice. The unaltered file should look something like:

```
current_dir = File.dirname(__FILE__)
log_level      :info
log_location   STDOUT
node_name      "mynode"
client_key     "#{current_dir}/user.pem"
chef_server_url "https://api.chef.io/organizations/myorg"
cookbook_path  ["#{current_dir}/cookbooks"]
```

Add the following information to your knife.rb, replacing the placeholders with your information:

```
validation_client_name  "myorg-validator"
validation_key          "#{current_dir}/myorg.pem"
knife[:azure_tenant_id] = "0000000-1111-aaaa-bbbb-222222222222"
knife[:azure_subscription_id] = "11111111-bbbb-cccc-1111-222222222222"
knife[:azure_client_id] = "11111111-bbbb-cccc-1111-222222222222"
knife[:azure_client_secret] = "#1234p$wdchef19"
```

These lines will ensure that Knife references the cookbooks directory under `c:\chef\cookbooks`.

Your `knife.rb` file should now look similar to the following example:

```
current_dir = File.dirname(__FILE__)
log_level      :info
log_location   STDOUT
node_name      "myorg"
client_key     "#{current_dir}/myorg.pem"
validation_client_name  "myorg-validator"
validation_key          "#{current_dir}/myorg-validator.pem"
chef_server_url "https://api.chef.io/organizations/myorg"
cookbook_path  ["#{current_dir}/..//cookbooks"]
knife[:azure_tenant_id] = "0000000-1111-aaaa-bbbb-222222222222"
knife[:azure_subscription_id] = "11111111-bbbb-cccc-1111-222222222222"
knife[:azure_client_id] = "0000000-1111-aaaa-bbbb-222222222222"
knife[:azure_client_secret] = "#1234p$wdchef19"
```

```
current_dir = File.dirname(__FILE__)
log_level :info
log_location STDOUT
node_name "myorg"
client_key "#{current_dir}/myorg.pem"
validation_client_name "myorg-validator"
validation_key "#{current_dir}/myorg-validator.pem"
chef_server_url "https://api.chef.io/organizations/myorg"
cookbook_path ["#{current_dir}/../cookbooks"]
knife[:azure_tenant_id] = "000000-1111-aaaa-bbbb-222222222222"
knife[:azure_subscription_id] = "11111111-bbbb-cccc-1111-222222222222"
knife[:azure_client_id] = "11111111-bbbb-cccc-1111-222222222222"
knife[:azure_client_secret] = "#1234p$wdchef19"
```

Install Chef Workstation

Next, [download, and install the Chef Workstation](#).

Install Chef Workstation to the default location.

On the desktop, you'll see a CW PowerShell. This tool is used to interact with Chef products. The CW PowerShell makes new commands available, such as `chef-run` and Chef CLI commands (such as `chef`). See your installed version of Chef Workstation and the Chef tools with `chef -v`. You can also check your Workstation version by selecting **About Chef Workstation** from the Chef Workstation App.

`chef --version` should return something like:

```
Chef Workstation: 0.4.2
chef-run: 0.3.0
chef-client: 15.0.300
delivery-cli: 0.0.52 (9d07501a3b347cc687c902319d23dc32dd5fa621)
berks: 7.0.8
test-kitchen: 2.2.5
inspec: 4.3.2
```

NOTE

The order of the path is important! If your opcode paths are not in the correct order, problems will result.

Reboot your workstation before you continue.

Install Knife Azure

This tutorial assumes that you're using the Azure Resource Manager to interact with your virtual machine.

Install the Knife Azure extension, which includes the Azure Plugin.

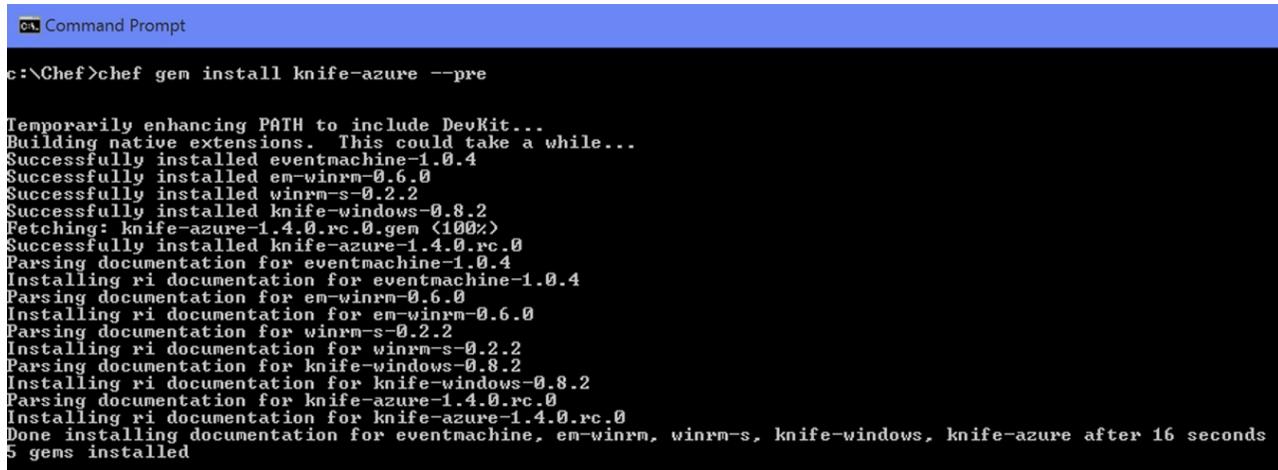
Run the following command.

```
chef gem install knife-azure --pre
```

NOTE

The `--pre` argument ensures you are receiving the latest RC version of the Knife Azure Plugin which provides access to the latest set of APIs.

It's likely that a number of dependencies will also be installed at the same time.



```
c:\Chef>chef gem install knife-azure --pre

Temporarily enhancing PATH to include DevKit...
Building native extensions. This could take a while...
Successfully installed eventmachine-1.0.4
Successfully installed em-winrm-0.6.0
Successfully installed winrm-s-0.2.2
Successfully installed knife-windows-0.8.2
Fetching: knife-azure-1.4.0.rc.0.gem (100%)
Successfully installed knife-azure-1.4.0.rc.0
Parsing documentation for eventmachine-1.0.4
Installing ri documentation for eventmachine-1.0.4
Parsing documentation for em-winrm-0.6.0
Installing ri documentation for em-winrm-0.6.0
Parsing documentation for winrm-s-0.2.2
Installing ri documentation for winrm-s-0.2.2
Parsing documentation for knife-windows-0.8.2
Installing ri documentation for knife-windows-0.8.2
Parsing documentation for knife-azure-1.4.0.rc.0
Installing ri documentation for knife-azure-1.4.0.rc.0
Done installing documentation for eventmachine, em-winrm, winrm-s, knife-windows, knife-azure after 16 seconds
5 gems installed
```

To ensure everything is configured correctly, run the following command.

```
knife azurerm server list
```

If everything is configured correctly, you will see a list of available Azure images scroll through.

Congratulations. Your workstation is set up!

Creating a cookbook

A cookbook is used by Chef to define a set of commands that you wish to run on your managed client. Creating a cookbook is straightforward, just use the `chef generate cookbook` command to generate the cookbook template. This cookbook is for a web server that automatically deploys IIS.

Under your `C:\Chef directory`, run the following command.

```
chef generate cookbook webserver
```

This command generates a set of files under the directory `C:\Chef\cookbooks\webserver`. Next, define the set of commands for the Chef client to run on the managed virtual machine.

The commands are stored in the file `default.rb`. In this file, define a set of commands that installs IIS, starts IIS, and copies a template file to the `wwwroot` folder.

Modify the `C:\Chef\cookbooks\webserver\recipes\default.rb` file and add the following lines.

```
powershell_script 'Install IIS' do
  action :run
  code 'add-windowsfeature Web-Server'
end

service 'w3svc' do
  action [ :enable, :start ]
end

template 'c:\inetpub\wwwroot\Default.htm' do
  source 'Default.htm.erb'
  rights :read, 'Everyone'
end
```

Save the file once you are done.

Creating a template

In this step, you'll generate a template file to use as the `default.html` page.

Run the following command to generate the template:

```
chef generate template webserver Default.htm
```

Navigate to the `c:\chef\cookbooks\webserver\templates\default\Default.htm.erb` file. Edit the file by adding some simple *Hello World* HTML code, and then save the file.

Upload the cookbook to the Chef Server

In this step, you make a copy of the cookbook that you have created on the local machine and upload it to the Chef Hosted Server. Once uploaded, the cookbook appears under the **Policy** tab.

```
knife cookbook upload webserver
```

Cookbook	Current Version
webserver	0.1.2

Deploy a virtual machine with Knife Azure

Deploy an Azure virtual machine and apply the `Webserver` cookbook using the `knife` command.

The `knife` command will also install the IIS web service and default web page.

```
knife azurerm server create `  
--azure-resource-group-name rg-chefdeployment `  
--azure-storage-account store `  
--azure-vm-name chefvm `  
--azure-vm-size 'Standard_DS2_v2' `  
--azure-service-location 'westus' `  
--azure-image-reference-offer 'WindowsServer' `  
--azure-image-reference-publisher 'MicrosoftWindowsServer' `  
--azure-image-reference-sku '2016-Datacenter' `  
--azure-image-reference-version 'latest' `  
-x myuser -P myPassword123 `  
--tcp-endpoints '80,3389' `  
--chef-daemon-interval 1 `  
-r "recipe[webserver]"
```

The `knife` command example creates a *Standard_DS2_v2* virtual machine with Windows Server 2016 installed within the West US region. Modify these values to per your app needs.

After running the command, browse to the Azure portal to see your machine begin to provision.

DEPLOYMENT NAME	STATUS	LAST MODIFIED	DURATION	RELATED EVENTS
chefvm_deploy	Deploying	7/9/2019, 1:43:33 PM	2 minutes 18 seconds	Related events

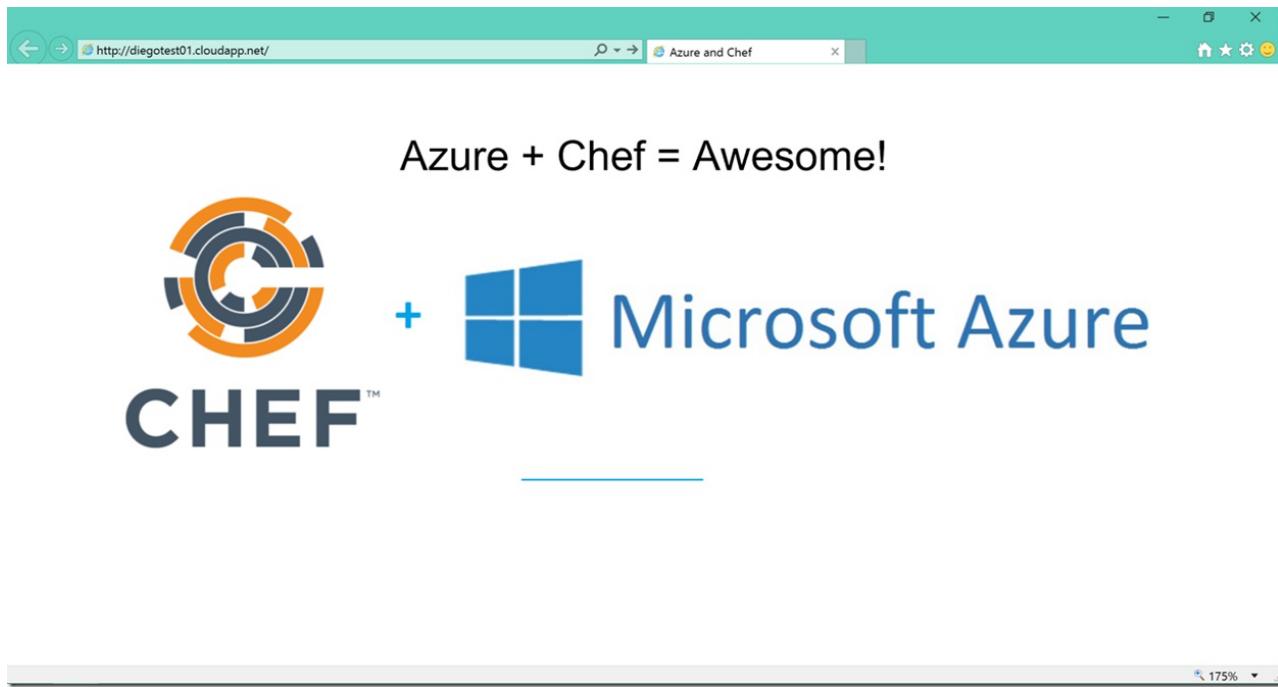
The command prompt appears next.

```

Administrator: Chef Workstation (sarah)
PS C:\chef> knife azurerm server create ` .
>>   --azure-resource-group-name rg-chefdeployment ` .
>>   --azure-storage-account 'vmstorage' ` .
>>   --azure-vm-name chefvm ` .
>>   --azure-vm-size 'Standard_DS2_v2' ` .
>>   --azure-service-location 'westus' ` .
>>   --azure-image-reference-offer 'WindowsServer' ` .
>>   --azure-image-reference-sku '2016-Datacenter' ` .
>>   --azure-image-reference-version 'latest' ` .
>>   -x myuser -P myPassword123 ` .
>>   --tcp-endpoints '80,3389' ` .
>>   --chef-daemon-interval 1 ` .
>>   -r "recipe[webserver]" ` .
Creating new client for chefvm
Creating new node for chefvm
Creating ResourceGroup....
Creating Virtual Machine....
Virtual Machine creation successful.
Deployment name is: chefvm_deploy
Deployment ID is: /subscriptions/05030f73-bb40-4f0e-867e-03e01a860992/resourceGroups/rg-chefdeployment/providers/Microsoft.Resources/deployments/chefvm_deploy
VM Details ...
-----
Virtual Machine name is: chefvm
Virtual Machine ID is: /subscriptions/05030f73-bb40-4f0e-867e-03e01a860992/resourceGroups/rg-chefdeployment/providers/Microsoft.Compute/virtualMachines/chefvm
Server Name           chefvm
Size                Standard_DS2_v2
Provisioning State  Succeeded
Location             westus
Publisher            MicrosoftWindowsServer
Offer               WindowsServer
Sku                 2016-Datacenter
Version              latest
OS Type             Windows
Public IP address   138.91.75.175
FQDN                chefvm.westus.cloudapp.azure.com
PS C:\chef> -

```

Once the deployment is complete, the public IP address of the new virtual machine is displayed. Paste this value into a web browser to view the new website. When we deployed the virtual machine, we opened port 80 so it should be available externally.



This example uses creative HTML code.

You can also view the node's status [Chef Manage](#).

A screenshot of the Chef Manage interface. The top navigation bar includes tabs for Nodes, Reports, Policy, and Administration. The Nodes tab is selected. A sub-menu on the left shows options like Delete, Manage Tags, Reset Key, Edit Run List, and Edit Attributes. The main content area is titled "Showing All Nodes" and contains a table with one row:

Node Name	Platform	FQDN	IP Address	Uptime	Last Check-In	Environment	Actions
chefvm	windows	chefvm	10.0.0.4	12 minutes	2 minutes ago	_default	

Don't forget you can also connect through an RDP session from the Azure portal via port 3389.

Next steps

[Chef on Azure](#)

Create and manage Windows VMs in Azure using Java

12/23/2019 • 7 minutes to read • [Edit Online](#)

An [Azure Virtual Machine](#) (VM) needs several supporting Azure resources. This article covers creating, managing, and deleting VM resources using Java. You learn how to:

- Create a Maven project
- Add dependencies
- Create credentials
- Create resources
- Perform management tasks
- Delete resources
- Run the application

It takes about 20 minutes to do these steps.

Create a Maven project

1. If you haven't already done so, install [Java](#).
2. Install [Maven](#).
3. Create a new folder and the project:

```
mkdir java-azure-test
cd java-azure-test

mvn archetype:generate -DgroupId=com.fabrikam -DartifactId=testAzureApp -DarchetypeArtifactId=maven-
archetype-quickstart -DinteractiveMode=false
```

Add dependencies

1. Under the `testAzureApp` folder, open the `pom.xml` file and add the build configuration to `<project>` to enable the building of your application:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>exec-maven-plugin</artifactId>
      <configuration>
        <mainClass>com.fabrikam.testAzureApp.App</mainClass>
      </configuration>
    </plugin>
  </plugins>
</build>
```

2. Add the dependencies that are needed to access the Azure Java SDK.

```

<dependency>
    <groupId>com.microsoft.azure</groupId>
    <artifactId>azure</artifactId>
    <version>1.1.0</version>
</dependency>
<dependency>
    <groupId>com.microsoft.azure</groupId>
    <artifactId>azure-mgmt-compute</artifactId>
    <version>1.1.0</version>
</dependency>
<dependency>
    <groupId>com.microsoft.azure</groupId>
    <artifactId>azure-mgmt-resources</artifactId>
    <version>1.1.0</version>
</dependency>
<dependency>
    <groupId>com.microsoft.azure</groupId>
    <artifactId>azure-mgmt-network</artifactId>
    <version>1.1.0</version>
</dependency>
<dependency>
    <groupId>com.squareup.okio</groupId>
    <artifactId>okio</artifactId>
    <version>1.13.0</version>
</dependency>
<dependency>
    <groupId>com.nimbusds</groupId>
    <artifactId>nimbus-jose-jwt</artifactId>
    <version>3.6</version>
</dependency>
<dependency>
    <groupId>net.minidev</groupId>
    <artifactId>json-smart</artifactId>
    <version>1.0.6.3</version>
</dependency>
<dependency>
    <groupId>javax.mail</groupId>
    <artifactId>mail</artifactId>
    <version>1.4.5</version>
</dependency>

```

- Save the file.

Create credentials

Before you start this step, make sure that you have access to an [Active Directory service principal](#). You should also record the application ID, the authentication key, and the tenant ID that you need in a later step.

Create the authorization file

- Create a file named `azureauth.properties` and add these properties to it:

```

subscription=<subscription-id>
client=<application-id>
key=<authentication-key>
tenant=<tenant-id>
managementURI=https://management.core.windows.net/
baseURL=https://management.azure.com/
authURL=https://login.windows.net/
graphURL=https://graph.windows.net/

```

Replace **<subscription-id>** with your subscription identifier, **<application-id>** with the Active Directory application identifier, **<authentication-key>** with the application key, and **<tenant-id>** with the tenant

identifier.

2. Save the file.
3. Set an environment variable named AZURE_AUTH_LOCATION in your shell with the full path to the authentication file.

Create the management client

1. Open the `App.java` file under `src\main\java\com\fabrikam` and make sure this package statement is at the top:

```
package com.fabrikam.testAzureApp;
```

2. Under the package statement, add these import statements:

```
import com.microsoft.azure.management.Azure;
import com.microsoft.azure.management.compute.AvailabilitySet;
import com.microsoft.azure.management.compute.AvailabilitySetSkuTypes;
import com.microsoft.azure.management.compute.CachingTypes;
import com.microsoft.azure.management.compute.InstanceViewStatus;
import com.microsoft.azure.management.compute.DiskInstanceView;
import com.microsoft.azure.management.compute.VirtualMachine;
import com.microsoft.azure.management.compute.VirtualMachineSizeTypes;
import com.microsoft.azure.management.network.PublicIPAddress;
import com.microsoft.azure.management.network.Network;
import com.microsoft.azure.management.network.NetworkInterface;
import com.microsoft.azure.management.resources.ResourceGroup;
import com.microsoft.azure.management.resources.fluentcore.arm.Region;
import com.microsoft.azure.management.resources.fluentcore.model.Creatable;
import com.microsoft.rest.LogLevel;
import java.io.File;
import java.util.Scanner;
```

3. To create the Active Directory credentials that you need to make requests, add this code to the main method of the App class:

```
try {
    final File credFile = new File(System.getenv("AZURE_AUTH_LOCATION"));
    Azure azure = Azure.configure()
        .withLogLevel(LogLevel.BASIC)
        .authenticate(credFile)
        .withDefaultSubscription();
} catch (Exception e) {
    System.out.println(e.getMessage());
    e.printStackTrace();
}
```

Create resources

Create the resource group

All resources must be contained in a [Resource group](#).

To specify values for the application and create the resource group, add this code to the try block in the main method:

```
System.out.println("Creating resource group...");
ResourceGroup resourceGroup = azure.resourceGroups()
    .define("myResourceGroup")
    .withRegion(Region.US_EAST)
    .create();
```

Create the availability set

[Availability sets](#) make it easier for you to maintain the virtual machines used by your application.

To create the availability set, add this code to the try block in the main method:

```
System.out.println("Creating availability set...");
AvailabilitySet availabilitySet = azure.availabilitySets()
    .define("myAvailabilitySet")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withSku(AvailabilitySetSkuTypes.MANAGED)
    .create();
```

Create the public IP address

A [Public IP address](#) is needed to communicate with the virtual machine.

To create the public IP address for the virtual machine, add this code to the try block in the main method:

```
System.out.println("Creating public IP address...");
PublicIPAddress publicIPAddress = azure.publicIPAddresses()
    .define("myPublicIP")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withDynamicIP()
    .create();
```

Create the virtual network

A virtual machine must be in a subnet of a [Virtual network](#).

To create a subnet and a virtual network, add this code to the try block in the main method:

```
System.out.println("Creating virtual network...");
Network network = azure.networks()
    .define("myVN")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withAddressSpace("10.0.0.0/16")
    .withSubnet("mySubnet", "10.0.0.0/24")
    .create();
```

Create the network interface

A virtual machine needs a network interface to communicate on the virtual network.

To create a network interface, add this code to the try block in the main method:

```

System.out.println("Creating network interface...");
NetworkInterface networkInterface = azure.networkInterfaces()
    .define("myNIC")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetwork(network)
    .withSubnet("mySubnet")
    .withPrimaryPrivateIPAddressDynamic()
    .withExistingPrimaryPublicIPAddress(publicIPAddress)
    .create();

```

Create the virtual machine

Now that you created all the supporting resources, you can create a virtual machine.

To create the virtual machine, add this code to the try block in the main method:

```

System.out.println("Creating virtual machine...");
VirtualMachine virtualMachine = azure.virtualMachines()
    .define("myVM")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetworkInterface(networkInterface)
    .withLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer", "2012-R2-Datacenter")
    .withAdminUsername("azureuser")
    .withAdminPassword("Azure12345678")
    .withComputerName("myVM")
    .withExistingAvailabilitySet(availabilitySet)
    .withSize("Standard_DS1")
    .create();

Scanner input = new Scanner(System.in);
System.out.println("Press enter to get information about the VM...");
input.nextLine();

```

NOTE

This tutorial creates a virtual machine running a version of the Windows Server operating system. To learn more about selecting other images, see [Navigate and select Azure virtual machine images with Windows PowerShell and the Azure CLI](#).

If you want to use an existing disk instead of a marketplace image, use this code:

```

ManagedDisk managedDisk = azure.disks.define("myosdisk")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withWindowsFromVhd("https://mystorage.blob.core.windows.net/vhds/myosdisk.vhd")
    .withSizeInGB(128)
    .withSku(DiskSkuTypes.PremiumLRS)
    .create();

azure.virtualMachines.define("myVM")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetworkInterface(networkInterface)
    .withSpecializedOSDisk(managedDisk, OperatingSystemTypes.Windows)
    .withExistingAvailabilitySet(availabilitySet)
    .withSize(VirtualMachineSizeTypes.StandardDS1)
    .create();

```

Perform management tasks

During the lifecycle of a virtual machine, you may want to run management tasks such as starting, stopping, or deleting a virtual machine. Additionally, you may want to create code to automate repetitive or complex tasks.

When you need to do anything with the VM, you need to get an instance of it. Add this code to the try block of the main method:

```
VirtualMachine vm = azure.virtualMachines().getByResourceGroup("myResourceGroup", "myVM");
```

Get information about the VM

To get information about the virtual machine, add this code to the try block in the main method:

```
System.out.println("hardwareProfile");
System.out.println("    vmSize: " + vm.size());
System.out.println("storageProfile");
System.out.println("    imageReference");
System.out.println("        publisher: " + vm.storageProfile().imageReference().publisher());
System.out.println("        offer: " + vm.storageProfile().imageReference().offer());
System.out.println("        sku: " + vm.storageProfile().imageReference().sku());
System.out.println("        version: " + vm.storageProfile().imageReference().version());
System.out.println("    osDisk");
System.out.println("        osType: " + vm.storageProfile().osDisk().osType());
System.out.println("        name: " + vm.storageProfile().osDisk().name());
System.out.println("        createOption: " + vm.storageProfile().osDisk().createOption());
System.out.println("        caching: " + vm.storageProfile().osDisk().caching());
System.out.println("osProfile");
System.out.println("    computerName: " + vm.osProfile().computerName());
System.out.println("    adminUserName: " + vm.osProfile().adminUsername());
System.out.println("    provisionVMAgent: " + vm.osProfile().windowsConfiguration().provisionVMAgent());
System.out.println("    enableAutomaticUpdates: " +
vm.osProfile().windowsConfiguration().enableAutomaticUpdates());
System.out.println("networkProfile");
System.out.println("    networkInterface: " + vm.primaryNetworkInterfaceId());
System.out.println("vmAgent");
System.out.println("    vmAgentVersion: " + vm.instanceView().vmAgent().vmAgentVersion());
System.out.println("    statuses");
for(InstanceViewStatus status : vm.instanceView().vmAgent().statuses()) {
    System.out.println("        code: " + status.code());
    System.out.println("        displayStatus: " + status.displayStatus());
    System.out.println("        message: " + status.message());
    System.out.println("        time: " + status.time());
}
System.out.println("disks");
for(DiskInstanceView disk : vm.instanceView().disks()) {
    System.out.println("    name: " + disk.name());
    System.out.println("    statuses");
    for(InstanceViewStatus status : disk.statuses()) {
        System.out.println("        code: " + status.code());
        System.out.println("        displayStatus: " + status.displayStatus());
        System.out.println("        time: " + status.time());
    }
}
System.out.println("VM general status");
System.out.println("    provisioningStatus: " + vm.provisioningState());
System.out.println("    id: " + vm.id());
System.out.println("    name: " + vm.name());
System.out.println("    type: " + vm.type());
System.out.println("VM instance status");
for(InstanceViewStatus status : vm.instanceView().statuses()) {
    System.out.println("        code: " + status.code());
    System.out.println("        displayStatus: " + status.displayStatus());
}
System.out.println("Press enter to continue...");
input.nextLine();
```

Stop the VM

You can stop a virtual machine and keep all its settings, but continue to be charged for it, or you can stop a virtual machine and deallocate it. When a virtual machine is deallocated, all resources associated with it are also deallocated and billing ends for it.

To stop the virtual machine without deallocating it, add this code to the try block in the main method:

```
System.out.println("Stopping vm...");
vm.powerOff();
System.out.println("Press enter to continue...");
input.nextLine();
```

If you want to deallocate the virtual machine, change the PowerOff call to this code:

```
vm.deallocate();
```

Start the VM

To start the virtual machine, add this code to the try block in the main method:

```
System.out.println("Starting vm...");
vm.start();
System.out.println("Press enter to continue...");
input.nextLine();
```

Resize the VM

Many aspects of deployment should be considered when deciding on a size for your virtual machine. For more information, see [VM sizes](#).

To change size of the virtual machine, add this code to the try block in the main method:

```
System.out.println("Resizing vm...");
vm.update()
    .withSize(VirtualMachineSizeTypes.STANDARD_DS2)
    .apply();
System.out.println("Press enter to continue...");
input.nextLine();
```

Add a data disk to the VM

To add a data disk to the virtual machine that is 2 GB in size, has a LUN of 0, and a caching type of ReadWrite, add this code to the try block in the main method:

```
System.out.println("Adding data disk...");
vm.update()
    .withNewDataDisk(2, 0, CachingTypes.READ_WRITE)
    .apply();
System.out.println("Press enter to delete resources...");
input.nextLine();
```

Delete resources

Because you are charged for resources used in Azure, it is always good practice to delete resources that are no longer needed. If you want to delete the virtual machines and all the supporting resources, all you have to do is delete the resource group.

1. To delete the resource group, add this code to the try block in the main method:

```
System.out.println("Deleting resources...");  
azure.resourceGroups().deleteByName("myResourceGroup");
```

2. Save the App.java file.

Run the application

It should take about five minutes for this console application to run completely from start to finish.

1. To run the application, use this Maven command:

```
mvn compile exec:java
```

2. Before you press **Enter** to start deleting resources, you could take a few minutes to verify the creation of the resources in the Azure portal. Click the deployment status to see information about the deployment.

Next steps

- Learn more about using the [Azure libraries for Java](#).

Create and manage Windows VMs in Azure using Python

12/23/2019 • 10 minutes to read • [Edit Online](#)

An [Azure Virtual Machine](#) (VM) needs several supporting Azure resources. This article covers creating, managing, and deleting VM resources using Python. You learn how to:

- Create a Visual Studio project
- Install packages
- Create credentials
- Create resources
- Perform management tasks
- Delete resources
- Run the application

It takes about 20 minutes to do these steps.

Create a Visual Studio project

1. If you haven't already, install [Visual Studio](#). Select **Python development** on the Workloads page, and then click **Install**. In the summary, you can see that **Python 3 64-bit (3.6.0)** is automatically selected for you. If you have already installed Visual Studio, you can add the Python workload using the Visual Studio Launcher.
2. After installing and starting Visual Studio, click **File > New > Project**.
3. Click **Templates > Python > Python Application**, enter *myPythonProject* for the name of the project, select the location of the project, and then click **OK**.

Install packages

1. In Solution Explorer, under *myPythonProject*, right-click **Python Environments**, and then select **Add virtual environment**.
2. On the Add Virtual Environment screen, accept the default name of *env*, make sure that *Python 3.6 (64-bit)* is selected for the base interpreter, and then click **Create**.
3. Right-click the *env* environment that you created, click **Install Python Package**, enter *azure* in the search box, and then press Enter.

You should see in the output windows that the azure packages were successfully installed.

Create credentials

Before you start this step, make sure that you have an [Active Directory service principal](#). You should also record the application ID, the authentication key, and the tenant ID that you need in a later step.

1. Open *myPythonProject.py* file that was created, and then add this code to enable your application to run:

```
if __name__ == "__main__":
```

2. To import the code that is needed, add these statements to the top of the .py file:

```
from azure.common.credentials import ServicePrincipalCredentials
from azure.mgmt.resource import ResourceManagementClient
from azure.mgmt.compute import ComputeManagementClient
from azure.mgmt.network import NetworkManagementClient
from azure.mgmt.compute.models import DiskCreateOption
```

3. Next in the .py file, add variables after the import statements to specify common values used in the code:

```
SUBSCRIPTION_ID = 'subscription-id'
GROUP_NAME = 'myResourceGroup'
LOCATION = 'westus'
VM_NAME = 'myVM'
```

Replace **subscription-id** with your subscription identifier.

4. To create the Active Directory credentials that you need to make requests, add this function after the variables in the .py file:

```
def get_credentials():
    credentials = ServicePrincipalCredentials(
        client_id = 'application-id',
        secret = 'authentication-key',
        tenant = 'tenant-id'
    )

    return credentials
```

Replace **application-id**, **authentication-key**, and **tenant-id** with the values that you previously collected when you created your Azure Active Directory service principal.

5. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
credentials = get_credentials()
```

Create resources

Initialize management clients

Management clients are needed to create and manage resources using the Python SDK in Azure. To create the management clients, add this code under the **if** statement at then end of the .py file:

```
resource_group_client = ResourceManagementClient(
    credentials,
    SUBSCRIPTION_ID
)
network_client = NetworkManagementClient(
    credentials,
    SUBSCRIPTION_ID
)
compute_client = ComputeManagementClient(
    credentials,
    SUBSCRIPTION_ID
)
```

Create the VM and supporting resources

All resources must be contained in a [Resource group](#).

1. To create a resource group, add this function after the variables in the .py file:

```
def create_resource_group(resource_group_client):
    resource_group_params = { 'location':LOCATION }
    resource_group_result = resource_group_client.resource_groups.create_or_update(
        GROUP_NAME,
        resource_group_params
    )
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
create_resource_group(resource_group_client)
input('Resource group created. Press enter to continue...')
```

A [Availability sets](#) make it easier for you to maintain the virtual machines used by your application.

1. To create an availability set, add this function after the variables in the .py file:

```
def create_availability_set(compute_client):
    avset_params = {
        'location': LOCATION,
        'sku': { 'name': 'Aligned' },
        'platform_fault_domain_count': 3
    }
    availability_set_result = compute_client.availability_sets.create_or_update(
        GROUP_NAME,
        'myAVSet',
        avset_params
    )
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
create_availability_set(compute_client)
print("-----")
input('Availability set created. Press enter to continue...')
```

A [Public IP address](#) is needed to communicate with the virtual machine.

1. To create a public IP address for the virtual machine, add this function after the variables in the .py file:

```
def create_public_ip_address(network_client):
    public_ip_address_params = {
        'location': LOCATION,
        'public_ip_allocation_method': 'Dynamic'
    }
    creation_result = network_client.public_ip_addresses.create_or_update(
        GROUP_NAME,
        'myIPAddress',
        public_ip_address_params
    )

    return creation_result.result()
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
creation_result = create_public_ip_address(network_client)
print("-----")
print(creation_result)
input('Press enter to continue...')
```

A virtual machine must be in a subnet of a [Virtual network](#).

1. To create a virtual network, add this function after the variables in the .py file:

```
def create_vnet(network_client):
    vnet_params = {
        'location': LOCATION,
        'address_space': {
            'address_prefixes': ['10.0.0.0/16']
        }
    }
    creation_result = network_client.virtual_networks.create_or_update(
        GROUP_NAME,
        'myVNet',
        vnet_params
    )
    return creation_result.result()
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
creation_result = create_vnet(network_client)
print("-----")
print(creation_result)
input('Press enter to continue...')
```

3. To add a subnet to the virtual network, add this function after the variables in the .py file:

```
def create_subnet(network_client):
    subnet_params = {
        'address_prefix': '10.0.0.0/24'
    }
    creation_result = network_client.subnets.create_or_update(
        GROUP_NAME,
        'myVNet',
        'mySubnet',
        subnet_params
    )
    return creation_result.result()
```

4. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
creation_result = create_subnet(network_client)
print("-----")
print(creation_result)
input('Press enter to continue...')
```

A virtual machine needs a network interface to communicate on the virtual network.

1. To create a network interface, add this function after the variables in the .py file:

```

def create_nic(network_client):
    subnet_info = network_client.subnets.get(
        GROUP_NAME,
        'myVNet',
        'mySubnet'
    )
    publicIPAddress = network_client.public_ip_addresses.get(
        GROUP_NAME,
        'myIPAddress'
    )
    nic_params = {
        'location': LOCATION,
        'ip_configurations': [
            {
                'name': 'myIPConfig',
                'public_ip_address': publicIPAddress,
                'subnet': {
                    'id': subnet_info.id
                }
            }
        ]
    }
    creation_result = network_client.network_interfaces.create_or_update(
        GROUP_NAME,
        'myNic',
        nic_params
    )

    return creation_result.result()

```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```

creation_result = create_nic(network_client)
print("-----")
print(creation_result)
input('Press enter to continue...')

```

Now that you created all the supporting resources, you can create a virtual machine.

1. To create the virtual machine, add this function after the variables in the .py file:

```

def create_vm(network_client, compute_client):
    nic = network_client.network_interfaces.get(
        GROUP_NAME,
        'myNic'
    )
    avset = compute_client.availability_sets.get(
        GROUP_NAME,
        'myAVSet'
    )
    vm_parameters = {
        'location': LOCATION,
        'os_profile': {
            'computer_name': VM_NAME,
            'admin_username': 'azureuser',
            'admin_password': 'Azure12345678'
        },
        'hardware_profile': {
            'vm_size': 'Standard_DS1'
        },
        'storage_profile': {
            'image_reference': {
                'publisher': 'MicrosoftWindowsServer',
                'offer': 'WindowsServer',
                'sku': '2012-R2-Datacenter',
                'version': 'latest'
            }
        },
        'network_profile': {
            'network_interfaces': [
                {
                    'id': nic.id
                }
            ]
        },
        'availability_set': {
            'id': avset.id
        }
    }
    creation_result = compute_client.virtual_machines.create_or_update(
        GROUP_NAME,
        VM_NAME,
        vm_parameters
    )

    return creation_result.result()

```

NOTE

This tutorial creates a virtual machine running a version of the Windows Server operating system. To learn more about selecting other images, see [Navigate and select Azure virtual machine images with Windows PowerShell and the Azure CLI](#).

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```

creation_result = create_vm(network_client, compute_client)
print("-----")
print(creation_result)
input('Press enter to continue...')

```

Perform management tasks

During the lifecycle of a virtual machine, you may want to run management tasks such as starting, stopping, or deleting a virtual machine. Additionally, you may want to create code to automate repetitive or complex tasks.

Get information about the VM

1. To get information about the virtual machine, add this function after the variables in the .py file:

```
def get_vm(compute_client):
    vm = compute_client.virtual_machines.get(GROUP_NAME, VM_NAME, expand='instanceView')
    print("hardwareProfile")
    print("  vmSize: ", vm.hardware_profile.vm_size)
    print("\nstorageProfile")
    print("  imageReference")
    print("    publisher: ", vm.storage_profile.image_reference.publisher)
    print("    offer: ", vm.storage_profile.image_reference.offer)
    print("    sku: ", vm.storage_profile.image_reference.sku)
    print("    version: ", vm.storage_profile.image_reference.version)
    print("  osDisk")
    print("    osType: ", vm.storage_profile.os_disk.os_type.value)
    print("    name: ", vm.storage_profile.os_disk.name)
    print("    createOption: ", vm.storage_profile.os_disk.create_option.value)
    print("    caching: ", vm.storage_profile.os_disk.caching.value)
    print("\nosProfile")
    print("  computerName: ", vm.os_profile.computer_name)
    print("  adminUsername: ", vm.os_profile.admin_username)
    print("  provisionVMAgent: {0}".format(vm.os_profile.windows_configuration.provision_vm_agent))
    print("  enableAutomaticUpdates:
{0}".format(vm.os_profile.windows_configuration.enable_automatic_updates))
    print("\nnetworkProfile")
    for nic in vm.network_profile.network_interfaces:
        print("  networkInterface id: ", nic.id)
    print("\nvmAgent")
    print("  vmAgentVersion", vm.instance_view.vm_agent.vm_agent_version)
    print("  statuses")
    for stat in vm_result.instance_view.vm_agent.statuses:
        print("    code: ", stat.code)
        print("    displayStatus: ", stat.display_status)
        print("    message: ", stat.message)
        print("    time: ", stat.time)
    print("\ndisks");
    for disk in vm.instance_view.disks:
        print("  name: ", disk.name)
        print("  statuses")
        for stat in disk.statuses:
            print("    code: ", stat.code)
            print("    displayStatus: ", stat.display_status)
            print("    time: ", stat.time)
    print("\nVM general status")
    print("  provisioningStatus: ", vm.provisioning_state)
    print("  id: ", vm.id)
    print("  name: ", vm.name)
    print("  type: ", vm.type)
    print("  location: ", vm.location)
    print("\nVM instance status")
    for stat in vm.instance_view.statuses:
        print("  code: ", stat.code)
        print("  displayStatus: ", stat.display_status)
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
get_vm(compute_client)
print("-----")
input('Press enter to continue...')
```

Stop the VM

You can stop a virtual machine and keep all its settings, but continue to be charged for it, or you can stop a virtual machine and deallocate it. When a virtual machine is deallocated, all resources associated with it are also

deallocated and billing ends for it.

1. To stop the virtual machine without deallocating it, add this function after the variables in the .py file:

```
def stop_vm(compute_client):
    compute_client.virtual_machines.power_off(GROUP_NAME, VM_NAME)
```

If you want to deallocate the virtual machine, change the power_off call to this code:

```
compute_client.virtual_machines.deallocate(GROUP_NAME, VM_NAME)
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
stop_vm(compute_client)
input('Press enter to continue...')
```

Start the VM

1. To start the virtual machine, add this function after the variables in the .py file:

```
def start_vm(compute_client):
    compute_client.virtual_machines.start(GROUP_NAME, VM_NAME)
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
start_vm(compute_client)
input('Press enter to continue...')
```

Resize the VM

Many aspects of deployment should be considered when deciding on a size for your virtual machine. For more information, see [VM sizes](#).

1. To change the size of the virtual machine, add this function after the variables in the .py file:

```
def update_vm(compute_client):
    vm = compute_client.virtual_machines.get(GROUP_NAME, VM_NAME)
    vm.hardware_profile.vm_size = 'Standard_DS3'
    update_result = compute_client.virtual_machines.create_or_update(
        GROUP_NAME,
        VM_NAME,
        vm
    )

    return update_result.result()
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
update_result = update_vm(compute_client)
print("-----")
print(update_result)
input('Press enter to continue...')
```

Add a data disk to the VM

Virtual machines can have one or more [data disks](#) that are stored as VHDs.

1. To add a data disk to the virtual machine, add this function after the variables in the .py file:

```
def add_datadisk(compute_client):
    disk_creation = compute_client.disks.create_or_update(
        GROUP_NAME,
        'myDataDisk1',
        {
            'location': LOCATION,
            'disk_size_gb': 1,
            'creation_data': {
                'create_option': DiskCreateOption.empty
            }
        }
    )
    data_disk = disk_creation.result()
    vm = compute_client.virtual_machines.get(GROUP_NAME, VM_NAME)
    add_result = vm.storage_profile.data_disks.append({
        'lun': 1,
        'name': 'myDataDisk1',
        'create_option': DiskCreateOption.attach,
        'managed_disk': {
            'id': data_disk.id
        }
    })
    add_result = compute_client.virtual_machines.create_or_update(
        GROUP_NAME,
        VM_NAME,
        vm)

return add_result.result()
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
add_result = add_datadisk(compute_client)
print("-----")
print(add_result)
input('Press enter to continue...')
```

Delete resources

Because you are charged for resources used in Azure, it's always a good practice to delete resources that are no longer needed. If you want to delete the virtual machines and all the supporting resources, all you have to do is delete the resource group.

1. To delete the resource group and all resources, add this function after the variables in the .py file:

```
def delete_resources(resource_group_client):
    resource_group_client.resource_groups.delete(GROUP_NAME)
```

2. To call the function that you previously added, add this code under the **if** statement at the end of the .py file:

```
delete_resources(resource_group_client)
```

3. Save *myPythonProject.py*.

Run the application

1. To run the console application, click **Start** in Visual Studio.

2. Press **Enter** after the status of each resource is returned. In the status information, you should see a **Succeeded** provisioning state. After the virtual machine is created, you have the opportunity to delete all the resources that you create. Before you press **Enter** to start deleting resources, you could take a few minutes to verify their creation in the Azure portal. If you have the Azure portal open, you might have to refresh the blade to see new resources.

It should take about five minutes for this console application to run completely from start to finish. It may take several minutes after the application has finished before all the resources and the resource group are deleted.

Next steps

- If there were issues with the deployment, a next step would be to look at [Troubleshooting resource group deployments with Azure portal](#)
- Learn more about the [Azure Python Library](#)

Create a Windows virtual machine from a Resource Manager template

11/13/2019 • 4 minutes to read • [Edit Online](#)

Learn how to create a Windows virtual machine by using an Azure Resource Manager template and Azure PowerShell from the Azure Cloud shell. The template used in this article deploys a single virtual machine running Windows Server in a new virtual network with a single subnet. For creating a Linux virtual machine, see [How to create a Linux virtual machine with Azure Resource Manager templates](#).

Create a virtual machine

Creating an Azure virtual machine usually includes two steps:

- Create a resource group. An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine.
- Create a virtual machine.

The following example creates a VM from an [Azure Quickstart template](#). Here is a copy of the template:

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "adminUsername": {  
      "type": "string",  
      "metadata": {  
        "description": "Username for the Virtual Machine."  
      }  
    },  
    "adminPassword": {  
      "type": "securestring",  
      "metadata": {  
        "description": "Password for the Virtual Machine."  
      }  
    },  
    "dnsLabelPrefix": {  
      "type": "string",  
      "metadata": {  
        "description": "Unique DNS Name for the Public IP used to access the Virtual Machine."  
      }  
    },  
    "windowsOSVersion": {  
      "type": "string",  
      "defaultValue": "2016-Datacenter",  
      "allowedValues": [  
        "2008-R2-SP1",  
        "2012-Datacenter",  
        "2012-R2-Datacenter",  
        "2016-Nano-Server",  
        "2016-Datacenter-with-Containers",  
        "2016-Datacenter",  
        "2019-Datacenter"  
      ],  
      "metadata": {  
        "description": "The Windows version for the VM. This will pick a fully patched image of this given  
        Windows version."  
      }  
    },  
  },  
  "resources": [  
    {  
      "type": "Microsoft.Compute/virtualMachines",  
      "name": "[parameters('dnsLabelPrefix')]",  
      "apiVersion": "2018-06-01",  
      "location": "West US",  
      "properties": {  
        "osProfile": {  
          "computerName": "[parameters('dnsLabelPrefix')]",  
          "adminUsername": "[parameters('adminUsername')]",  
          "adminPassword": "[parameters('adminPassword')]",  
          "osType": "Windows",  
          "osTypeHandlerVersion": "2018-06-01",  
          "windowsConfiguration": {  
            "version": "[parameters('windowsOSVersion')]"  
          }  
        },  
        "hardwareProfile": {  
          "vmSize": "Standard_DS1_v2"  
        },  
        "networkProfile": {  
          "networkInterfaces": [  
            {  
              "id": "[resourceId('Microsoft.Network/networkInterfaces', 'VM-NIC')]",  
              "primary": true,  
              "ipConfigurations": [  
                {  
                  "name": "VM-NIC",  
                  "subnet": {  
                    "id": "[resourceId('Microsoft.Network/subnets', 'Subnet-1')]"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "storageProfile": {  
          "imageReference": {  
            "id": "[resourceId('Microsoft.Compute/images', 'Windows-Image')]"  
          },  
          "osDisk": {  
            "caching": "None",  
            "createOption": "FromImage",  
            "name": "VM-OSSD",  
            "vhd": {  
              "uri": "[concat('https://', parameters('dnsLabelPrefix'), '.blob.core.windows.net/vhds/', parameters('dnsLabelPrefix'), '-os.vhd')]"  
            }  
          },  
          "dataDisks": [  
            {  
              "count": 1,  
              "lun": 0,  
              "name": "VM-DataSSD",  
              "sizeGB": 100,  
              "vhd": {  
                "uri": "[concat('https://', parameters('dnsLabelPrefix'), '.blob.core.windows.net/vhds/', parameters('dnsLabelPrefix'), '-data.vhd')]"  
              }  
            }  
          ]  
        }  
      }  
    }  
  ]  
}
```

```

"vmSize": {
    "type": "string",
    "defaultValue": "Standard_A2_v2",
    "metadata": {
        "description": "Size of the virtual machine."
    }
},
"location": {
    "type": "string",
    "defaultValue": "[resourceGroup().location]",
    "metadata": {
        "description": "Location for all resources."
    }
}
},
"variables": {
    "storageAccountName": "[concat(uniqueString(resourceGroup().id), 'sawinvm')]",
    "nicName": "myVMNic",
    "addressPrefix": "10.0.0.0/16",
    "subnetName": "Subnet",
    "subnetPrefix": "10.0.0.0/24",
    "publicIPAddressName": "myPublicIP",
    "vmName": "SimpleWinVM",
    "virtualNetworkName": "MyVNET",
    "subnetRef": "[resourceId('Microsoft.Network/virtualNetworks/subnets', variables('virtualNetworkName'), variables('subnetName'))]",
    "networkSecurityGroupName": "default-NSG"
},
"resources": [
    {
        "type": "Microsoft.Storage/storageAccounts",
        "apiVersion": "2018-11-01",
        "name": "[variables('storageAccountName')]",
        "location": "[parameters('location')]",
        "sku": {
            "name": "Standard_LRS"
        },
        "kind": "Storage",
        "properties": {}
    },
    {
        "type": "Microsoft.Network/publicIPAddresses",
        "apiVersion": "2018-11-01",
        "name": "[variables('publicIPAddressName')]",
        "location": "[parameters('location')]",
        "properties": {
            "publicIPAllocationMethod": "Dynamic",
            "dnsSettings": {
                "domainNameLabel": "[parameters('dnsLabelPrefix')]"
            }
        }
    },
    {
        "comments": "Default Network Security Group for template",
        "type": "Microsoft.Network/networkSecurityGroups",
        "apiVersion": "2019-08-01",
        "name": "[variables('networkSecurityGroupName')]",
        "location": "[parameters('location')]",
        "properties": {
            "securityRules": [
                {
                    "name": "default-allow-3389",
                    "properties": {
                        "priority": 1000,
                        "access": "Allow",
                        "direction": "Inbound",
                        "destinationPortRange": "3389",
                        "protocol": "Tcp",
                        "sourcePortRange": "*"
                    }
                }
            ]
        }
    }
]

```

```

        "sourceAddressPrefix": "*",
        "destinationAddressPrefix": "*"
    }
}
]
}
},
{
    "type": "Microsoft.Network/virtualNetworks",
    "apiVersion": "2018-11-01",
    "name": "[variables('virtualNetworkName')]",
    "location": "[parameters('location')]",
    "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups', variables('networkSecurityGroupName'))]"
    ],
    "properties": {
        "addressSpace": {
            "addressPrefixes": [
                "[variables('addressPrefix')]"
            ]
        },
        "subnets": [
            {
                "name": "[variables('subnetName')]",
                "properties": {
                    "addressPrefix": "[variables('subnetPrefix')]",
                    "networkSecurityGroup": {
                        "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
                    }
                }
            }
        ]
    }
},
{
    "type": "Microsoft.Network/networkInterfaces",
    "apiVersion": "2018-11-01",
    "name": "[variables('nicName')]",
    "location": "[parameters('location')]",
    "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses/', variables('publicIPPropertyName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks/', variables('virtualNetworkName'))]"
    ],
    "properties": {
        "ipConfigurations": [
            {
                "name": "ipconfig1",
                "properties": {
                    "privateIPAllocationMethod": "Dynamic",
                    "publicIPAddress": {
                        "id": "[resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPPropertyName'))]"
                    },
                    "subnet": {
                        "id": "[variables('subnetRef')]"
                    }
                }
            }
        ]
    }
},
{
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2018-10-01",
    "name": "[variables('vmName')]",
    "location": "[parameters('location')]",
    "dependsOn": [
        "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
        "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
    ]
}

```

```

    ],
    "properties": {
        "hardwareProfile": {
            "vmSize": "[parameters('vmSize')]"
        },
        "osProfile": {
            "computerName": "[variables('vmName')]",
            "adminUsername": "[parameters('adminUsername')]",
            "adminPassword": "[parameters('adminPassword')]"
        },
        "storageProfile": {
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "[parameters('windowsOSVersion')]",
                "version": "latest"
            },
            "osDisk": {
                "createOption": "FromImage"
            },
            "dataDisks": [
                {
                    "diskSizeGB": 1023,
                    "lun": 0,
                    "createOption": "Empty"
                }
            ]
        },
        "networkProfile": {
            "networkInterfaces": [
                {
                    "id": "[resourceId('Microsoft.Network/networkInterfaces',variables('nicName'))]"
                }
            ]
        },
        "diagnosticsProfile": {
            "bootDiagnostics": {
                "enabled": true,
                "storageUri": "[reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob]"
            }
        }
    }
},
"outputs": {
    "hostname": {
        "type": "string",
        "value": "[reference(variables('publicIPAddressName')).dnsSettings.fqdn]"
    }
}
}

```

To run the PowerShell script, Select **Try it** to open the Azure Cloud shell. To paste the script, right-click the shell, and then select **Paste**:

```

$resourceGroupName = Read-Host -Prompt "Enter the Resource Group name"
.setLocation = Read-Host -Prompt "Enter the location (i.e. centralus)"
$adminUsername = Read-Host -Prompt "Enter the administrator username"
$adminPassword = Read-Host -Prompt "Enter the administrator password" -AsSecureString
$dnsLabelPrefix = Read-Host -Prompt "Enter an unique DNS name for the public IP"

New-AzResourceGroup -Name $resourceGroupName -Location "$location"
New-AzResourceGroupDeployment ` 
    -ResourceGroupName $resourceGroupName ` 
    -TemplateUri "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-` 
windows/azuredeploy.json" ` 
    -adminUsername $adminUsername ` 
    -adminPassword $adminPassword ` 
    -dnsLabelPrefix $dnsLabelPrefix

(Get-AzVm -ResourceGroupName $resourceGroupName).name

```

If you choose to install and use the PowerShell locally instead of from the Azure Cloud shell, this tutorial requires the Azure PowerShell module. Run `Get-Module -ListAvailable Az` to find the version. If you need to upgrade, see [Install Azure PowerShell module](#). If you're running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

In the previous example, you specified a template stored in GitHub. You can also download or create a template and specify the local path with the `--template-file` parameter.

Here are some additional resources:

- To learn how to develop Resource Manager templates, see [Azure Resource Manager documentation](#).
- To see the Azure virtual machine schemas, see [Azure template reference](#).
- To see more virtual machine template samples, see [Azure Quickstart templates](#).

Connect to the virtual machine

The last PowerShell command from the previous script shows the virtual machine name. To connect to the virtual machine, see [How to connect and sign on to an Azure virtual machine running Windows](#).

Next Steps

- If there were issues with the deployment, you might take a look at [Troubleshoot common Azure deployment errors with Azure Resource Manager](#).
- Learn how to create and manage a virtual machine in [Create and manage Windows VMs with the Azure PowerShell module](#).

To learn more about creating templates, view the JSON syntax and properties for the resources types you deployed:

- [Microsoft.Network/publicIPAddresses](#)
- [Microsoft.Network/virtualNetworks](#)
- [Microsoft.Network/networkInterfaces](#)
- [Microsoft.Compute/virtualMachines](#)

How to connect and sign on to an Azure virtual machine running Windows

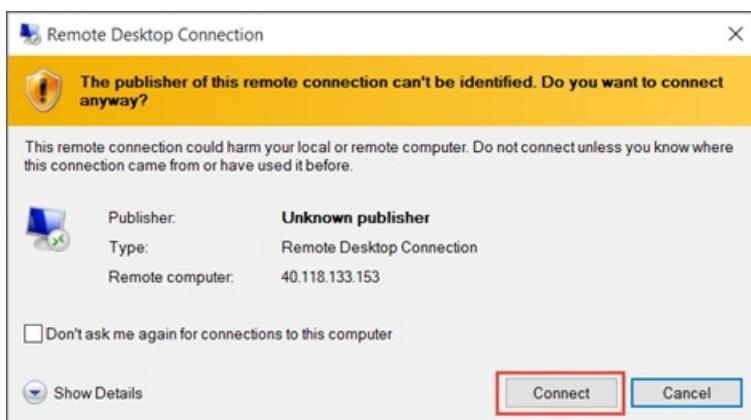
12/4/2019 • 2 minutes to read • [Edit Online](#)

You'll use the **Connect** button in the Azure portal to start a Remote Desktop (RDP) session from a Windows desktop. First you connect to the virtual machine, and then you sign on.

To connect to a Windows VM from a Mac, you will need to install an RDP client for Mac such as [Microsoft Remote Desktop](#).

Connect to the virtual machine

1. Go to the [Azure portal](#) to connect to a VM. Search for and select **Virtual machines**.
2. Select the virtual machine from the list.
3. At the beginning of the virtual machine page, select **Connect**.
4. On the **Connect to virtual machine** page, select **RDP**, and then select the appropriate **IP address** and **Port number**. In most cases, the default IP address and port should be used. Select **Download RDP File**. If the VM has a just-in-time policy set, you first need to select the **Request access** button to request access before you can download the RDP file. For more information about the just-in-time policy, see [Manage virtual machine access using the just in time policy](#).
5. Open the downloaded RDP file and select **Connect** when prompted. You will get a warning that the **.rdp** file is from an unknown publisher. This is expected. In the **Remote Desktop Connection** window, select **Connect** to continue.



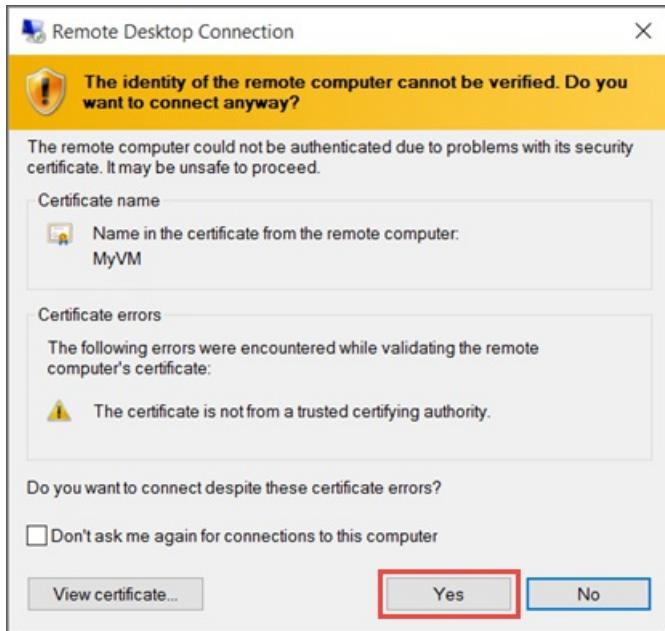
6. In the **Windows Security** window, select **More choices** and then **Use a different account**. Enter the credentials for an account on the virtual machine and then select **OK**.

Local account: This is usually the local account user name and password that you specified when you created the virtual machine. In this case, the domain is the name of the virtual machine and it is entered as `vmname\username`.

Domain joined VM: If the VM belongs to a domain, enter the user name in the format `Domain\Username`. The account also needs to either be in the Administrators group or have been granted remote access privileges to the VM.

Domain controller: If the VM is a domain controller, enter the user name and password of a domain administrator account for that domain.

7. Select **Yes** to verify the identity of the virtual machine and finish logging on.



TIP

If the **Connect** button in the portal is grayed-out and you are not connected to Azure via an [Express Route](#) or [Site-to-Site VPN](#) connection, you will need to create and assign your VM a public IP address before you can use RDP. For more information, see [Public IP addresses in Azure](#).

Connect to the virtual machine using PowerShell

If you are using PowerShell and have the Azure PowerShell module installed you may also connect using the `Get-AzRemoteDesktopFile` cmdlet, as shown below.

This example will immediately launch the RDP connection, taking you through similar prompts as above.

```
Get-AzRemoteDesktopFile -ResourceGroupName "RgName" -Name "VmName" -Launch
```

You may also save the RDP file for future use.

```
Get-AzRemoteDesktopFile -ResourceGroupName "RgName" -Name "VmName" -LocalPath "C:\Path\to\folder"
```

Next steps

If you have difficulty connecting, see [Troubleshoot Remote Desktop connections](#).

Azure Hybrid Benefit for Windows Server

2/28/2020 • 5 minutes to read • [Edit Online](#)

For customers with Software Assurance, Azure Hybrid Benefit for Windows Server allows you to use your on-premises Windows Server licenses and run Windows virtual machines on Azure at a reduced cost. You can use Azure Hybrid Benefit for Windows Server to deploy new virtual machines with Windows OS. This article goes over the steps on how to deploy new VMs with Azure Hybrid Benefit for Windows Server and how you can update existing running VMs. For more information about Azure Hybrid Benefit for Windows Server licensing and cost savings, see the [Azure Hybrid Benefit for Windows Server licensing page](#).

Each 2-processor license or each set of 16-core licenses are entitled to two instances of up to 8 cores, or one instance of up to 16 cores. The Azure Hybrid Benefit for Standard Edition licenses can only be used once either on-premises or in Azure. Datacenter Edition benefits allow for simultaneous usage both on-premises and in Azure.

Using Azure Hybrid Benefit for Windows Server with any VMs running Windows Server OS are now supported in all regions, including VMs with additional software such as SQL Server or third-party marketplace software.

Classic VMs

For classic VMs, only deploying new VM from on premises custom images is supported. To take advantage of the capabilities supported in this article, you must first migrate classic VMs to Resource Manager model.

IMPORTANT

Classic VMs will be retired on March 1, 2023.

If you use IaaS resources from ASM, please complete your migration by March 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.

For more information, see [Migrate your IaaS resources to Azure Resource Manager by March 1, 2023](#).

Ways to use Azure Hybrid Benefit for Windows Server

There are few ways to use Windows virtual machines with the Azure Hybrid Benefit:

1. You can deploy VMs from one of the provided Windows Server images on the Azure Marketplace
2. You can upload a custom VM and deploy using a Resource Manager template or Azure PowerShell
3. You can toggle and convert existing VM between running with Azure Hybrid Benefit or pay on-demand cost for Windows Server
4. You can also apply Azure Hybrid Benefit for Windows Server on virtual machine scale set as well

Create a VM with Azure Hybrid Benefit for Windows Server

All Windows Server OS based images are supported for Azure Hybrid Benefit for Windows Server. You can use Azure platform support images or upload your own custom Windows Server images.

Portal

To create a VM with Azure Hybrid Benefit for Windows Server, use the toggle under the "Save money" section.

PowerShell

```
New-AzVm ` 
    -ResourceGroupName "myResourceGroup" ` 
    -Name "myVM" ` 
    -Location "East US" ` 
    -ImageName "Win2016Datacenter" ` 
    -LicenseType "Windows_Server"
```

CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --location eastus \
    --license-type Windows_Server
```

Template

Within your Resource Manager templates, an additional parameter `licenseType` must be specified. You can read more about [authoring Azure Resource Manager templates](#)

```
"properties": { 
    "licenseType": "Windows_Server", 
    "hardwareProfile": { 
        "vmSize": "[variables('vmSize')]" 
    } 
}
```

Convert an existing VM using Azure Hybrid Benefit for Windows Server

If you have an existing VM that you would like to convert to take advantage of Azure Hybrid Benefit for Windows Server, you can update your VM's license type by following the instructions below.

NOTE

Changing the license type on the VM does not cause the system to reboot or cause a service interruption. It is simply an update to a metadata flag.

Portal

From portal VM blade, you can update the VM to use Azure Hybrid Benefit by selecting "Configuration" option and toggle the "Azure hybrid benefit" option

PowerShell

- Convert existing Windows Server VMs to Azure Hybrid Benefit for Windows Server

```
$vm = Get-AzVM -ResourceGroup "rg-name" -Name "vm-name"
$vm.LicenseType = "Windows_Server"
Update-AzVM -ResourceGroupName rg-name -VM $vm
```

- Convert Windows Server VMs with benefit back to pay-as-you-go

```
$vm = Get-AzVM -ResourceGroup "rg-name" -Name "vm-name"
$vm.LicenseType = "None"
Update-AzVM -ResourceGroupName rg-name -VM $vm
```

CLI

- Convert existing Windows Server VMs to Azure Hybrid Benefit for Windows Server

```
az vm update --resource-group myResourceGroup --name myVM --set licenseType=Windows_Server
```

How to verify your VM is utilizing the licensing benefit

Once you have deployed your VM through either PowerShell, Resource Manager template or portal, you can verify the setting in the following methods.

Portal

From portal VM blade, you can view the toggle for Azure Hybrid Benefit for Windows Server by selecting "Configuration" tab.

PowerShell

The following example shows the license type for a single VM

```
Get-AzVM -ResourceGroup "myResourceGroup" -Name "myVM"
```

Output:

```
Type          : Microsoft.Compute/virtualMachines
Location     : westus
LicenseType  : Windows_Server
```

This output contrasts with the following VM deployed without Azure Hybrid Benefit for Windows Server licensing:

```
Type          : Microsoft.Compute/virtualMachines
Location     : westus
LicenseType  :
```

CLI

```
az vm get-instance-view -g MyResourceGroup -n MyVM --query "[?licenseType=='Windows_Server']" -o table
```

NOTE

Changing the license type on the VM does not cause the system to reboot or cause a service interruption. It is a metadata licensing flag only.

List all VMs with Azure Hybrid Benefit for Windows Server in a subscription

To see and count all virtual machines deployed with Azure Hybrid Benefit for Windows Server, you can run the following command from your subscription:

Portal

From the Virtual Machine or Virtual machine scale sets resource blade, you can view a list of all your VM(s) and licensing type by configuring the table column to include "Azure Hybrid Benefit". The VM setting can either be in "Enabled", "Not enabled" or "Not supported" state.

PowerShell

```
$vms = Get-AzVM  
$vms | ?{$_.LicenseType -like "Windows_Server"} | select ResourceGroupName, Name, LicenseType
```

CLI

```
az vm list --query "[?licenseType=='Windows_Server']" -o table
```

Deploy a Virtual Machine Scale Set with Azure Hybrid Benefit for Windows Server

Within your virtual machine scale set Resource Manager templates, an additional parameter `licenseType` must be specified within your `VirtualMachineProfile` property. You can do this during create or update for your scale set through ARM template, PowerShell, Azure CLI or REST.

The following example uses ARM template with a Windows Server 2016 Datacenter image:

```
"virtualMachineProfile": {  
    "storageProfile": {  
        "osDisk": {  
            "createOption": "FromImage"  
        },  
        "imageReference": {  
            "publisher": "MicrosoftWindowsServer",  
            "offer": "WindowsServer",  
            "sku": "2016-Datacenter",  
            "version": "latest"  
        }  
    },  
    "licenseType": "Windows_Server",  
    "osProfile": {  
        "computerNamePrefix": "[parameters('vmssName')]",  
        "adminUsername": "[parameters('adminUsername')]",  
        "adminPassword": "[parameters('adminPassword')]"  
    }  
}
```

You can also learn more about how to [Modify a virtual machine scale set](#) for more ways to update your scale set.

Next steps

- Read more about [How to save money with the Azure Hybrid Benefit](#)
- Read more about [Frequently asked questions for Azure Hybrid Benefit](#)
- Learn more about [Azure Hybrid Benefit for Windows Server licensing detailed guidance](#)
- Learn more about [Azure Hybrid Benefit for Windows Server and Azure Site Recovery make migrating applications to Azure even more cost-effective](#)
- Learn more about [Windows 10 on Azure with Multitenant Hosting Right](#)
- Learn more about [Using Resource Manager templates](#)

How to deploy Windows 10 on Azure with Multitenant Hosting Rights

8/28/2019 • 3 minutes to read • [Edit Online](#)

For customers with Windows 10 Enterprise E3/E5 per user or Windows Virtual Desktop Access per user (User Subscription Licenses or Add-on User Subscription Licenses), Multitenant Hosting Rights for Windows 10 allows you to bring your Windows 10 Licenses to the cloud and run Windows 10 Virtual Machines on Azure without paying for another license. For more information, please see [Multitenant Hosting for Windows 10](#).

NOTE

This article shows you to implement the licensing benefit for Windows 10 Pro Desktop images on Azure Marketplace.

- For Windows 7, 8.1, 10 Enterprise (x64) images on Azure Marketplace for MSDN Subscriptions, please refer to [Windows client in Azure for dev/test scenarios](#)
- For Windows Server licensing benefits, please refer to [Azure Hybrid use benefits for Windows Server images](#).

Deploying Windows 10 Image from Azure Marketplace

For Powershell, CLI and Azure Resource Manager template deployments, the Windows 10 image can be found with the following publishername, offer, sku.

OS	PUBLISHERNAME	OFFER	SKU
Windows 10 Pro	MicrosoftWindowsDesktop	Windows-10	RS2-Pro
Windows 10 Pro N	MicrosoftWindowsDesktop	Windows-10	RS2-ProN
Windows 10 Pro	MicrosoftWindowsDesktop	Windows-10	RS3-Pro
Windows 10 Pro N	MicrosoftWindowsDesktop	Windows-10	RS3-ProN

Uploading Windows 10 VHD to Azure

If you are uploading a generalized Windows 10 VHD, please note Windows 10 does not have built-in administrator account enabled by default. To enable the built-in administrator account, include the following command as part of the Custom Script extension.

```
Net user <username> /active:yes
```

The following powershell snippet is to mark all administrator accounts as active, including the built-in administrator. This example is useful if the built-in administrator username is unknown.

```
$adminAccount = Get-WmiObject Win32_UserAccount -filter "LocalAccount=True" | ? {$_.SID -Like "S-1-5-21-*-500"}
if($adminAccount.Disabled)
{
    $adminAccount.Disabled = $false
    $adminAccount.Put()
}
```

For more information:

- [How to upload VHD to Azure](#)
- [How to prepare a Windows VHD to upload to Azure](#)

Deploying Windows 10 with Multitenant Hosting Rights

Make sure you have [installed and configured the latest Azure PowerShell](#). Once you have prepared your VHD, upload the VHD to your Azure Storage account using the `Add-AzVhd` cmdlet as follows:

```
Add-AzVhd -ResourceGroupName "myResourceGroup" -LocalFilePath "C:\Path\To\myvhd.vhd" ` 
-Destination "https://mystorageaccount.blob.core.windows.net/vhds/myvhd.vhd"
```

Deploy using Azure Resource Manager Template Deployment Within your Resource Manager templates, an additional parameter for `licenseType` can be specified. You can read more about [authoring Azure Resource Manager templates](#). Once you have your VHD uploaded to Azure, edit your Resource Manager template to include the license type as part of the compute provider and deploy your template as normal:

```
"properties": {
    "licenseType": "Windows_Client",
    "hardwareProfile": {
        "vmSize": "[variables('vmSize')]"
    }
}
```

Deploy via PowerShell When deploying your Windows Server VM via PowerShell, you have an additional parameter for `-LicenseType`. Once you have your VHD uploaded to Azure, you create a VM using `New-AzVM` and specify the licensing type as follows:

```
New-AzVM -ResourceGroupName "myResourceGroup" -Location "West US" -VM $vm -LicenseType "Windows_Client"
```

Verify your VM is utilizing the licensing benefit

Once you have deployed your VM through either the PowerShell or Resource Manager deployment method, verify the license type with `Get-AzVM` as follows:

```
Get-AzVM -ResourceGroup "myResourceGroup" -Name "myVM"
```

The output is similar to the following example for Windows 10 with correct license type:

Type	:	Microsoft.Compute/virtualMachines
Location	:	westus
LicenseType	:	Windows_Client

This output contrasts with the following VM deployed without Azure Hybrid Use Benefit licensing, such as a VM deployed straight from the Azure Gallery:

```
Type : Microsoft.Compute/virtualMachines  
Location : westus  
LicenseType :
```

Additional Information about joining Azure AD

NOTE

Azure provisions all Windows VMs with built-in administrator account, which cannot be used to join AAD. For example, *Settings > Account > Access Work or School > +Connect* will not work. You must create and log on as a second administrator account to join Azure AD manually. You can also configure Azure AD using a provisioning package, use the link is the *Next Steps* section to learn more.

Next Steps

- Learn more about [Configuring VDA for Windows 10](#)
- Learn more about [Multitenant Hosting for Windows 10](#)

Migrate your IaaS resources to Azure Resource Manager by March 1, 2023

2/28/2020 • 2 minutes to read • [Edit Online](#)

In 2014, we launched IaaS on Azure Resource Manager, and have been enhancing capabilities ever since. Because [Azure Resource Manager](#) now has full IaaS capabilities and other advancements, we deprecated the management of IaaS VMs through Azure Service Manager on February 28, 2020 and this functionality will be fully retired on March 1, 2023.

Today, about 90% of the IaaS VMs are using Azure Resource Manager. If you use IaaS resources through Azure Service Manager (ASM), start planning your migration now and complete it by March 1, 2023 to take advantage of [Azure Resource Manager](#).

Classic VMs will be following the [Modern Lifecycle Policy](#) for deprecation.

How does this affect me?

1. Starting February 28, 2020, customers who did not utilize IaaS VMs through Azure Service Manager (ASM) in the month of February 2020 will no longer be able to create classic VMs.
2. On March 1, 2023, customers will no longer be able to start IaaS VMs using Azure Service Manager and any that are still running or allocated will be stopped and deallocated.
3. On March 1, 2023, subscriptions who have not migrated to Azure Resource Manager will be informed regarding timelines for deleting any remaining Classic VMs.

The following Azure services and functionality will **NOT** be impacted by this retirement:

- Cloud Services
- Storage accounts **not** used by classic VMs
- Virtual networks (VNets) **not** used by classic VMs.
- Other classic resources

What actions should I take?

- Start planning your migration to Azure Resource Manager, today.
- [Learn more](#) about migrating your classic [Linux](#) and [Windows](#) VMs to Azure Resource Manager.
- For more information, refer to the [Frequently asked questions about classic to Azure Resource Manager migration](#)
- For technical questions and issues, [contact support](#).
- For other questions not part of FAQ and feedback, comment below.

Platform-supported migration of IaaS resources from classic to Azure Resource Manager

2/28/2020 • 8 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

This article describes how to migrate infrastructure as a service (IaaS) resources from the Classic to Resource Manager deployment models and details how to connect resources from the two deployment models that coexist in your subscription by using virtual network site-to-site gateways. You can read more about [Azure Resource Manager features and benefits](#).

Goal for migration

Resource Manager enables deploying complex applications through templates, configures virtual machines by using VM extensions, and incorporates access management and tagging. Azure Resource Manager includes scalable, parallel deployment for virtual machines into availability sets. The new deployment model also provides lifecycle management of compute, network, and storage independently. Finally, there's a focus on enabling security by default with the enforcement of virtual machines in a virtual network.

Almost all the features from the classic deployment model are supported for compute, network, and storage under Azure Resource Manager. To benefit from the new capabilities in Azure Resource Manager, you can migrate existing deployments from the Classic deployment model.

Supported resources for migration

These classic IaaS resources are supported during migration

- Virtual Machines
- Availability Sets
- Storage Accounts
- Virtual Networks
- VPN Gateways
- Express Route Gateways (*in the same subscription as Virtual Network only*)
- Network Security Groups
- Route Tables
- Reserved IPs

Supported scopes of migration

There are four different ways to complete migration of compute, network, and storage resources:

- [Migration of virtual machines \(NOT in a virtual network\)](#)
- [Migration of virtual machines \(in a virtual network\)](#)
- [Migration of storage accounts](#)
- [Migration of unattached resources](#)

Migration of virtual machines (NOT in a virtual network)

In the Resource Manager deployment model, security is enforced for your applications by default. All VMs need to be in a virtual network in the Resource Manager model. The Azure platform restarts (`Stop` , `Deallocate` , and `Start`) the VMs as part of the migration. You have two options for the virtual networks that the Virtual Machines will be migrated to:

- You can request the platform to create a new virtual network and migrate the virtual machine into the new virtual network.
- You can migrate the virtual machine into an existing virtual network in Resource Manager.

NOTE

In this migration scope, both the management-plane operations and the data-plane operations may not be allowed for a period of time during the migration.

Migration of virtual machines (in a virtual network)

For most VM configurations, only the metadata is migrating between the Classic and Resource Manager deployment models. The underlying VMs are running on the same hardware, in the same network, and with the same storage. The management-plane operations may not be allowed for a certain period of time during the migration. However, the data plane continues to work. That is, your applications running on top of VMs (classic) do not incur downtime during the migration.

The following configurations are not currently supported. If support is added in the future, some VMs in this configuration might incur downtime (go through stop, deallocate, and restart VM operations).

- You have more than one availability set in a single cloud service.
- You have one or more availability sets and VMs that are not in an availability set in a single cloud service.

NOTE

In this migration scope, the management plane may not be allowed for a period of time during the migration. For certain configurations as described earlier, data-plane downtime occurs.

Migration of storage accounts

To allow seamless migration, you can deploy Resource Manager VMs in a classic storage account. With this capability, compute and network resources can and should be migrated independently of storage accounts. Once you migrate over your Virtual Machines and Virtual Network, you need to migrate over your storage accounts to complete the migration process.

If your storage account does not have any associated disks or Virtual Machines data and only has blobs, files, tables, and queues then the migration to Azure Resource Manager can be done as a standalone migration without dependencies.

NOTE

The Resource Manager deployment model doesn't have the concept of Classic images and disks. When the storage account is migrated, Classic images and disks are not visible in the Resource Manager stack but the backing VHDs remain in the storage account.

The following screenshots show how to upgrade a Classic storage account to an Azure Resource Manager storage account using Azure portal:

1. Sign in to the [Azure portal](#).

2. Navigate to your storage account.
3. In the **Settings** section, click **Migrate to ARM**.
4. Click on **Validate** to determine migration feasibility.
5. If validation passes, click on **Prepare** to create a migrated storage account.
6. Type **yes** to confirm migration and click **Commit** to finish the migration.

Migrate to ARM

testclassicaccount2 - Migrate to ARM
Storage account (classic)

To benefit from new capabilities in Azure Resource Manager, you can migrate existing deployments from the Classic deployment model. extensions, incorporates role-based access management, and tagging. [Learn more](#)

Take the following steps to complete migration:

1. Validate if the resource is capable of migration

[Validate](#)
2. Prepare
3. Commit or abort

testclassicaccount2 - Migrate to ARM
Storage account (classic)

Validation passed.

To benefit from new capabilities in Azure Resource Manager, you can migrate existing deployments from the Classic deployment model. extensions, incorporates role-based access management, and tagging. [Learn more](#)

Take the following steps to complete migration:

1. Validate if the resource is capable of migration

Validation passed.

1 storage account will be migrated.

[View details](#)
2. Prepare

Simulate the transformation of classic resources into Resource Manager resources.

If you see any issues with the results of 'Prepare', you will be able to abort migration in the next step.

[Prepare](#)
3. Commit or abort

Migration of unattached resources

Storage Accounts with no associated disks or Virtual Machines data may be migrated independently.

Network Security Groups, Route Tables & Reserved IPs that are not attached to any Virtual Machines and Virtual Networks can also be migrated independently.

Unsupported features and configurations

Some features and configurations are not currently supported; the following sections describe our recommendations around them.

Unsupported features

The following features are not currently supported. You can optionally remove these settings, migrate the VMs, and then re-enable the settings in the Resource Manager deployment model.

RESOURCE PROVIDER	FEATURE	RECOMMENDATION
Compute	Unassociated virtual machine disks.	The VHD blobs behind these disks will get migrated when the Storage Account is migrated
Compute	Virtual machine images.	The VHD blobs behind these disks will get migrated when the Storage Account is migrated
Network	Endpoint ACLs.	Remove Endpoint ACLs and retry migration.
Network	Application Gateway	Remove the Application Gateway before beginning migration and then recreate the Application Gateway once migration is complete.
Network	Virtual networks using VNet Peering.	Migrate Virtual Network to Resource Manager, then peer. Learn more about VNet Peering .

Unsupported configurations

The following configurations are not currently supported.

Service	Configuration	Recommendation
Resource Manager	Role-Based Access Control (RBAC) for classic resources	Because the URI of the resources is modified after migration, it is recommended that you plan the RBAC policy updates that need to happen after migration.
Compute	Multiple subnets associated with a VM	Update the subnet configuration to reference only one subnet. This may require you to remove a secondary NIC (that is referring to another subnet) from the VM and reattach it after migration has completed.
Compute	Virtual machines that belong to a virtual network but don't have an explicit subnet assigned	You can optionally delete the VM.
Compute	Virtual machines that have alerts, Autoscale policies	The migration goes through and these settings are dropped. It is highly recommended that you evaluate your environment before you do the migration. Alternatively, you can reconfigure the alert settings after migration is complete.
Compute	XML VM extensions (BGInfo 1.* , Visual Studio Debugger, Web Deploy, and Remote Debugging)	This is not supported. It is recommended that you remove these extensions from the virtual machine to continue migration or they will be dropped automatically during the migration process.
Compute	Boot diagnostics with Premium storage	Disable Boot Diagnostics feature for the VMs before continuing with migration. You can re-enable boot diagnostics in the Resource Manager stack after the migration is complete. Additionally, blobs that are being used for screenshot and serial logs should be deleted so you are no longer charged for those blobs.
Compute	Cloud services that contain web/worker roles	This is currently not supported.
Compute	Cloud services that contain more than one availability set or multiple availability sets.	This is currently not supported. Please move the Virtual Machines to the same availability set before migrating.

Service	Configuration	Recommendation
Compute	VM with Azure Security Center extension	Azure Security Center automatically installs extensions on your Virtual Machines to monitor their security and raise alerts. These extensions usually get installed automatically if the Azure Security Center policy is enabled on the subscription. To migrate the Virtual Machines, disable the security center policy on the subscription, which will remove the Security Center monitoring extension from the Virtual Machines.
Compute	VM with backup or snapshot extension	These extensions are installed on a Virtual Machine configured with the Azure Backup service. While the migration of these VMs is not supported, follow the guidance here to keep backups that were taken prior to migration.
Compute	VM with Azure Site Recovery extension	These extensions are installed on a Virtual Machine configured with the Azure Site Recovery service. While the migration of storage used with Site Recovery will work, current replication will be impacted. You need to disable and enable VM replication after storage migration.
Network	Virtual networks that contain virtual machines and web/worker roles	This is currently not supported. Please move the Web/Worker roles to their own Virtual Network before migrating. Once the classic Virtual Network is migrated, the migrated Azure Resource Manager Virtual Network can be peered with the classic Virtual Network to achieve similar configuration as before.
Network	Classic Express Route circuits	This is currently not supported. These circuits need to be migrated to Azure Resource Manager before beginning IaaS migration. To learn more, see Moving ExpressRoute circuits from the classic to the Resource Manager deployment model .
Azure App Service	Virtual networks that contain App Service environments	This is currently not supported.
Azure HDInsight	Virtual networks that contain HDInsight services	This is currently not supported.
Microsoft Dynamics Lifecycle Services	Virtual networks that contain virtual machines that are managed by Dynamics Lifecycle Services	This is currently not supported.

Service	Configuration	Recommendation
Azure AD Domain Services	Virtual networks that contain Azure AD Domain services	This is currently not supported.
Azure API Management	Virtual networks that contain Azure API Management deployments	This is currently not supported. To migrate the IaaS VNET, change the VNET of the API Management deployment, which is a no downtime operation.

Next steps

- [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
- [Planning for migration of IaaS resources from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [VPN Gateway classic to Resource Manager migration](#)
- [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#)
- [Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager](#)
- [Review most common migration errors](#)
- [Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager](#)

Technical deep dive on platform-supported migration from classic to Azure Resource Manager

2/28/2020 • 13 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

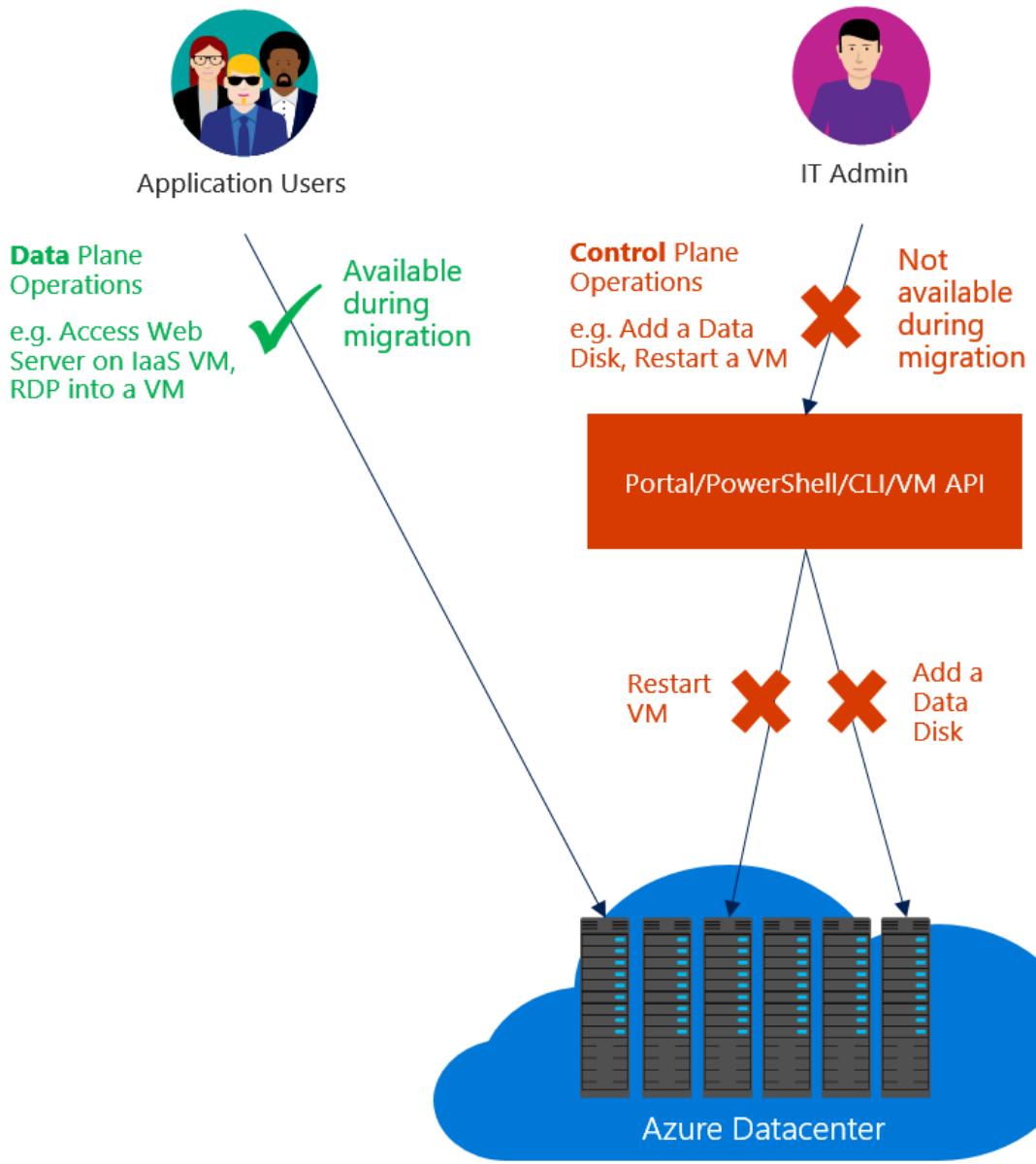
Let's take a deep-dive on migrating from the Azure classic deployment model to the Azure Resource Manager deployment model. We look at resources at a resource and feature level to help you understand how the Azure platform migrates resources between the two deployment models. For more information, please read the service announcement article: [Platform-supported migration of IaaS resources from classic to Azure Resource Manager](#).

Migrate IaaS resources from the classic deployment model to Azure Resource Manager

First, it's important to understand the difference between data-plane and management-plane operations on the infrastructure as a service (IaaS) resources.

- *Management/control plane* describes the calls that come into the management/control plane or the API for modifying resources. For example, operations like creating a VM, restarting a VM, and updating a virtual network with a new subnet manage the running resources. They don't directly affect connecting to the VMs.
- *Data plane* (application) describes the runtime of the application itself, and involves interaction with instances that don't go through the Azure API. For example, accessing your website, or pulling data from a running SQL Server instance or a MongoDB server, are data plane or application interactions. Other examples include copying a blob from a storage account, and accessing a public IP address to use Remote Desktop Protocol (RDP) or Secure Shell (SSH) into the virtual machine. These operations keep the application running across compute, networking, and storage.

The data plane is the same between the classic deployment model and Resource Manager stacks. The difference is that during the migration process, Microsoft translates the representation of the resources from the classic deployment model to that in the Resource Manager stack. As a result, you need to use new tools, APIs, and SDKs to manage your resources in the Resource Manager stack.



NOTE

In some migration scenarios, the Azure platform stops, deallocates, and restarts your virtual machines. This causes a brief data-plane downtime.

The migration experience

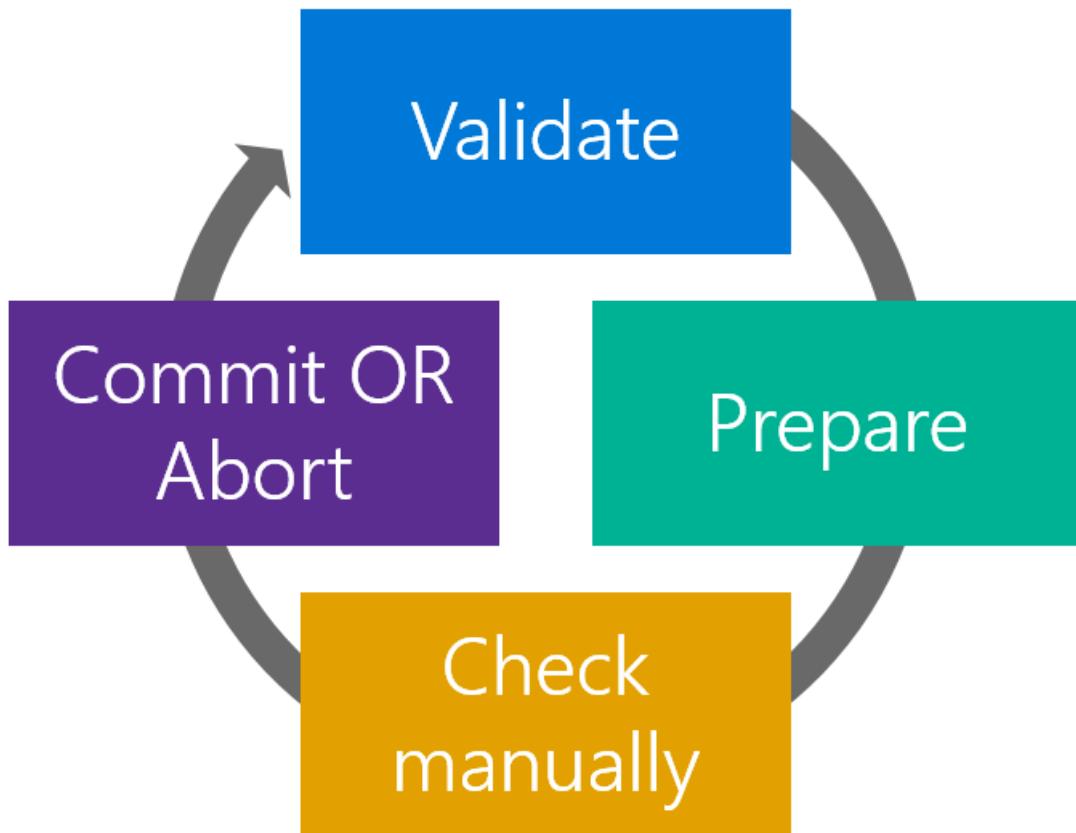
Before you start the migration:

- Ensure that the resources that you want to migrate don't use any unsupported features or configurations. Usually the platform detects these issues and generates an error.
- If you have VMs that are not in a virtual network, they are stopped and deallocated as part of the prepare operation. If you don't want to lose the public IP address, consider reserving the IP address before triggering the prepare operation. If the VMs are in a virtual network, they are not stopped and deallocated.
- Plan your migration during non-business hours to accommodate for any unexpected failures that might happen during migration.
- Download the current configuration of your VMs by using PowerShell, command-line interface (CLI) commands, or REST APIs to make it easier for validation after the prepare step is complete.
- Update your automation and operationalization scripts to handle the Resource Manager deployment model,

before you start the migration. You can optionally do GET operations when the resources are in the prepared state.

- Evaluate the Role-Based Access Control (RBAC) policies that are configured on the IaaS resources in the classic deployment model, and plan for after the migration is complete.

The migration workflow is as follows:



NOTE

The operations described in the following sections are all idempotent. If you have a problem other than an unsupported feature or a configuration error, retry the prepare, abort, or commit operation. Azure tries the action again.

Validate

The validate operation is the first step in the migration process. The goal of this step is to analyze the state of the resources you want to migrate in the classic deployment model. The operation evaluates whether the resources are capable of migration (success or failure).

You select the virtual network or a cloud service (if it's not in a virtual network) that you want to validate for migration. If the resource is not capable of migration, Azure lists the reasons why.

Checks not done in the validate operation

The validate operation only analyzes the state of the resources in the classic deployment model. It can check for all failures and unsupported scenarios due to various configurations in the classic deployment model. It is not possible to check for all issues that the Azure Resource Manager stack might impose on the resources during migration. These issues are only checked when the resources undergo transformation in the next step of migration (the prepare operation). The following table lists all the issues not checked in the validate operation:

NETWORKING CHECKS NOT IN THE VALIDATE OPERATION

A virtual network having both ER and VPN gateways.

A virtual network gateway connection in a disconnected state.

All ER circuits are pre-migrated to Azure Resource Manager stack.

Azure Resource Manager quota checks for networking resources. For example: static public IP, dynamic public IPs, load balancer, network security groups, route tables, and network interfaces.

All load balancer rules are valid across deployment and the virtual network.

Conflicting private IPs between stop-deallocated VMs in the same virtual network.

Prepare

The prepare operation is the second step in the migration process. The goal of this step is to simulate the transformation of the IaaS resources from the classic deployment model to Resource Manager resources. Further, the prepare operation presents this side-by-side for you to visualize.

NOTE

Your resources in the classic deployment model are not modified during this step. It's a safe step to run if you're trying out migration.

You select the virtual network or the cloud service (if it's not a virtual network) that you want to prepare for migration.

- If the resource is not capable of migration, Azure stops the migration process and lists the reason why the prepare operation failed.
- If the resource is capable of migration, Azure locks down the management-plane operations for the resources under migration. For example, you are not able to add a data disk to a VM under migration.

Azure then starts the migration of metadata from the classic deployment model to Resource Manager for the migrating resources.

After the prepare operation is complete, you have the option of visualizing the resources in both the classic deployment model and Resource Manager. For every cloud service in the classic deployment model, the Azure platform creates a resource group name that has the pattern `cloud-service-name>-Migrated`.

NOTE

It is not possible to select the name of a resource group created for migrated resources (that is, "-Migrated"). After migration is complete, however, you can use the move feature of Azure Resource Manager to move resources to any resource group you want. For more information, see [Move resources to new resource group or subscription](#).

The following two screenshots show the result after a successful prepare operation. The first one shows a resource group that contains the original cloud service. The second one shows the new "-Migrated" resource group that contains the equivalent Azure Resource Manager resources.

portalmigrate
Resource group

Essentials

Subscription name (change)	Subscription ID
Deployments No deployments	Location East US

Filter by name...

2 items

NAME	TYPE	LOCATION
portalmigrate	Cloud service (class...)	East US
portalmigrate	Virtual machine (cl...)	East US

portalmigrate-Migrated
Resource group

Essentials

Subscription name (change)	Subscription ID
Deployments 2 Succeeded	Location East US

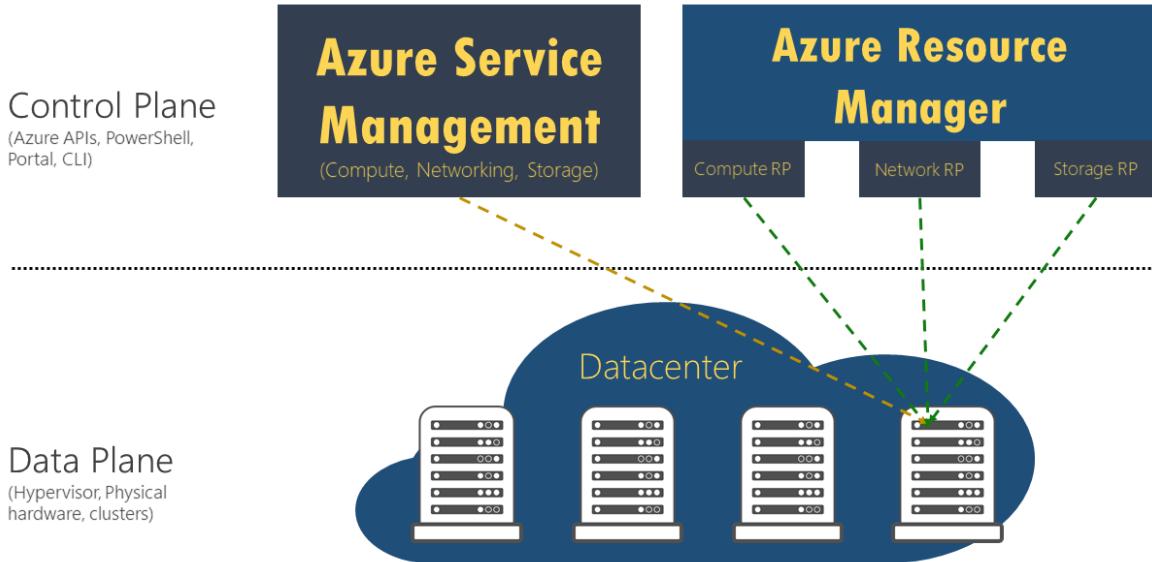
Filter by name...

5 items

NAME	TYPE	LOCATION
portalmigrate	Virtual machine	East US
portalmigrate-PrimaryNic	Network interface	East US
portalmigrate-PrimaryVirtualIP	Public IP address	East US
portalmigrate-PublicLoadBalancer	Load balancer	East US
portalmigrate-VirtualNetwork	Virtual network	East US

Here is a behind-the-scenes look at your resources after the completion of the prepare phase. Note that the resource in the data plane is the same. It's represented in both the management plane (classic deployment model) and the control plane (Resource Manager).

Prepare



NOTE

VMs that are not in a virtual network in the classic deployment model are stopped and deallocated in this phase of migration.

Check (manual or scripted)

In the check step, you have the option to use the configuration that you downloaded earlier to validate that the migration looks correct. Alternatively, you can sign in to the portal, and spot check the properties and resources to validate that metadata migration looks good.

If you are migrating a virtual network, most configuration of virtual machines is not restarted. For applications on those VMs, you can validate that the application is still running.

You can test your monitoring and operational scripts to see if the VMs are working as expected, and if your updated scripts work correctly. Only GET operations are supported when the resources are in the prepared state.

There is no set window of time before which you need to commit the migration. You can take as much time as you want in this state. However, the management plane is locked for these resources until you either abort or commit.

If you see any issues, you can always abort the migration and go back to the classic deployment model. After you go back, Azure opens the management-plane operations on the resources, so that you can resume normal operations on those VMs in the classic deployment model.

Abort

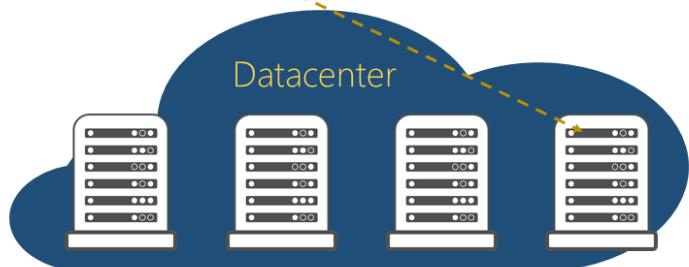
This is an optional step if you want to revert your changes to the classic deployment model and stop the migration. This operation deletes the Resource Manager metadata (created in the prepare step) for your resources.

Abort

Control Plane
(Azure APIs, PowerShell, Portal, CLI)



Data Plane
(Hypervisor, Physical hardware, clusters)



NOTE

This operation can't be done after you have triggered the commit operation.

Commit

After you finish the validation, you can commit the migration. Resources do not appear anymore in the classic deployment model, and are available only in the Resource Manager deployment model. The migrated resources can be managed only in the new portal.

NOTE

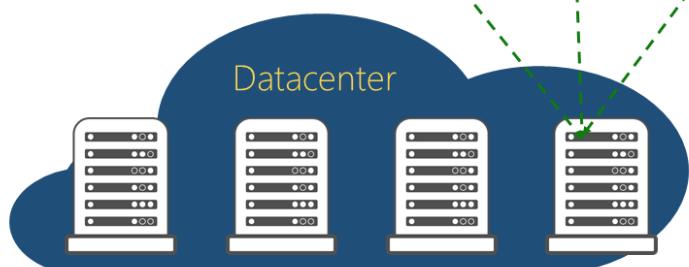
This is an idempotent operation. If it fails, retry the operation. If it continues to fail, create a support ticket or create a forum on [Microsoft Q&A](#)

Commit

Control Plane
(Azure APIs, PowerShell, Portal, CLI)



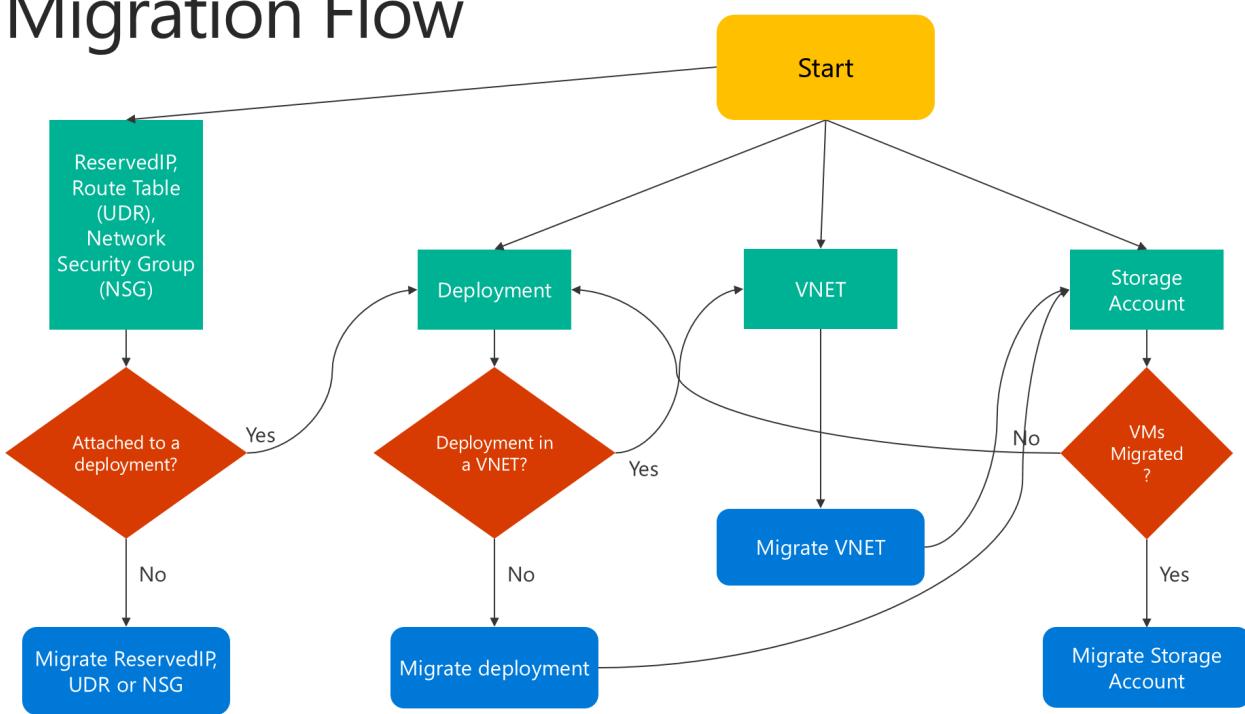
Data Plane
(Hypervisor, Physical hardware, clusters)



Migration flowchart

Here is a flowchart that shows how to proceed with migration:

Migration Flow



Translation of the classic deployment model to Resource Manager resources

You can find the classic deployment model and Resource Manager representations of the resources in the following table. Other features and resources are not currently supported.

CLASSIC REPRESENTATION	RESOURCE MANAGER REPRESENTATION	NOTES
Cloud service name	DNS name	During migration, a new resource group is created for every cloud service with the naming pattern <code><cloudservicename>-migrated</code> . This resource group contains all your resources. The cloud service name becomes a DNS name that is associated with the public IP address.
Virtual machine	Virtual machine	VM-specific properties are migrated unchanged. Certain osProfile information, like computer name, is not stored in the classic deployment model, and remains empty after migration.

CLASSIC REPRESENTATION	RESOURCE MANAGER REPRESENTATION	NOTES
Disk resources attached to VM	Implicit disks attached to VM	Disks are not modeled as top-level resources in the Resource Manager deployment model. They are migrated as implicit disks under the VM. Only disks that are attached to a VM are currently supported. Resource Manager VMs can now use storage accounts in the classic deployment model, which allows the disks to be easily migrated without any updates.
VM extensions	VM extensions	All the resource extensions, except XML extensions, are migrated from the classic deployment model.
Virtual machine certificates	Certificates in Azure Key Vault	If a cloud service contains service certificates, the migration creates a new Azure key vault per cloud service, and moves the certificates into the key vault. The VMs are updated to reference the certificates from the key vault. Do not delete the key vault. This can cause the VM to go into a failed state.
WinRM configuration	WinRM configuration under osProfile	Windows Remote Management configuration is moved unchanged, as part of the migration.
Availability-set property	Availability-set resource	Availability-set specification is a property on the VM in the classic deployment model. Availability sets become a top-level resource as part of the migration. The following configurations are not supported: multiple availability sets per cloud service, or one or more availability sets along with VMs that are not in any availability set in a cloud service.
Network configuration on a VM	Primary network interface	Network configuration on a VM is represented as the primary network interface resource after migration. For VMs that are not in a virtual network, the internal IP address changes during migration.
Multiple network interfaces on a VM	Network interfaces	If a VM has multiple network interfaces associated with it, each network interface becomes a top-level resource as part of the migration, along with all the properties.

CLASSIC REPRESENTATION	RESOURCE MANAGER REPRESENTATION	NOTES
Load-balanced endpoint set	Load balancer	In the classic deployment model, the platform assigned an implicit load balancer for every cloud service. During migration, a new load-balancer resource is created, and the load-balancing endpoint set becomes load-balancer rules.
Inbound NAT rules	Inbound NAT rules	Input endpoints defined on the VM are converted to inbound network address translation rules under the load balancer during the migration.
VIP address	Public IP address with DNS name	The virtual IP address becomes a public IP address, and is associated with the load balancer. A virtual IP can only be migrated if there is an input endpoint assigned to it.
Virtual network	Virtual network	The virtual network is migrated, with all its properties, to the Resource Manager deployment model. A new resource group is created with the name <code>-migrated</code> .
Reserved IPs	Public IP address with static allocation method	Reserved IPs associated with the load balancer are migrated, along with the migration of the cloud service or the virtual machine. Unassociated reserved IP migration is not currently supported.
Public IP address per VM	Public IP address with dynamic allocation method	The public IP address associated with the VM is converted as a public IP address resource, with the allocation method set to static.
NSGs	NSGs	Network security groups associated with a subnet are cloned as part of the migration to the Resource Manager deployment model. The NSG in the classic deployment model is not removed during the migration. However, the management-plane operations for the NSG are blocked when the migration is in progress.
DNS servers	DNS servers	DNS servers associated with a virtual network or the VM are migrated as part of the corresponding resource migration, along with all the properties.

CLASSIC REPRESENTATION	RESOURCE MANAGER REPRESENTATION	NOTES
UDRs	UDRs	User-defined routes associated with a subnet are cloned as part of the migration to the Resource Manager deployment model. The UDR in the classic deployment model is not removed during the migration. The management-plane operations for the UDR are blocked when the migration is in progress.
IP forwarding property on a VM's network configuration	IP forwarding property on the NIC	The IP forwarding property on a VM is converted to a property on the network interface during the migration.
Load balancer with multiple IPs	Load balancer with multiple public IP resources	Every public IP associated with the load balancer is converted to a public IP resource, and associated with the load balancer after migration.
Internal DNS names on the VM	Internal DNS names on the NIC	During migration, the internal DNS suffixes for the VMs are migrated to a read-only property named "InternalDomainNameSuffix" on the NIC. The suffix remains unchanged after migration, and VM resolution should continue to work as previously.
Virtual network gateway	Virtual network gateway	Virtual network gateway properties are migrated unchanged. The VIP associated with the gateway does not change either.
Local network site	Local network gateway	Local network site properties are migrated unchanged to a new resource called a local network gateway. This represents on-premises address prefixes and the remote gateway IP.
Connections references	Connection	Connectivity references between the gateway and the local network site in network configuration is represented by a new resource called Connection. All properties of connectivity reference in network configuration files are copied unchanged to the Connection resource. Connectivity between virtual networks in the classic deployment model is achieved by creating two IPsec tunnels to local network sites representing the virtual networks. This is transformed to the virtual-network-to-virtual-network connection type in the Resource Manager model, without requiring local network gateways.

Changes to your automation and tooling after migration

As part of migrating your resources from the classic deployment model to the Resource Manager deployment

model, you must update your existing automation or tooling to ensure that it continues to work after the migration.

Next steps

- [Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
- [Planning for migration of IaaS resources from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [VPN Gateway classic to Resource Manager migration](#)
- [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#)
- [Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager](#)
- [Review most common migration errors](#)
- [Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager](#)

Planning for migration of IaaS resources from classic to Azure Resource Manager

2/28/2020 • 13 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

While Azure Resource Manager offers many amazing features, it is critical to plan out your migration journey to make sure things go smoothly. Spending time on planning will ensure that you do not encounter issues while executing migration activities.

There are four general phases of the migration journey:



Plan

Technical considerations and tradeoffs

Depending on your technical requirements size, geographies and operational practices, you might want to consider:

1. Why is Azure Resource Manager desired for your organization? What are the business reasons for a migration?
2. What are the technical reasons for Azure Resource Manager? What (if any) additional Azure services would you like to leverage?
3. Which application (or sets of virtual machines) is included in the migration?
4. Which scenarios are supported with the migration API? Review the [unsupported features and configurations](#).
5. Will your operational teams now support applications/VMs in both Classic and Azure Resource Manager?
6. How (if at all) does Azure Resource Manager change your VM deployment, management, monitoring, and reporting processes? Do your deployment scripts need to be updated?
7. What is the communications plan to alert stakeholders (end users, application owners, and infrastructure owners)?
8. Depending on the complexity of the environment, should there be a maintenance period where the application is unavailable to end users and to application owners? If so, for how long?
9. What is the training plan to ensure stakeholders are knowledgeable and proficient in Azure Resource Manager?
10. What is the program management or project management plan for the migration?
11. What are the timelines for the Azure Resource Manager migration and other related technology road maps? Are they optimally aligned?

Patterns of success

Successful customers have detailed plans where the preceding questions are discussed, documented and governed. Ensure the migration plans are broadly communicated to sponsors and stakeholders. Equip yourself

with knowledge about your migration options; reading through this migration document set below is highly recommended.

- [Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
- [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
- [Planning for migration of IaaS resources from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager](#)
- [Review most common migration errors](#)
- [Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager](#)

Pitfalls to avoid

- Failure to plan. The technology steps of this migration are proven and the outcome is predictable.
- Assumption that the platform supported migration API will account for all scenarios. Read the [unsupported features and configurations](#) to understand what scenarios are supported.
- Not planning potential application outage for end users. Plan enough buffer to adequately warn end users of potentially unavailable application time.

Lab Test

Replicate your environment and do a test migration

NOTE

Exact replication of your existing environment is executed by using a community-contributed tool which is not officially supported by Microsoft Support. Therefore, it is an **optional** step but it is the best way to find out issues without touching your production environments. If using a community-contributed tool is not an option, then read about the Validate/Prepare/Abort Dry Run recommendation below.

Conducting a lab test of your exact scenario (compute, networking, and storage) is the best way to ensure a smooth migration. This will help ensure:

- A wholly separate lab or an existing non-production environment to test. We recommend a wholly separate lab that can be migrated repeatedly and can be destructively modified. Scripts to collect/hydrate metadata from the real subscriptions are listed below.
- It's a good idea to create the lab in a separate subscription. The reason is that the lab will be torn down repeatedly, and having a separate, isolated subscription will reduce the chance that something real will get accidentally deleted.

This can be accomplished by using the AsmMetadataParser tool. [Read more about this tool here](#)

Patterns of success

The following were issues discovered in many of the larger migrations. This is not an exhaustive list and you should refer to the [unsupported features and configurations](#) for more detail. You may or may not encounter these technical issues but if you do solving these before attempting migration will ensure a smoother experience.

- **Do a Validate/Prepare/Abort Dry Run** - This is perhaps the most important step to ensure Classic to Azure Resource Manager migration success. The migration API has three main steps: Validate, Prepare and Commit. Validate will read the state of your classic environment and return a result of all issues. However, because some issues might exist in the Azure Resource Manager stack, Validate will not catch everything. The next step in migration process, Prepare will help expose those issues. Prepare will move the metadata

from Classic to Azure Resource Manager, but will not commit the move, and will not remove or change anything on the Classic side. The dry run involves preparing the migration, then aborting (**not committing**) the migration prepare. The goal of validate/prepare/abort dry run is to see all of the metadata in the Azure Resource Manager stack, examine it (*programmatically or in Portal*), and verify that everything migrates correctly, and work through technical issues. It will also give you a sense of migration duration so you can plan for downtime accordingly. A validate/prepare/abort does not cause any user downtime; therefore, it is non-disruptive to application usage.

- The items below will need to be solved before the dry run, but a dry run test will also safely flush out these preparation steps if they are missed. During enterprise migration, we've found the dry run to be a safe and invaluable way to ensure migration readiness.
- When prepare is running, the control plane (Azure management operations) will be locked for the whole virtual network, so no changes can be made to VM metadata during validate/prepare/abort. But otherwise any application function (RD, VM usage, etc.) will be unaffected. Users of the VMs will not know that the dry run is being executed.
- **Express Route Circuits and VPN.** Currently Express Route Gateways with authorization links cannot be migrated without downtime. For the workaround, see [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#).
- **VM Extensions** - Virtual Machine extensions are potentially one of the biggest roadblocks to migrating running VMs. Remediation of VM Extensions could take upwards of 1-2 days, so plan accordingly. A working Azure agent is needed to report back VM Extension status of running VMs. If the status comes back as bad for a running VM, this will halt migration. The agent itself does not need to be in working order to enable migration, but if extensions exist on the VM, then both a working agent AND outbound internet connectivity (with DNS) will be needed for migration to move forward.
 - If connectivity to a DNS server is lost during migration, all VM Extensions except BGInfo version 1.* need to first be removed from every VM before migration prepare, and subsequently re-added back to the VM after Azure Resource Manager migration. **This is only for VMs that are running.** If the VMs are stopped deallocated, VM Extensions do not need to be removed.

NOTE

Many extensions like Azure diagnostics and security center monitoring will reinstall themselves after migration, so removing them is not a problem.

- In addition, make sure Network Security Groups are not restricting outbound internet access. This can happen with some Network Security Groups configurations. Outbound internet access (and DNS) is needed for VM Extensions to be migrated to Azure Resource Manager.
- Two versions of the BGInfo extension exist and are called versions 1 and 2.
 - If the VM is using the BGInfo version 1 extension, you can leave this extension as is. The migration API skips this extension. The BGInfo extension can be added after migration.
 - If the VM is using the JSON-based BGInfo version 2 extension, the VM was created using the Azure portal. The migration API includes this extension in the migration to Azure Resource Manager, provided the agent is working and has outbound internet access (and DNS).
- **Remediation Option 1.** If you know your VMs will not have outbound internet access, a working DNS service, and working Azure agents on the VMs, then uninstall all VM extensions as part of the migration before Prepare, then reinstall the VM Extensions after migration.
- **Remediation Option 2.** If VM extensions are too big of a hurdle, another option is to shutdown/deallocate all VMs before migration. Migrate the deallocated VMs, then restart them on the Azure Resource Manager side. The benefit here is that VM extensions will migrate. The downside

is that all public facing Virtual IPs will be lost (this may be a non-starter), and obviously the VMs will shut down causing a much greater impact on working applications.

NOTE

If an Azure Security Center policy is configured against the running VMs being migrated, the security policy needs to be stopped before removing extensions, otherwise the security monitoring extension will be reinstalled automatically on the VM after removing it.

- **Availability Sets** - For a virtual network (vNet) to be migrated to Azure Resource Manager, the Classic deployment (i.e. cloud service) contained VMs must all be in one availability set, or the VMs must all not be in any availability set. Having more than one availability set in the cloud service is not compatible with Azure Resource Manager and will halt migration. Additionally, there cannot be some VMs in an availability set, and some VMs not in an availability set. To resolve this, you will need to remediate or reshuffle your cloud service. Plan accordingly as this might be time consuming.
- **Web/Worker Role Deployments** - Cloud Services containing web and worker roles cannot migrate to Azure Resource Manager. To migrate the contents of your web and worker roles, you will need to migrate the code itself to newer PaaS App Services (this discussion is beyond the scope of this document). If you want to leave the web/worker roles as is but migrate classic VMs to the Resource Manager deployment model, the web/worker roles must first be removed from the virtual network before migration can start. A typical solution is to just move web/worker role instances to a separate Classic virtual network that is also linked to an ExpressRoute circuit. In the former redeploy case, create a new Classic virtual network, move/redeploy the web/worker roles to that new virtual network, and then delete the deployments from the virtual network being moved. No code changes required. The new [Virtual Network Peering](#) capability can be used to peer together the classic virtual network containing the web/worker roles and other virtual networks in the same Azure region such as the virtual network being migrated (**after virtual network migration is completed as peered virtual networks cannot be migrated**), hence providing the same capabilities with no performance loss and no latency/bandwidth penalties. Given the addition of [Virtual Network Peering](#), web/worker role deployments can now easily be mitigated and not block the migration to Azure Resource Manager.
- **Azure Resource Manager Quotas** - Azure regions have separate quotas/limits for both Classic and Azure Resource Manager. Even though in a migration scenario new hardware isn't being consumed (*we're swapping existing VMs from Classic to Azure Resource Manager*), Azure Resource Manager quotas still need to be in place with enough capacity before migration can start. Listed below are the major limits we've seen cause problems. Open a quota support ticket to raise the limits.

NOTE

These limits need to be raised in the same region as your current environment to be migrated.

- Network Interfaces
- Load Balancers
- Public IPs
- Static Public IPs
- Cores
- Network Security Groups
- Route Tables

You can check your current Azure Resource Manager quotas using the following commands with the latest version of Azure PowerShell.

Compute (Cores, Availability Sets)

```
Get-AzVMUsage -Location <azure-region>
```

Network (Virtual Networks, Static Public IPs, Public IPs, Network Security Groups, Network Interfaces, Load Balancers, Route Tables)

```
Get-AzUsage /subscriptions/<subscription-id>/providers/Microsoft.Network/locations/<azure-region> -ApiVersion 2016-03-30 | Format-Table
```

Storage (Storage Account)

```
Get-AzStorageUsage
```

- **Azure Resource Manager API throttling limits** - If you have a large enough environment (eg. > 400 VMs in a VNET), you might hit the default API throttling limits for writes (currently `1200 writes/hour`) in Azure Resource Manager. Before starting migration, you should raise a support ticket to increase this limit for your subscription.
- **Provisioning Timed Out VM Status** - If any VM has the status of `provisioning timed out`, this needs to be resolved pre-migration. The only way to do this is with downtime by deprovisioning/reprovisioning the VM (delete it, keep the disk, and recreate the VM).
- **RoleStateUnknown VM Status** - If migration halts due to a `role state unknown` error message, inspect the VM using the portal and ensure it is running. This error will typically go away on its own (no remediation required) after a few minutes and is often a transient type often seen during a Virtual Machine `start`, `stop`, `restart` operations. **Recommended practice:** re-try migration again after a few minutes.
- **Fabric Cluster does not exist** - In some cases, certain VMs cannot be migrated for various odd reasons. One of these known cases is if the VM was recently created (within the last week or so) and happened to land an Azure cluster that is not yet equipped for Azure Resource Manager workloads. You will get an error that says `fabric cluster does not exist` and the VM cannot be migrated. Waiting a couple of days will usually resolve this particular problem as the cluster will soon get Azure Resource Manager enabled. However, one immediate workaround is to `stop-deallocate` the VM, then continue forward with migration, and start the VM back up in Azure Resource Manager after migrating.

Pitfalls to avoid

- Do not take shortcuts and omit the validate/prepare/abort dry run migrations.
- Most, if not all, of your potential issues will surface during the validate/prepare/abort steps.

Migration

Technical considerations and tradeoffs

Now you are ready because you have worked through the known issues with your environment.

For the real migrations, you might want to consider:

1. Plan and schedule the virtual network (smallest unit of migration) with increasing priority. Do the simple virtual networks first, and progress with the more complicated virtual networks.
2. Most customers will have non-production and production environments. Schedule production last.

3. **(OPTIONAL)** Schedule a maintenance downtime with plenty of buffer in case unexpected issues arise.
4. Communicate with and align with your support teams in case issues arise.

Patterns of success

The technical guidance from the *Lab Test* section should be considered and mitigated prior to a real migration. With adequate testing, the migration is actually a non-event. For production environments, it might be helpful to have additional support, such as a trusted Microsoft partner or Microsoft Premier services.

Pitfalls to avoid

Not fully testing may cause issues and delay in the migration.

Beyond Migration

Technical considerations and tradeoffs

Now that you are in Azure Resource Manager, maximize the platform. Read the [overview of Azure Resource Manager](#) to find out about additional benefits.

Things to consider:

- Bundling the migration with other activities. Most customers opt for an application maintenance window. If so, you might want to use this downtime to enable other Azure Resource Manager capabilities like encryption and migration to Managed Disks.
- Revisit the technical and business reasons for Azure Resource Manager; enable the additional services available only on Azure Resource Manager that apply to your environment.
- Modernize your environment with PaaS services.

Patterns of success

Be purposeful on what services you now want to enable in Azure Resource Manager. Many customers find the below compelling for their Azure environments:

- [Role Based Access Control](#).
- [Azure Resource Manager templates for easier and more controlled deployment](#).
- [Tags](#).
- [Activity Control](#)
- [Azure Policies](#)

Pitfalls to avoid

Remember why you started this Classic to Azure Resource Manager migration journey. What were the original business reasons? Did you achieve the business reason?

Next steps

- [Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
- [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [VPN Gateway classic to Resource Manager migration](#)
- [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#)
- [Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager](#)
- [Review most common migration errors](#)
- [Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource](#)

Manager

Migrate IaaS resources from classic to Azure Resource Manager by using PowerShell

2/28/2020 • 10 minutes to read • [Edit Online](#)

IMPORTANT

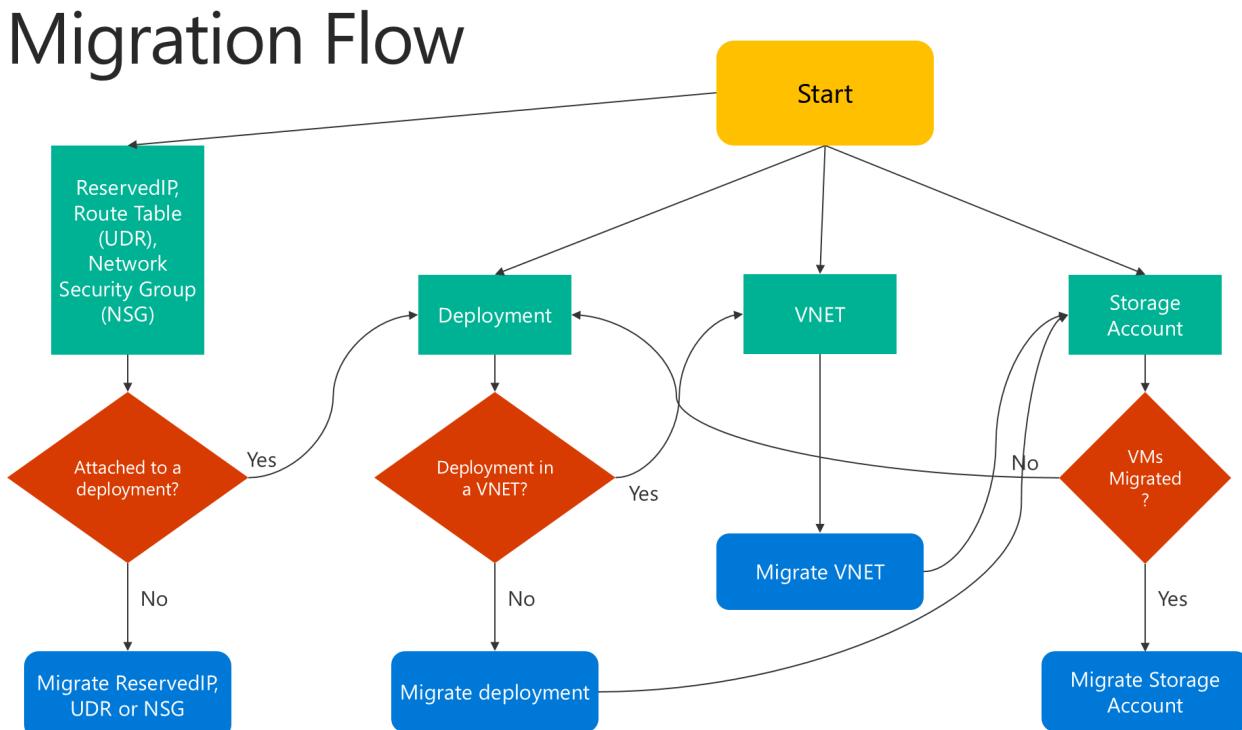
Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

These steps show you how to use Azure PowerShell commands to migrate infrastructure as a service (IaaS) resources from the classic deployment model to the Azure Resource Manager deployment model.

If you want, you can also migrate resources by using the [Azure CLI](#).

- For background on supported migration scenarios, see [Platform-supported migration of IaaS resources from classic to Azure Resource Manager](#).
- For detailed guidance and a migration walkthrough, see [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#).
- [Review the most common migration errors](#).

Here's a flowchart to identify the order in which steps need to be executed during a migration process.



Step 1: Plan for migration

Here are a few best practices that we recommend as you evaluate whether to migrate IaaS resources from classic to Resource Manager:

- Read through the [supported and unsupported features and configurations](#). If you have virtual machines that use unsupported configurations or features, wait for the configuration or feature support to be announced.

Alternatively, if it suits your needs, remove that feature or move out of that configuration to enable migration.

- If you have automated scripts that deploy your infrastructure and applications today, try to create a similar test setup by using those scripts for migration. Alternatively, you can set up sample environments by using the Azure portal.

IMPORTANT

Application gateways aren't currently supported for migration from classic to Resource Manager. To migrate a virtual network with an application gateway, remove the gateway before you run a Prepare operation to move the network. After you complete the migration, reconnect the gateway in Azure Resource Manager.

Azure ExpressRoute gateways that connect to ExpressRoute circuits in another subscription can't be migrated automatically. In such cases, remove the ExpressRoute gateway, migrate the virtual network, and re-create the gateway. For more information, see [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#).

Step 2: Install the latest version of PowerShell

There are two main options to install Azure PowerShell: [PowerShell Gallery](#) or [Web Platform Installer \(WebPI\)](#). WebPI receives monthly updates. PowerShell Gallery receives updates on a continuous basis. This article is based on Azure PowerShell version 2.1.0.

For installation instructions, see [How to install and configure Azure PowerShell](#).

Step 3: Ensure that you're an administrator for the subscription

To perform this migration, you must be added as a coadministrator for the subscription in the [Azure portal](#).

1. Sign in to the [Azure portal](#).
2. On the **Hub** menu, select **Subscription**. If you don't see it, select **All services**.
3. Find the appropriate subscription entry, and then look at the **MY ROLE** field. For a coadministrator, the value should be *Account admin*.

If you're not able to add a coadministrator, contact a service administrator or coadministrator for the subscription to get yourself added.

Step 4: Set your subscription, and sign up for migration

First, start a PowerShell prompt. For migration, set up your environment for both classic and Resource Manager.

Sign in to your account for the Resource Manager model.

```
Connect-AzAccount
```

Get the available subscriptions by using the following command:

```
Get-AzSubscription | Sort Name | Select Name
```

Set your Azure subscription for the current session. This example sets the default subscription name to **My Azure Subscription**. Replace the example subscription name with your own.

```
Select-AzSubscription -SubscriptionName "My Azure Subscription"
```

NOTE

Registration is a one-time step, but you must do it once before you attempt migration. Without registering, you see the following error message:

BadRequest : Subscription is not registered for migration.

Register with the migration resource provider by using the following command:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.ClassicInfrastructureMigrate
```

Wait five minutes for the registration to finish. Check the status of the approval by using the following command:

```
Get-AzResourceProvider -ProviderNamespace Microsoft.ClassicInfrastructureMigrate
```

Make sure that RegistrationState is `Registered` before you proceed.

Before switching to the classic deployment model, make sure that you have enough Azure Resource Manager virtual machine vCPUs in the Azure region of your current deployment or virtual network. You can use the following PowerShell command to check the current number of vCPUs you have in Azure Resource Manager. To learn more about vCPU quotas, see [Limits and the Azure Resource Manager](#).

This example checks the availability in the **West US** region. Replace the example region name with your own.

```
Get-AzVMUsage -Location "West US"
```

Now, sign in to your account for the classic deployment model.

```
Add-AzureAccount
```

Get the available subscriptions by using the following command:

```
Get-AzureSubscription | Sort SubscriptionName | Select SubscriptionName
```

Set your Azure subscription for the current session. This example sets the default subscription to **My Azure Subscription**. Replace the example subscription name with your own.

```
Select-AzureSubscription -SubscriptionName "My Azure Subscription"
```

Step 5: Run commands to migrate your IaaS resources

- [Migrate VMs in a cloud service \(not in a virtual network\)](#)
- [Migrate VMs in a virtual network](#)
- [Migrate a storage account](#)

NOTE

All the operations described here are idempotent. If you have a problem other than an unsupported feature or a configuration error, we recommend that you retry the prepare, abort, or commit operation. The platform then tries the action again.

Step 5.1: Option 1 - Migrate virtual machines in a cloud service (not in a virtual network)

Get the list of cloud services by using the following command. Then pick the cloud service that you want to migrate. If the VMs in the cloud service are in a virtual network or if they have web or worker roles, the command returns an error message.

```
Get-AzureService | ft Servicename
```

Get the deployment name for the cloud service. In this example, the service name is **My Service**. Replace the example service name with your own service name.

```
$serviceName = "My Service"  
$deployment = Get-AzureDeployment -ServiceName $serviceName  
$deploymentName = $deployment.DeploymentName
```

Prepare the virtual machines in the cloud service for migration. You have two options to choose from.

- **Option 1: Migrate the VMs to a platform-created virtual network.**

First, validate that you can migrate the cloud service by using the following commands:

```
$validate = Move-AzureService -Validate -ServiceName $serviceName `  
    -DeploymentName $deploymentName -CreateNewVirtualNetwork  
$validate.ValidationMessages
```

The following command displays any warnings and errors that block migration. If validation is successful, you can move on to the Prepare step.

```
Move-AzureService -Prepare -ServiceName $serviceName `  
    -DeploymentName $deploymentName -CreateNewVirtualNetwork
```

- **Option 2: Migrate to an existing virtual network in the Resource Manager deployment model.**

This example sets the resource group name to **myResourceGroup**, the virtual network name to **myVirtualNetwork**, and the subnet name to **mySubNet**. Replace the names in the example with the names of your own resources.

```
$existingVnetRGName = "myResourceGroup"  
$vnetName = "myVirtualNetwork"  
$subnetName = "mySubNet"
```

First, validate that you can migrate the virtual network by using the following command:

```
$validate = Move-AzureService -Validate -ServiceName $serviceName `  
    -DeploymentName $deploymentName -UseExistingVirtualNetwork -VirtualNetworkResourceGroupName  
$existingVnetRGName -VirtualNetworkName $vnetName -SubnetName $subnetName  
$validate.ValidationMessages
```

The following command displays any warnings and errors that block migration. If validation is successful, you can proceed with the following Prepare step:

```
Move-AzureService -Prepare -ServiceName $serviceName -DeploymentName $deploymentName  
-UseExistingVirtualNetwork -VirtualNetworkResourceGroupName $existingVnetRGName  
-VirtualNetworkName $vnetName -SubnetName $subnetName
```

After the Prepare operation succeeds with either of the preceding options, query the migration state of the VMs. Ensure that they're in the **Prepared** state.

This example sets the VM name to **myVM**. Replace the example name with your own VM name.

```
$vmName = "myVM"  
$vm = Get-AzureVM -ServiceName $serviceName -Name $vmName  
$vm.VM.MigrationState
```

Check the configuration for the prepared resources by using either PowerShell or the Azure portal. If you're not ready for migration and you want to go back to the old state, use the following command:

```
Move-AzureService -Abort -ServiceName $serviceName -DeploymentName $deploymentName
```

If the prepared configuration looks good, you can move forward and commit the resources by using the following command:

```
Move-AzureService -Commit -ServiceName $serviceName -DeploymentName $deploymentName
```

Step 5.1: Option 2 - Migrate virtual machines in a virtual network

To migrate virtual machines in a virtual network, you migrate the virtual network. The virtual machines automatically migrate with the virtual network. Pick the virtual network that you want to migrate.

NOTE

[Migrate a single virtual machine](#) created using the classic deployment model by creating a new Resource Manager virtual machine with Managed Disks by using the VHD (OS and data) files of the virtual machine.

NOTE

The virtual network name might be different from what is shown in the new portal. The new Azure portal displays the name as `[vnet-name]`, but the actual virtual network name is of type `Group [resource-group-name] [vnet-name]`. Before you start the migration, look up the actual virtual network name by using the command

```
Get-AzureVnetSite | Select -Property Name
```

This example sets the virtual network name to **myVnet**. Replace the example virtual network name with your own.

```
$vnetName = "myVnet"
```

NOTE

If the virtual network contains web or worker roles, or VMs with unsupported configurations, you get a validation error message.

First, validate that you can migrate the virtual network by using the following command:

```
Move-AzureVirtualNetwork -Validate -VirtualNetworkName $vnetName
```

The following command displays any warnings and errors that block migration. If validation is successful, you can proceed with the following Prepare step:

```
Move-AzureVirtualNetwork -Prepare -VirtualNetworkName $vnetName
```

Check the configuration for the prepared virtual machines by using either Azure PowerShell or the Azure portal. If you're not ready for migration and you want to go back to the old state, use the following command:

```
Move-AzureVirtualNetwork -Abort -VirtualNetworkName $vnetName
```

If the prepared configuration looks good, you can move forward and commit the resources by using the following command:

```
Move-AzureVirtualNetwork -Commit -VirtualNetworkName $vnetName
```

Step 5.2: Migrate a storage account

After you're done migrating the virtual machines, perform the following prerequisite checks before you migrate the storage accounts.

NOTE

If your storage account has no associated disks or VM data, you can skip directly to the "Validate storage accounts and start migration" section.

- Prerequisite checks if you migrated any VMs or your storage account has disk resources:

- Migrate virtual machines whose disks are stored in the storage account.

The following command returns RoleName and DiskName properties of all the VM disks in the storage account. RoleName is the name of the virtual machine to which a disk is attached. If this command returns disks, then ensure that virtual machines to which these disks are attached are migrated before you migrate the storage account.

```
$storageAccountName = 'yourStorageAccountName'  
Get-AzureDisk | where-Object {$_.MediaLink.Host.Contains($storageAccountName)} | Select-Object  
-ExpandProperty AttachedTo -Property `'  
DiskName | Format-List -Property RoleName, DiskName
```

- Delete unattached VM disks stored in the storage account.

Find unattached VM disks in the storage account by using the following command:

```
$storageAccountName = 'yourStorageAccountName'  
Get-AzureDisk | where-Object {$_.MediaLink.Host.Contains($storageAccountName)} | Where-  
Object -Property AttachedTo -EQ $null | Format-List -Property DiskName
```

If the previous command returns disks, delete these disks by using the following command:

```
Remove-AzureDisk -DiskName 'yourDiskName'
```

- Delete VM images stored in the storage account.

The following command returns all the VM images with OS disks stored in the storage account.

```
Get-AzureVmImage | Where-Object { $_.OSDiskConfiguration.MediaLink -ne $null -and  
$_.OSDiskConfiguration.MediaLink.Host.Contains($storageAccountName)`  
} | Select-Object -Property ImageName, ImageLabel
```

The following command returns all the VM images with data disks stored in the storage account.

```
Get-AzureVmImage | Where-Object {$_ .DataDiskConfigurations -ne $null `  
-and ($_.DataDiskConfigurations | Where-Object {$_ .MediaLink  
-ne $null -and $_ .MediaLink.Host.Contains($storageAccountName)}).Count -gt 0 `  
} | Select-Object -Property ImageName, ImageLabel
```

Delete all the VM images returned by the previous commands by using this command:

```
Remove-AzureVmImage -ImageName 'yourImageName'
```

- Validate storage accounts and start migration.

Validate each storage account for migration by using the following command. In this example, the storage account name is **myStorageAccount**. Replace the example name with the name of your own storage account.

```
$storageAccountName = "myStorageAccount"  
Move-AzureStorageAccount -Validate -StorageAccountName $storageAccountName
```

The next step is to prepare the storage account for migration.

```
$storageAccountName = "myStorageAccount"  
Move-AzureStorageAccount -Prepare -StorageAccountName $storageAccountName
```

Check the configuration for the prepared storage account by using either Azure PowerShell or the Azure portal. If you're not ready for migration and you want to go back to the old state, use the following command:

```
Move-AzureStorageAccount -Abort -StorageAccountName $storageAccountName
```

If the prepared configuration looks good, you can move forward and commit the resources by using the following command:

```
Move-AzureStorageAccount -Commit -StorageAccountName $storageAccountName
```

Next steps

- Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager
- Technical deep dive on platform-supported migration from classic to Azure Resource Manager
- Planning for migration of IaaS resources from classic to Azure Resource Manager
- Use CLI to migrate IaaS resources from classic to Azure Resource Manager
- Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager
- Review most common migration errors
- Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager

Common errors during Classic to Azure Resource Manager migration

2/28/2020 • 9 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

This article catalogs the most common errors and mitigations during the migration of IaaS resources from Azure classic deployment model to the Azure Resource Manager stack.

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

List of errors

ERROR STRING	MITIGATION
Internal server error	In some cases, this is a transient error that goes away with a retry. If it continues to persist, contact Azure support as it needs investigation of platform logs. NOTE: Once the incident is tracked by the support team, please do not attempt any self-mitigation as this might have unintended consequences on your environment.
Migration is not supported for Deployment {deployment-name} in HostedService {hosted-service-name} because it is a PaaS deployment (Web/Worker).	This happens when a deployment contains a web/worker role. Since migration is only supported for Virtual Machines, please remove the web/worker role from the deployment and try migration again.
Template {template-name} deployment failed. CorrelationId= {guid}	In the backend of migration service, we use Azure Resource Manager templates to create resources in the Azure Resource Manager stack. Since templates are idempotent, usually you can safely retry the migration operation to get past this error. If this error continues to persist, please contact Azure support and give them the CorrelationId. NOTE: Once the incident is tracked by the support team, please do not attempt any self-mitigation as this might have unintended consequences on your environment.
The virtual network {virtual-network-name} does not exist.	This can happen if you created the Virtual Network in the new Azure portal. The actual Virtual Network name follows the pattern "Group * < VNET name >"

ERROR STRING	MITIGATION
VM {vm-name} in HostedService {hosted-service-name} contains Extension {extension-name} which is not supported in Azure Resource Manager. It is recommended to uninstall it from the VM before continuing with migration.	XML extensions such as BGInfo 1.* are not supported in Azure Resource Manager. Therefore, these extensions cannot be migrated. If these extensions are left installed on the virtual machine, they are automatically uninstalled before completing the migration.
VM {vm-name} in HostedService {hosted-service-name} contains Extension VMSnapshot/VMSnapshotLinux, which is currently not supported for Migration. Uninstall it from the VM and add it back using Azure Resource Manager after the Migration is Complete	This is the scenario where the virtual machine is configured for Azure Backup. Since this is currently an unsupported scenario, please follow the workaround at https://aka.ms/vmbakcupmigration
VM {vm-name} in HostedService {hosted-service-name} contains Extension {extension-name} whose Status is not being reported from the VM. Hence, this VM cannot be migrated. Ensure that the Extension status is being reported or uninstall the extension from the VM and retry migration.	Azure guest agent & VM Extensions need outbound internet access to the VM storage account to populate their status. Common causes of status failure include <ul style="list-style-type: none"> • a Network Security Group that blocks outbound access to the internet • If the VNET has on premises DNS servers and DNS connectivity is lost
VM {vm-name} in HostedService {hosted-service-name} contains Extension {extension-name} reporting Handler Status: {handler-status}. Hence, the VM cannot be migrated. Ensure that the Extension handler status being reported is {handler-status} or uninstall it from the VM and retry migration.	If you continue to see an unsupported status, you can uninstall the extensions to skip this check and move forward with migration.
VM Agent for VM {vm-name} in HostedService {hosted-service-name} is reporting the overall agent status as Not Ready. Hence, the VM may not be migrated, if it has a migratable extension. Ensure that the VM Agent is reporting overall agent status as Ready. Refer to https://aka.ms/classiciaasmigrationfaqs .	
Migration is not supported for Deployment {deployment-name} in HostedService {hosted-service-name} because it has multiple Availability Sets.	Currently, only hosted services that have 1 or less Availability sets can be migrated. To work around this problem, please move the additional Availability sets and Virtual machines in those Availability sets to a different hosted service.
Migration is not supported for Deployment {deployment-name} in HostedService {hosted-service-name} because it has VMs that are not part of the Availability Set even though the HostedService contains one.	The workaround for this scenario is to either move all the virtual machines into a single Availability set or remove all Virtual machines from the Availability set in the hosted service.
Storage account/HostedService/Virtual Network {virtual-network-name} is in the process of being migrated and hence cannot be changed	This error happens when the "Prepare" migration operation has been completed on the resource and an operation that would make a change to the resource is triggered. Because of the lock on the management plane after "Prepare" operation, any changes to the resource are blocked. To unlock the management plane, you can run the "Commit" migration operation to complete migration or the "Abort" migration operation to roll back the "Prepare" operation.
Migration is not allowed for HostedService {hosted-service-name} because it has VM {vm-name} in State: RoleStateUnknown. Migration is allowed only when the VM is in one of the following states - Running, Stopped, Stopped Deallocated.	The VM might be undergoing through a state transition, which usually happens when during an update operation on the HostedService such as a reboot, extension installation etc. It is recommended for the update operation to complete on the HostedService before trying migration.

ERROR STRING	MITIGATION
Deployment {deployment-name} in HostedService {hosted-service-name} contains a VM {vm-name} with Data Disk {data-disk-name} whose physical blob size {size-of-the-vhd-blob-backing-the-data-disk} bytes does not match the VM Data Disk logical size {size-of-the-data-disk-specified-in-the-vm-api} bytes. Migration will proceed without specifying a size for the data disk for the Azure Resource Manager VM.	This error happens if you've resized the VHD blob without updating the size in the VM API model. Detailed mitigation steps are outlined below .
A storage exception occurred while validating data disk {data-disk-name} with media link {data-disk-uri} for VM {VM name} in Cloud Service {Cloud Service name}. Please ensure that the VHD media link is accessible for this virtual machine	This error can happen if the disks of the VM have been deleted or are not accessible anymore. Please make sure the disks for the VM exist.
VM {vm-name} in HostedService {cloud-service-name} contains Disk with MediaLink {vhd-uri} which has blob name {vhd-blob-name} that is not supported in Azure Resource Manager.	This error occurs when the name of the blob has a "/" in it which is not supported in Compute Resource Provider currently.
Migration is not allowed for Deployment {deployment-name} in HostedService {cloud-service-name} as it is not in the regional scope. Please refer to https://aka.ms/regionscope for moving this deployment to regional scope.	In 2014, Azure announced that networking resources will move from a cluster level scope to regional scope. See https://aka.ms/regionscope for more details . This error happens when the deployment being migrated has not had an update operation, which automatically moves it to a regional scope. Best workaround is to either add an endpoint to a VM or a data disk to the VM and then retry migration. See How to set up endpoints on a classic Windows virtual machine in Azure or Attach a data disk to a Windows virtual machine created with the classic deployment model
Migration is not supported for Virtual Network {vnet-name} because it has non-gateway PaaS deployments.	This error occurs when you have non-gateway PaaS deployments such as Application Gateway or API Management services that are connected to the Virtual Network.

Detailed mitigations

VM with Data Disk whose physical blob size bytes does not match the VM Data Disk logical size bytes.

This happens when the Data disk logical size can get out of sync with the actual VHD blob size. This can be easily verified using the following commands:

Verifying the issue

```

# Store the VM details in the VM object
$vm = Get-AzureVM -ServiceName $servicename -Name $vmname

# Display the data disk properties
# NOTE the data disk LogicalDiskSizeInGB below which is 11GB. Also note the MediaLink Uri of the VHD blob as
we'll use this in the next step
$vm.VM.DataVirtualHardDisks


HostCaching      : None
DiskLabel        :
DiskName         : coreosvm-coreosvm-0-201611230636240687
Lun              : 0
LogicalDiskSizeInGB : 11
MediaLink        : https://contosostorage.blob.core.windows.net/vhds/coreosvm-dd1.vhd
SourceMediaLink   :
IOType           : Standard
ExtensionData    :

# Now get the properties of the blob backing the data disk above
# NOTE the size of the blob is about 15 GB which is different from LogicalDiskSizeInGB above
blob = Get-AzStorageblob -Blob "coreosvm-dd1.vhd" -Container vhds

blob

ICloudBlob      : Microsoft.WindowsAzure.Storage.Blob.CloudPageBlob
BlobType        : PageBlob
Length          : 16106127872
ContentType     : application/octet-stream
LastModified    : 11/23/2016 7:16:22 AM +00:00
SnapshotTime    :
ContinuationToken :
Context          : Microsoft.WindowsAzure.Commands.Common.Storage.AzureStorageContext
Name            : coreosvm-dd1.vhd

```

Mitigating the issue

```

# Convert the blob size in bytes to GB into a variable which we'll use later
$newSize = [int]($blob.Length / 1GB)

# See the calculated size in GB
$newSize

15

# Store the disk name of the data disk as we'll use this to identify the disk to be updated
$diskName = $vm.VM.DataVirtualHardDisks[0].DiskName

# Identify the LUN of the data disk to remove
$lunToRemove = $vm.VM.DataVirtualHardDisks[0].Lun

# Now remove the data disk from the VM so that the disk isn't leased by the VM and it's size can be updated
Remove-AzureDataDisk -LUN $lunToRemove -VM $vm | Update-AzureVm -Name $vmname -ServiceName $servicename

OperationDescription OperationId          OperationStatus
----- ----- -----
Update-AzureVM      213xx1-b44b-1v6n-23gg-591f2a13cd16 Succeeded

# Verify we have the right disk that's going to be updated
Get-AzureDisk -DiskName $diskName

AffinityGroup      :
AttachedTo        :
IsCorrupted      : False
Label             :
Location          : East US
DiskSizeInGB     : 11

```

```

DISKSIZEINGB          : 11
MediaLink              : https://contosostorage.blob.core.windows.net/vhds/coreosvm-dd1.vhd
DiskName               : coreosvm-coreosvm-0-201611230636240687
SourceImageName        :
OS                     :
IOType                 : Standard
OperationDescription   : Get-AzureDisk
OperationId            : 0c56a2b7-a325-123b-7043-74c27d5a61fd
OperationStatus         : Succeeded

# Now update the disk to the new size
Update-AzureDisk -DiskName $diskName -ResizedSizeInGB $newSize -Label $diskName

OperationDescription OperationId          OperationStatus
-----  -----  -----
Update-AzureDisk      cv134b65-1b6n-8908-abuo-ce9e395ac3e7 Succeeded

# Now verify that the "DiskSizeInGB" property of the disk matches the size of the blob
Get-AzureDisk -DiskName $diskName

AffinityGroup          :
AttachedTo              :
IsCorrupted             : False
Label                  : coreosvm-coreosvm-0-201611230636240687
Location                : East US
DiskSizeInGB            : 15
MediaLink               : https://contosostorage.blob.core.windows.net/vhds/coreosvm-dd1.vhd
DiskName                : coreosvm-coreosvm-0-201611230636240687
SourceImageName         :
OS                     :
IOType                 : Standard
OperationDescription   : Get-AzureDisk
OperationId            : 1v53bde5-cv56-5621-9078-16b9c8a0bad2
OperationStatus         : Succeeded

# Now we'll add the disk back to the VM as a data disk. First we need to get an updated VM object
$vm = Get-AzureVM -ServiceName $servicename -Name $vmname

Add-AzureDataDisk -Import -DiskName $diskName -LUN 0 -VM $vm -HostCaching ReadWrite | Update-AzureVm -Name
$vmname -ServiceName $servicename

OperationDescription OperationId          OperationStatus
-----  -----  -----
Update-AzureVM       b0ad3d4c-4v68-45vb-xxc1-134fd010d0f8 Succeeded

```

Moving a VM to a different subscription after completing migration

After you complete the migration process, you may want to move the VM to another subscription. However, if you have a secret/certificate on the VM that references a Key Vault resource, the move is currently not supported. The below instructions will allow you to workaround the issue.

PowerShell

```

$vm = Get-AzVM -ResourceGroupName "MyRG" -Name "MyVM"
Remove-AzVMSecret -VM $vm
Update-AzVM -ResourceGroupName "MyRG" -VM $vm

```

Azure CLI

```
az vm update -g "myrg" -n "myvm" --set osProfile.Secrets=[]
```

Next steps

- Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager
- Technical deep dive on platform-supported migration from classic to Azure Resource Manager
- Planning for migration of IaaS resources from classic to Azure Resource Manager
- Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager
- Use CLI to migrate IaaS resources from classic to Azure Resource Manager
- Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager
- Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager

Community tools to migrate IaaS resources from classic to Azure Resource Manager

2/28/2020 • 2 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

This article catalogs the tools that have been provided by the community to assist with migration of IaaS resources from classic to the Azure Resource Manager deployment model.

NOTE

These tools are not officially supported by Microsoft Support. Therefore they are open sourced on GitHub and we're happy to accept PRs for fixes or additional scenarios. To report an issue, use the GitHub issues feature.

Migrating with these tools will cause downtime for your classic Virtual Machine. If you're looking for platform supported migration, visit

- [Platform supported migration of IaaS resources from Classic to Azure Resource Manager stack](#)
- [Technical Deep Dive on Platform supported migration from Classic to Azure Resource Manager](#)
- [Migrate IaaS resources from Classic to Azure Resource Manager using Azure PowerShell](#)

AsmMetadataParser

This is a collection of helper tools created as part of enterprise migrations from Azure Service Management to Azure Resource Manager. This tool allows you to replicate your infrastructure into another subscription which can be used for testing migration and iron out any issues before running the migration on your Production subscription.

[Link to the tool documentation](#)

migAz

migAz is an additional option to migrate a complete set of classic IaaS resources to Azure Resource Manager IaaS resources. The migration can occur within the same subscription or between different subscriptions and subscription types (ex: CSP subscriptions).

[Link to the tool documentation](#)

Next Steps

- [Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
- [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
- [Planning for migration of IaaS resources from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Review most common migration errors](#)

- Review the most frequently asked questions about migrating IaaS resources from classic to Azure Resource Manager

Frequently asked questions about classic to Azure Resource Manager migration

2/28/2020 • 6 minutes to read • [Edit Online](#)

IMPORTANT

Today, about 90% of IaaS VMs are using [Azure Resource Manager](#). As of February 28, 2020, classic VMs have been deprecated and will be fully retired on March 1, 2023. [Learn more](#) about this deprecation and [how it affects you](#).

What is the time required for migration?

Planning and execution of migration greatly depends on the complexity of the architecture and could take couple of months.

What is the definition of a new customer on IaaS VMs (classic)?

Customers who did not have IaaS VMs (classic) in their subscriptions in the month of February 2020 (a month before deprecation started) are considered as new customers.

Does this migration plan affect any of my existing services or applications that run on Azure virtual machines?

Not until March 1st, 2023 for IaaS VMs (classic). The IaaS VMs (classic) are fully supported services in general availability. You can continue to use these resources to expand your footprint on Microsoft Azure. On March 1st, 2023, these VMs will be fully retired and any active or allocated VMs will be stopped & deallocated. There will be no impact to other classic resources like Cloud Services (Classic), Storage Accounts (Classic), etc.

What happens to my VMs if I don't plan on migrating in the near future?

On March 1st, 2023, the IaaS VMs (Classic) will be fully retired and any active or allocated VMs will be stopped & deallocated. To prevent business impact, we highly recommend to start planning your migration today and complete it before March 1st, 2023. We are not deprecating the existing classic APIs, Cloud Services and resource model. We want to make migration easy, considering the advanced features that are available in the Resource Manager deployment model. We recommend that you start planning to migrate these resources to Azure Resource Manager.

What does this migration plan mean for my existing tooling?

Updating your tooling to the Resource Manager deployment model is one of the most important changes that you have to account for in your migration plans.

How long will the management-plane downtime be?

It depends on the number of resources that are being migrated. For smaller deployments (a few tens of VMs), the whole migration should take less than an hour. For large-scale deployments (hundreds of VMs), the migration can take a few hours.

Can I roll back after my migrating resources are committed in Resource Manager?

You can abort your migration as long as the resources are in the prepared state. Rollback is not supported after the resources have been successfully migrated through the commit operation.

Can I roll back my migration if the commit operation fails?

You cannot abort migration if the commit operation fails. All migration operations, including the commit operation, are idempotent. So we recommend that you retry the operation after a short time. If you still face an error, create a support ticket.

Do I have to buy another express route circuit if I have to use IaaS under Resource Manager?

No. We recently enabled [moving ExpressRoute circuits from the classic to the Resource Manager deployment model](#). You don't have to buy a new ExpressRoute circuit if you already have one.

What if I had configured Role-Based Access Control policies for my classic IaaS resources?

During migration, the resources transform from classic to Resource Manager. So we recommend that you plan the RBAC policy updates that need to happen after migration.

I backed up my classic VMs in a vault. Can I migrate my VMs from classic mode to Resource Manager mode and protect them in a Recovery Services vault?

When you move a VM from classic to Resource Manager mode, backups taken prior to migration will not migrate to newly migrated Resource Manager VM. However, if you wish to keep your backups of classic VMs, follow these steps before the migration.

1. In the Recovery Services vault, go to the **Protected Items** tab and select the VM.
2. Click Stop Protection. Leave *Delete associated backup data* option **unchecked**.

NOTE

You will be charged backup instance cost till you retain data. Backup copies will be pruned as per retention range. However, last backup copy is always kept until you explicitly delete backup data. It is advised to check your retention range of the Virtual machine and trigger "Delete Backup Data" on the protected item in the vault once the retention range is over.

To migrate the virtual machine to Resource Manager mode,

1. Delete the backup/snapshot extension from the VM.
2. Migrate the virtual machine from classic mode to Resource Manager mode. Make sure the storage and network information corresponding to the virtual machine is also migrated to Resource Manager mode.

Additionally, if you want to back up the migrated VM, go to Virtual Machine management blade to [enable backup](#).

Can I validate my subscription or resources to see if they're capable of migration?

Yes. In the platform-supported migration option, the first step in preparing for migration is to validate that the resources are capable of migration. In case the validate operation fails, you receive messages for all the reasons the migration cannot be completed.

What happens if I run into a quota error while preparing the IaaS resources for migration?

We recommend that you abort your migration and then log a support request to increase the quotas in the region where you are migrating the VMs. After the quota request is approved, you can start executing the migration steps again.

How do I report an issue?

Post your issues and questions about migration to our [VM forum](#), with the keyword ClassicIaaSMigration. We recommend posting all your questions on this forum. If you have a support contract, you're welcome to log a support ticket as well.

What if I don't like the names of the resources that the platform chose during migration?

All the resources that you explicitly provide names for in the classic deployment model are retained during migration. In some cases, new resources are created. For example: a network interface is created for every VM. We currently don't support the ability to control the names of these new resources created during migration. Log your votes for this feature on the [Azure feedback forum](#).

Can I migrate ExpressRoute circuits used across subscriptions with authorization links?

ExpressRoute circuits which use cross-subscription authorization links cannot be migrated automatically without downtime. We have guidance on how these can be migrated using manual steps. See [Migrate ExpressRoute circuits and associated virtual networks from the classic to the Resource Manager deployment model](#) for steps and more information.

I got the message "VM is reporting the overall agent status as Not Ready. Hence, the VM cannot be migrated. Ensure that the VM Agent is reporting overall agent status as Ready" or "VM contains Extension whose Status is not being reported from the VM. Hence, this VM cannot be migrated."

This message is received when the VM does not have outbound connectivity to the internet. The VM agent uses outbound connectivity to reach the Azure storage account for updating the agent status every five minutes.

Next steps

- [Overview of platform-supported migration of IaaS resources from classic to Azure Resource Manager](#)
- [Technical deep dive on platform-supported migration from classic to Azure Resource Manager](#)
- [Planning for migration of IaaS resources from classic to Azure Resource Manager](#)
- [Use PowerShell to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Use CLI to migrate IaaS resources from classic to Azure Resource Manager](#)
- [Community tools for assisting with migration of IaaS resources from classic to Azure Resource Manager](#)

- Review most common migration errors

Security recommendations for Windows virtual machines in Azure

11/13/2019 • 3 minutes to read • [Edit Online](#)

This article contains security recommendations for Azure Virtual Machines. Follow these recommendations to help fulfill the security obligations described in our model for shared responsibility. The recommendations will also help you improve overall security for your web app solutions. For more information about what Microsoft does to fulfill service-provider responsibilities, see [Shared responsibilities for cloud computing](#).

Some of this article's recommendations can be automatically addressed by Azure Security Center. Azure Security Center is the first line of defense for your resources in Azure. It periodically analyzes the security state of your Azure resources to identify potential security vulnerabilities. It then recommends how to address the vulnerabilities. For more information, see [Security recommendations in Azure Security Center](#).

For general information about Azure Security Center, see [What is Azure Security Center?](#).

General

RECOMMENDATION	COMMENTS	SECURITY CENTER
When you build custom VM images, apply the latest updates.	Before you create images, install the latest updates for the operating system and for all applications that will be part of your image.	-
Keep your VMs current.	You can use the Update Management solution in Azure Automation to manage operating system updates for your Windows and Linux computers in Azure.	Yes
Back up your VMs.	Azure Backup helps protect your application data and has minimal operating costs. Application errors can corrupt your data, and human errors can introduce bugs into your applications. Azure Backup protects your VMs that run Windows and Linux.	-
Use multiple VMs for greater resilience and availability.	If your VM runs applications that must be highly available, use multiple VMs or availability sets .	-
Adopt a business continuity and disaster recovery (BCDR) strategy.	Azure Site Recovery allows you to choose from different options designed to support business continuity. It supports different replication and failover scenarios. For more information, see About Site Recovery .	-

Data security

RECOMMENDATION	COMMENTS	SECURITY CENTER
Encrypt operating system disks.	Azure Disk Encryption helps you encrypt your Windows and Linux IaaS VM disks. Without the necessary keys, the contents of encrypted disks are unreadable. Disk encryption protects stored data from unauthorized access that would otherwise be possible if the disk were copied.	Yes
Encrypt data disks.	Azure Disk Encryption helps you encrypt your Windows and Linux IaaS VM disks. Without the necessary keys, the contents of encrypted disks are unreadable. Disk encryption protects stored data from unauthorized access that would otherwise be possible if the disk were copied.	-
Limit installed software.	Limit installed software to what is required to successfully apply your solution. This guideline helps reduce your solution's attack surface.	-
Use antivirus or antimalware.	In Azure, you can use antimalware software from security vendors such as Microsoft, Symantec, Trend Micro, and Kaspersky. This software helps protect your VMs from malicious files, adware, and other threats. You can deploy Microsoft Antimalware based on your application workloads. Use either basic secure-by-default or advanced custom configuration. For more information, see Microsoft Antimalware for Azure Cloud Services and Virtual Machines .	-
Securely store keys and secrets.	Simplify the management of your secrets and keys by providing your application owners with a secure, centrally managed option. This management reduces the risk of an accidental compromise or leak. Azure Key Vault can securely store your keys in hardware security modules (HSMs) that are certified to FIPS 140-2 Level 2. If you need to use FIPs 140.2 Level 3 to store your keys and secrets, you can use Azure Dedicated HSM .	-

Identity and access management

RECOMMENDATION	COMMENTS	SECURITY CENTER
Centralize VM authentication.	You can centralize the authentication of your Windows and Linux VMs by using Azure Active Directory authentication .	-

Monitoring

RECOMMENDATION	COMMENTS	SECURITY CENTER
Monitor your VMs.	You can use Azure Monitor for VMs to monitor the state of your Azure VMs and virtual machine scale sets. Performance issues with a VM can lead to service disruption, which violates the security principle of availability.	-

Networking

RECOMMENDATION	COMMENTS	SECURITY CENTER
Restrict access to management ports.	Attackers scan public cloud IP ranges for open management ports and attempt "easy" attacks like common passwords and known unpatched vulnerabilities. You can use just-in-time (JIT) VM access to lock down inbound traffic to your Azure VMs, reducing exposure to attacks while providing easy connections to VMs when they're needed.	-
Limit network access.	Network security groups allow you to restrict network access and control the number of exposed endpoints. For more information, see Create, change, or delete a network security group .	-

Next steps

Check with your application provider to learn about additional security requirements. For more information about developing secure applications, see [Secure-development documentation](#).

Secure your management ports with just-in-time access

2/25/2020 • 11 minutes to read • [Edit Online](#)

If you're on Security Center's standard pricing tier (see [pricing](#)), you can lock down inbound traffic to your Azure VMs with just-in-time (JIT) virtual machine (VM) access. This reduces exposure to attacks while providing easy access to connect to VMs when needed.

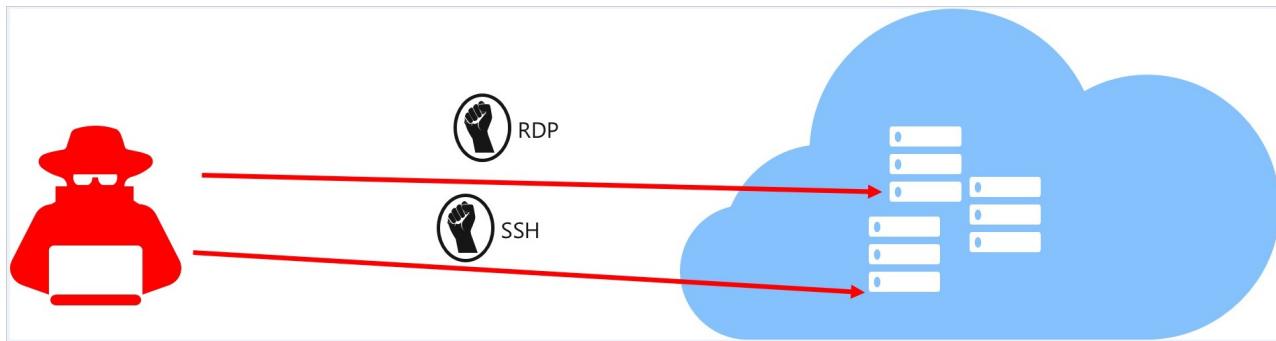
NOTE

Security Center just-in-time VM access currently supports only VMs deployed through Azure Resource Manager. To learn more about the classic and Resource Manager deployment models see [Azure Resource Manager vs. classic deployment](#).

Attack scenario

Brute force attacks commonly target management ports as a means to gain access to a VM. If successful, an attacker can take control over the VM and establish a foothold into your environment.

One way to reduce exposure to a brute force attack is to limit the amount of time that a port is open. Management ports don't need to be open at all times. They only need to be open while you're connected to the VM, for example to perform management or maintenance tasks. When just-in-time is enabled, Security Center uses [network security group](#) (NSG) and Azure Firewall rules, which restrict access to management ports so they cannot be targeted by attackers.



How does JIT access work?

When just-in-time is enabled, Security Center locks down inbound traffic to your Azure VMs by creating an NSG rule. You select the ports on the VM to which inbound traffic will be locked down. These ports are controlled by the just-in-time solution.

When a user requests access to a VM, Security Center checks that the user has [Role-Based Access Control \(RBAC\)](#) permissions for that VM. If the request is approved, Security Center automatically configures the Network Security Groups (NSGs) and Azure Firewall to allow inbound traffic to the selected ports and requested source IP addresses or ranges, for the amount of time that was specified. After the time has expired, Security Center restores the NSGs to their previous states. Those connections that are already established are not being interrupted, however.

NOTE

If a JIT access request is approved for a VM behind an Azure Firewall, then Security Center automatically changes both the NSG and firewall policy rules. For the amount of time that was specified, the rules allow inbound traffic to the selected ports and requested source IP addresses or ranges. After the time is over, Security Center restores the firewall and NSG rules to their previous states.

Permissions needed to configure and use JIT

TO ENABLE A USER TO:	PERMISSIONS TO SET
Configure or edit a JIT policy for a VM	<p><i>Assign these actions to the role:</i></p> <ul style="list-style-type: none"> On the scope of a subscription or resource group that is associated with the VM: <code>Microsoft.Security/locations/jitNetworkAccessPolicies/write</code> On the scope of a subscription or resource group of VM: <code>Microsoft.Compute/virtualMachines/write</code>
Request JIT access to a VM	<p><i>Assign these actions to the user:</i></p> <ul style="list-style-type: none"> On the scope of a subscription or resource group that is associated with the VM: <code>Microsoft.Security/locations/jitNetworkAccessPolicies/initiate/*</code> On the scope of a subscription or resource group that is associated with the VM: <code>Microsoft.Security/locations/jitNetworkAccessPolicies/*/read</code> On the scope of a subscription or resource group or VM: <code>Microsoft.Compute/virtualMachines/read</code> On the scope of a subscription or resource group or VM: <code>Microsoft.Network/networkInterfaces/*/read</code>

Configure JIT on a VM

There are three ways to configure a JIT policy on a VM:

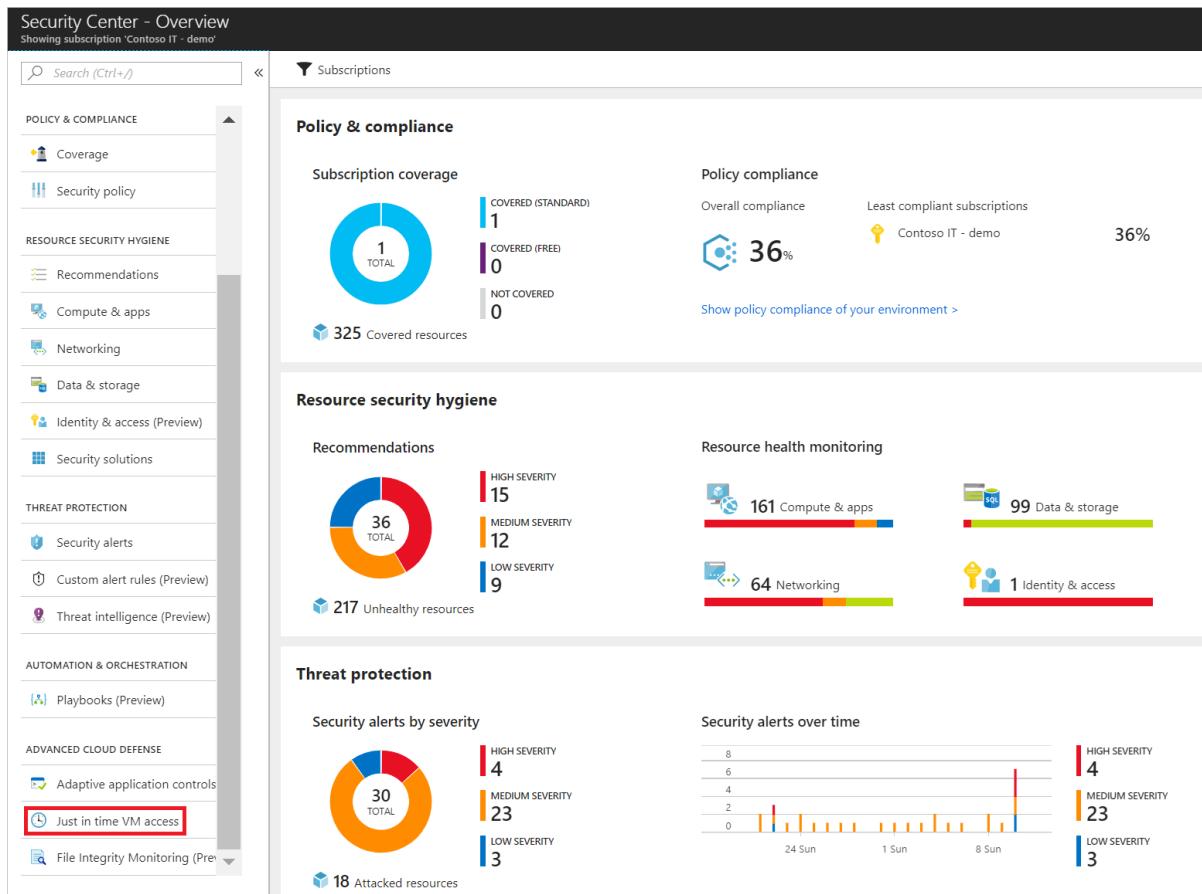
- Configure JIT access in Azure Security Center
- Configure JIT access in an Azure VM page
- Configure a JIT policy on a VM programmatically

Configure JIT in Azure Security Center

From Security Center, you can configure a JIT policy and request access to a VM using a JIT policy

Configure JIT access on a VM in Security Center

- Open the **Security Center** dashboard.
- In the left pane, select **Just-in-time VM access**.



The **Just-in-time VM access** window opens and shows information on the state of your VMs:

- **Configured** - VMs that have been configured to support just-in-time VM access. The data presented is for the last week and includes for each VM the number of approved requests, last access date and time, and last user.
- **Recommended** - VMs that can support just-in-time VM access but haven't been configured to. We recommend that you enable just-in-time VM access control for these VMs.
- **No recommendation** - Reasons that can cause a VM not to be recommended are:
 - Missing NSG - The just-in-time solution requires an NSG to be in place.
 - Classic VM - Security Center just-in-time VM access currently supports only VMs deployed through Azure Resource Manager. A classic deployment is not supported by the just-in-time solution.
 - Other - A VM is in this category if the just-in-time solution is turned off in the security policy of the subscription or the resource group, or if the VM is missing a public IP and doesn't have an NSG in place.

3. Select the **Recommended** tab.
4. Under **VIRTUAL MACHINE**, click the VMs that you want to enable. This puts a checkmark next to a VM.

Virtual machines

Configured Recommended No recommendation

VMs for which we recommend you to apply the just in time VM access control.

61 VMs

[Enable JIT on 2 VMs](#)

Search to filter items...

VIRTUAL MACHINE	STATE	SEVERITY
AA-Contoso-01	Open	! High
<input checked="" type="checkbox"/> App01	Open	! High
App03	Open	! High
App04	Open	! High
App05	Open	! High
App06	Open	! High
<input checked="" type="checkbox"/> App07	Open	! High
App08	Open	! High
App09	Open	! High

5. Click **Enable JIT on VMs**. A pane opens displaying the default ports recommended by Azure Security Center:

- 22 - SSH
- 3389 - RDP
- 5985 - WinRM
- 5986 - WinRM

6. Optionally, you can add custom ports to the list:

- a. Click **Add**. The **Add port configuration** window opens.
- b. For each port you choose to configure, both default and custom, you can customize the following settings:
 - **Protocol type**- The protocol that is allowed on this port when a request is approved.
 - **Allowed source IP addresses**- The IP ranges that are allowed on this port when a request is approved.
 - **Maximum request time**- The maximum time window during which a specific port can be opened.
- c. Click **OK**.

7. Click **Save**.

NOTE

When JIT VM Access is enabled for a VM, Azure Security Center creates "deny all inbound traffic" rules for the selected ports in the network security groups associated and Azure Firewall with it. If other rules had been created for the selected ports, then the existing rules take priority over the new "deny all inbound traffic" rules. If there are no existing rules on the selected ports, then the new "deny all inbound traffic" rules take top priority in the Network Security Groups and Azure Firewall.

Request JIT access via Security Center

To request access to a VM via Security Center:

1. Under **Just-in-time VM access**, select the **Configured** tab.

2. Under **Virtual Machine**, click the VMs that you want to request access for. This puts a checkmark next to the VM.

- The icon in the **Connection Details** column indicates whether JIT is enabled on the NSG or FW. If it's enabled on both, only the Firewall icon appears.
- The **Connection Details** column provides the information required to connect the VM, and its open ports.

Virtual machines						
Configured Recommended No recommendation						
VMs for which the just in time VM access control is already in place. Presented data is for the last week.						
92 VMs						Request access
VIRTUAL MACHINE	APPROVED	LAST ACCESS	CONNECTION DETAILS	LAST USER		
af-vm	1 Requests	5/7/19, 11:05 AM	13.64.24.215:13389	user@contoso.com	...	
srvworkload2	1 Requests	5/7/19, 11:30 AM	20.185.107.87:10022	user@contoso.com	...	
sc2019	2 Requests	5/7/19, 11:39 AM	3 Ports	user@contoso.com	...	
bengr-jit-mul-1	1 Requests	Active now	Ports: 5986, 22, 3389	user@contoso.com	...	
LBWeb0	0 Requests	N/A	-	N/A	...	
LBWeb1	0 Requests	N/A	-	N/A	...	
muliport1	0 Requests	N/A	-	N/A	...	
<input checked="" type="checkbox"/> muliport0	0 Requests	N/A	-	N/A	...	
WebApp1	0 Requests	N/A	-	N/A	...	
WinVM	0 Requests	N/A	-	N/A	...	
vm2	0 Requests	N/A	-	N/A	...	
BarWaFT2Jun3	0 Requests	N/A	-	N/A	...	
bengr-jit-mul-2	0 Requests	N/A	-	N/A	...	
ChkpJun3	0 Requests	N/A	-	N/A	...	

3. Click **Request access**. The **Request access** window opens.

Request access

Please select the ports that you would like to open per virtual machine.

PORT	TOGGLE	ALLOWED SOURCE IP	IP RANGE	TIMERANGE
▼ vm1				
22	<input checked="" type="button"/> On <input type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
3389	<input checked="" type="button"/> On <input type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
5985	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
5986	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
▼ vm2				
22	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
3389	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	2
5985	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3
5986	<input type="button"/> On <input checked="" type="button"/> Off	<input type="button"/> My IP <input type="button"/> IP Range	<input type="button"/> No range	3

Open ports

4. Under **Request access**, for each VM, configure the ports that you want to open and the source IP addresses that the port is opened on and the time window for which the port will be open. It will only be possible to request access to the ports that are configured in the just-in-time policy. Each port has a maximum allowed time derived from the just-in-time policy.

5. Click **Open ports**.

NOTE

If a user who is requesting access is behind a proxy, the option **My IP** may not work. You may need to define the full IP address range of the organization.

Edit a JIT access policy via Security Center

You can change a VM's existing just-in-time policy by adding and configuring a new port to protect for that VM, or by changing any other setting related to an already protected port.

To edit an existing just-in-time policy of a VM:

1. In the **Configured** tab, under **VMs**, select a VM to which to add a port by clicking on the three dots within the row for that VM.
2. Select **Edit**.
3. Under **JIT VM access configuration**, you can either edit the existing settings of an already protected port or add a new custom port.

The screenshot displays two windows from the Azure Security Center interface:

- Just in time VM access**: This window shows a summary of configured VMs. It includes sections for "What is just in time VM access?", "How does it work?", and a table of VMs. The "Configured" tab is selected, highlighted with a red box. A tooltip for "Edit" is shown over the three-dot menu for the first VM, "vm1".
- JIT VM access configuration**: This window shows the configuration details for a specific VM. It has tabs for "Add", "Save", and "Discard". The table lists ports 22 and 3389 with their respective settings: Any protocol, Per request allowed source, and N/A IP range, all valid for 3 hours.

Audit JIT access activity in Security Center

You can gain insights into VM activities using log search. To view logs:

1. Under **Just-in-time VM access**, select the **Configured** tab.
2. Under **VMs**, select a VM to view information about by clicking on the three dots within the row for that VM and select **Activity Log** from the menu. The **Activity log** opens.

Just in time VM access

Last week

What is just-in-time VM access?

Just in time VM access enables you to lock down your VMs in the network level by blocking inbound traffic to specific ports. It enables you to control the access and reduce the attack surface to your VMs, by allowing access only upon a specific need.

How does it work?

Upon a user request, based on Azure RBAC, Security Center will decide whether to grant access. If a request is approved, Security Center automatically configures the NSGs to allow inbound traffic to these ports, for only 3 hours, after which it restores the NSGs to their previous states.

[For more information go to the documentation >](#)

Virtual machines

Configured	Recommended	No recommendation	
5 VMs			
<input type="text"/> Search to filter items...			
VIRTUAL MACHINE	APPROVED	LAST ACCESS	LAST USER
vm1	6 Requests	17/07/17, 15:27	Properties Activity Log  Edit Remove
vm2	4 Requests	13/07/17, 17:37	
vm3	0 Requests	N/A	
vm2WL	0 Requests	N/A	
vm3WL	0 Requests	N/A	

Activity log provides a filtered view of previous operations for that VM along with time, date, and subscription.

You can download the log information by selecting [Click here to download all the items as CSV](#).

Modify the filters and click **Apply** to create a search and log.

Configure JIT access from an Azure VM's page

For your convenience, you can connect to a VM using JIT directly from within the VM's page in Security Center.

Configure JIT access on a VM via the Azure VM page

To make it easy to roll out just-in-time access across your VMs, you can set a VM to allow only just-in-time access directly from within the VM.

- From the [Azure portal](#), search for and select **Virtual machines**.
- Select the virtual machine you want to limit to just-in-time access.
- In the menu, select **Configuration**.
- Under **Just-in-time access**, select **Enable just-in-time**.

This enables just-in-time access for the VM using the following settings:

- Windows servers:
 - RDP port 3389
 - Three hours of maximum allowed access
 - Allowed source IP addresses is set to Any
- Linux servers:
 - SSH port 22
 - Three hours of maximum allowed access
 - Allowed source IP addresses is set to Any

If a VM already has just-in-time enabled, when you go to its configuration page you will be able to see that just-in-time is

enabled and you can use the link to open the policy in Azure Security Center to view and change the settings.

The screenshot shows the Azure portal interface for managing virtual machines. On the left, a sidebar lists several VMs: vm-usa, vm-contoso-us (selected), vm-europe, vm-contoso1, vm-contoso2, vm-contoso-3, vm-contoso4, vm-london, vm-marketing, vm-hr, vm-uni, vm-contoso-hr, and vm-contoso. The main pane is titled "vm-contoso-us - Configuration". It includes a search bar, save and discard buttons, and a "Just-in-time access" section. This section contains a message: "To improve security, enable a just-in-time access policy." with a "Enable just-in-time policy" button. Below this is an "Azure hybrid benefit" section with a "Use existing Windows license" toggle and "No" and "Yes" buttons. A vertical sidebar on the right lists various configuration options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Networking, Disks, Size, Security, Extensions, Continuous delivery (Preview), Availability set, Configuration (selected), and Identity (Preview). The "Configuration" option is highlighted with a red box.

Request JIT access to a VM via an Azure VM's page

In the Azure portal, when you try to connect to a VM, Azure checks to see if you have a just-in-time access policy configured on that VM.

- If you have a JIT policy configured on the VM, you can click **Request access** to grant access in accordance with the JIT policy set for the VM.

Connect to virtual machine

X

vm1



This VM has a just-in-time access policy. Select "Request access" before connecting.

RDP

SSH

You need to request access to connect to your virtual machine. Select an IP address, optionally change the port number, and select "Request access". [Learn more](#)

o

* IP address

Public IP address (52.161.18.9)



* Port number

3389

Request access

[Download RDP file anyway](#)

Having trouble connecting to this VM?

- [Diagnose and solve problems](#)
- [Troubleshoot connection](#)
- [Serial console](#)

Access is requested with the following default parameters:

- o **source IP:** 'Any' (*) (cannot be changed)
- o **time range:** Three hours (cannot be changed)
- o **port number** RDP port 3389 for Windows / port 22 for Linux (can be changed)

NOTE

After a request is approved for a VM protected by Azure Firewall, Security Center provides the user with the proper connection details (the port mapping from the DNAT table) to use to connect to the VM.

- If you do not have JIT configured on a VM, you will be prompted to configure a JIT policy on it.

Connect to virtual machine

ContosoAppSrv2

To improve security, enable just-in-time access on this VM.

RDP **SSH**

To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

* IP address
Public IP address (40.124.37.238)

* Port number
3389

Download RDP File

 Inbound traffic to the Public IP address may be blocked. You can update inbound port rules in the **VM Networking** page.

 You can troubleshoot VM connection issues by opening the **Diagnose and solve problems** page.

Configure a JIT policy on a VM programmatically

You can set up and use just-in-time via REST APIs and via PowerShell.

JIT VM access via REST APIs

The just-in-time VM access feature can be used via the Azure Security Center API. You can get information about configured VMs, add new ones, request access to a VM, and more, via this API. See [Jit Network Access Policies](#), to learn more about the just-in-time REST API.

JIT VM access via PowerShell

To use the just-in-time VM access solution via PowerShell, use the official Azure Security Center PowerShell cmdlets, and specifically `Set-AzJitNetworkAccessPolicy`.

The following example sets a just-in-time VM access policy on a specific VM, and sets the following:

1. Close ports 22 and 3389.
2. Set a maximum time window of 3 hours for each so they can be opened per approved request.
3. Allows the user who is requesting access to control the source IP addresses and allows the user to establish a successful session upon an approved just-in-time access request.

Run the following in PowerShell to accomplish this:

1. Assign a variable that holds the just-in-time VM access policy for a VM:

```
$JitPolicy = @{
    id="/subscriptions/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Microsoft.Compute/virtualMachines/VMNAME"
    "
    ports=@{
        number=22;
        protocol="*";
        allowedSourceAddressPrefix=@("*.");
        maxRequestAccessDuration="PT3H"},

        @{
            number=3389;
            protocol="*";
            allowedSourceAddressPrefix=@("*.");
            maxRequestAccessDuration="PT3H"})})
```

2. Insert the VM just-in-time VM access policy to an array:

```
$JitPolicyArr=@($JitPolicy)
```

3. Configure the just-in-time VM access policy on the selected VM:

```
Set-AzJitNetworkAccessPolicy -Kind "Basic" -Location "LOCATION" -Name "default" -ResourceGroupName "RESOURCEGROUP"
-VirtualMachine $JitPolicyArr
```

Request access to a VM via PowerShell

In the following example, you can see a just-in-time VM access request to a specific VM in which port 22 is requested to be opened for a specific IP address and for a specific amount of time:

Run the following in PowerShell:

1. Configure the VM request access properties

```
$JitPolicyVm1 = (@{
    id="/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Microsoft.Compute/virtualMachines/VMNAME"
    ports=@{
        number=22;
        endTimeUtc="2018-09-17T17:00:00.3658798Z";
        allowedSourceAddressPrefix=@("IPV4ADDRESS"))})
```

2. Insert the VM access request parameters in an array:

```
$JitPolicyArr=@($JitPolicyVm1)
```

3. Send the request access (use the resource ID you got in step 1)

```
Start-AzJitNetworkAccessPolicy -ResourceId
"/subscriptions/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Microsoft.Security/locations/LOCATION/jitNet
workAccessPolicies/default" -VirtualMachine $JitPolicyArr
```

For more information, see the [PowerShell cmdlet documentation](#).

Automatic cleanup of redundant JIT rules

Whenever you update a JIT policy, a cleanup tool automatically runs to check the validity of your entire ruleset. The tool looks for mismatches between rules in your policy and rules in the NSG. If the cleanup tool finds a mismatch, it determines the cause and, when it's safe to do so, removes built-in rules that aren't needed any more. The cleaner never deletes rules that you've created.

Examples scenarios when the cleaner might remove a built-in rule:

- When two rules with identical definitions exist and one has a higher priority than the other (meaning, the lower priority

rule will never be used)

- When a rule description includes the name of a VM which doesn't match the destination IP in the rule

Next steps

In this article, you learned how just-in-time VM access in Security Center helps you control access to your Azure virtual machines.

To learn more about Security Center, see the following:

- [Setting security policies](#) — Learn how to configure security policies for your Azure subscriptions and resource groups.
- [Managing security recommendations](#) — Learn how recommendations help you protect your Azure resources.
- [Security health monitoring](#) — Learn how to monitor the health of your Azure resources.

Azure Disk Encryption scenarios on Windows VMs

10/29/2019 • 11 minutes to read • [Edit Online](#)

Azure Disk Encryption uses the BitLocker external key protector to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets. For an overview of the service, see [Azure Disk Encryption for Windows VMs](#).

There are many disk encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Windows VMs.

You can only apply disk encryption to virtual machines of [supported VM sizes and operating systems](#). You must also meet the following prerequisites:

- [Networking requirements](#)
- [Group Policy requirements](#)
- [Encryption key storage requirements](#)

IMPORTANT

- If you have previously used Azure Disk Encryption with Azure AD to encrypt a VM, you must continue use this option to encrypt your VM. See [Azure Disk Encryption with Azure AD \(previous release\)](#) for details.
- You should [take a snapshot](#) and/or create a backup before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the [Set-AzVMDiskEncryptionExtension cmdlet](#) to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see [Back up and restore encrypted Azure VM](#).
- Encrypting or disabling encryption may cause a VM to reboot.

Install tools and connect to Azure

Azure Disk Encryption can be enabled and managed through the [Azure CLI](#) and [Azure PowerShell](#). To do so you must install the tools locally and connect to your Azure subscription.

Azure CLI

The [Azure CLI 2.0](#) is a command-line tool for managing Azure resources. The CLI is designed to flexibly query data, support long-running operations as non-blocking processes, and make scripting easy. You can install it locally by following the steps in [Install the Azure CLI](#).

To [Sign in to your Azure account with the Azure CLI](#), use the `az login` command.

```
az login
```

If you would like to select a tenant to sign in under, use:

```
az login --tenant <tenant>
```

If you have multiple subscriptions and want to specify a specific one, get your subscription list with [az account list](#) and specify with [az account set](#).

```
az account list  
az account set --subscription "<subscription name or ID>"
```

For more information, see [Get started with Azure CLI 2.0](#).

Azure PowerShell

The [Azure PowerShell az module](#) provides a set of cmdlets that uses the [Azure Resource Manager](#) model for managing your Azure resources. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine using the instructions in [Install the Azure PowerShell module](#).

If you already have it installed locally, make sure you use the latest version of Azure PowerShell SDK version to configure Azure Disk Encryption. Download the latest version of [Azure PowerShell release](#).

To [Sign in to your Azure account with Azure PowerShell](#), use the `Connect-AzAccount` cmdlet.

```
Connect-AzAccount
```

If you have multiple subscriptions and want to specify one, use the `Get-AzSubscription` cmdlet to list them, followed by the `Set-AzContext` cmdlet:

```
Set-AzContext -Subscription -Subscription <SubscriptionId>
```

Running the `Get-AzContext` cmdlet will verify that the correct subscription has been selected.

To confirm the Azure Disk Encryption cmdlets are installed, use the `Get-command` cmdlet:

```
Get-command *diskencryption*
```

For more information, see [Getting started with Azure PowerShell](#).

Enable encryption on an existing or running Windows VM

In this scenario, you can enable encryption by using the Resource Manager template, PowerShell cmdlets, or CLI commands. If you need schema information for the virtual machine extension, see the [Azure Disk Encryption for Windows extension](#) article.

Enable encryption on existing or running IaaS Windows VMs

You can enable encryption by using a template, PowerShell cmdlets, or CLI commands. If you need schema information for the virtual machine extension, see the [Azure Disk Encryption for Windows extension](#) article.

Enable encryption on existing or running VMs with Azure PowerShell

Use the `Set-AzVMDiskEncryptionExtension` cmdlet to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:** The script below initializes your variables and runs the `Set-AzVMDiskEncryptionExtension` cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace `MyKeyVaultResourceGroup`, `MyVirtualMachineResourceGroup`, `MySecureVM`, and `MySecureVault` with your values.

```

$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -DiskEncryptionKeyVaultUrl
$diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId;

```

- **Encrypt a running VM using KEK:**

```

$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName).Key.kid;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -DiskEncryptionKeyVaultUrl
$diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the [Get-AzVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

- **Disable disk encryption:** To disable the encryption, use the [Disable-AzVMDiskEncryption](#) cmdlet. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

```
Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

Enable encryption on existing or running VMs with the Azure CLI

Use the [az vm encryption enable](#) command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- **Encrypt a running VM using KEK:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

```
az vm encryption disable --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
```

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the [Resource Manager template to encrypt a running Windows VM](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, location, settings, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs:

PARAMETER	DESCRIPTION
vmName	Name of the VM to run the encryption operation.
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet <pre>(Get-AzKeyVault -ResourceGroupName <MyKeyVaultResourceGroupName>).VaultName</pre> or the Azure CLI command <pre>az keyvault list --resource-group "MyKeyVaultResourceGroup"</pre>
keyVaultResourceGroup	Name of the resource group that contains the key vault
keyEncryptionKeyURL	The URL of the key encryption key, in the format https://<keyvault-name>.vault.azure.net/key/<key-name>. If you do not wish to use a KEK, leave this field blank.

PARAMETER	DESCRIPTION
volumeType	Type of volume that the encryption operation is performed on. Valid values are <i>OS</i> , <i>Data</i> , and <i>All</i> .
forceUpdateTag	Pass in a unique value like a GUID every time the operation needs to be force run.
resizeOSDisk	Should the OS partition be resized to occupy full OS VHD before splitting system volume.
location	Location for all resources.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can create a new VM from a pre-encrypted VHD and the associated encryption keys using PowerShell cmdlets or CLI commands.

Use the instructions in [Prepare a pre-encrypted Windows VHD](#). After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

Encrypt VMs with pre-encrypted VHDS with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzVMOSDisk](#). The example below gives you some common parameters.

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching ReadWrite -
Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
00000000/resourceGroups/myKVresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can [add a new disk to a Windows VM using PowerShell](#), or [through the Azure portal](#).

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the [Set-AzVmDiskEncryptionExtension](#) cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- **Encrypt a running VM:** The script below initializes your variables and runs the [Set-AzVMDiskEncryptionExtension](#) cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, and MySecureVault with your values. This example uses "All" for the *-VolumeType* parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the *-VolumeType* parameter.

```

$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$sequenceVersion = [Guid]::NewGuid();

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -DiskEncryptionKeyVaultUrl
$diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -VolumeType "All" -
SequenceVersion $sequenceVersion;

```

- **Encrypt a running VM using KEK:** This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```

$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName).Key.kid;
$sequenceVersion = [Guid]::NewGuid();

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -DiskEncryptionKeyVaultUrl
$diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyId $keyVaultResourceId -VolumeType "All" -SequenceVersion
$sequenceVersion;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. The example uses "All" for the volume-type parameter. You may need to change the volume-type parameter to OS if you're only encrypting the OS disk. In contrast to Powershell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt a running VM:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type "All"
```

- **Encrypt a running VM using KEK:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "All"
```

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzVMDiskEncryption](#) cmdlet.

```
Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM' -VolumeType "all"
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup" --volume-type "all"
```

- **Disable encryption with a Resource Manager template:**

1. Click **Deploy to Azure** from the [Disable disk encryption on running Windows VM](#) template.
2. Select the subscription, resource group, location, VM, volume type, legal terms, and agreement.
3. Click **Purchase** to disable disk encryption on a running Windows VM.

Unsupported scenarios

Azure Disk Encryption does not work for the following scenarios, features, and technology:

- Encrypting basic tier VM or VMs created through the classic VM creation method.
- Encrypting VMs configured with software-based RAID systems.
- Encrypting VMs configured with Storage Spaces Direct (S2D), or Windows Server versions before 2016 configured with Windows Storage Spaces.
- Integration with an on-premises key management system.
- Azure Files (shared file system).
- Network File System (NFS).
- Dynamic volumes.
- Windows Server containers, which create dynamic volumes for each container.
- Ephemeral OS disks.
- Encryption of shared/distributed file systems like (but not limited to) DFS, GFS, DRDB, and CephFS.

Next steps

- [Azure Disk Encryption overview](#)
- [Azure Disk Encryption sample scripts](#)
- [Azure Disk Encryption troubleshooting](#)

Quickstart: Create and encrypt a Windows VM with the Azure CLI

10/9/2019 • 3 minutes to read • [Edit Online](#)

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quickstart shows you how to use the Azure CLI to create and encrypt a Windows Server 2016 virtual machine (VM).

If you don't have an Azure subscription, create a [free account](#) before you begin.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser.	
Select the Cloud Shell button on the menu bar at the upper right in the Azure portal .	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use the CLI locally, this quickstart requires that you are running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see [Install Azure CLI](#).

Create a resource group

Create a resource group with the `az group create` command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named `myResourceGroup` in the `eastus` location:

```
az group create --name myResourceGroup --location eastus
```

Create a virtual machine

Create a VM with [az vm create](#). The following example creates a VM named *myVM*. This example uses *azureuser* for an administrative user name and *myPassword12* as the password.

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password myPassword12
```

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

```
{
  "fqdns": "",
  "id": "/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroup"
}
```

Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with [az keyvault create](#). To enable the Key Vault to store encryption keys, use the `--enabled-for-disk-encryption` parameter.

IMPORTANT

Each Key Vault must have a unique name. The following example creates a Key Vault named *myKV*, but you must name yours something different.

```
az keyvault create --name "myKV" --resource-group "myResourceGroup" --location eastus --enabled-for-disk-encryption
```

Encrypt the virtual machine

Encrypt your VM with [az vm encryption](#), providing your unique Key Vault name to the `--disk-encryption-keyvault` parameter.

```
az vm encryption enable -g MyResourceGroup --name MyVM --disk-encryption-keyvault myKV
```

You can verify that encryption is enabled on your VM with [az vm show](#)

```
az vm show --name MyVM -g MyResourceGroup
```

You will see the following in the returned output:

```
"EncryptionOperation": "EnableEncryption"
```

Clean up resources

When no longer needed, you can use the [az group delete](#) command to remove the resource group, VM, and Key Vault.

```
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you created a virtual machine, created a Key Vault that was enable for encryption keys, and encrypted the VM. Advance to the next article to learn more about Azure Disk Encryption prerequisites for IaaS VMs.

[Azure Disk Encryption overview](#)

Quickstart: Create and encrypt a Windows virtual machine in Azure with PowerShell

10/9/2019 • 2 minutes to read • [Edit Online](#)

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to create a Windows virtual machine (VM), create a Key Vault for the storage of encryption keys, and encrypt the VM.

If you don't have an Azure subscription, create a [free account](#) before you begin.

Create a resource group

Create an Azure resource group with [New-AzResourceGroup](#). A resource group is a logical container into which Azure resources are deployed and managed:

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create a virtual machine

Create an Azure virtual machine with [New-AzVM](#). You must supply credentials to the cmdlet.

```
$cred = Get-Credential  
  
New-AzVM -Name MyVm -Credential $cred -ResourceGroupName MyResourceGroup -Image win2016datacenter -Size Standard_D2S_V3
```

It will take a few minutes for your VM to be deployed.

Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with [New-AzKeyvault](#). To enable the Key Vault to store encryption keys, use the `-EnabledForDiskEncryption` parameter.

IMPORTANT

Each Key Vault must have a unique name. The following example creates a Key Vault named *myKV*, but you must name yours something different.

```
New-AzKeyvault -name MyKV -ResourceGroupName myResourceGroup -Location EastUS -EnabledForDiskEncryption
```

Encrypt the virtual machine

Encrypt your VM with [Set-AzVmDiskEncryptionExtension](#).

`Set-AzVmDiskEncryptionExtension` requires some values from your Key Vault object. You can obtain these values by passing the unique name of your key vault to [Get-AzKeyvault](#).

```
$KeyVault = Get-AzKeyVault -VaultName MyKV -ResourceGroupName MyResourceGroup

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyResourceGroup -VMName MyVM -DiskEncryptionKeyVaultUrl
$KeyVault.VaultUri -DiskEncryptionKeyId $KeyVault.ResourceId
```

After a few minutes the process will return the following:

RequestId	IsSuccess	Status	StatusCode	ReasonPhrase
	True	OK	OK	

You can verify the encryption process by running [Get-AzVmDiskEncryptionStatus](#).

```
Get-AzVmDiskEncryptionStatus -VMName MyVM -ResourceGroupName MyResourceGroup
```

When encryption is enabled, you will see the following in the returned output:

OsVolumeEncrypted	:	Encrypted
DataVolumesEncrypted	:	NoDiskFound
OsVolumeEncryptionSettings	:	Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage	:	Provisioning succeeded

Clean up resources

When no longer needed, you can use the [Remove-AzResourceGroup](#) cmdlet to remove the resource group, VM, and all related resources:

```
Remove-AzResourceGroup -Name "myResourceGroup"
```

Next steps

In this quickstart, you created a virtual machine, created a Key Vault that was enable for encryption keys, and encrypted the VM. Advance to the next article to learn more about Azure Disk Encryption prerequisites for IaaS VMs.

[Azure Disk Encryption overview](#)

Quickstart: Create and encrypt a Windows virtual machine with the Azure portal

11/17/2019 • 2 minutes to read • [Edit Online](#)

Azure virtual machines (VMs) can be created through the Azure portal. The Azure portal is a browser-based user interface to create VMs and their associated resources. In this quickstart you will use the Azure portal to deploy a Windows virtual machine (VM) running Ubuntu 18.04 LTS, create a key vault for the storage of encryption keys, and encrypt the VM.

If you don't have an Azure subscription, create a [free account](#) before you begin.

Sign in to Azure

Sign in to the [Azure portal](#).

Create a virtual machine

1. Choose **Create a resource** in the upper left corner of the Azure portal.
2. In the New page, under Popular, select **Windows Server 2016 Datacenter**.
3. In the Basics tab, under Project details, make sure the correct subscription is selected and then choose to **Create new resource group**. Enter *myResourceGroup* as the name.
4. For **Virtual machine name**, enter *MyVM*.
5. For **Region**, select the same region you used when making your key vault above (e.g., *East US*).
6. Make sure the **Size** is *Standard D2s v3*.
7. Under **Administrator account**, select **Password**. Enter a user name and a password.

Create a virtual machine

Complete the Basics tab then Review + create to provision a virtual machine with default customization.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups to group your resources.

Subscription * ⓘ

Free Trial

Resource group * ⓘ

myResourceGroup

[Create new](#)

Instance details

Virtual machine name * ⓘ

myVM

Region * ⓘ

(US) East US

Availability options ⓘ

No infrastructure redundancy required

Image * ⓘ

Windows Server 2016 Datacenter

[Browse all public and private images](#)

Size * ⓘ

Standard DS1 v2

1 vcpu, 3.5 GiB memory

[Change size](#)

Administrator account

Username * ⓘ

azureUser

Password * ⓘ

.....

Confirm password * ⓘ

.....

Review + create

< Previous

Next : Disks >

8. Select the "Management" tab and verify that you have a Diagnostics Storage Account. If you have no storage accounts, select "Create New", give your new account a name, and select "Ok"

Create a virtual machine

Basics Disks Networking **Management** Advanced Tags Review +

Configure monitoring and management options for your VM.

Azure Security Center

Azure Security Center provides unified security management and advanced threat protection. [Learn more](#)

Your subscription is protected by Azure Security Center basic plan.

Monitoring

Enable detailed monitoring (preview) On Off

Boot diagnostics On Off

OS guest diagnostics On Off

* Diagnostics storage account No existing storage accounts in current subscription [Create new](#)

The value must not be empty.

Create storage account

* Name .core.windows.net (4)

Account kind

Performance Standard Premium

Replication

9. Click "Review + Create".

10. On the **Create a virtual machine** page, you can see the details about the VM you are about to create. When you are ready, select **Create**.

It will take a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

Encrypt the virtual machine

1. When the VM deployment is complete, select **Go to resource**.

2. On the left-hand sidebar, select **Disks**.

3. On the Disks screen, select **Encryption**.

myVM - Disks

Virtual machine

Search (Ctrl+J)

Edit Refresh Encryption Swap OS Disk

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Networking Disks Size Security Extensions Continuous delivery (Preview) Availability + scaling

Managed disks created since June 10, 2017 are encrypted at rest with Storage Service Encryption (SSE). You may also want to enable Azure Disk Encryption.

Ultra Disk compatibility is not available for this location.

Disk settings

Enable Ultra Disk compatibility Yes No

OS disk

Name	Size	Storage account type	Encryption
myVM_OsDisk_1_8df8426fc5464c6d8cb2b26f8658e9b0	30 GiB	Premium SSD	Not enabled

Data disks

None

+ Add data disk

4. On the encryption screen, under **Disks to encrypt**, choose **OS and data disks**.

5. Under **Encryption settings**, choose **Select a key vault and key for encryption**.

6. On the **Select key from Azure Key Vault** screen, select **Create New**.

Home > myVM - Disks > Encryption > Select key from Azure Key Vault

Select key from Azure Key Vault

Key vault *

Select the key vault.

Create new

Key

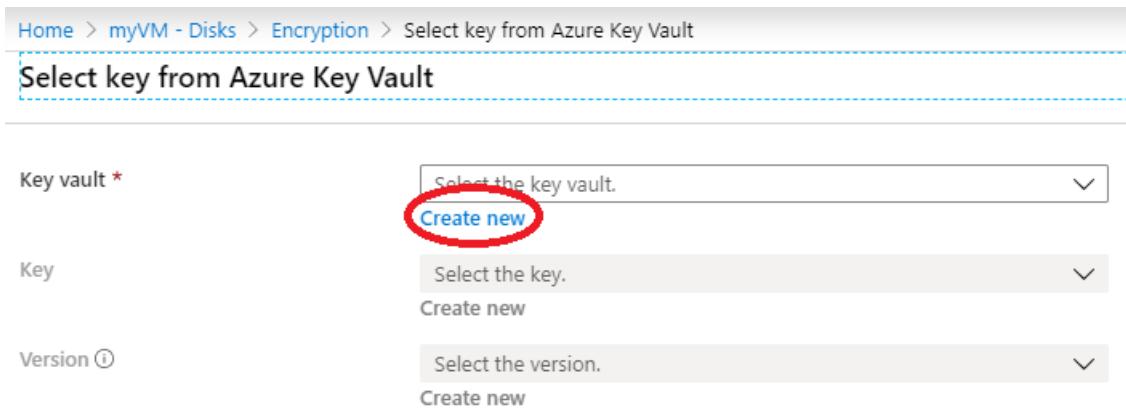
Select the key.

Create new

Version ⓘ

Select the version.

Create new



7. On the **Create key vault** screen, ensure that the Resource Group is the same as the one you used to create the VM.

8. Give your key vault a name. Every key vault across Azure must have an unique name.

9. On the **Access Policies** tab, check the **Azure Disk Encryption for volume encryption** box.

Home > myVM - Disks > Encryption > Select key from Azure Key Vault > Create key vault

Create key vault

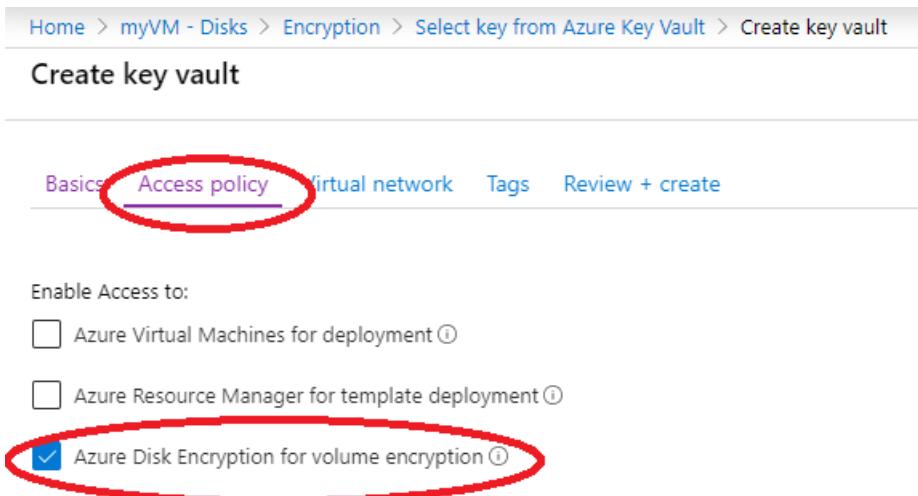
Basics **Access policy** virtual network Tags Review + create

Enable Access to:

Azure Virtual Machines for deployment ⓘ

Azure Resource Manager for template deployment ⓘ

Azure Disk Encryption for volume encryption ⓘ



10. Select **Review + create**.

11. After the key vault has passed validation, select **Create**. This will return you to the **Select key from Azure Key Vault** screen.

12. Leave the **Key** field blank and choose **Select**.

13. At the top of the encryption screen, click **Save**. A popup will warn you that the VM will reboot. Click **Yes**.

Clean up resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select Delete, then confirm the name of the resource group to delete.

Next steps

In this quickstart, you created a Key Vault that was enable for encryption keys, created a virtual machine, and enabled the virtual machine for encryption.

[Azure Disk Encryption overview](#)

Creating and configuring a key vault for Azure Disk Encryption

10/9/2019 • 6 minutes to read • [Edit Online](#)

Azure Disk Encryption uses Azure Key Vault to control and manage disk encryption keys and secrets. For more information about key vaults, see [Get started with Azure Key Vault](#) and [Secure your key vault](#).

WARNING

- If you have previously used Azure Disk Encryption with Azure AD to encrypt a VM, you must continue use this option to encrypt your VM. See [Creating and configuring a key vault for Azure Disk Encryption with Azure AD \(previous release\)](#) for details.

Creating and configuring a key vault for use with Azure Disk Encryption involves three steps:

1. Creating a resource group, if needed.
2. Creating a key vault.
3. Setting key vault advanced access policies.

These steps are illustrated in the following quickstarts:

- [Create and encrypt a Windows VM with Azure CLI](#)
- [Create and encrypt a Windows VM with Azure PowerShell](#)

You may also, if you wish, generate or import a key encryption key (KEK).

NOTE

The steps in this article are automated in the [Azure Disk Encryption prerequisites CLI script](#) and [Azure Disk Encryption prerequisites PowerShell script](#).

Install tools and connect to Azure

The steps in this article can be completed with the [Azure CLI](#), the [Azure PowerShell Az module](#), or the [Azure portal](#).

While the portal is accessible through your browser, Azure CLI and Azure PowerShell require local installation; see [Azure Disk Encryption for Windows: Install tools](#) for details.

Connect to your Azure account

Before using the Azure CLI or Azure PowerShell, you must first connect to your Azure subscription. You do so by [Signing in with Azure CLI](#), [Signing in with Azure Powershell](#), or supplying your credentials to the Azure portal when prompted.

```
az login
```

```
Connect-AzAccount
```

Create a resource group

If you already have a resource group, you can skip to [Create a key vault](#).

A resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group using the [az group create](#) Azure CLI command, the [New-AzResourceGroup](#) Azure PowerShell command, or from the [Azure portal](#).

Azure CLI

```
az group create --name "myResourceGroup" --location eastus
```

Azure PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create a key vault

If you already have a key vault, you can skip to [Set key vault advanced access policies](#).

Create a key vault using the [az keyvault create](#) Azure CLI command, the [New-AzKeyVault](#) Azure Powershell command, the [Azure portal](#), or a [Resource Manager template](#).

WARNING

To ensure that encryption secrets don't cross regional boundaries, Azure Disk Encryption requires the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VMs to be encrypted.

Each Key Vault must have a unique name. Replace with the name of your key vault in the following examples.

Azure CLI

When creating a key vault using Azure CLI, add the "--enabled-for-disk-encryption" flag.

```
az keyvault create --name "<your-unique-keyvault-name>" --resource-group "myResourceGroup" --location "eastus" --enabled-for-disk-encryption
```

Azure PowerShell

When creating a key vault using Azure PowerShell, add the "-EnabledForDiskEncryption" flag.

```
New-AzKeyVault -name "<your-unique-keyvault-name>" -ResourceGroupName "myResourceGroup" -Location "eastus" -EnabledForDiskEncryption
```

Resource Manager template

You can also create a key vault by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then click **Purchase**.

Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the

VM for booting and decrypting the volumes.

If you did not enable your key vault for disk encryption, deployment, or template deployment at the time of creation (as demonstrated in the previous step), you must update its advanced access policies.

Azure CLI

Use [az keyvault update](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

```
az keyvault update --name "<your-unique-keyvault-name>" --resource-group "MyResourceGroup" --enabled-for-disk-encryption "true"
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

```
az keyvault update --name "<your-unique-keyvault-name>" --resource-group "MyResourceGroup" --enabled-for-deployment "true"
```

- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

```
az keyvault update --name "<your-unique-keyvault-name>" --resource-group "MyResourceGroup" --enabled-for-template-deployment "true"
```

Azure PowerShell

Use the key vault PowerShell cmdlet [Set-AzKeyVaultAccessPolicy](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

```
Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -ResourceGroupName "MyResourceGroup" -EnabledForDiskEncryption
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

```
Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -ResourceGroupName "MyResourceGroup" -EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

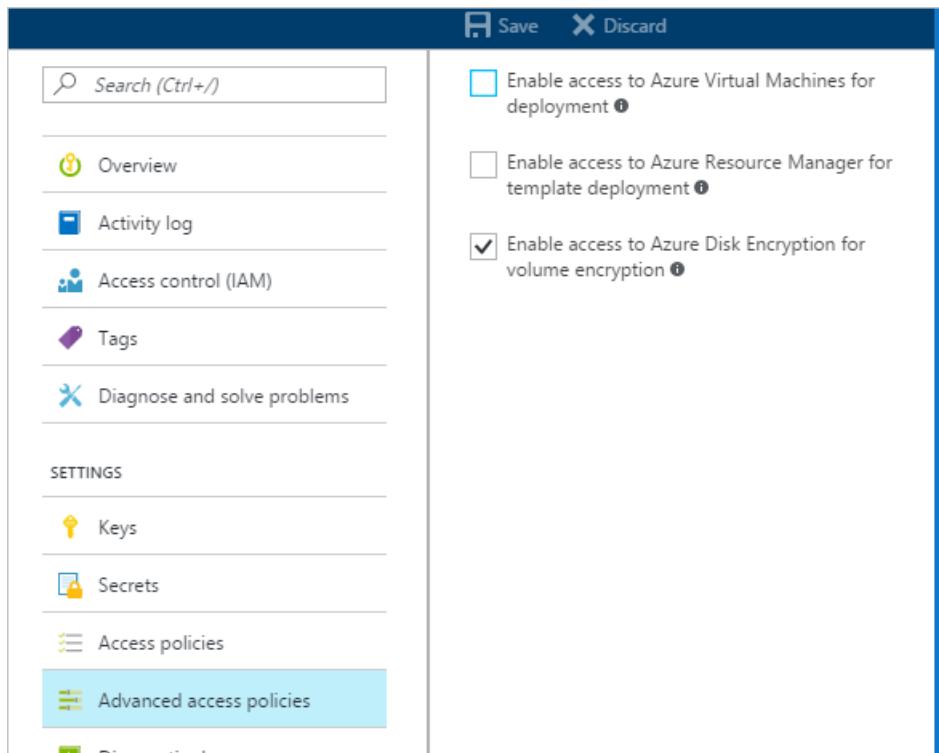
```
Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -ResourceGroupName "MyResourceGroup" -EnabledForTemplateDeployment
```

Azure portal

1. Select your key vault, go to **Access Policies**, and [Click to show advanced access policies](#).
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure**

Resource Manager for template deployment, if needed.

4. Click **Save**.



Set up a key encryption key (KEK)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

You can generate a new KEK using the Azure CLI [az keyvault key create](#) command, the Azure PowerShell [Add-AzKeyVaultKey](#) cmdlet, or the [Azure portal](#). You must generate an RSA key type; Azure Disk Encryption does not yet support using Elliptic Curve keys.

You can instead import a KEK from your on-premises key management HSM. For more information, see [Key Vault Documentation](#).

Your key vault KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:

- Example of a valid secret URL:
<https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
- Example of a valid KEK URL:
<https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:

- Acceptable key vault URL:
<https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
- Unacceptable key vault URL:
<https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Azure CLI

Use the Azure CLI [az keyvault key create](#) command to generate a new KEK and store it in your key vault.

```
az keyvault key create --name "myKEK" --vault-name "<your-unique-keyvault-name>" --kty RSA-HSM
```

You may instead import a private key using the Azure CLI [az keyvault key import](#) command:

In either case, you will supply the name of your KEK to the Azure CLI [az vm encryption enable](#) --key-encryption-key parameter.

```
az vm encryption enable -g "MyResourceGroup" --name "myVM" --disk-encryption-keyvault "<your-unique-keyvault-name>" --key-encryption-key "myKEK"
```

Azure PowerShell

Use the Azure PowerShell [Add-AzKeyVaultKey](#) cmdlet to generate a new KEK and store it in your key vault.

```
Add-AzKeyVaultKey -Name "myKEK" -VaultName "<your-unique-keyvault-name>" -Destination "HSM"
```

You may instead import a private key using the Azure PowerShell [az keyvault key import](#) command.

In either case, you will supply the ID of your KEK key Vault and the URL of your KEK to the Azure PowerShell [Set-AzVMDiskEncryptionExtension](#) -KeyEncryptionKeyVaultId and -KeyEncryptionKeyUrl parameters. Note that this example assumes that you are using the same key vault for both the disk encryption key and the KEK.

```
$KeyVault = Get-AzKeyVault -VaultName "<your-unique-keyvault-name>" -ResourceGroupName "myResourceGroup"
$KEK = Get-AzKeyVaultKey -VaultName "<your-unique-keyvault-name>" -Name "myKEK"

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyResourceGroup -VMName "MyVM" -DiskEncryptionKeyVaultUrl
$KeyVault.VaultUri -DiskEncryptionKeyVaultId $KeyVault.ResourceId -KeyEncryptionKeyVaultId $KeyVault.ResourceId
-KeyEncryptionKeyUrl $KEK.Id -SkipVmBackup -VolumeType All
```

Next steps

- [Azure Disk Encryption prerequisites CLI script](#)
- [Azure Disk Encryption prerequisites PowerShell script](#)
- Learn [Azure Disk Encryption scenarios on Windows VMs](#)
- Learn how to [troubleshoot Azure Disk Encryption](#)
- Read the [Azure Disk Encryption sample scripts](#)

Azure Disk Encryption sample scripts

11/7/2019 • 7 minutes to read • [Edit Online](#)

This article provides sample scripts for preparing pre-encrypted VHDs and other tasks.

List VMs and secrets

List all encrypted VMs in your subscription:

```
$osVolEncrypted = {(Get-AzVMDiskEncryptionStatus -ResourceGroupName $_.ResourceGroupName -VMName
$_.Name).OsVolumeEncrypted}
$dataVolEncrypted= {(Get-AzVMDiskEncryptionStatus -ResourceGroupName $_.ResourceGroupName -VMName
$_.Name).DataVolumesEncrypted}
Get-AzVm | Format-Table @{Label="MachineName"; Expression={$_.Name}}, @{Label="OsVolumeEncrypted";
Expression=$osVolEncrypted}, @{Label="DataVolumesEncrypted"; Expression=$dataVolEncrypted}
```

List all disk encryption secrets used for encrypting VMs in a key vault:

```
Get-AzKeyVaultSecret -VaultName $KeyVaultName | where {$_.Tags.ContainsKey('DiskEncryptionKeyFileName')} |
format-table @{Label="MachineName"; Expression={$_.Tags['MachineName']}}, @{Label="VolumeLetter"; Expression=
{$_.Tags['VolumeLetter']}}, @{Label="EncryptionKeyURL"; Expression={$_.Id}}
```

The Azure Disk Encryption prerequisites scripts

If you're already familiar with the prerequisites for Azure Disk Encryption, you can use the [Azure Disk Encryption prerequisites PowerShell script](#). For an example of using this PowerShell script, see the [Encrypt a VM Quickstart](#). You can remove the comments from a section of the script, starting at line 211, to encrypt all disks for existing VMs in an existing resource group.

The following table shows which parameters can be used in the PowerShell script:

PARAMETER	DESCRIPTION	MANDATORY?
\$resourceGroupName	Name of the resource group to which the KeyVault belongs to. A new resource group with this name will be created if one doesn't exist.	True
\$keyVaultName	Name of the KeyVault in which encryption keys are to be placed. A new vault with this name will be created if one doesn't exist.	True
\$location	Location of the KeyVault. Make sure the KeyVault and VMs to be encrypted are in the same location. Get a location list with Get-AzLocation .	True
\$subscriptionId	Identifier of the Azure subscription to be used. You can get your Subscription ID with Get-AzSubscription .	True

PARAMETER	DESCRIPTION	MANDATORY?
\$aadAppName	Name of the Azure AD application that will be used to write secrets to KeyVault. A new application with this name will be created if one doesn't exist. If this app already exists, pass <code>aadClientSecret</code> parameter to the script.	False
\$aadClientSecret	Client secret of the Azure AD application that was created earlier.	False
\$keyEncryptionKeyName	Name of optional key encryption key in KeyVault. A new key with this name will be created if one doesn't exist.	False

Resource Manager templates

Encrypt or decrypt VMs without an Azure AD app

- [Enable disk encryption on an existing or running Windows VM](#)
- [Disable encryption on a running Windows VM](#)

Encrypt or decrypt VMs with an Azure AD app (previous release)

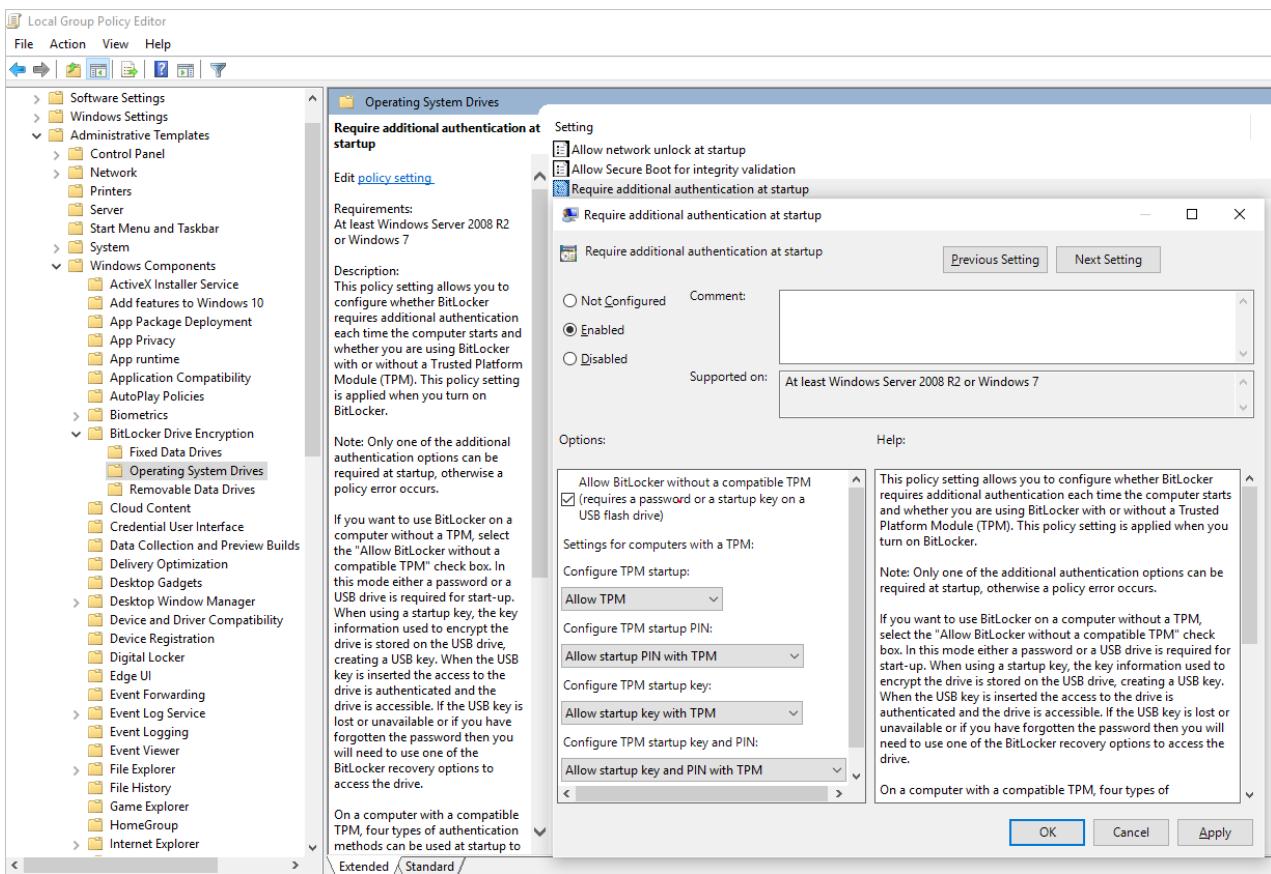
- [Enable disk encryption on an existing or running Windows VM](#)
- [Disable encryption on a running Windows VM](#)
- [Create a new encrypted managed disk from a pre-encrypted VHD/storage blob](#)
 - Creates a new encrypted managed disk provided a pre-encrypted VHD and its corresponding encryption settings

Prepare a pre-encrypted Windows VHD

The sections that follow are necessary to prepare a pre-encrypted Windows VHD for deployment as an encrypted VHD in Azure IaaS. Use the information to prepare and boot a fresh Windows VM (VHD) on Azure Site Recovery or Azure. For more information on how to prepare and upload a VHD, see [Upload a generalized VHD and use it to create new VMs in Azure](#).

Update group policy to allow non-TPM for OS protection

Configure the BitLocker Group Policy setting **BitLocker Drive Encryption**, which you'll find under **Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components**. Change this setting to **Operating System Drives > Require additional authentication at startup > Allow BitLocker without a compatible TPM**, as shown in the following figure:



Install BitLocker feature components

For Windows Server 2012 and later, use the following command:

```
dism /online /Enable-Feature /all /FeatureName:BitLocker /quiet /norestart
```

For Windows Server 2008 R2, use the following command:

```
ServerManagerCmd -install BitLockers
```

Prepare the OS volume for BitLocker by using [bdehdcfg](#)

To compress the OS partition and prepare the machine for BitLocker, execute the [bdehdcfg](#) if needed:

```
bdehdcfg -target c: shrink -quiet
```

Protect the OS volume by using BitLocker

Use the [manage-bde](#) command to enable encryption on the boot volume using an external key protector. Also place the external key (.bek file) on the external drive or volume. Encryption is enabled on the system/boot volume after the next reboot.

```
manage-bde -on %systemdrive% -sk [ExternalDriveOrVolume]
reboot
```

NOTE

Prepare the VM with a separate data/resource VHD for getting the external key by using BitLocker.

Upload encrypted VHD to an Azure storage account

After DM-Crypt encryption is enabled, the local encrypted VHD needs to be uploaded to your storage account.

```
Add-AzVhd [-Destination] <Uri> [-LocalFilePath] <FileInfo> [[-NumberOfUploaderThreads] <Int32> ] [[-BaseImageUriToPatch] <Uri> ] [[-OverWrite]] [ <CommonParameters>]
```

Upload the secret for the pre-encrypted VM to your key vault

The disk encryption secret that you obtained previously must be uploaded as a secret in your key vault. This requires granting the set secret permission and the wrapkey permission to the account that will upload the secrets.

```
# Typically, account Id is the user principal name (in user@domain.com format)
$upn = (Get-AzureRmContext).Account.Id
Set-AzKeyVaultAccessPolicy -VaultName $kvname -UserPrincipalName $acctid -PermissionsToKeys wrapKey -
PermissionsToSecrets set

# In cloud shell, the account ID is a managed service identity, so specify the username directly
# $upn = "user@domain.com"
# Set-AzKeyVaultAccessPolicy -VaultName $kvname -UserPrincipalName $acctid -PermissionsToKeys wrapKey -
PermissionsToSecrets set

# When running as a service principal, retrieve the service principal ID from the account ID, and set access
policy to that
# $acctid = (Get-AzureRmContext).Account.Id
# $spoid = (Get-AzureRmADServicePrincipal -ServicePrincipalName $acctid).Id
# Set-AzKeyVaultAccessPolicy -VaultName $kvname -ObjectId $spoid -BypassObjectIdValidation -PermissionsToKeys
wrapKey -PermissionsToSecrets set
```

Disk encryption secret not encrypted with a KEK

To set up the secret in your key vault, use [Set-AzKeyVaultSecret](#). The passphrase is encoded as a base64 string and then uploaded to the key vault. In addition, make sure that the following tags are set when you create the secret in the key vault.

```
# This is the passphrase that was provided for encryption during the distribution installation
$passphrase = "contoso-password"

$tags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP"; "DiskEncryptionKeyFileName" =
"LinuxPassPhraseFileName"}
$secretName = [guid]::NewGuid().ToString()
$secretValue = [Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($passphrase))
$secureSecretValue = ConvertTo-SecureString $secretValue -AsPlainText -Force

$secret = Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $secretName -SecretValue $secureSecretValue -
tags $tags
$secretUrl = $secret.Id
```

Use the `$secretUrl` in the next step for [attaching the OS disk without using KEK](#).

Disk encryption secret encrypted with a KEK

Before you upload the secret to the key vault, you can optionally encrypt it by using a key encryption key. Use the wrap API to first encrypt the secret using the key encryption key. The output of this wrap operation is a base64 URL encoded string, which you can then upload as a secret by using the [Set-AzKeyVaultSecret](#) cmdlet.

```
# This is the passphrase that was provided for encryption during the distribution installation
$passphrase = "contoso-password"
```

```

Add-AzKeyVaultKey -VaultName $KeyVaultName -Name "keyencryptionkey" -Destination Software
$keyEncryptionKey = Get-AzKeyVaultKey -VaultName $KeyVault.OriginalVault.Name -Name "keyencryptionkey"

$apiversion = "2015-06-01"

#####
# Get Auth URI
#####

$uri = $KeyVault.VaultUri + "/keys"
$headers = @{}

$response = try { Invoke-RestMethod -Method GET -Uri $uri -Headers $headers } catch {
$__.Exception.Response }

$authHeader = $response.Headers["www-authenticate"]
$authUri = [regex]::match($authHeader, 'authorization="(.*?)"').Groups[1].Value

Write-Host "Got Auth URI successfully"

#####
# Get Auth Token
#####

$uri = $authUri + "/oauth2/token"
$body = "grant_type=client_credentials"
$body += "&client_id=" + $AadClientId
$body += "&client_secret=" + [Uri]::EscapeDataString($AadClientSecret)
$body += "&resource=" + [Uri]::EscapeDataString("https://vault.azure.net")
$headers = @{}

$response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body $body

$access_token = $response.access_token

Write-Host "Got Auth Token successfully"

#####
# Get KEK info
#####

$uri = $KeyEncryptionKey.Id + "?api-version=" + $apiversion
$headers = @{"Authorization" = "Bearer " + $access_token}

$response = Invoke-RestMethod -Method GET -Uri $uri -Headers $headers

$keyid = $response.key.kid

Write-Host "Got KEK info successfully"

#####
# Encrypt passphrase using KEK
#####

$passphraseB64 = [Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Passphrase))
$uri = $keyid + "/encrypt?api-version=" + $apiversion
$headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" = "application/json"}
$bodyObj = @{"alg" = "RSA-OAEP"; "value" = $passphraseB64}
$body = $bodyObj | ConvertTo-Json

$response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body $body

$wrappedSecret = $response.value

Write-Host "Encrypted passphrase successfully"

#####
# Store secret
#####

```

```

#####
$secretName = [guid]::NewGuid().ToString()
$uri = $KeyVault.VaultUri + "/secrets/" + $secretName + "?api-version=" + $apiversion
$secretAttributes = @{"enabled" = $true}
$secretTags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP"; "DiskEncryptionKeyFileName" =
"LinuxPassPhraseFileName"}
$headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" = "application/json"}
$bodyObj = @{"value" = $wrappedSecret; "attributes" = $secretAttributes; "tags" = $secretTags}
$body = $bodyObj | ConvertTo-Json

$response = Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body $body

Write-Host "Stored secret successfully"

$secretUrl = $response.id

```

Use `$keyEncryptionKey` and `$secretUrl` in the next step for [attaching the OS disk using KEK](#).

Specify a secret URL when you attach an OS disk

Without using a KEK

While you're attaching the OS disk, you need to pass `$secretUrl`. The URL was generated in the "Disk-encryption secret not encrypted with a KEK" section.

```

Set-AzVMOSDisk `

-VM $VirtualMachine `

-Name $OSDiskName `

-SourceImageUri $VhdUri `

-VhdUri $OSDiskUri `

-Windows `

-CreateOption FromImage `

-DiskEncryptionKeyVaultId $KeyVault.ResourceId `

-DiskEncryptionKeyUrl $SecretUrl

```

Using a KEK

When you attach the OS disk, pass `$keyEncryptionKey` and `$secretUrl`. The URL was generated in the "Disk encryption secret encrypted with a KEK" section.

```

Set-AzVMOSDisk `

-VM $VirtualMachine `

-Name $OSDiskName `

-SourceImageUri $CopiedTemplateBlobUri `

-VhdUri $OSDiskUri `

-Windows `

-CreateOption FromImage `

-DiskEncryptionKeyVaultId $KeyVault.ResourceId `

-DiskEncryptionKeyUrl $SecretUrl `

-KeyEncryptionKeyVaultId $KeyVault.ResourceId `

-KeyEncryptionKeyURL $KeyEncryptionKey.Id

```

Azure Disk Encryption troubleshooting guide

11/7/2019 • 3 minutes to read • [Edit Online](#)

This guide is for IT professionals, information security analysts, and cloud administrators whose organizations use Azure Disk Encryption. This article is to help with troubleshooting disk-encryption-related problems.

Before taking any of the steps below, first ensure that the VMs you are attempting to encrypt are among the [supported VM sizes and operating systems](#), and that you have met all the prerequisites:

- [Networking requirements](#)
- [Group policy requirements](#)
- [Encryption key storage requirements](#)

Troubleshooting Azure Disk Encryption behind a firewall

When connectivity is restricted by a firewall, proxy requirement, or network security group (NSG) settings, the ability of the extension to perform needed tasks might be disrupted. This disruption can result in status messages such as "Extension status not available on the VM." In expected scenarios, the encryption fails to finish. The sections that follow have some common firewall problems that you might investigate.

Network security groups

Any network security group settings that are applied must still allow the endpoint to meet the documented network configuration [prerequisites](#) for disk encryption.

Azure Key Vault behind a firewall

When encryption is being enabled with [Azure AD credentials](#), the target VM must allow connectivity to both Azure Active Directory endpoints and Key Vault endpoints. Current Azure Active Directory authentication endpoints are maintained in sections 56 and 59 of the [Office 365 URLs and IP address ranges](#) documentation. Key Vault instructions are provided in the documentation on how to [Access Azure Key Vault behind a firewall](#).

Azure Instance Metadata Service

The VM must be able to access the [Azure Instance Metadata service](#) endpoint which uses a well-known non-routable IP address (`169.254.169.254`) that can be accessed only from within the VM. Proxy configurations that alter local HTTP traffic to this address (for example, adding an X-Forwarded-For header) are not supported.

Troubleshooting Windows Server 2016 Server Core

On Windows Server 2016 Server Core, the bdehdcfg component isn't available by default. This component is required by Azure Disk Encryption. It's used to split the system volume from OS volume, which is done only once for the life time of the VM. These binaries aren't required during later encryption operations.

To work around this issue, copy the following four files from a Windows Server 2016 Data Center VM to the same location on Server Core:

```
\windows\system32\bdehdcfg.exe  
\windows\system32\bdehdcfglib.dll  
\windows\system32\en-US\bdehdcfglib.dll.mui  
\windows\system32\en-US\bdehdcfg.exe.mui
```

1. Enter the following command:

```
bdehdcfg.exe -target default
```

2. This command creates a 550-MB system partition. Reboot the system.

3. Use DiskPart to check the volumes, and then proceed.

For example:

```
DISKPART> list vol

Volume ### Ltr Label Fs Type Size Status Info
----- -----
Volume 0 C NTFS Partition 126 GB Healthy Boot
Volume 1 NTFS Partition 550 MB Healthy System
Volume 2 D Temporary S NTFS Partition 13 GB Healthy Pagefile
```

Troubleshooting encryption status

The portal may display a disk as encrypted even after it has been unencrypted within the VM. This can occur when low-level commands are used to directly unencrypt the disk from within the VM, instead of using the higher level Azure Disk Encryption management commands. The higher level commands not only unencrypt the disk from within the VM, but outside of the VM they also update important platform level encryption settings and extension settings associated with the VM. If these are not kept in alignment, the platform will not be able to report encryption status or provision the VM properly.

To disable Azure Disk Encryption with PowerShell, use [Disable-AzVMDiskEncryption](#) followed by [Remove-AzVMDiskEncryptionExtension](#). Running Remove-AzVMDiskEncryptionExtension before the encryption is disabled will fail.

To disable Azure Disk Encryption with CLI, use [az vm encryption disable](#).

Next steps

In this document, you learned more about some common problems in Azure Disk Encryption and how to troubleshoot those problems. For more information about this service and its capabilities, see the following articles:

- [Apply disk encryption in Azure Security Center](#)
- [Azure data encryption at rest](#)

Azure Disk Encryption for Windows VMs FAQ

11/22/2019 • 6 minutes to read • [Edit Online](#)

This article provides answers to frequently asked questions (FAQ) about Azure Disk Encryption for Windows VMs. For more information about this service, see [Azure Disk Encryption overview](#).

Where is Azure Disk Encryption in general availability (GA)?

Azure Disk Encryption is in general availability in all Azure public regions.

What user experiences are available with Azure Disk Encryption?

Azure Disk Encryption GA supports Azure Resource Manager templates, Azure PowerShell, and Azure CLI. The different user experiences give you flexibility. You have three different options for enabling disk encryption for your VMs. For more information on the user experience and step-by-step guidance available in Azure Disk Encryption, see [Azure Disk Encryption scenarios for Windows](#).

How much does Azure Disk Encryption cost?

There's no charge for encrypting VM disks with Azure Disk Encryption, but there are charges associated with the use of Azure Key Vault. For more information on Azure Key Vault costs, see the [Key Vault pricing](#) page.

How can I start using Azure Disk Encryption?

To get started, read the [Azure Disk Encryption overview](#).

What VM sizes and operating systems support Azure Disk Encryption?

The [Azure Disk Encryption overview](#) article lists the [VM sizes](#) and [VM operating systems](#) that support Azure Disk Encryption.

Can I encrypt both boot and data volumes with Azure Disk Encryption?

You can encrypt both boot and data volumes, but you can't encrypt the data without first encrypting the OS volume.

After you've encrypted the OS volume, disabling encryption on the OS volume isn't supported.

Can I encrypt an unmounted volume with Azure Disk Encryption?

No, Azure Disk Encryption only encrypts mounted volumes.

How do I rotate secrets or encryption keys?

To rotate secrets, just call the same command you used originally to enable disk encryption, specifying a different Key Vault. To rotate the key encryption key, call the same command you used originally to enable disk encryption, specifying the new key encryption.

WARNING

- If you have previously used [Azure Disk Encryption with Azure AD app](#) by specifying Azure AD credentials to encrypt this VM, you will have to continue use this option to encrypt your VM. You can't use Azure Disk Encryption on this encrypted VM as this isn't a supported scenario, meaning switching away from AAD application for this encrypted VM isn't supported yet.

How do I add or remove a key encryption key if I didn't originally use one?

To add a key encryption key, call the enable command again passing the key encryption key parameter. To remove a key encryption key, call the enable command again without the key encryption key parameter.

Does Azure Disk Encryption allow you to bring your own key (BYOK)?

Yes, you can supply your own key encryption keys. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption keys support scenarios, see [Creating and configuring a key vault for Azure Disk Encryption](#).

Can I use an Azure-created key encryption key?

Yes, you can use Azure Key Vault to generate a key encryption key for Azure disk encryption use. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption key, see [Creating and configuring a key vault for Azure Disk Encryption](#).

Can I use an on-premises key management service or HSM to safeguard the encryption keys?

You can't use the on-premises key management service or HSM to safeguard the encryption keys with Azure Disk Encryption. You can only use the Azure Key Vault service to safeguard the encryption keys. For more information on the key encryption key support scenarios, see [Creating and configuring a key vault for Azure Disk Encryption](#).

What are the prerequisites to configure Azure Disk Encryption?

There are prerequisites for Azure Disk Encryption. See the [Creating and configuring a key vault for Azure Disk Encryption](#) article to create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see [Creating and configuring a key vault for Azure Disk Encryption](#).

What are the prerequisites to configure Azure Disk Encryption with an Azure AD app (previous release)?

There are prerequisites for Azure Disk Encryption. See the [Azure Disk Encryption with Azure AD](#) content to create an Azure Active Directory application, create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see [Creating and configuring a key vault for Azure Disk Encryption with Azure AD](#).

Is Azure Disk Encryption using an Azure AD app (previous release) still supported?

Yes. Disk encryption using an Azure AD app is still supported. However, when encrypting new VMs it's

recommended that you use the new method rather than encrypting with an Azure AD app.

Can I migrate VMs that were encrypted with an Azure AD app to encryption without an Azure AD app?

Currently, there isn't a direct migration path for machines that were encrypted with an Azure AD app to encryption without an Azure AD app. Additionally, there isn't a direct path from encryption without an Azure AD app to encryption with an AD app.

What version of Azure PowerShell does Azure Disk Encryption support?

Use the latest version of the Azure PowerShell SDK to configure Azure Disk Encryption. Download the latest version of [Azure PowerShell](#). Azure Disk Encryption is *not* supported by Azure SDK version 1.1.0.

What is the disk "Bek Volume" or "/mnt/azure_bek_disk"?

The "Bek volume" is a local data volume that securely stores the encryption keys for Encrypted Azure VMs.

NOTE

Do not delete or edit any contents in this disk. Do not unmount the disk since the encryption key presence is needed for any encryption operations on the IaaS VM.

What encryption method does Azure Disk Encryption use?

Azure Disk Encryption selects the encryption method in BitLocker based on the version of Windows as follows:

WINDOWS VERSIONS	VERSION	ENCRYPTION METHOD
Windows Server 2012, Windows 10, or greater	>=1511	XTS-AES 256 bit
Windows Server 2012, Windows 8, 8.1, 10	< 1511	AES 256 bit *
Windows Server 2008R2		AES 256 bit with Diffuser

* AES 256 bit with Diffuser isn't supported in Windows 2012 and later.

To determine Windows OS version, run the 'winver' tool in your virtual machine.

If I use EncryptFormatAll and specify all volume types, will it erase the data on the data drives that we already encrypted?

No, data won't be erased from data drives that are already encrypted using Azure Disk Encryption. Similar to how EncryptFormatAll didn't re-encrypt the OS drive, it won't re-encrypt the already encrypted data drive.

Can I backup and restore an encrypted VM?

Azure Backup provides a mechanism to backup and restore encrypted VM's within the same subscription and region. For instructions, please see [Back up and restore encrypted virtual machines with Azure Backup](#). Restoring an encrypted VM to a different region is not currently supported.

Where can I go to ask questions or provide feedback?

You can ask questions or provide feedback on the [Azure Disk Encryption forum](#).

Next steps

In this document, you learned more about the most frequent questions related to Azure Disk Encryption. For more information about this service, see the following articles:

- [Azure Disk Encryption Overview](#)
- [Apply disk encryption in Azure Security Center](#)
- [Azure data encryption at rest](#)

Azure Disk Encryption with Azure AD (previous release)

10/9/2019 • 2 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption for Windows VMs](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

This article supplements [Azure Disk Encryption for Windows VMs](#) with additional requirements and prerequisites for Azure Disk Encryption with Azure AD (previous release). The [Supported VMs and operating systems](#) section remains the same.

Networking and Group Policy

To enable the Azure Disk Encryption feature using the older AAD parameter syntax, the IaaS VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the IaaS VM must be able to connect to an Azure Active Directory endpoint, [login.microsoftonline.com].
- To write the encryption keys to your key vault, the IaaS VM must be able to connect to the key vault endpoint.
- The IaaS VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see [Azure Key Vault behind a firewall](#).
- On Windows, if TLS 1.0 has been explicitly disabled and the .NET version has not been updated to 4.6 or higher, the following registry change will enable ADE to select the more recent TLS version:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001`
```

Group Policy:

- The Azure Disk Encryption solution uses the BitLocker external key protector for Windows IaaS VMs. For domain joined VMs, don't push any group policies that enforce TPM protectors. For information about the group policy for "Allow BitLocker without a compatible TPM," see [BitLocker Group Policy Reference](#).
- BitLocker policy on domain joined virtual machines with custom group policy must include the following

setting: [Configure user storage of BitLocker recovery information -> Allow 256-bit recovery key](#). Azure Disk Encryption will fail when custom group policy settings for BitLocker are incompatible. On machines that didn't have the correct policy setting, apply the new policy, force the new policy to update (gpupdate.exe /force), and then restarting may be required.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see [Creating and configuring a key vault for Azure Disk Encryption with Azure AD \(previous release\)](#).

Next steps

- [Creating and configuring a key vault for Azure Disk Encryption with Azure AD \(previous release\)](#)
- [Enable Azure Disk Encryption with Azure AD on Windows VMs \(previous release\)](#)
- [Azure Disk Encryption prerequisites CLI script](#)
- [Azure Disk Encryption prerequisites PowerShell script](#)

Creating and configuring a key vault for Azure Disk Encryption with Azure AD (previous release)

10/9/2019 • 15 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

Azure Disk Encryption uses Azure Key Vault to control and manage disk encryption keys and secrets. For more information about key vaults, see [Get started with Azure Key Vault](#) and [Secure your key vault](#).

Creating and configuring a key vault for use with Azure Disk Encryption with Azure AD (previous release) involves three steps:

1. Create a key vault.
2. Set up an Azure AD application and service principal.
3. Set the key vault access policy for the Azure AD app.
4. Set key vault advanced access policies.

You may also, if you wish, generate or import a key encryption key (KEK).

See the main [Creating and configuring a key vault for Azure Disk Encryption](#) article for steps on how to [Install tools and connect to Azure](#).

NOTE

The steps in this article are automated in the [Azure Disk Encryption prerequisites CLI script](#) and [Azure Disk Encryption prerequisites PowerShell script](#).

Create a key vault

Azure Disk Encryption is integrated with [Azure Key Vault](#) to help you control and manage the disk-encryption keys and secrets in your key vault subscription. You can create a key vault or use an existing one for Azure Disk Encryption. For more information about key vaults, see [Get started with Azure Key Vault](#) and [Secure your key vault](#). You can use a Resource Manager template, Azure PowerShell, or the Azure CLI to create a key vault.

WARNING

In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Create a key vault with PowerShell

You can create a key vault with Azure PowerShell using the [New-AzKeyVault](#) cmdlet. For additional cmdlets for Key Vault, see [Az.KeyVault](#).

1. Create a new resource group, if needed, with [New-AzResourceGroup](#). To list data center locations, use [Get-AzLocation](#).

```
# Get-AzLocation  
New-AzResourceGroup -Name 'MyKeyVaultResourceGroup' -Location 'East US'
```

2. Create a new key vault using [New-AzKeyVault](#)

```
New-AzKeyVault -VaultName 'MySecureVault' -ResourceGroupName 'MyKeyVaultResourceGroup' -Location 'East US'
```

3. Note the **Vault Name**, **Resource Group Name**, **Resource ID**, **Vault URI**, and the **Object ID** that are returned for later use when you encrypt the disks.

Create a key vault with Azure CLI

You can manage your key vault with Azure CLI using the [az keyvault](#) commands. To create a key vault, use [az keyvault create](#).

1. Create a new resource group, if needed, with [az group create](#). To list locations, use [az account list-locations](#)

```
# To list locations: az account list-locations --output table  
az group create -n "MyKeyVaultResourceGroup" -l "East US"
```

2. Create a new key vault using [az keyvault create](#).

```
az keyvault create --name "MySecureVault" --resource-group "MyKeyVaultResourceGroup" --location "East US"
```

3. Note the **Vault Name** (name), **Resource Group Name**, **Resource ID** (ID), **Vault URI**, and the **Object ID** that are returned for use later.

Create a key vault with a Resource Manager template

You can create a key vault by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then click **Purchase**.

Set up an Azure AD app and service principal

When you need encryption to be enabled on a running VM in Azure, Azure Disk Encryption generates and writes the encryption keys to your key vault. Managing encryption keys in your key vault requires Azure AD authentication. Create an Azure AD application for this purpose. For authentication purposes, you can use either client secret-based authentication or [client certificate-based Azure AD authentication](#).

Set up an Azure AD app and service principal with Azure PowerShell

To execute the following commands, get and use the [Azure AD PowerShell module](#).

1. Use the [New-AzADApplication](#) PowerShell cmdlet to create an Azure AD application. MyApplicationHomePage and the MyApplicationUri can be any values you wish.

```

$aadClientSecret = "My AAD client secret"
$aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -AsPlainText -Force
$azureAdApplication = New-AzADApplication -DisplayName "My Application Display Name" -HomePage
"https://MyApplicationHomePage" -IdentifierUris "https://MyApplicationUri" -Password
$aadClientSecretSec
$servicePrincipal = New-AzADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId

```

2. The \$azureAdApplication.ApplicationId is the Azure AD ClientID and the \$aadClientSecret is the client secret that you will use later to enable Azure Disk Encryption. Safeguard the Azure AD client secret appropriately. Running `$azureAdApplication.ApplicationId` will show you the ApplicationID.

Set up an Azure AD app and service principal with Azure CLI

You can manage your service principals with Azure CLI using the `az ad sp` commands. For more information, see [Create an Azure service principal](#).

1. Create a new service principal.

```

az ad sp create-for-rbac --name "ServicePrincipalName" --password "My-AAD-client-secret" --skip-
assignment

```

2. The appId returned is the Azure AD ClientID used in other commands. It's also the SPN you'll use for az keyvault set-policy. The password is the client secret that you should use later to enable Azure Disk Encryption. Safeguard the Azure AD client secret appropriately.

Set up an Azure AD app and service principal though the Azure portal

Use the steps from the [Use portal to create an Azure Active Directory application and service principal that can access resources](#) article to create an Azure AD application. Each step listed below will take you directly to the article section to complete.

1. [Verify required permissions](#)
2. [Create an Azure Active Directory application](#)
 - You can use any name and sign-on URL you would like when creating the application.
3. [Get the application ID and the authentication key](#).
 - The authentication key is the client secret and is used as the AadClientSecret for Set-AzVMDiskEncryptionExtension.
 - The authentication key is used by the application as a credential to sign in to Azure AD. In the Azure portal, this secret is called keys, but has no relation to key vaults. Secure this secret appropriately.
 - The application ID will be used later as the AadClientId for Set-AzVMDiskEncryptionExtension and as the ServicePrincipalName for Set-AzKeyVaultAccessPolicy.

Set the key vault access policy for the Azure AD app

To write encryption secrets to a specified Key Vault, Azure Disk Encryption needs the Client ID and the Client Secret of the Azure Active Directory application that has permissions to write secrets to the Key Vault.

NOTE

Azure Disk Encryption requires you to configure the following access policies to your Azure AD client application: *WrapKey* and *Set* permissions.

Set the key vault access policy for the Azure AD app with Azure PowerShell

Your Azure AD application needs rights to access the keys or secrets in the vault. Use the `Set-`

[AzKeyVaultAccessPolicy](#) cmdlet to grant permissions to the application, using the client ID (which was generated when the application was registered) as the *-ServicePrincipalName* parameter value. To learn more, see the blog post [Azure Key Vault - Step by Step](#).

1. Set the key vault access policy for the AD application with PowerShell.

```
$keyVaultName = 'MySecureVault'  
$aadClientID = 'MyAadAppClientID'  
$KVRGname = 'MyKeyVaultResourceGroup'  
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -  
PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $KVRGname
```

Set the key vault access policy for the Azure AD app with Azure CLI

Use [az keyvault set-policy](#) to set the access policy. For more information, see [Manage Key Vault using CLI 2.0](#).

Give the service principal you created via the Azure CLI access to get secrets and wrap keys with the following command:

```
```azurecli-interactive  
az keyvault set-policy --name "MySecureVault" --spn "<spn created with CLI/the Azure AD ClientID>" --key-
permissions wrapKey --secret-permissions set
```
```

Set the key vault access policy for the Azure AD app with the portal

1. Open the resource group with your key vault.
2. Select your key vault, go to **Access Policies**, then click **Add new**.
3. Under **Select principal**, search for the Azure AD application you created and select it.
4. For **Key permissions**, check **Wrap Key** under **Cryptographic Operations**.
5. For **Secret permissions**, check **Set** under **Secret Management Operations**.
6. Click **OK** to save the access policy.

Add new permissions -

Add a new access policy - PREVIEW

★ Select principal
vmencrypt >

Configure from template (optional)

Key permissions
1 selected >

Secret permissions
1 selected >

Authorized application ⓘ

None selected

Key permissions

All Key Operations

All

Key Management Operations

Get

List

Update

Create

Import

Delete

Backup

Restore

Cryptographic Operations

Decrypt

Encrypt

UnwrapKey

WrapKey

Verify

Sign

Add new permissions -

Add a new access policy - PREVIEW

★ Select principal
vmencrypt >

Configure from template (optional)

Key permissions
1 selected >

Secret permissions
1 selected >

Authorized application ⓘ

None selected

Secret permissions

All Secret Operations

All

Secret Management Operations

Get

List

Set

Delete

Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes. Enable disk encryption on the key vault or deployments will fail.

Set key vault advanced access policies with Azure PowerShell

Use the key vault PowerShell cmdlet [Set-AzKeyVaultAccessPolicy](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

```
Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MyKeyVaultResourceGroup' -  
EnabledForDiskEncryption
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

```
Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MyKeyVaultResourceGroup' -  
EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

```
Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MyKeyVaultResourceGroup' -  
EnabledForTemplateDeployment
```

Set key vault advanced access policies using the Azure CLI

Use [az keyvault update](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

```
az keyvault update --name "MySecureVault" --resource-group "MyKeyVaultResourceGroup" --enabled-for-  
disk-encryption "true"
```

- **Enable Key Vault for deployment, if needed:** Allow Virtual Machines to retrieve certificates stored as secrets from the vault.

```
az keyvault update --name "MySecureVault" --resource-group "MyKeyVaultResourceGroup" --enabled-for-  
deployment "true"
```

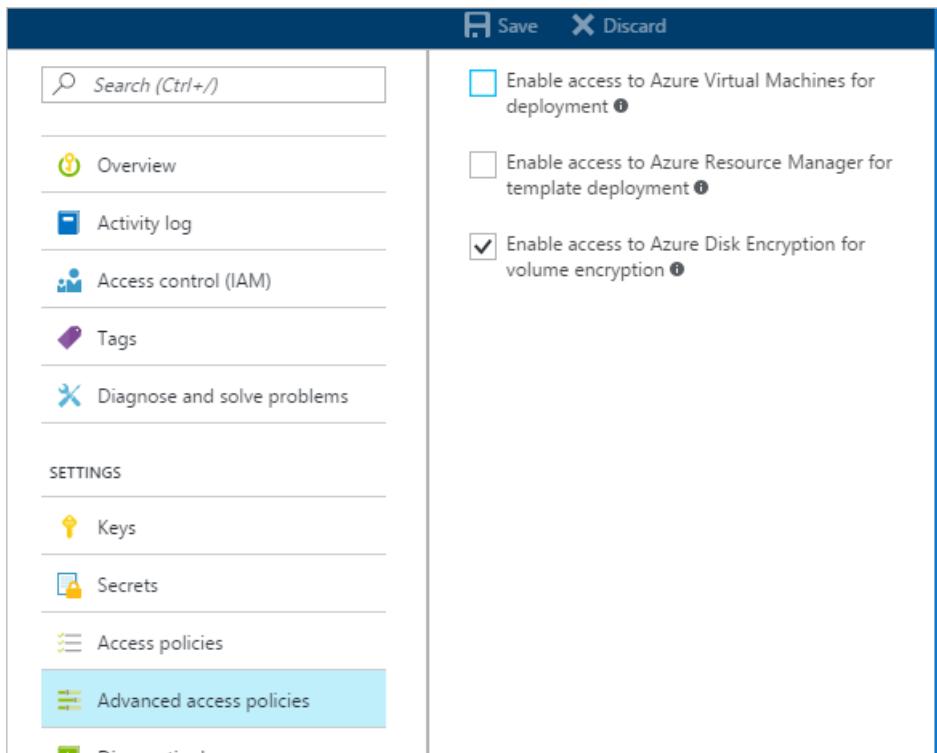
- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

```
az keyvault update --name "MySecureVault" --resource-group "MyKeyVaultResourceGroup" --enabled-for-  
template-deployment "true"
```

Set key vault advanced access policies through the Azure portal

1. Select your keyvault, go to **Access Policies**, and [Click to show advanced access policies](#).
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure Resource Manager for template deployment**, if needed.

4. Click **Save**.



Set up a key encryption key (optional)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. Use the [Add-AzKeyVaultKey](#) cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises key management HSM. For more information, see [Key Vault Documentation](#). When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

- When generating keys, use an RSA key type. Azure Disk Encryption does not yet support using Elliptic Curve keys.
- Your key vault secret and KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:
 - Example of a valid secret URL:
<https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Example of a valid KEK URL:
<https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
- Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:
 - Unacceptable key vault URL
<https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Acceptable key vault URL:
<https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Set up a key encryption key with Azure PowerShell

Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment. This script creates all Azure Disk Encryption prerequisites and encrypts an existing IaaS VM, wrapping the disk encryption key by using a key encryption key.

```

# Step 1: Create a new resource group and key vault in the same location.
# Fill in 'MyLocation', 'MyKeyVaultResourceGroup', and 'MySecureVault' with your values.
# Use Get-AzLocation to get available locations and use the DisplayName.
# To use an existing resource group, comment out the line for New-AzResourceGroup

$Loc = 'MyLocation';
$KVRGname = 'MyKeyVaultResourceGroup';
$keyVaultName = 'MySecureVault';
New-AzResourceGroup -Name $KVRGname -Location $Loc;
New-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname -Location $Loc;
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$keyVaultResourceId = (Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname).ResourceId;
$diskEncryptionKeyVaultUrl = (Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname).VaultUri;

# Step 2: Create the AD application and service principal.
# Fill in 'MyAADClientSecret', "<My Application Display Name>", "<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your values.
# MyApplicationHomePage and the MyApplicationUri can be any values you wish.

$aadClientSecret = 'MyAADClientSecret';
$aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -AsPlainText -Force;
$azureAdApplication = New-AzADApplication -DisplayName "<My Application Display Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "<https://MyApplicationUri>" -Password $aadClientSecretSec
$servicePrincipal = New-AzADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId;
$aadClientID = $azureAdApplication.ApplicationId;

#Step 3: Enable the vault for disk encryption and set the access policy for the Azure AD application.

Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $KVRGname -EnabledForDiskEncryption;
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $KVRGname;

#Step 4: Create a new key in the key vault with the Add-AzKeyVaultKey cmdlet.
# Fill in 'MyKeyEncryptionKey' with your value.

$keyEncryptionKeyName = 'MyKeyEncryptionKey';
Add-AzKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName -Destination 'Software';
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName).Key.kid;

#Step 5: Encrypt the disks of an existing IaaS VM
# Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

$VMName = 'MySecureVM';
$VMRGName = 'MyVirtualMachineResourceGroup';
Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyId $keyVaultResourceId;

```

Certificate-based authentication (optional)

If you would like to use certificate authentication, you can upload one to your key vault and deploy it to the client. Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

```

# Fill in "MyKeyVaultResourceGroup", "MySecureVault", and 'MyLocation' ('My location' only if needed)

$KVRGname = 'MyKeyVaultResourceGroup'
$keyVaultName= 'MySecureVault'

# Create a key vault and set enabledForDiskEncryption property on it.
# Comment out the next three lines if you already have an existing key vault enabled for encryption. No need

```

```

to set 'My location' in this case.

$Loc = 'MyLocation'
New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location $Loc
Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $KVRGname -EnabledForDiskEncryption

#Setting some variables with the key vault information
$keyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname
$DiskEncryptionKeyVaultUrl = $keyVault.VaultUri
$keyVaultResourceId = $keyVault.ResourceId

# Create the Azure AD application and associate the certificate with it.
# Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display Name>", "
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your values.
# MyApplicationHomePage and the MyApplicationUri can be any values you wish

$CertPath = "C:\certificates\mycert.pfx"
$CertPassword = "Password"
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)
$CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

$AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display Name>" -HomePage "
<https://MyApplicationHomePage>" -IdentifierUris "<https://MyApplicationUri>" -CertValue $CertValue
$ServicePrincipal = New-AzADServicePrincipal -ApplicationId $AzureAdApplication.ApplicationId

$AADClientID = $AzureAdApplication.ApplicationId
$aadClientCertThumbprint= $cert.Thumbprint

Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -PermissionsToKeys
'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $KVRGname

# Upload the pfx file to the key vault.
# Fill in "MyAADCert".

$keyVaultSecretName = "MyAADCert"
$fileContentBytes = get-content $CertPath -Encoding Byte
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
$jsonObject = @"
{
    "data" : "$fileContentEncoded",
    "dataType" : "pfx",
    "password" : "$CertPassword"
}
"@

$jsonObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
$jsonEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

#Set the secret and set the key vault policy for -EnabledForDeployment

$Secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText -Force
Set-AzKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName -SecretValue $Secret
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $KVRGname -EnabledForDeployment

# Deploy the certificate to the VM
# Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

$VMName = 'MySecureVM'
$VMRGName = 'MyVirtualMachineResourceGroup'
$CertUrl = (Get-AzKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName).Id
$SourceVaultId = (Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname).ResourceId
$VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
$VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore "My" -CertificateUrl $CertUrl
Update-AzVM -VM $VM -ResourceGroupName $VMRGName

#Enable encryption on the VM using Azure AD client ID and the client certificate thumbprint

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $aadClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -

```

```
DiskEncryptionKeyVaultId $KeyVaultResourceId
```

Certificate-based authentication and a KEK (optional)

If you would like to use certificate authentication and wrap the encryption key with a KEK, you can use the below script as an example. Before using the PowerShell script, you should be familiar with all of the previous Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

```
# Fill in 'MyKeyVaultResourceGroup', 'MySecureVault', and 'MyLocation' (if needed)

$KVRGname = 'MyKeyVaultResourceGroup'
$keyVaultName= 'MySecureVault'

# Create a key vault and set enabledForDiskEncryption property on it.
# Comment out the next three lines if you already have an existing key vault enabled for encryption.

$Loc = 'MyLocation'
New-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname -Location $Loc
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $KVRGname -EnabledForDiskEncryption

# Create the Azure AD application and associate the certificate with it.
# Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display Name>", "
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your values.
# MyApplicationHomePage and the MyApplicationUri can be any values you wish

$CertPath = "C:\certificates\mycert.pfx"
$CertPassword = "Password"
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)
$CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

$AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display Name>" -HomePage "
<https://MyApplicationHomePage>" -IdentifierUris "<https://MyApplicationUri>" -CertValue $CertValue
$ServicePrincipal = New-AzADServicePrincipal -ApplicationId $AzureAdApplication.ApplicationId

$AADClientID = $AzureAdApplication.ApplicationId
$aadClientCertThumbprint= $cert.Thumbprint

## Give access for setting secrets and wrapping keys
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -PermissionsToKeys
'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $KVRGname

# Upload the pfx file to the key vault.
# Fill in "MyAADCert".

$keyVaultSecretName = "MyAADCert"
$fileContentBytes = get-content $CertPath -Encoding Byte
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
$jsonObject = @"
{
    "data" : "$fileContentEncoded",
    "dataType" : "pfx",
    "password" : "$CertPassword"
}
"@

$jsonObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
$jsonEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

#Set the secret and set the key vault policy for deployment

$Secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText -Force
Set-AzKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName -SecretValue $Secret
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $KVRGname -EnabledForDeployment
```

```

#Setting some variables with the key vault information and generating a KEK
# Fill in 'KEKName'

$KEKName = 'KEKName'
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $kVRGname
$DiskEncryptionKeyVaultUrl = $keyVault.VaultUri
$keyVaultResourceId = $keyVault.ResourceId
$KEK = Add-AzKeyVaultKey -VaultName $keyVaultName -Name $KEKName -Destination "Software"
$keyEncryptionKeyUrl = $KEK.Key.kid


# Deploy the certificate to the VM
# Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

$VMName = 'MySecureVM';
$VMRGName = 'MyVirtualMachineResourceGroup';
$CertUrl = (Get-AzKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName).Id
$SourceVaultId = (Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $kVRGname).ResourceId
$VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
$VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore "My" -CertificateUrl $CertUrl
Update-AzVM -VM $VM -ResourceGroupName $VMRGName

#Enable encryption on the VM using Azure AD client ID and the client certificate thumbprint

Set-AzMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $AADClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -
DiskEncryptionKeyId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyId $keyVaultResourceId

```

Next steps

[Enable Azure Disk Encryption with Azure AD on Windows VMs \(previous release\)](#)

Azure Disk Encryption with Azure AD for Windows VMs (previous release)

10/9/2019 • 13 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption for Windows VMs](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Windows IaaS VMs. Before you can use disk encryption, the [Azure Disk Encryption prerequisites](#) need to be completed.

IMPORTANT

- You should [take a snapshot](#) and/or create a backup before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the [Set-AzVMDiskEncryptionExtension cmdlet](#) to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see [Back up and restore encrypted Azure VM](#).
- Encrypting or disabling encryption may cause a VM to reboot.

Enable encryption on new IaaS VMs created from the Marketplace

You can enable disk encryption on new IaaS Windows VM from the Marketplace in Azure using a Resource Manager template. The template creates a new encrypted Windows VM using the Windows Server 2012 gallery image.

1. On the [Resource Manager template](#), click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to deploy a new IaaS VM where encryption is enabled.
3. After you deploy the template, verify the VM encryption status using your preferred method:

- Verify with the Azure CLI by using the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup"
```

- Verify with Azure PowerShell by using the [Get-AzVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

- Select the VM, then click on **Disks** under the **Settings** heading to verify encryption status in the portal. In the chart under **Encryption**, you'll see if it's enabled.

The following table lists the Resource Manager template parameters for new VMs from the Marketplace scenario using Azure AD client ID:

| PARAMETER | DESCRIPTION |
|-----------------------|---|
| adminUserName | Admin user name for the virtual machine. |
| adminPassword | Admin user password for the virtual machine. |
| newStorageAccountName | Name of the storage account to store OS and data VHDs. |
| vmSize | Size of the VM. Currently, only Standard A, D, and G series are supported. |
| virtualNetworkName | Name of the VNet that the VM NIC should belong to. |
| subnetName | Name of the subnet in the VNet that the VM NIC should belong to. |
| AADClientID | Client ID of the Azure AD application that has permissions to write secrets to your key vault. |
| AADClientSecret | Client secret of the Azure AD application that has permissions to write secrets to your key vault. |
| keyVaultURL | <p>URL of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet
 <code>(Get-AzKeyVault -VaultName "MyKeyVault" -ResourceGroupName "MyKeyVaultResourceGroupName").VaultURI</code></p> <p>or the Azure CLI
 <code>az keyvault show --name "MySecureVault" --query properties.vaultUri</code></p> |
| keyEncryptionKeyURL | <p>URL of the key encryption key that's used to encrypt the generated BitLocker key (optional).</p> <p>KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (Passphrase secret) in your key vault.</p> |
| keyVaultResourceGroup | Resource group of the key vault. |

| PARAMETER | DESCRIPTION |
|-----------|---|
| vmName | Name of the VM that the encryption operation is to be performed on. |

Enable encryption on existing or running IaaS Windows VMs

In this scenario, you can enable encryption by using a template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail how to enable Azure Disk Encryption.

Enable encryption on existing or running VMs with Azure PowerShell

Use the [Set-AzVMDiskEncryptionExtension](#) cmdlet to enable encryption on a running IaaS virtual machine in Azure. For information about enabling encryption with Azure Disk Encryption by using PowerShell cmdlets, see the blog posts [Explore Azure Disk Encryption with Azure PowerShell - Part 1](#) and [Explore Azure Disk Encryption with Azure PowerShell - Part 2](#).

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values.

```
$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -AadClientID $aadClientID
-AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId;
```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

```

$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -AadClientID $aadClientID
-AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-
name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-
name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the [Get-AzVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

- Disable disk encryption:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

Enable encryption on existing or running VMs with Azure CLI

Use the [az vm encryption enable](#) command to enable encryption on a running IaaS virtual machine in Azure.

- Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-
client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret"
--disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-
client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-
client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-
encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
```

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the [Resource Manager template to encrypt a running Windows VM](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs that use an Azure AD client ID:

| PARAMETER | DESCRIPTION |
|---------------------|---|
| AADClientID | Client ID of the Azure AD application that has permissions to write secrets to the key vault. |
| AADClientSecret | Client secret of the Azure AD application that has permissions to write secrets to the key vault. |
| keyVaultName | Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet
<pre>(Get-AzKeyVault -ResourceGroupName <MyKeyVaultResourceGroupName>).Vaultname</pre> or the Azure CLI command
<pre>az keyvault list --resource-group "MySecureGroup"</pre> |
| keyEncryptionKeyURL | URL of the key encryption key that's used to encrypt the generated BitLocker key. This parameter is optional if you select nokek in the UseExistingKek drop-down list. If you select kek in the UseExistingKek drop-down list, you must enter the <i>keyEncryptionKeyURL</i> value. |
| volumeType | Type of volume that the encryption operation is performed on. Valid values are <i>OS</i> , <i>Data</i> , and <i>All</i> . |

| PARAMETER | DESCRIPTION |
|-----------------|--|
| sequenceVersion | Sequence version of the BitLocker operation. Increment this version number every time a disk-encryption operation is performed on the same VM. |
| vmName | Name of the VM that the encryption operation is to be performed on. |

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using the Resource Manager template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail the Resource Manager template and CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- [Prepare a pre-encrypted Windows VHD](#)

Encrypt VMs with pre-encrypted VHDS with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzVMOSDisk](#). The example below gives you some common parameters.

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching ReadWrite -
Windows -CreateOption "Attach" -DiskEncryptionKeyIdUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myKVresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can [add a new disk to a Windows VM using PowerShell](#), or [through the Azure portal](#).

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the `Set-AzVmDiskEncryptionExtension` cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the `Set-AzVMDiskEncryptionExtension` cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace `MyKeyVaultResourceGroup`, `MyVirtualMachineResourceGroup`, `MySecureVM`, `MySecureVault`, `My-AAD-client-ID`, and `My-AAD-client-secret` with your values. This example uses "All" for the `-VolumeType` parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the `-VolumeType` parameter.

```

$sequenceVersion = [Guid]::NewGuid();
$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -AadClientID $aadClientID
-AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -VolumeType 'all' -SequenceVersion $sequenceVersion;

```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```

$sequenceVersion = [Guid]::NewGuid();
$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -AadClientID $aadClientID
-AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId -VolumeType 'all' -SequenceVersion $sequenceVersion;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: [https://\[keyvault-name\].vault.azure.net/keys/\[kekname\]/\[kek-unique-id\]](https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id])

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. Acceptable values for the volume-type parameter are All, OS, and Data. You may need to change the volume-type parameter to OS or Data if you're only encrypting one type of disk for the VM. The examples use "All" for the volume-type parameter.

- **Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type "All"
```

- **Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "all"
```

Enable encryption using Azure AD client certificate-based authentication.

You can use client certificate authentication with or without KEK. Before using the PowerShell scripts, you should already have the certificate uploaded to the key vault and deployed to the VM. If you're using KEK too, the KEK should already exist. For more information, see the [Certificate-based authentication for Azure AD](#) section of the prerequisites article.

Enable encryption using certificate-based authentication with Azure PowerShell

```
## Fill in 'MyVirtualMachineResourceGroup', 'MyKeyVaultResourceGroup', 'My-AAD-client-ID', 'MySecureVault', and 'MySecureVM'.
```

```
$VMRGName = 'MyVirtualMachineResourceGroup'
$KVRGname = 'MyKeyVaultResourceGroup';
$AADClientID ='My-AAD-client-ID';
$KeyVaultName = 'MySecureVault';
$VMName = 'MySecureVM';
$KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$keyVaultResourceId = $KeyVault.ResourceId;

# Fill in the certificate path and the password so the thumbprint can be set as a variable.

$certPath = '$CertPath = "C:\certificates\mycert.pfx";
$certPassword = 'Password'
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)
$aadClientCertThumbprint = $cert.Thumbprint;

# Enable disk encryption using the client certificate thumbprint

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $AADClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId
```

Enable encryption using certificate-based authentication and a KEK with Azure PowerShell

```

# Fill in 'MyVirtualMachineResourceGroup', 'MyKeyVaultResourceGroup', 'My-AAD-client-ID', 'MySecureVault,,,
'MySecureVM', and "KEKName.

$VMRGName = 'MyVirtualMachineResourceGroup';
$KVRGname = 'MyKeyVaultResourceGroup';
$AADClientID ='My-AAD-client-ID';
$keyVaultName = 'MySecureVault';
$VMName = 'MySecureVM';
$keyEncryptionKeyName ='KEKName';
$KeyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$keyVaultResourceId = $KeyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName).Key.kid;

## Fill in the certificate path and the password so the thumbprint can be read and set as a variable.

$certPath = '$CertPath = "C:\certificates\mycert.pfx";
$CertPassword ='Password'
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)
$aadClientCertThumbprint = $cert.Thumbprint;

# Enable disk encryption using the client certificate thumbprint and a KEK

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $aadClientCertThumbprint -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId

```

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
```

- **Disable encryption with a Resource Manager Template:**

1. Click **Deploy to Azure** from the [Disable disk encryption on running Windows VM](#) template.
2. Select the subscription, resource group, location, VM, legal terms, and agreement.
3. Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

[Azure Disk Encryption overview](#)

Setting up WinRM access for Virtual Machines in Azure Resource Manager

11/13/2019 • 3 minutes to read • [Edit Online](#)

Here are the steps you need to take to set up a VM with WinRM connectivity

1. Create a Key Vault
2. Create a self-signed certificate
3. Upload your self-signed certificate to Key Vault
4. Get the URL for your self-signed certificate in the Key Vault
5. Reference your self-signed certificates URL while creating a VM

Step 1: Create a Key Vault

You can use the below command to create the Key Vault

```
New-AzKeyVault -VaultName "<vault-name>" -ResourceGroupName "<rg-name>" -Location "<vault-location>" -EnabledForDeployment -EnabledForTemplateDeployment
```

Step 2: Create a self-signed certificate

You can create a self-signed certificate using this PowerShell script

```
$certificateName = "somename"

$thumbprint = (New-SelfSignedCertificate -DnsName $certificateName -CertStoreLocation Cert:\CurrentUser\My -KeySpec KeyExchange).Thumbprint

$cert = (Get-ChildItem -Path cert:\CurrentUser\My\$thumbprint)

$password = Read-Host -Prompt "Please enter the certificate password." -AsSecureString

Export-PfxCertificate -Cert $cert -FilePath ".\$certificateName.pfx" -Password $password
```

Step 3: Upload your self-signed certificate to the Key Vault

Before uploading the certificate to the Key Vault created in step 1, it needs to converted into a format the Microsoft.Compute resource provider will understand. The below PowerShell script will allow you do that

```

$fileName = "<Path to the .pfx file>"
$fileContentBytes = Get-Content $fileName -Encoding Byte
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)

$jsonObject = @"
{
    "data": "$fileContentEncoded",
    "dataType" : "pfx",
    "password": "<password>"
}
"@

$jsonObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
$jsonEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

$secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText -Force
Set-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret name>" -SecretValue $secret

```

Step 4: Get the URL for your self-signed certificate in the Key Vault

The Microsoft.Compute resource provider needs a URL to the secret inside the Key Vault while provisioning the VM. This enables the Microsoft.Compute resource provider to download the secret and create the equivalent certificate on the VM.

NOTE

The URL of the secret needs to include the version as well. An example URL looks like below

<https://contosovault.vault.azure.net:443/secrets/contososecret/01h9db0df2cd4300a20ence585a6s7ve>

Templates

You can get the link to the URL in the template using the below code

```
"certificateUrl": "[reference(resourceId(resourceGroup().name, 'Microsoft.KeyVault/vaults/secrets', '<vault-name>', '<secret-name>'), '2015-06-01').secretUriWithVersion]"
```

PowerShell

You can get this URL using the below PowerShell command

```
$secretURL = (Get-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret name>").Id
```

Step 5: Reference your self-signed certificates URL while creating a VM

Azure Resource Manager Templates

While creating a VM through templates, the certificate gets referenced in the secrets section and the winRM section as below:

```

"osProfile": {
    ...
    "secrets": [
        {
            "sourceVault": {
                "id": "<resource id of the Key Vault containing the secret>"
            },
            "vaultCertificates": [
                {
                    "certificateUrl": "<URL for the certificate you got in Step 4>",
                    "certificateStore": "<Name of the certificate store on the VM>"
                }
            ]
        }
    ],
    "windowsConfiguration": {
        ...
        "winRM": {
            "listeners": [
                {
                    "protocol": "http"
                },
                {
                    "protocol": "https",
                    "certificateUrl": "<URL for the certificate you got in Step 4>"
                }
            ]
        },
        ...
    }
},

```

A sample template for the above can be found here at [201-vm-winrm-keyvault-windows](#)

Source code for this template can be found on [GitHub](#)

PowerShell

```

$vm = New-AzVMConfig -VMName "<VM name>" -VMSize "<VM Size>"
$credential = Get-Credential
$secretURL = (Get-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret name>").Id
$vm = Set-AzVMOperatingSystem -VM $vm -Windows -ComputerName "<Computer Name>" -Credential $credential -
WinRMHttp -WinRMHttps -WinRMCertificateUrl $secretURL
$sourceVaultId = (Get-AzKeyVault -ResourceGroupName "<Resource Group name>" -VaultName "<Vault
Name>").ResourceId
$CertificateStore = "My"
$vm = Add-AzVMSecret -VM $vm -SourceVaultId $sourceVaultId -CertificateStore $CertificateStore -CertificateUrl
$secretURL

```

Step 6: Connecting to the VM

Before you can connect to the VM you'll need to make sure your machine is configured for WinRM remote management. Start PowerShell as an administrator and execute the below command to make sure you're set up.

```
Enable-PSRemoting -Force
```

NOTE

You might need to make sure the WinRM service is running if the above does not work. You can do that using

```
Get-Service WinRM
```

Once the setup is done, you can connect to the VM using the below command

```
Enter-PSSession -ConnectionUri https://<public-ip-dns-of-the-vm>:5986 -Credential $cred -SessionOption (New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck) -Authentication Negotiate
```

What is role-based access control (RBAC) for Azure resources?

12/23/2019 • 7 minutes to read • [Edit Online](#)

Access management for cloud resources is a critical function for any organization that is using the cloud. Role-based access control (RBAC) helps you manage who has access to Azure resources, what they can do with those resources, and what areas they have access to.

RBAC is an authorization system built on [Azure Resource Manager](#) that provides fine-grained access management of Azure resources.

What can I do with RBAC?

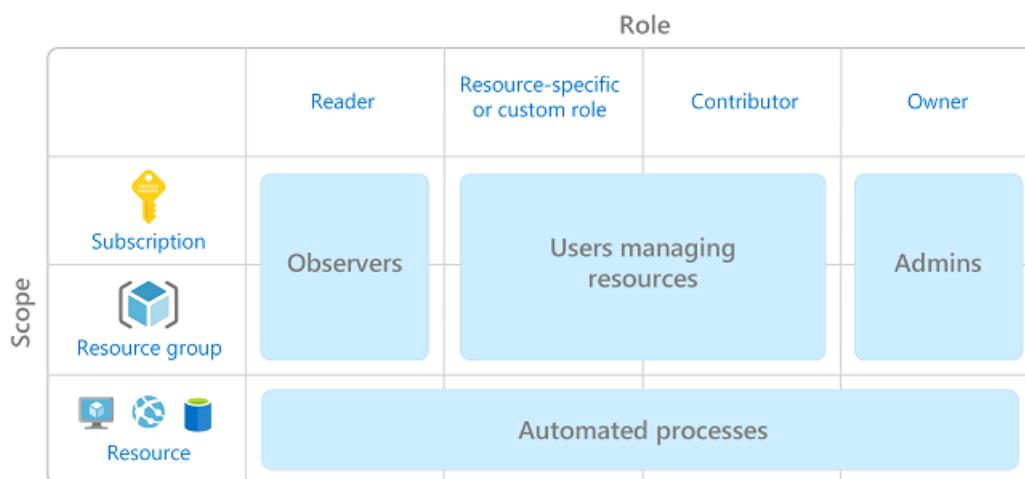
Here are some examples of what you can do with RBAC:

- Allow one user to manage virtual machines in a subscription and another user to manage virtual networks
- Allow a DBA group to manage SQL databases in a subscription
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets
- Allow an application to access all resources in a resource group

Best practice for using RBAC

Using RBAC, you can segregate duties within your team and grant only the amount of access to users that they need to perform their jobs. Instead of giving everybody unrestricted permissions in your Azure subscription or resources, you can allow only certain actions at a particular scope.

When planning your access control strategy, it's a best practice to grant users the least privilege to get their work done. The following diagram shows a suggested pattern for using RBAC.



How RBAC works

The way you control access to resources using RBAC is to create role assignments. This is a key concept to understand – it's how permissions are enforced. A role assignment consists of three elements: security principal, role definition, and scope.

Security principal

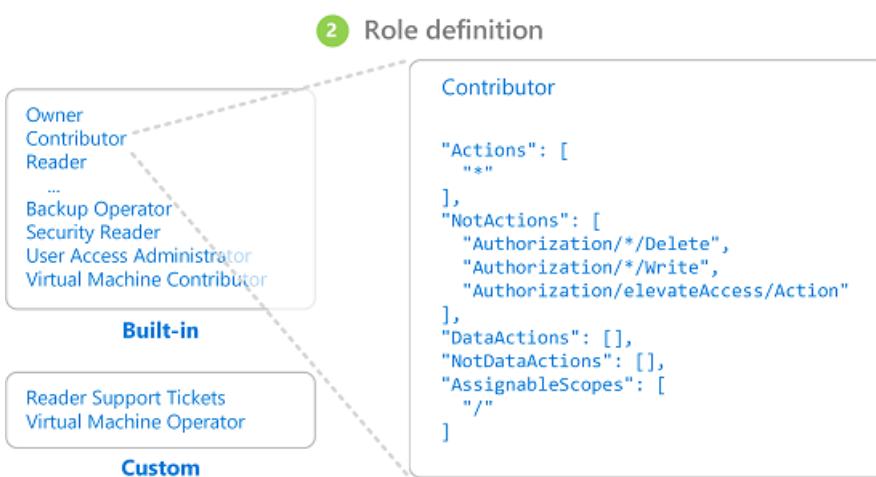
A *security principal* is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources.



- User - An individual who has a profile in Azure Active Directory. You can also assign roles to users in other tenants. For information about users in other organizations, see [Azure Active Directory B2B](#).
- Group - A set of users created in Azure Active Directory. When you assign a role to a group, all users within that group have that role.
- Service principal - A security identity used by applications or services to access specific Azure resources. You can think of it as a *user identity* (username and password or certificate) for an application.
- Managed identity - An identity in Azure Active Directory that is automatically managed by Azure. You typically use [managed identities](#) when developing cloud applications to manage the credentials for authenticating to Azure services.

Role definition

A *role definition* is a collection of permissions. It's typically just called a *role*. A role definition lists the operations that can be performed, such as read, write, and delete. Roles can be high-level, like owner, or specific, like virtual machine reader.



Azure includes several [built-in roles](#) that you can use. The following lists four fundamental built-in roles. The first three apply to all resource types.

- **Owner** - Has full access to all resources including the right to delegate access to others.
- **Contributor** - Can create and manage all types of Azure resources but can't grant access to others.
- **Reader** - Can view existing Azure resources.
- **User Access Administrator** - Lets you manage user access to Azure resources.

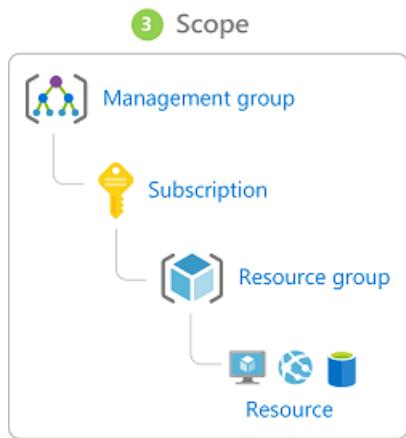
The rest of the built-in roles allow management of specific Azure resources. For example, the [Virtual Machine Contributor](#) role allows a user to create and manage virtual machines. If the built-in roles don't meet the specific needs of your organization, you can create your own [custom roles for Azure resources](#).

Azure has data operations that enable you to grant access to data within an object. For example, if a user has read data access to a storage account, then they can read the blobs or messages within that storage account. For more information, see [Understand role definitions for Azure resources](#).

Scope

Scope is the set of resources that the access applies to. When you assign a role, you can further limit the actions allowed by defining a scope. This is helpful if you want to make someone a [Website Contributor](#), but only for one resource group.

In Azure, you can specify a scope at multiple levels: [management group](#), subscription, resource group, or resource. Scopes are structured in a parent-child relationship.



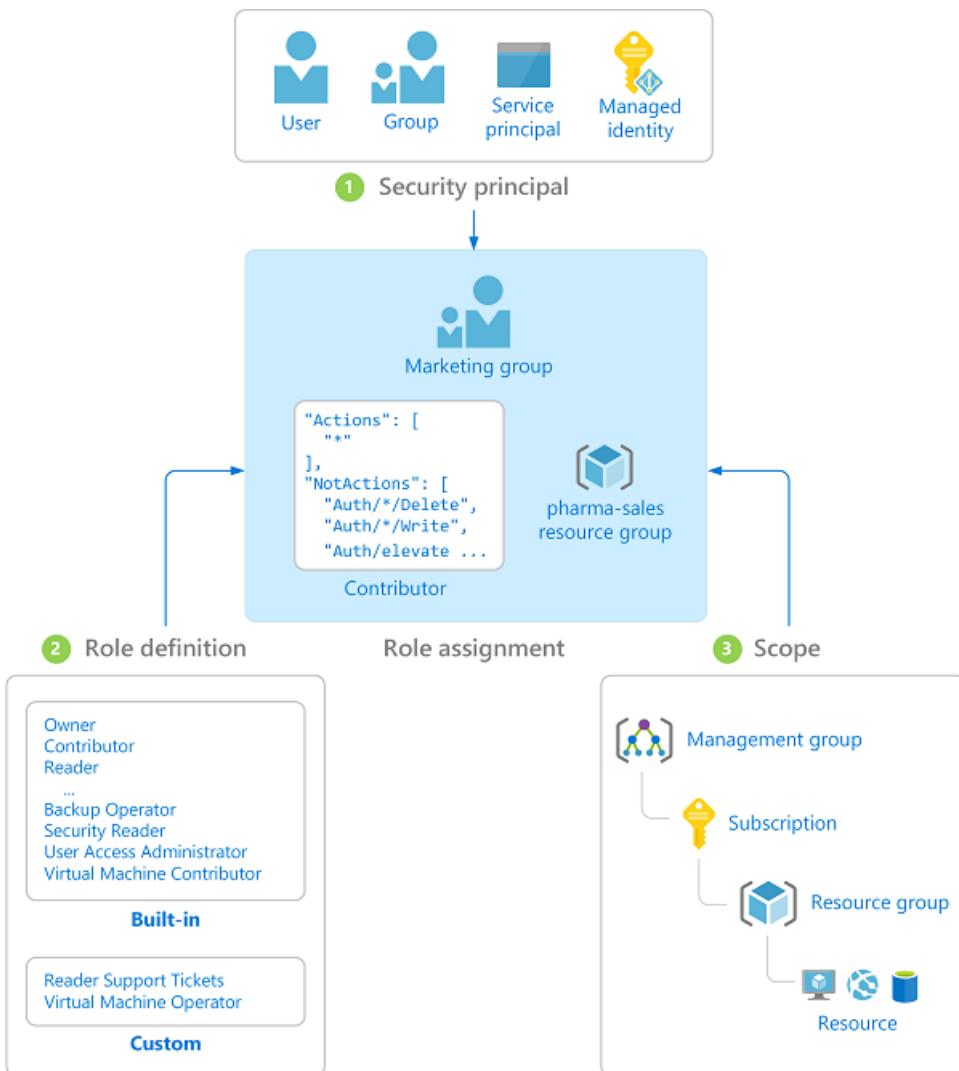
When you grant access at a parent scope, those permissions are inherited to the child scopes. For example:

- If you assign the [Owner](#) role to a user at the management group scope, that user can manage everything in all subscriptions in the management group.
- If you assign the [Reader](#) role to a group at the subscription scope, the members of that group can view every resource group and resource in the subscription.
- If you assign the [Contributor](#) role to an application at the resource group scope, it can manage resources of all types in that resource group, but not other resource groups in the subscription.

Role assignments

A *role assignment* is the process of attaching a role definition to a user, group, service principal, or managed identity at a particular scope for the purpose of granting access. Access is granted by creating a role assignment, and access is revoked by removing a role assignment.

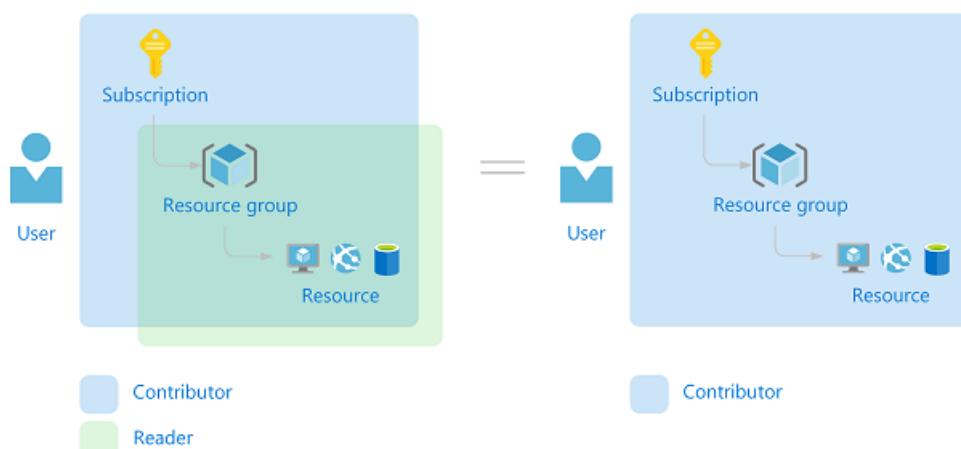
The following diagram shows an example of a role assignment. In this example, the Marketing group has been assigned the [Contributor](#) role for the pharma-sales resource group. This means that users in the Marketing group can create or manage any Azure resource in the pharma-sales resource group. Marketing users do not have access to resources outside the pharma-sales resource group, unless they are part of another role assignment.



You can create role assignments using the Azure portal, Azure CLI, Azure PowerShell, Azure SDKs, or REST APIs. You can have up to **2000** role assignments in each subscription and **500** role assignments in each management group. To create and remove role assignments, you must have `Microsoft.Authorization/roleAssignments/*` permission. This permission is granted through the [Owner](#) or [User Access Administrator](#) roles.

Multiple role assignments

So what happens if you have multiple overlapping role assignments? RBAC is an additive model, so your effective permissions are the addition of your role assignments. Consider the following example where a user is granted the Contributor role at the subscription scope and the Reader role on a resource group. The addition of the Contributor permissions and the Reader permissions is effectively the Contributor role for the resource group. Therefore, in this case, the Reader role assignment has no impact.



Deny assignments

Previously, RBAC was an allow-only model with no deny, but now RBAC supports deny assignments in a limited way. Similar to a role assignment, a *deny assignment* attaches a set of deny actions to a user, group, service principal, or managed identity at a particular scope for the purpose of denying access. A role assignment defines a set of actions that are *allowed*, while a deny assignment defines a set of actions that are *not allowed*. In other words, deny assignments block users from performing specified actions even if a role assignment grants them access. Deny assignments take precedence over role assignments. For more information, see [Understand deny assignments for Azure resources](#).

How RBAC determines if a user has access to a resource

The following are the high-level steps that RBAC uses to determine if you have access to a resource on the management plane. This is helpful to understand if you are trying to troubleshoot an access issue.

1. A user (or service principal) acquires a token for Azure Resource Manager.

The token includes the user's group memberships (including transitive group memberships).

2. The user makes a REST API call to Azure Resource Manager with the token attached.
3. Azure Resource Manager retrieves all the role assignments and deny assignments that apply to the resource upon which the action is being taken.
4. Azure Resource Manager narrows the role assignments that apply to this user or their group and determines what roles the user has for this resource.
5. Azure Resource Manager determines if the action in the API call is included in the roles the user has for this resource.
6. If the user doesn't have a role with the action at the requested scope, access is not granted. Otherwise, Azure Resource Manager checks if a deny assignment applies.
7. If a deny assignment applies, access is blocked. Otherwise access is granted.

License requirements

Using this feature is free and included in your Azure subscription.

Next steps

- [Quickstart: View the access a user has to Azure resources using the Azure portal](#)
- [Manage access to Azure resources using RBAC and the Azure portal](#)
- [Understand the different roles in Azure](#)
- [Enterprise Cloud Adoption: Resource access management in Azure](#)

Apply policies to Windows VMs with Azure Resource Manager

11/13/2019 • 3 minutes to read • [Edit Online](#)

By using policies, an organization can enforce various conventions and rules throughout the enterprise. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization. In this article, we describe how you can use Azure Resource Manager policies to define the desired behavior for your organization's Virtual Machines.

For an introduction to policies, see [What is Azure Policy?](#).

Permitted Virtual Machines

To ensure that virtual machines for your organization are compatible with an application, you can restrict the permitted operating systems. In the following policy example, you allow only Windows Server 2012 R2 Datacenter Virtual Machines to be created:

```
{
  "if": {
    "allOf": [
      {
        "field": "type",
        "in": [
          "Microsoft.Compute/disks",
          "Microsoft.Compute/virtualMachines",
          "Microsoft.Compute/VirtualMachineScaleSets"
        ]
      },
      {
        "not": {
          "allOf": [
            {
              "field": "Microsoft.Compute/imagePublisher",
              "in": [
                "MicrosoftWindowsServer"
              ]
            },
            {
              "field": "Microsoft.Compute/imageOffer",
              "in": [
                "WindowsServer"
              ]
            },
            {
              "field": "Microsoft.Compute/imageSku",
              "in": [
                "2012-R2-Datacenter"
              ]
            },
            {
              "field": "Microsoft.Compute/imageVersion",
              "in": [
                "latest"
              ]
            }
          ]
        }
      }
    ],
    "then": {
      "effect": "deny"
    }
  }
}
```

Use a wild card to modify the preceding policy to allow any Windows Server Datacenter image:

```
{
  "field": "Microsoft.Compute/imageSku",
  "like": "*Datacenter"
}
```

Use anyOf to modify the preceding policy to allow any Windows Server 2012 R2 Datacenter or higher image:

```
{
  "anyOf": [
    {
      "field": "Microsoft.Compute/imageSku",
      "like": "2012-R2-Datacenter*"
    },
    {
      "field": "Microsoft.Compute/imageSku",
      "like": "2016-Datacenter*"
    }
  ]
}
```

For information about policy fields, see [Policy aliases](#).

Managed disks

To require the use of managed disks, use the following policy:

```
{
  "if": {
    "anyOf": [
      {
        "allOf": [
          {
            "field": "type",
            "equals": "Microsoft.Compute/virtualMachines"
          },
          {
            "field": "Microsoft.Compute/virtualMachines/osDisk.uri",
            "exists": true
          }
        ]
      },
      {
        "allOf": [
          {
            "field": "type",
            "equals": "Microsoft.Compute/VirtualMachineScaleSets"
          },
          {
            "anyOf": [
              {
                "field": "Microsoft.Compute/VirtualMachineScaleSets/osDisk.vhdContainers",
                "exists": true
              },
              {
                "field": "Microsoft.Compute/VirtualMachineScaleSets/osdisk.imageUrl",
                "exists": true
              }
            ]
          }
        ]
      }
    ],
    "then": {
      "effect": "deny"
    }
  }
}
```

Images for Virtual Machines

For security reasons, you can require that only approved custom images are deployed in your environment. You can specify either the resource group that contains the approved images, or the specific approved images.

The following example requires images from an approved resource group:

```
{  
    "if": {  
        "allof": [  
            {  
                "field": "type",  
                "in": [  
                    "Microsoft.Compute/virtualMachines",  
                    "Microsoft.Compute/VirtualMachineScaleSets"  
                ]  
            },  
            {  
                "not": {  
                    "field": "Microsoft.Compute/imageId",  
                    "contains": "resourceGroups/CustomImage"  
                }  
            }  
        ]  
    },  
    "then": {  
        "effect": "deny"  
    }  
}
```

The following example specifies the approved image IDs:

```
{  
    "field": "Microsoft.Compute/imageId",  
    "in": ["{imageId1}","{imageId2}"]  
}
```

Virtual Machine extensions

You may want to forbid usage of certain types of extensions. For example, an extension may not be compatible with certain custom virtual machine images. The following example shows how to block a specific extension. It uses publisher and type to determine which extension to block.

```
{
  "if": {
    "allOf": [
      {
        "field": "type",
        "equals": "Microsoft.Compute/virtualMachines/extensions"
      },
      {
        "field": "Microsoft.Compute/virtualMachines/extensions/publisher",
        "equals": "Microsoft.Compute"
      },
      {
        "field": "Microsoft.Compute/virtualMachines/extensions/type",
        "equals": "{extension-type}"
      }
    ]
  },
  "then": {
    "effect": "deny"
  }
}
```

Azure Hybrid Use Benefit

When you have an on-premises license, you can save the license fee on your virtual machines. When you don't have the license, you should forbid the option. The following policy forbids usage of Azure Hybrid Use Benefit (AHUB):

```
{
  "if": {
    "allOf": [
      {
        "field": "type",
        "in": [ "Microsoft.Compute/virtualMachines", "Microsoft.Compute/VirtualMachineScaleSets" ]
      },
      {
        "field": "Microsoft.Compute/licenseType",
        "exists": true
      }
    ]
  },
  "then": {
    "effect": "deny"
  }
}
```

Next steps

- After defining a policy rule (as shown in the preceding examples), you need to create the policy definition and assign it to a scope. The scope can be a subscription, resource group, or resource. To assign policies, see [Use Azure portal to assign and manage resource policies](#), [Use PowerShell to assign policies](#), or [Use Azure CLI to assign policies](#).
- For an introduction to resource policies, see [What is Azure Policy?](#).
- For guidance on how enterprises can use Resource Manager to effectively manage subscriptions, see [Azure enterprise scaffold - prescriptive subscription governance](#).

Set up Key Vault for virtual machines in Azure Resource Manager

11/13/2019 • 2 minutes to read • [Edit Online](#)

NOTE

Azure has two different deployment models you can use to create and work with resources: [Azure Resource Manager](#) and [classic](#). This article covers the use of the Resource Manager deployment model. We recommend the Resource Manager deployment model for new deployments instead of the classic deployment model.

In Azure Resource Manager stack, secrets/certificates are modeled as resources that are provided by the resource provider of Key Vault. To learn more about Key Vault, see [What is Azure Key Vault?](#)

NOTE

1. In order for Key Vault to be used with Azure Resource Manager virtual machines, the **EnabledForDeployment** property on Key Vault must be set to true. You can do this in various clients.
2. The Key Vault needs to be created in the same subscription and location as the Virtual Machine.

Use PowerShell to set up Key Vault

To create a key vault by using PowerShell, see [Set and retrieve a secret from Azure Key Vault using PowerShell](#).

For new key vaults, you can use this PowerShell cmdlet:

```
New-AzKeyVault -VaultName 'ContosoKeyVault' -ResourceGroupName 'ContosoResourceGroup' -Location 'East Asia' -EnabledForDeployment
```

For existing key vaults, you can use this PowerShell cmdlet:

```
Set-AzKeyVaultAccessPolicy -VaultName 'ContosoKeyVault' -EnabledForDeployment
```

Use CLI to set up Key Vault

To create a key vault by using the command-line interface (CLI), see [Manage Key Vault using CLI](#).

For CLI, you have to create the key vault before you assign the deployment policy. You can do this by using the following command:

```
az keyvault create --name "ContosoKeyVault" --resource-group "ContosoResourceGroup" --location "EastAsia"
```

Then to enable Key Vault for use with template deployment, run the following command:

```
az keyvault update --name "ContosoKeyVault" --resource-group "ContosoResourceGroup" --enabled-for-deployment "true"
```

Use templates to set up Key Vault

While you use a template, you need to set the `enabledForDeployment` property to `true` for the Key Vault resource.

```
{  
  "type": "Microsoft.KeyVault/vaults",  
  "name": "ContosoKeyVault",  
  "apiVersion": "2015-06-01",  
  "location": "<location-of-key-vault>",  
  "properties": {  
    "enabledForDeployment": "true",  
    ....  
    ....  
  }  
}
```

For other options that you can configure when you create a key vault by using templates, see [Create a key vault](#).

What if an Azure service disruption impacts Azure VMs

2/10/2020 • 3 minutes to read • [Edit Online](#)

At Microsoft, we work hard to make sure that our services are always available to you when you need them. Forces beyond our control sometimes impact us in ways that cause unplanned service disruptions.

Microsoft provides a Service Level Agreement (SLA) for its services as a commitment for uptime and connectivity. The SLA for individual Azure services can be found at [Azure Service Level Agreements](#).

Azure already has many built-in platform features that support highly available applications. For more about these services, read [Disaster recovery and high availability for Azure applications](#).

This article covers a true disaster recovery scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. These are rare occurrences, but you must prepare for the possibility that there is an outage of an entire region. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you have enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region is not recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

To help you handle these rare occurrences, we provide the following guidance for Azure virtual machines in the case of a service disruption of the entire region where your Azure virtual machine application is deployed.

Option 1: Initiate a failover by using Azure Site Recovery

You can configure Azure Site Recovery for your VMs so that you can recover your application with a single click in matter of minutes. You can replicate to Azure region of your choice and not restricted to paired regions. You can get started by [replicating your virtual machines](#). You can [create a recovery plan](#) so that you can automate the entire failover process for your application. You can [test your failovers](#) beforehand without impacting production application or the ongoing replication. In the event of a primary region disruption, you just [initiate a failover](#) and bring your application in target region.

Option 2: Wait for recovery

In this case, no action on your part is required. Know that we are working diligently to restore service availability. You can see the current service status on our [Azure Service Health Dashboard](#).

This is the best option if you have not set up Azure Site Recovery, read-access geo-redundant storage, or geo-redundant storage prior to the disruption. If you have set up geo-redundant storage or read-access geo-redundant storage for the storage account where your VM virtual hard drives (VHDs) are stored, you can look to recover the base image VHD and try to provision a new VM from it. This is not a preferred option because there are no guarantees of synchronization of data. Consequently, this option is not guaranteed to work.

NOTE

Be aware that you do not have any control over this process, and it will only occur for region-wide service disruptions. Because of this, you must also rely on other application-specific backup strategies to achieve the highest level of availability. For more information, see the section on [Data strategies for disaster recovery](#).

Next steps

- Start protecting your applications running on [Azure virtual machines](#) using Azure Site Recovery
- To learn more about how to implement a disaster recovery and high availability strategy, see [Disaster recovery and high availability for Azure applications](#).
- To develop a detailed technical understanding of a cloud platform's capabilities, see [Azure resiliency technical guidance](#).
- If the instructions are not clear, or if you would like Microsoft to do the operations on your behalf, contact [Customer Support](#).

2 minutes to read

Back up a virtual machine in Azure with the CLI

11/18/2019 • 5 minutes to read • [Edit Online](#)

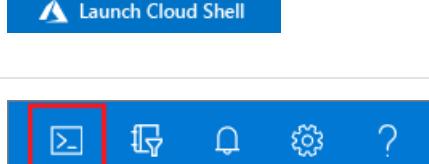
The Azure CLI is used to create and manage Azure resources from the command line or in scripts. You can protect your data by taking backups at regular intervals. Azure Backup creates recovery points that can be stored in geo-redundant recovery vaults. This article details how to back up a virtual machine (VM) in Azure with the Azure CLI. You can also perform these steps with [Azure PowerShell](#) or in the [Azure portal](#).

This quickstart enables backup on an existing Azure VM. If you need to create a VM, you can [create a VM with the Azure CLI](#).

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

| OPTION | EXAMPLE/LINK |
|---|--|
| Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell. |  |
| Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser. |  |
| Select the Cloud Shell button on the menu bar at the upper right in the Azure portal . |  |

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

To install and use the CLI locally, you must run Azure CLI version 2.0.18 or later. To find the CLI version, run `az --version`. If you need to install or upgrade, see [Install the Azure CLI](#).

Create a recovery services vault

A Recovery Services vault is a logical container that stores the backup data for each protected resource, such as Azure VMs. When the backup job for a protected resource runs, it creates a recovery point inside the Recovery Services vault. You can then use one of these recovery points to restore data to a given point in time.

Create a Recovery Services vault with `az backup vault create`. Specify the same resource group and location as the VM you wish to protect. If you used the [VM quickstart](#), then you created:

- a resource group named *myResourceGroup*,
- a VM named *myVM*,
- resources in the *eastus* location.

```
az backup vault create --resource-group myResourceGroup \
--name myRecoveryServicesVault \
--location eastus
```

By default, the Recovery Services vault is set for Geo-Redundant storage. Geo-Redundant storage ensures your backup data is replicated to a secondary Azure region that is hundreds of miles away from the primary region. If the storage redundancy setting needs to be modified, use [az backup vault backup-properties set](#) cmdlet.

```
az backup vault backup-properties set \
--name myRecoveryServicesVault \
--resource-group myResourceGroup \
--backup-storage-redundancy "LocallyRedundant/GeoRedundant"
```

Enable backup for an Azure VM

Create a protection policy to define: when a backup job runs, and how long the recovery points are stored. The default protection policy runs a backup job each day and retains recovery points for 30 days. You can use these default policy values to quickly protect your VM. To enable backup protection for a VM, use [az backup protection enable-for-vm](#). Specify the resource group and VM to protect, then the policy to use:

```
az backup protection enable-for-vm \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--vm myVM \
--policy-name DefaultPolicy
```

NOTE

If the VM is not in the same resource group as that of vault, then myResourceGroup refers to the resource group where vault was created. Instead of VM name, provide the VM ID as indicated below.

```
az backup protection enable-for-vm \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--vm $(az vm show -g VMResourceGroup -n MyVm --query id | tr -d '') \
--policy-name DefaultPolicy
```

IMPORTANT

While using CLI to enable backup for multiple VMs at once, ensure that a single policy doesn't have more than 100 VMs associated with it. This is a [recommended best practice](#). Currently, the PS client doesn't explicitly block if there are more than 100 VMs but the check is planned to be added in the future.

Start a backup job

To start a backup now rather than wait for the default policy to run the job at the scheduled time, use [az backup protection backup-now](#). This first backup job creates a full recovery point. Each backup job after this initial backup

creates incremental recovery points. Incremental recovery points are storage and time-efficient, as they only transfer changes made since the last backup.

The following parameters are used to back up the VM:

- `--container-name` is the name of your VM
- `--item-name` is the name of your VM
- `--retain-until` value should be set to the last available date, in UTC time format (**dd-mm-yyyy**), that you wish the recovery point to be available

The following example backs up the VM named *myVM* and sets the expiration of the recovery point to October 18, 2017:

```
az backup protection backup-now \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --container-name myVM \
    --item-name myVM \
    --retain-until 18-10-2017
```

Monitor the backup job

To monitor the status of backup jobs, use [az backup job list](#):

```
az backup job list \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --output table
```

The output is similar to the following example, which shows the backup job is *InProgress*:

| Name | Operation | Status | Item Name | Start Time UTC | Duration |
|----------|-----------------|------------|-----------|---------------------|----------------|
| a0a8e5e6 | Backup | InProgress | myvm | 2017-09-19T03:09:21 | 0:00:48.718366 |
| fe5d0414 | ConfigureBackup | Completed | myvm | 2017-09-19T03:03:57 | 0:00:31.191807 |

When the *Status* of the backup job reports *Completed*, your VM is protected with Recovery Services and has a full recovery point stored.

Clean up deployment

When no longer needed, you can disable protection on the VM, remove the restore points and Recovery Services vault, then delete the resource group and associated VM resources. If you used an existing VM, you can skip the final [az group delete](#) command to leave the resource group and VM in place.

If you want to try a Backup tutorial that explains how to restore data for your VM, go to [Next steps](#).

```
az backup protection disable \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--delete-backup-data true
az backup vault delete \
--resource-group myResourceGroup \
--name myRecoveryServicesVault \
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you created a Recovery Services vault, enabled protection on a VM, and created the initial recovery point. To learn more about Azure Backup and Recovery Services, continue to the tutorials.

[Back up multiple Azure VMs](#)

Use Azure portal to back up multiple virtual machines

11/18/2019 • 6 minutes to read • [Edit Online](#)

When you back up data in Azure, you store that data in an Azure resource called a Recovery Services vault. The Recovery Services vault resource is available from the Settings menu of most Azure services. The benefit of having the Recovery Services vault integrated into the Settings menu of most Azure services makes it easy to back up data. However, individually working with each database or virtual machine in your business is tedious. What if you want to back up the data for all virtual machines in one department, or in one location? It is easy to back up multiple virtual machines by creating a backup policy and applying that policy to the desired virtual machines. This tutorial explains how to:

- Create a Recovery Services vault
- Define a backup policy
- Apply the backup policy to protect multiple virtual machines
- Trigger an on-demand backup job for the protected virtual machines

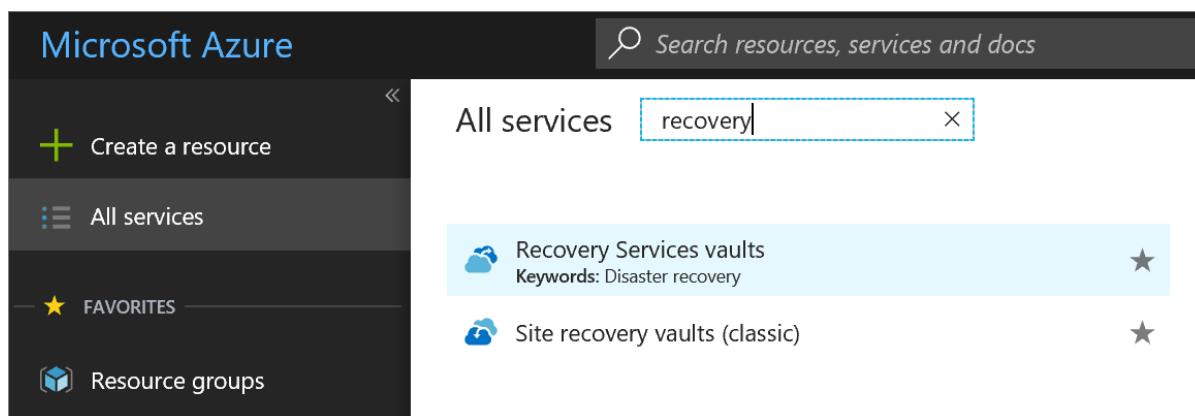
Sign in to the Azure portal

Sign in to the [Azure portal](#).

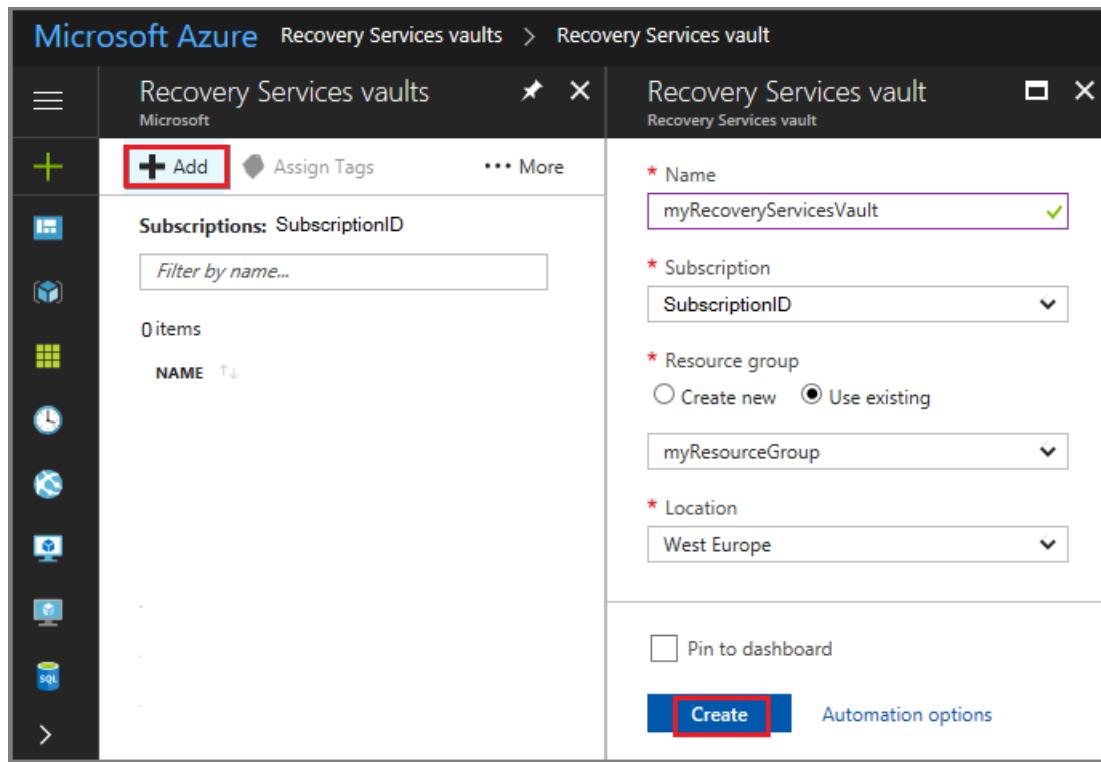
Create a Recovery Services vault

The Recovery Services vault contains the backup data, and the backup policy applied to the protected virtual machines. Backing up virtual machines is a local process. You cannot back up a virtual machine from one location to a Recovery Services vault in another location. So, for each Azure location that has virtual machines to be backed up, at least one Recovery Services vault must exist in that location.

1. On the left-hand menu, select **All services** and in the services list, type *Recovery Services*. As you type, the list of resources filters. When you see Recovery Services vaults in the list, select it to open the Recovery Services vaults menu.



2. In the **Recovery Services vaults** menu, click **Add** to open the Recovery Services vault menu.



3. In the Recovery Services vault menu,

- Type *myRecoveryServicesVault* in **Name**.
- The current subscription ID appears in **Subscription**. If you have additional subscriptions, you could choose another subscription for the new vault.
- For **Resource group**, select **Use existing** and choose *myResourceGroup*. If *myResourceGroup* doesn't exist, select **Create new** and type *myResourceGroup*.
- From the **Location** drop-down menu, choose *West Europe*.
- Click **Create** to create your Recovery Services vault.

A Recovery Services vault must be in the same location as the virtual machines being protected. If you have virtual machines in multiple regions, create a Recovery Services vault in each region. This tutorial creates a Recovery Services vault in *West Europe* because that is where *myVM* (the virtual machine created with the quickstart) was created.

It can take several minutes for the Recovery Services vault to be created. Monitor the status notifications in the upper right-hand area of the portal. Once your vault is created, it appears in the list of Recovery Services vaults.

When you create a Recovery Services vault, by default the vault has geo-redundant storage. To provide data resiliency, geo-redundant storage replicates the data multiple times across two Azure regions.

Set backup policy to protect VMs

After creating the Recovery Services vault, the next step is to configure the vault for the type of data, and to set the backup policy. Backup policy is the schedule for how often and when recovery points are taken. Policy also includes the retention range for the recovery points. For this tutorial, let's assume your business is a sports complex with a hotel, stadium, and restaurants and concessions, and you are protecting the data on the virtual machines. The following steps create a backup policy for the financial data.

1. From the list of Recovery Services vaults, select **myRecoveryServicesVault** to open its dashboard.

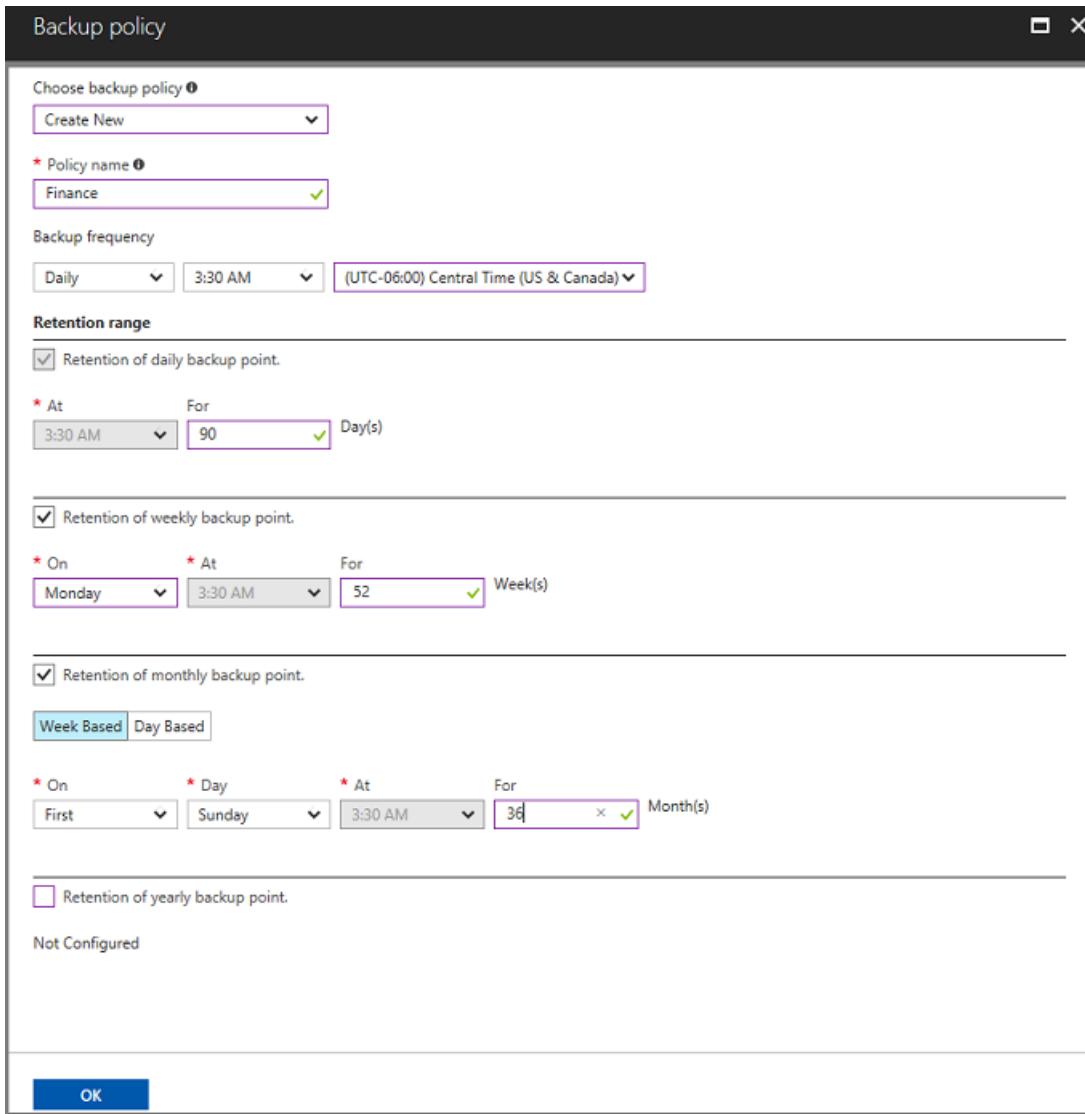
2. On the vault dashboard menu, click **Backup** to open the Backup menu.
3. On the Backup Goal menu, in the **Where is your workload running?** drop-down menu, choose *Azure*. From the **What do you want to backup?** drop-down, choose *Virtual machine*, and click **Backup**.

These actions prepare the Recovery Services vault for interacting with a virtual machine. Recovery Services vaults have a default policy that creates a restore point each day, and retains the restore points for 30 days.

4. To create a new policy, on the Backup policy menu, from the **Choose backup policy** drop-down menu, select *Create New*.

5. In the **Backup policy** menu, for **Policy Name** type *Finance*. Enter the following changes for the Backup policy:

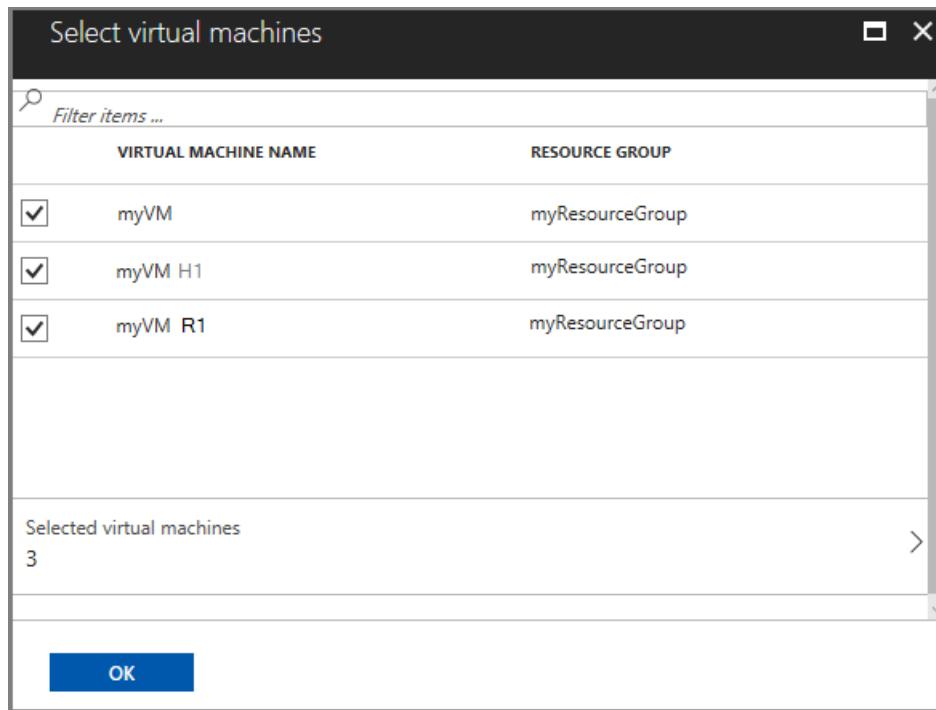
- For **Backup frequency** set the timezone for *Central Time*. Since the sports complex is in Texas, the owner wants the timing to be local. Leave the backup frequency set to Daily at 3:30AM.
- For **Retention of daily backup point**, set the period to 90 days.
- For **Retention of weekly backup point**, use the *Monday* restore point and retain it for 52 weeks.
- For **Retention of monthly backup point**, use the restore point from First Sunday of the month, and retain it for 36 months.
- Deselect the **Retention of yearly backup point** option. The leader of Finance doesn't want to keep data longer than 36 months.
- Click **OK** to create the backup policy.



After creating the backup policy, associate the policy with the virtual machines.

6. In the **Select virtual machines** dialog, select *myVM* and click **OK** to deploy the backup policy to the virtual machines.

All virtual machines that are in the same location, and are not already associated with a backup policy, appear. *myVMH1* and *myVMR1* are selected to be associated with the *Finance* policy.



When the deployment completes, you receive a notification that deployment successfully completed.

Initial backup

You have enabled backup for the Recovery Services vaults, but an initial backup has not been created. It is a disaster recovery best practice to trigger the first backup, so that your data is protected.

To run an on-demand backup job:

1. On the vault dashboard, click **3** under **Backup Items**, to open the Backup Items menu.

The **Backup Items** menu opens.

2. On the **Backup Items** menu, click **Azure Virtual Machine** to open the list of virtual machines associated with the vault.

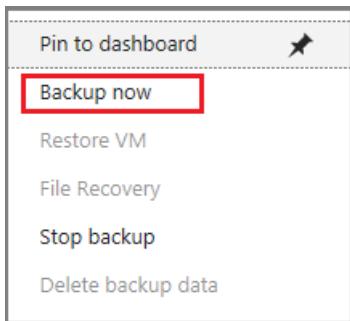
| NAME | RESOURCE GROUP | BACKUP PRE-CHECK | LAST BACKUP STATUS | LATEST RESTORE POI |
|-------|-------------------|------------------|---------------------------|--------------------|
| myVM | myResourceGroup | Passed | Warning(Initial backu...) | Context menu |
| buntu | rasquill-security | Passed | Warning(Initial backu...) | ... |
| ops | rhelpfiles | Passed | Warning(Initial backu...) | ... |

The **Backup Items** list opens.

| NAME | RESOURCE GROUP | BACKUP PRE-CHECK | LAST BACKUP STATUS | LATEST RESTORE POI | Context menu |
|-------|-------------------|------------------|---------------------------|--------------------|--------------|
| myVM | myResourceGroup | Passed | Warning(Initial backu...) | ... | ... |
| buntu | rasquill-security | Passed | Warning(Initial backu...) | ... | ... |
| ops | rhelpfiles | Passed | Warning(Initial backu...) | ... | ... |

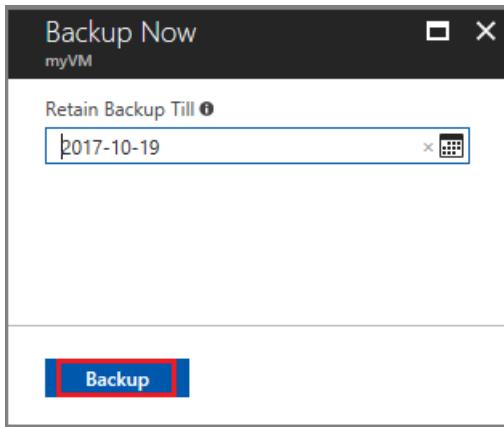
3. On the **Backup Items** list, click the ellipses ... to open the Context menu.

4. On the Context menu, select **Backup now**.



The Backup Now menu opens.

5. On the Backup Now menu, enter the last day to retain the recovery point, and click **Backup**.



Deployment notifications let you know the backup job has been triggered, and that you can monitor the progress of the job on the Backup jobs page. Depending on the size of your virtual machine, creating the initial backup may take a while.

When the initial backup job completes, you can see its status in the Backup job menu. The on-demand backup job created the initial restore point for *myVM*. If you want to back up other virtual machines, repeat these steps for each virtual machine.

| NAME | RESOURCE GROUP | BACKUP PRE-CHECK | LAST BACKUP STATUS | LATEST RESTORE POINT | ... |
|------|-----------------|------------------|--------------------|----------------------|-----|
| myVM | myResourceGroup | Passed | Success | 9/19/2017 6:52:32 PM | ... |

Clean up resources

If you plan to continue on to work with subsequent tutorials, do not clean up the resources created in this tutorial. If you do not plan to continue, use the following steps to delete all resources created by this tutorial in the Azure portal.

1. On the **myRecoveryServicesVault** dashboard, click **3** under **Backup Items**, to open the Backup Items menu.

The screenshot shows the Azure Recovery Services vault overview page. The left sidebar contains a navigation menu with various options such as Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Properties, Locks, Automation script, Backup, Site Recovery, Jobs, Alerts and Events, and Backup Reports. The main content area is divided into sections: 'Essentials' and 'Monitoring'. In the 'Essentials' section, it shows a resource group named 'myResourceGroup' with a status of 'Active', located in 'West Europe', and associated with a specific subscription. It also displays 'Backup items' (3), 'Backup management servers' (0), and 'Replicated items' (0). The 'Monitoring' section includes a 'Backup Alerts (last 24...)' table showing 0 Critical and 0 Warning alerts, and a 'Backup Pre-Check Status (Azure VMs)' chart indicating 0 total critical and 0 warning issues. A callout at the top right encourages users to take a survey about their experience with Azure Backup and Site Recovery.

2. On the **Backup Items** menu, click **Azure Virtual Machine** to open the list of virtual machines associated with the vault.

| BACKUP MANAGEMENT TYPE | BACKUP ITEM COUNT |
|------------------------|-------------------|
| Azure Virtual Machine | 3 |
| Azure Backup Agent | 0 |
| Azure Backup Server | 0 |

The **Backup Items** list opens.

3. In the **Backup Items** menu, click the ellipsis to open the Context menu.

Backup Items (Azure Virtual Machine)

myRecoveryServicesVault

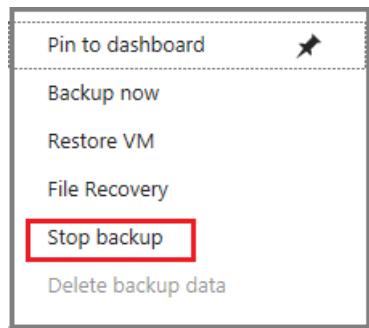
Refresh Add Filter

Fetching data from service completed.

Filter items ...

| NAME | RESOURCE GROUP | BACKUP PRE-CHECK | LAST BACKUP STATUS | LATEST RESTORE POINT | ... |
|---------|-----------------|------------------|--------------------|----------------------|---|
| myVM | myResourceGroup | Passed | Success | 9/19/2017 6:22:37 PM | ... |
| myVM H1 | myResourceGroup | Passed | Success | 9/19/2017 6:32:35 PM | ... |
| myVM R1 | myResourceGroup | Passed | Success | 9/19/2017 6:56:17 PM | ... |

4. On the context menu, select **Stop backup** to open Stop Backup menu.



5. In the **Stop Backup** menu, select the upper drop-down menu and choose **Delete Backup Data**.
6. In the **Type the name of the Backup item** dialog, type **myVM**.
7. Once the backup item is verified (a check mark appears), **Stop backup** button is enabled. Click **Stop Backup** to stop the policy and delete the restore points.

Stop Backup

myVM

Delete Backup Data

i This option will stop all scheduled backup jobs, deletes backup data and can't be undone.

* Type the name of Backup Item
myVM

Reason (optional)
Others

Comments

Stop backup

8. In the **myRecoveryServicesVault** menu, click **Delete**.

The screenshot shows the Azure portal interface for a Recovery Services vault named 'myRecoveryServicesVault'. The left sidebar includes options like 'Overview', 'Activity log', 'Access control (IAM)', and 'Tags'. The main content area has tabs for 'Backup', 'Replicate', and 'Delete', with 'Delete' being the active tab and highlighted with a red box. A survey prompt at the top right encourages users to take a survey about their experience with Azure Backup and Site Recovery. Below the prompt, the 'Essentials' section displays resource group information: 'Resource group (change) myResourceGroup', 'Status Active', 'Backup items 0', and 'Backup management servers 0'.

Once the vault is deleted, you return to the list of Recovery Services vaults.

Next steps

In this tutorial, you used the Azure portal to:

- Create a Recovery Services vault
- Set the vault to protect virtual machines
- Create a custom backup and retention policy
- Assign the policy to protect multiple virtual machines
- Trigger an on-demand back up for virtual machines

Continue to the next tutorial to restore an Azure virtual machine from disk.

[Restore VMs using CLI](#)

Restore a disk and create a recovered VM in Azure

2/10/2020 • 8 minutes to read • [Edit Online](#)

Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore the whole VM or individual files. This article explains how to restore a complete VM using CLI. In this tutorial you learn how to:

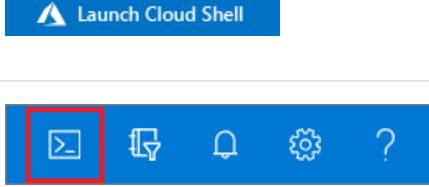
- List and select recovery points
- Restore a disk from a recovery point
- Create a VM from the restored disk

For information on using PowerShell to restore a disk and create a recovered VM, see [Back up and restore Azure VMs with PowerShell](#).

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

| OPTION | EXAMPLE/LINK |
|---|--|
| Select Try It in the upper-right corner of a code block. Selecting Try It doesn't automatically copy the code to Cloud Shell. |  |
| Go to https://shell.azure.com , or select the Launch Cloud Shell button to open Cloud Shell in your browser. |  |
| Select the Cloud Shell button on the menu bar at the upper right in the Azure portal . | |

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.18 or later. Run `az --version` to find the version. If you need to install or upgrade, see [Install the Azure CLI](#).

Prerequisites

This tutorial requires a Linux VM that has been protected with Azure Backup. To simulate an accidental VM deletion and recovery process, you create a VM from a disk in a recovery point. If you need a Linux VM that has

been protected with Azure Backup, see [Back up a virtual machine in Azure with the CLI](#).

Backup overview

When Azure initiates a backup, the backup extension on the VM takes a point-in-time snapshot. The backup extension is installed on the VM when the first backup is requested. Azure Backup can also take a snapshot of the underlying storage if the VM is not running when the backup takes place.

By default, Azure Backup takes a file system consistent backup. Once Azure Backup takes the snapshot, the data is transferred to the Recovery Services vault. To maximize efficiency, Azure Backup identifies and transfers only the blocks of data that have changed since the previous backup.

When the data transfer is complete, the snapshot is removed and a recovery point is created.

List available recovery points

To restore a disk, you select a recovery point as the source for the recovery data. As the default policy creates a recovery point each day and retains them for 30 days, you can keep a set of recovery points that allows you to select a particular point in time for recovery.

To see a list of available recovery points, use `az backup recoverypoint list`. The recovery point **name** is used to recover disks. In this tutorial, we want the most recent recovery point available. The `--query [0].name` parameter selects the most recent recovery point name as follows:

```
az backup recoverypoint list \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--backup-management-type AzureIaaSVM \
--container-name myVM \
--item-name myVM \
--query [0].name \
--output tsv
```

Restore a VM disk

IMPORTANT

It is very strongly recommended to use Az CLI version 2.0.74 or later to get all the benefits of a quick restore including managed disk restore. It is best if user always uses the latest version.

Managed disk restore

If the backed up VM has managed disks and if the intent is to restore managed disks from the recovery point, you first provide an Azure storage account. This storage account is used to store the VM configuration and the deployment template that can be later used to deploy the VM from the restored disks. Then, you also provide a target resource group for the managed disks to be restored into.

1. To create a storage account, use `az storage account create`. The storage account name must be all lowercase, and be globally unique. Replace *mystorageaccount* with your own unique name:

```
az storage account create \
--resource-group myResourceGroup \
--name mystorageaccount \
--sku Standard_LRS
```

2. Restore the disk from your recovery point with `az backup restore restore-disks`. Replace *mystorageaccount*

with the name of the storage account you created in the preceding command. Replace *myRecoveryPointName* with the recovery point name you obtained in the output from the previous [az backup recoverypoint list](#) command. **Also provide the target resource group to which the managed disks are restored into.**

```
az backup restore restore-disks \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--storage-account mystorageaccount \
--rp-name myRecoveryPointName \
--target-resource-group targetRG
```

WARNING

If target-resource-group is not provided then the managed disks will be restored as unmanaged disks to the given storage account. This will have significant consequences to the restore time since the time taken to restore the disks entirely depends on the given storage account.

Unmanaged disks restore

If the backed up VM has unmanaged disks and if the intent is to restore disks from the recovery point, you first provide an Azure storage account. This storage account is used to store the VM configuration and the deployment template that can be later used to deploy the VM from the restored disks. By default, the unmanaged disks will be restored to their original storage accounts. If user wishes to restore all unmanaged disks to one single place, then the given storage account can also be used as a staging location for those disks too.

In additional steps, the restored disk is used to create a VM.

1. To create a storage account, use [az storage account create](#). The storage account name must be all lowercase, and be globally unique. Replace *mystorageaccount* with your own unique name:

```
az storage account create \
--resource-group myResourceGroup \
--name mystorageaccount \
--sku Standard_LRS
```

2. Restore the disk from your recovery point with [az backup restore restore-disks](#). Replace *mystorageaccount* with the name of the storage account you created in the preceding command. Replace *myRecoveryPointName* with the recovery point name you obtained in the output from the previous [az backup recoverypoint list](#) command:

```
az backup restore restore-disks \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--storage-account mystorageaccount \
--rp-name myRecoveryPointName
```

As mentioned above, the unmanaged disks will be restored to their original storage account. This provides the best restore performance. But if all unmanaged disks need to be restored to given storage account, then use the relevant flag as shown below.

```

az backup restore restore-disks \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--storage-account mystorageaccount \
--rp-name myRecoveryPointName
--restore-to-staging-storage-account
```
Monitor the restore job

To monitor the status of restore job, use [az backup job list](https://docs.microsoft.com/cli/azure/backup/job?view=azure-cli-latest#az-backup-job-list):

```azurecli-interactive
az backup job list \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--output table
```

```

The output is similar to the following example, which shows the restore job is *InProgress*:

| Name     | Operation       | Status     | Item Name | Start Time UTC      | Duration       |
|----------|-----------------|------------|-----------|---------------------|----------------|
| 7f2ad916 | Restore         | InProgress | myvm      | 2017-09-19T19:39:52 | 0:00:34.520850 |
| a0a8e5e6 | Backup          | Completed  | myvm      | 2017-09-19T03:09:21 | 0:15:26.155212 |
| fe5d0414 | ConfigureBackup | Completed  | myvm      | 2017-09-19T03:03:57 | 0:00:31.191807 |

When the *Status* of the restore job reports *Completed*, the necessary information (VM configuration and the deployment template) has been restored to the storage account.

## Create a VM from the restored disk

The final step is to create a VM from the restored disks. You can use the deployment template downloaded to the given storage account to create the VM.

### Fetch the Job details

The resultant job details give the template URI that can be queried and deployed. Use the job show command to get more details for the triggered restored job.

```

az backup job show \
-v myRecoveryServicesVault \
-g myResourceGroup \
-n 1fc2d55d-f0dc-4ca6-ad48-aca0fe5d0414

```

The output of this query will give all details but we are interested only in the storage account contents. We can use the [query capability](#) of Azure CLI to fetch the relevant details

```

az backup job show \
-v myRecoveryServicesVault \
-g myResourceGroup \
-n 1fc2d55d-f0dc-4ca6-ad48-aca0fe5d0414 \
--query properties.extendedInfo.propertyBag

{
 "Config Blob Container Name": "myVM-daa1931199fd4a22ae601f46d8812276",
 "Config Blob Name": "config-myVM-1fc2d55d-f0dc-4ca6-ad48-aca0fe5d0414.json",
 "Config Blob Uri": "https://mystorageaccount.blob.core.windows.net/myVM-
daa1931199fd4a22ae601f46d8812276/config-appvm8-1fc2d55d-f0dc-4ca6-ad48-aca0519c0232.json",
 "Job Type": "Recover disks",
 "Recovery point time": "12/25/2019 10:07:11 PM",
 "Target Storage Account Name": "mystorageaccount",
 "Target resource group": "mystorageaccountRG",
 "Template Blob Uri": "https://mystorageaccount.blob.core.windows.net/myVM-
daa1931199fd4a22ae601f46d8812276/azuredeploy1fc2d55d-f0dc-4ca6-ad48-aca0519c0232.json"
}

```

## Fetch the deployment template

The template is not directly accessible since it is under a customer's storage account and the given container. We need the complete URL (along with a temporary SAS token) to access this template.

First, extract the template blob Uri from job details

```

az backup job show \
-v myRecoveryServicesVault \
-g myResourceGroup \
-n 1fc2d55d-f0dc-4ca6-ad48-aca0fe5d0414 \
--query properties.extendedInfo.propertyBag."Template Blob Uri"

"https://mystorageaccount.blob.core.windows.net/myVM-daa1931199fd4a22ae601f46d8812276/azuredeploy1fc2d55d-f0dc-
4ca6-ad48-aca0519c0232.json"

```

The template blob Uri will be of this format and extract the template name

```
https://<storageAccountName.blob.core.windows.net>/<containerName>/<templateName>
```

So, the template name from the above example will be `azuredeploy1fc2d55d-f0dc-4ca6-ad48-aca0519c0232.json` and the container name is `myVM-daa1931199fd4a22ae601f46d8812276`

Now get the SAS token for this container and template as detailed [here](#)

```
expiretime=$(date -u -d '30 minutes' +%Y-%m-%dT%H:%MZ)
connection=$(az storage account show-connection-string \
 --resource-group mystorageaccountRG \
 --name mystorageaccount \
 --query connectionString)
token=$(az storage blob generate-sas \
 --container-name myVM-daa1931199fd4a22ae601f46d8812276 \
 --name azuredeploy1fc2d55d-f0dc-4ca6-ad48-aca0519c0232.json \
 --expiry $expiretime \
 --permissions r \
 --output tsv \
 --connection-string $connection)
url=$(az storage blob url \
 --container-name myVM-daa1931199fd4a22ae601f46d8812276 \
 --name azuredeploy1fc2d55d-f0dc-4ca6-ad48-aca0519c0232.json \
 --output tsv \
 --connection-string $connection)
```

## Deploy the template to create the VM

Now deploy the template to create the VM as explained [here](#).

```
az group deployment create \
 --resource-group ExampleGroup \
 --template-uri $url?$token
```

To confirm that your VM has been created from your recovered disk, list the VMs in your resource group with [az vm list](#) as follows:

```
az vm list --resource-group myResourceGroup --output table
```

## Next steps

In this tutorial, you restored a disk from a recovery point and then created a VM from the disk. You learned how to:

- List and select recovery points
- Restore a disk from a recovery point
- Create a VM from the restored disk

Advance to the next tutorial to learn about restoring individual files from a recovery point.

[Restore files to a virtual machine in Azure](#)

# Restore files to a virtual machine in Azure

11/18/2019 • 6 minutes to read • [Edit Online](#)

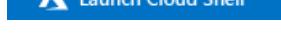
Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore the whole VM or individual files. This article details how to restore individual files. In this tutorial you learn how to:

- List and select recovery points
- Connect a recovery point to a VM
- Restore files from a recovery point

## Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

| OPTION                                                                                                                                                    | EXAMPLE/LINK                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Select <b>Try It</b> in the upper-right corner of a code block.<br>Selecting <b>Try It</b> doesn't automatically copy the code to Cloud Shell.            |  |
| Go to <a href="https://shell.azure.com">https://shell.azure.com</a> , or select the <b>Launch Cloud Shell</b> button to open Cloud Shell in your browser. |  |
| Select the <b>Cloud Shell</b> button on the menu bar at the upper right in the <a href="#">Azure portal</a> .                                             |  |

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.18 or later. Run `az --version` to find the version. If you need to install or upgrade, see [Install the Azure CLI](#).

## Prerequisites

This tutorial requires a Linux VM that has been protected with Azure Backup. To simulate an accidental file deletion and recovery process, you delete a page from a web server. If you need a Linux VM that runs a webserver and has been protected with Azure Backup, see [Back up a virtual machine in Azure with the CLI](#).

## Backup overview

When Azure initiates a backup, the backup extension on the VM takes a point-in-time snapshot. The backup extension is installed on the VM when the first backup is requested. Azure Backup can also take a snapshot of the underlying storage if the VM is not running when the backup takes place.

By default, Azure Backup takes a file system consistent backup. Once Azure Backup takes the snapshot, the data is transferred to the Recovery Services vault. To maximize efficiency, Azure Backup identifies and transfers only the blocks of data that have changed since the previous backup.

When the data transfer is complete, the snapshot is removed and a recovery point is created.

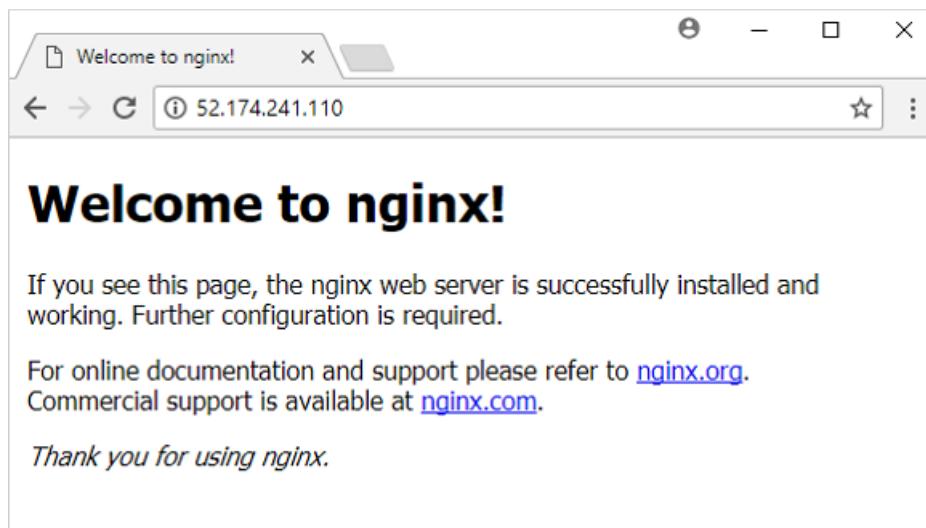
## Delete a file from a VM

If you accidentally delete or make changes to a file, you can restore individual files from a recovery point. This process allows you to browse the files backed up in a recovery point and restore only the files you need. In this example, we delete a file from a web server to demonstrate the file-level recovery process.

1. To connect to your VM, obtain the IP address of your VM with [az vm show](#):

```
az vm show --resource-group myResourceGroup --name myVM -d --query [publicIps] --o tsv
```

2. To confirm that your web site currently works, open a web browser to the public IP address of your VM. Leave the web browser window open.



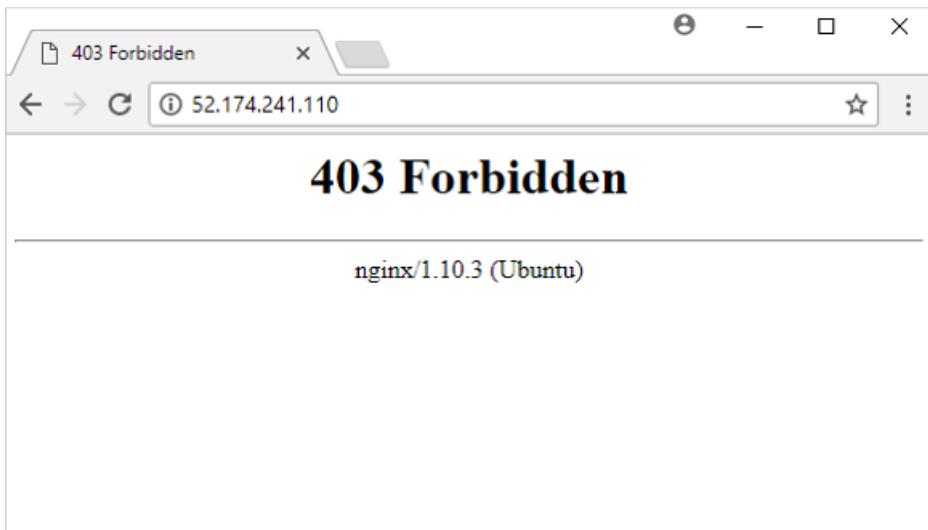
3. Connect to your VM with SSH. Replace *publicIpAddress* with the public IP address that you obtained in a previous command:

```
ssh publicIpAddress
```

4. Delete the default page from the web server at */var/www/html/index.nginx-debian.html* as follows:

```
sudo rm /var/www/html/index.nginx-debian.html
```

5. In your web browser, refresh the web page. The web site no longer loads the page, as shown in the following example:



6. Close the SSH session to your VM as follows:

```
exit
```

## Generate file recovery script

To restore your files, Azure Backup provides a script to run on your VM that connects your recovery point as a local drive. You can browse this local drive, restore files to the VM itself, then disconnect the recovery point. Azure Backup continues to back up your data based on the assigned policy for schedule and retention.

1. To list recovery points for your VM, use [az backup recoverypoint list](#). In this example, we select the most recent recovery point for the VM named *myVM* that is protected in *myRecoveryServicesVault*:

```
az backup recoverypoint list \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--query [0].name \
--output tsv
```

2. To obtain the script that connects, or mounts, the recovery point to your VM, use [az backup restore files mount-rp](#). The following example obtains the script for the VM named *myVM* that is protected in *myRecoveryServicesVault*.

Replace *myRecoveryPointName* with the name of the recovery point that you obtained in the preceding command:

```
az backup restore files mount-rp \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--rp-name myRecoveryPointName
```

The script is downloaded and a password is displayed, as in the following example:

```
File downloaded: myVM_we_1571974050985163527.sh. Use password c068a041ce12465
```

3. To transfer the script to your VM, use Secure Copy (SCP). Provide the name of your downloaded script, and

replace *publicIpAddress* with the public IP address of your VM. Make sure you include the trailing `:` at the end of the SCP command as follows:

```
scp myVM_we_1571974050985163527.sh 52.174.241.110:
```

## Restore file to your VM

With the recovery script copied to your VM, you can now connect the recovery point and restore files.

1. Connect to your VM with SSH. Replace *publicIpAddress* with the public IP address of your VM as follows:

```
ssh publicIpAddress
```

2. To allow your script to run correctly, add execute permissions with **chmod**. Enter the name of your own script:

```
chmod +x myVM_we_1571974050985163527.sh
```

3. To mount the recovery point, run the script. Enter the name of your own script:

```
./myVM_we_1571974050985163527.sh
```

As the script runs, you are prompted to enter a password to access the recovery point. Enter the password shown in the output from the previous [az backup restore files mount-rp](#) command that generated the recovery script.

The output from the script gives you the path for the recovery point. The following example output shows that the recovery point is mounted at `/home/azureuser/myVM-20170919213536/Volume1`:

```
Microsoft Azure VM Backup - File Recovery

Please enter the password as shown on the portal to securely connect to the recovery point. :
c068a041ce12465

Connecting to recovery point using ISCSI service...

Connection succeeded!

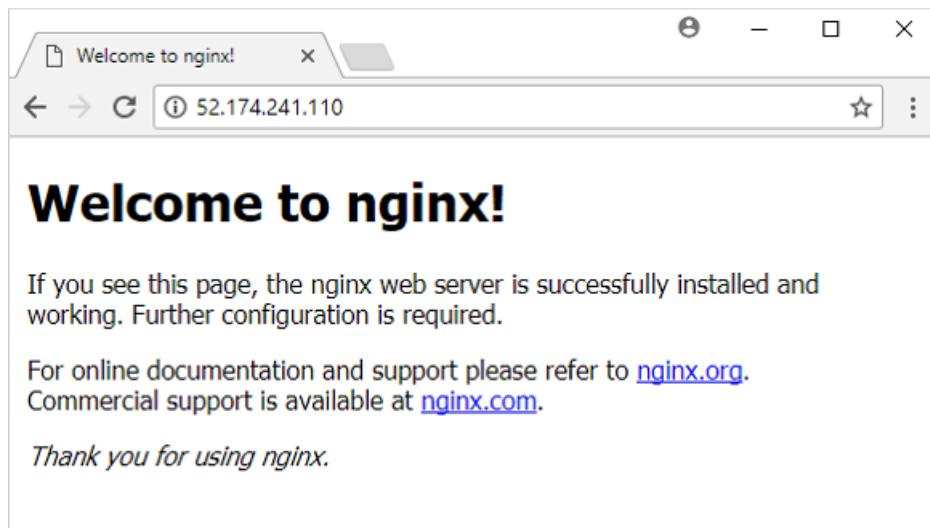
Please wait while we attach volumes of the recovery point to this machine...

***** Volumes of the recovery point and their mount paths on this machine *****
Sr.No. | Disk | Volume | MountPath
1) | /dev/sdc | /dev/sdc1 | /home/azureuser/myVM-20170919213536/Volume1
***** Open File Explorer to browse for files. *****
```

4. Use **cp** to copy the NGINX default web page from the mounted recovery point back to the original file location. Replace the `/home/azureuser/myVM-20170919213536/Volume1` mount point with your own location:

```
sudo cp /home/azureuser/myVM-20170919213536/Volume1/var/www/html/index.nginx-debian.html /var/www/html/
```

5. In your web browser, refresh the web page. The web site now loads correctly again, as shown in the following example:



6. Close the SSH session to your VM as follows:

```
exit
```

7. Unmount the recovery point from your VM with `az backup restore files unmount-rp`. The following example unmounts the recovery point from the VM named *myVM* in *myRecoveryServicesVault*.

Replace *myRecoveryPointName* with the name of your recovery point that you obtained in the previous commands:

```
az backup restore files unmount-rp \
--resource-group myResourceGroup \
--vault-name myRecoveryServicesVault \
--container-name myVM \
--item-name myVM \
--rp-name myRecoveryPointName
```

## Next steps

In this tutorial, you connected a recovery point to a VM and restored files for a web server. You learned how to:

- List and select recovery points
- Connect a recovery point to a VM
- Restore files from a recovery point

Advance to the next tutorial to learn about how to back up Windows Server to Azure.

[Back up Windows Server to Azure](#)

# About Site Recovery

9/9/2019 • 3 minutes to read • [Edit Online](#)

Welcome to the Azure Site Recovery service! This article provides a quick service overview.

As an organization you need to adopt a business continuity and disaster recovery (BCDR) strategy that keeps your data safe, and your apps and workloads up and running, when planned and unplanned outages occur.

Azure Recovery Services contribute to your BCDR strategy:

- **Site Recovery service:** Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to secondary location, and access apps from there. After the primary location is running again, you can fail back to it.
- **Backup service:** The [Azure Backup](#) service keeps your data safe and recoverable by backing it up to Azure.

Site Recovery can manage replication for:

- Azure VMs replicating between Azure regions.
- On-premises VMs, Azure Stack VMs and physical servers.

## What does Site Recovery provide?

| FEATURE                           | DETAILS                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Simple BCDR solution</b>       | Using Site Recovery, you can set up and manage replication, failover, and fallback from a single location in the Azure portal.                                                                                                                                                                                                       |
| <b>Azure VM replication</b>       | You can set up disaster recovery of Azure VMs from a primary region to a secondary region.                                                                                                                                                                                                                                           |
| <b>On-premises VM replication</b> | You can replicate on-premises VMs and physical servers to Azure, or to a secondary on-premises datacenter. Replication to Azure eliminates the cost and complexity of maintaining a secondary datacenter.                                                                                                                            |
| <b>Workload replication</b>       | Replicate any workload running on supported Azure VMs, on-premises Hyper-V and VMware VMs, and Windows/Linux physical servers.                                                                                                                                                                                                       |
| <b>Data resilience</b>            | Site Recovery orchestrates replication without intercepting application data. When you replicate to Azure, data is stored in Azure storage, with the resilience that provides. When failover occurs, Azure VMs are created, based on the replicated data.                                                                            |
| <b>RTO and RPO targets</b>        | Keep recovery time objectives (RTO) and recovery point objectives (RPO) within organizational limits. Site Recovery provides continuous replication for Azure VMs and VMware VMs, and replication frequency as low as 30 seconds for Hyper-V. You can reduce RTO further by integrating with <a href="#">Azure Traffic Manager</a> . |

| FEATURE                                   | DETAILS                                                                                                                                                                                                                                                                                        |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Keep apps consistent over failover</b> | You can replicate using recovery points with application-consistent snapshots. These snapshots capture disk data, all data in memory, and all transactions in process.                                                                                                                         |
| <b>Testing without disruption</b>         | You can easily run disaster recovery drills, without affecting ongoing replication.                                                                                                                                                                                                            |
| <b>Flexible failovers</b>                 | You can run planned failovers for expected outages with zero-data loss, or unplanned failovers with minimal data loss (depending on replication frequency) for unexpected disasters. You can easily fail back to your primary site when it's available again.                                  |
| <b>Customized recovery plans</b>          | Using recovery plans, can customize and sequence the failover and recovery of multi-tier applications running on multiple VMs. You group machines together in a recovery plan, and optionally add scripts and manual actions. Recovery plans can be integrated with Azure automation runbooks. |
| <b>BCDR integration</b>                   | Site Recovery integrates with other BCDR technologies. For example, you can use Site Recovery to protect the SQL Server backend of corporate workloads, with native support for SQL Server AlwaysOn, to manage the failover of availability groups.                                            |
| <b>Azure automation integration</b>       | A rich Azure Automation library provides production-ready, application-specific scripts that can be downloaded and integrated with Site Recovery.                                                                                                                                              |
| <b>Network integration</b>                | Site Recovery integrates with Azure for simple application network management, including reserving IP addresses, configuring load-balancers, and integrating Azure Traffic Manager for efficient network switchovers.                                                                          |

## What can I replicate?

| SUPPORTED                    | DETAILS                                                                                                                                                                                   |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Replication scenarios</b> | <p>Replicate Azure VMs from one Azure region to another.</p> <p>Replicate on-premises VMware VMs, Hyper-V VMs, physical servers (Windows and Linux), Azure Stack VMs to Azure.</p>        |
|                              | <p>Replicate AWS Windows instances to Azure.</p> <p>Replicate on-premises VMware VMs, Hyper-V VMs managed by System Center VMM, and physical servers to a secondary site.</p>             |
| <b>Regions</b>               | Review <a href="#">supported regions</a> for Site Recovery.                                                                                                                               |
| <b>Replicated machines</b>   | Review the replication requirements for <a href="#">Azure VM</a> replication, <a href="#">on-premises VMware VMs and physical servers</a> , and <a href="#">on-premises Hyper-V VMs</a> . |

| SUPPORTED        | DETAILS                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Workloads</b> | You can replicate any workload running on a machine that's supported for replication. In addition, the Site Recovery team have performed app-specific testing for a <a href="#">number of apps</a> . |

## Next steps

- Read more about [workload support](#).
- Get started with [Azure VM replication between regions](#).

# Set up disaster recovery for Azure VMs

1/24/2020 • 9 minutes to read • [Edit Online](#)

The [Azure Site Recovery](#) service contributes to your disaster recovery strategy by managing and orchestrating replication, failover, and failback of on-premises machines and Azure virtual machines (VMs).

This tutorial shows you how to set up disaster recovery for Azure VMs by replicating them from one Azure region to another. In this tutorial, you learn how to:

- Create a Recovery Services vault
- Verify target resource settings
- Set up outbound network connectivity for VMs
- Enable replication for a VM

## NOTE

This article provides instructions for deploying disaster recovery with the simplest settings. If you want to learn about customized settings, review the articles in the [How To section](#).

## Prerequisites

To complete this tutorial:

- Review the [scenario architecture and components](#).
- Review the [support requirements](#) before you start.

## Create a Recovery Services vault

Create the vault in any region, except the source region.

1. Sign in to the [Azure portal](#).
2. On the Azure portal menu or from the **Home** page, select **Create a resource**. Then, select **IT & Management Tools > Backup and Site Recovery**.
3. In **Name**, specify a friendly name to identify the vault. If you have more than one subscription, select the appropriate one.
4. Create a resource group or select an existing one. Specify an Azure region. To check supported regions, see geographic availability in [Azure Site Recovery Pricing Details](#).
5. To access the vault from the dashboard, select **Pin to dashboard** and then select **Create**.

The screenshot shows two windows side-by-side. On the left, the 'Recovery Services vaults' blade under 'Microsoft' shows a list of existing vaults: contosoassessment-Migr..., ContosoCorporation-Rec..., ContosoDemo, ContosoEmpty, ContosoScale, ContosoVMVault, ContosoVMVault, and Demo. A 'Filter by name...' search bar is at the top. On the right, a 'Recovery Services vault' configuration dialog is open, prompting the user to 'Click here to try new preview create vault experience with Tags support.' It contains fields for 'Name' (Vault1), 'Subscription' (<subscription-name>), 'Resource group' (RG1), and 'Location' (West Central US).

The new vault is added to the **Dashboard** under **All resources**, and on the main **Recovery Services vaults** page.

## Verify target resource settings

Check your Azure subscription for the target region.

- Verify that your Azure subscription allows you to create VMs in the target region. Contact support to enable the required quota.
- Make sure your subscription has enough resources to support VM sizes that match your source VMs. Site Recovery picks the same size, or the closest possible size, for the target VM.

## Set up outbound network connectivity for VMs

For Site Recovery to work as expected, you need to modify outbound network connectivity from the VMs that you want to replicate.

### NOTE

Site Recovery doesn't support using an authentication proxy to control network connectivity.

### Outbound connectivity for URLs

If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these URLs:

| URL                       | DETAILS                                                                                  |
|---------------------------|------------------------------------------------------------------------------------------|
| *.blob.core.windows.net   | Allows data to be written from the VM to the cache storage account in the source region. |
| login.microsoftonline.com | Provides authorization and authentication to Site Recovery service URLs.                 |

| URL                                      | DETAILS                                                               |
|------------------------------------------|-----------------------------------------------------------------------|
| *.hypervrecoverymanager.windowsazure.com | Allows the VM to communicate with the Site Recovery service.          |
| *.servicebus.windows.net                 | Allows the VM to write Site Recovery monitoring and diagnostics data. |

## Outbound connectivity for IP address ranges

If you're using a network security group (NSG), create service-tag based NSG rules for access to Azure Storage, Azure Active Directory, Site Recovery service, and Site Recovery monitoring. [Learn more](#).

## Verify Azure VM certificates

Check that the VMs you want to replicate have the latest root certificates. If they don't, the VM can't be registered to Site Recovery because of security constraints.

- For Windows VMs, install all the latest Windows updates on the VM, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows Update and certificate update processes for your organization.
- For Linux VMs, follow the guidance provided by your Linux distributor, to get the latest trusted root certificates and certificate revocation list on the VM.

## Set permissions on the account

Azure Site Recovery provides three built-in roles to control Site Recovery management operations.

- Site Recovery Contributor** - This role has all permissions required to manage Azure Site Recovery operations in a Recovery Services vault. A user with this role, however, can't create or delete a Recovery Services vault or assign access rights to other users. This role is best suited for disaster recovery administrators who can enable and manage disaster recovery for applications or entire organizations.
- Site Recovery Operator** - This role has permissions to execute and manage Failover and Failback operations. A user with this role can't enable or disable replication, create or delete vaults, register new infrastructure, or assign access rights to other users. This role is best suited for a disaster recovery operator who can fail over virtual machines or applications when instructed by application owners and IT administrators. Post resolution of the disaster, the disaster recovery operator can reprotect and failback the virtual machines.
- Site Recovery Reader** - This role has permissions to view all Site Recovery management operations. This role is best suited for an IT monitoring executive who can monitor the current state of protection and raise support tickets.

Learn more about [Azure RBAC built-in roles](#).

## Enable replication for a VM

The following sections describe how to enable replication.

### Select the source

To begin the replication set up, choose the source where your Azure VMs are running.

- Go to **Recovery Services vaults**, select the vault name, then select **+Replicate**.
- For the **Source**, select **Azure**.
- In **Source location**, select the source Azure region where your VMs are currently running.

- Select the **Source subscription** where the virtual machines are running. This can be any subscription within the same Azure Active Directory tenant where your recovery services vault exists.
- Select the **Source resource group**, and select **OK** to save the settings.

The image shows two overlapping windows. On the left is the 'Enable replication' wizard with three steps: 1. Source Configure, 2. Virtual machines Select, and 3. Replication settings Configure replication settings. Step 1 is highlighted. On the right is the 'Source' configuration dialog, which has a dropdown for 'Source' set to 'Azure'. It also contains fields for 'Source location' (West US), 'Azure virtual machine deployment model' (Resource Manager), 'Source subscription' (<subscription-ID>), and 'Source resource group' (123). Both windows have an 'X' button in the top right corner.

### Select the VMs

Site Recovery retrieves a list of the VMs associated with the subscription and resource group/cloud service.

- In **Virtual Machines**, select the VMs you want to replicate.
- Select **OK**.

### Configure replication settings

Site Recovery creates default settings and replication policy for the target region. You can change these settings as required.

- Select **Settings** to view the target and replication settings.
- To override the default target settings, select **Customize** next to **Resource group, Network, Storage and Availability**.

The image shows the 'Configure settings' dialog. It includes a 'Target location' dropdown set to 'East US 2', a 'Target subscription' section with a 'Customize' link, and a note about customizing resource groups, network, storage, and availability sets. Below this is a table for customizing target settings.

| SETTING | DETAILS |
|---------|---------|
|         |         |

| SETTING                                                           | DETAILS                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Target subscription</b>                                        | By default, the target subscription is the same as the source subscription. Select <b>Customize</b> to select a different target subscription within the same Azure Active Directory tenant.                                                                                                                                                                                                              |
| <b>Target location</b>                                            | <p>The target region used for disaster recovery.</p> <p>We recommend that the target location matches the location of the Site Recovery vault.</p>                                                                                                                                                                                                                                                        |
| <b>Target resource group</b>                                      | <p>The resource group in the target region that holds Azure VMs after failover.</p> <p>By default, Site Recovery creates a new resource group in the target region with an <code>asr</code> suffix. The location of the target resource group can be any region except the region in which your source virtual machines are hosted.</p>                                                                   |
| <b>Target virtual network</b>                                     | <p>The network in the target region that VMs are located after failover.</p> <p>By default, Site Recovery creates a new virtual network (and subnets) in the target region with an <code>asr</code> suffix.</p>                                                                                                                                                                                           |
| <b>Cache storage accounts</b>                                     | <p>Site Recovery uses a storage account in the source region. Changes to source VMs are sent to this account before replication to the target location.</p> <p>If you're using a firewall-enabled cache storage account, make sure that you enable <b>Allow trusted Microsoft services</b>. <a href="#">Learn more</a>. Also, ensure that you allow access to at least one subnet of the source Vnet.</p> |
| <b>Target storage accounts (source VM uses non-managed disks)</b> | <p>By default, Site Recovery creates a new storage account in the target region to mirror the source VM storage account.</p> <p>Enable <b>Allow trusted Microsoft services</b> if you're using a firewall-enabled cache storage account.</p>                                                                                                                                                              |
| <b>Replica managed disks (If source VM uses managed disks)</b>    | By default, Site Recovery creates replica managed disks in the target region to mirror the source VM's managed disks with the same storage type (standard or premium) as the source VM's managed disk. You can only customize Disk type.                                                                                                                                                                  |
| <b>Target availability sets</b>                                   | By default, Azure Site Recovery creates a new availability set in the target region with name having <code>asr</code> suffix for the VMs part of an availability set in source region. In case availability set created by Azure Site Recovery already exists, it's reused.                                                                                                                               |

| SETTING                          | DETAILS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Target availability zones</b> | <p>By default, Site Recovery assigns the same zone number as the source region in target region if the target region supports availability zones.</p> <p>If the target region doesn't support availability zones, the target VMs are configured as single instances by default.</p> <p>Select <b>Customize</b> to configure VMs as part of an availability set in the target region.</p> <p>You can't change the availability type (single instance, availability set, or availability zone) after you enable replication. To change the availability type, disable and enable replication.</p> |

- To customize replication policy settings, select **Customize** next to **Replication policy**, and modify the settings as needed.

| SETTING                                  | DETAILS                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Replication policy name</b>           | Policy name.                                                                                                                                                                                                                                                                                                                                                |
| <b>Recovery point retention</b>          | By default, Site Recovery keeps recovery points for 24 hours. You can configure a value between 1 and 72 hours.                                                                                                                                                                                                                                             |
| <b>App-consistent snapshot frequency</b> | <p>By default, Site Recovery takes an app-consistent snapshot every 4 hours. You can configure any value between 1 and 12 hours.</p> <p>An app-consistent snapshot is a point-in-time snapshot of the application data inside the VM. Volume Shadow Copy Service (VSS) ensures that app on the VM are in a consistent state when the snapshot is taken.</p> |
| <b>Replication group</b>                 | If your application needs multi-VM consistency across VMs, you can create a replication group for those VMs. By default, the selected VMs are not part of any replication group.                                                                                                                                                                            |

- In **Customize**, select **Yes** for multi-VM consistency if you want to add VMs to a new or existing replication group. Then select **OK**.

#### NOTE

- All the machines in a replication group have shared crash consistent and app-consistent recovery points when failed over.
- Enabling multi-VM consistency can impact workload performance (it's CPU intensive). It should be used only if machines are running the same workload, and you need consistency across multiple machines.
- You can have a maximum of 16 VMs in a replication group.
- If you enable multi-VM consistency, machines in the replication group communicate with each other over port 20004. Make sure there's no firewall blocking the internal communication between the VMs over this port.
- For Linux VMs in a replication group, ensure the outbound traffic on port 20004 is manually opened in accordance with guidance for the Linux version.

## Configure encryption settings

If the source VM has Azure disk encryption (ADE) enabled, review the settings.

1. Verify the settings:
  - a. **Disk encryption key vaults:** By default, Site Recovery creates a new key vault on the source VM disk encryption keys, with an `asr` suffix. If the key vault already exists, it's reused.
  - b. **Key encryption key vaults:** By default, Site Recovery creates a new key vault in the target region. The name has an `asr` suffix, and is based on the source VM key encryption keys. If the key vault created by Site Recovery already exists, it's reused.
2. Select **Customize** to select custom key vaults.

#### NOTE

Only Azure VMs running Windows operating systems and [enabled for encryption with Azure AD app](#) are currently supported by Azure Site Recovery.

## Track replication status

After replication is enabled, you can track the job's status.

1. In **Settings**, select **Refresh** to get the latest status.
2. Track progress and status as follows:
  - a. Track progress of the **Enable protection** job in **Settings > Jobs > Site Recovery Jobs**.
  - b. In **Settings > Replicated Items**, you can view the status of VMs and the initial replication progress.  
Select the VM to drill down into its settings.

## Next steps

In this tutorial, you configured disaster recovery for an Azure VM. Now you can run a disaster recovery drill to check that failover works as expected.

[Run a disaster recovery drill](#)

# Run a disaster recovery drill to a secondary region for Azure VMs

1/17/2020 • 2 minutes to read • [Edit Online](#)

The [Azure Site Recovery](#) service contributes to your business continuity and disaster recovery (BCDR) strategy by keeping your business apps up and running available during planned and unplanned outages. Site Recovery manages and orchestrates disaster recovery of on-premises machines and Azure virtual machines (VMs), including replication, failover, and recovery.

This tutorial shows you how to run a disaster recovery drill for an Azure VM, from one Azure region to another, with a test failover. A drill validates your replication strategy without data loss or downtime, and doesn't affect your production environment. In this tutorial, you learn how to:

- Check the prerequisites
- Run a test failover for a single VM

## NOTE

This tutorial helps you to perform a disaster recovery drill with minimal steps. To learn more about the various functions related to doing a disaster recovery drill, see the documentation for Azure VMs [replication](#), [networking](#), [automation](#), or [troubleshooting](#).

## Prerequisites

Check the following items before you do this tutorial:

- Before you run a test failover, we recommend that you check the VM's properties to make sure it's configured for disaster recovery. Go to the VM's **Operations > Disaster Recovery > Properties** to view the replication and failover properties.
- **We recommend you use a separate Azure VM network for the test failover**, and not the default network that was set up when you enabled replication.
- Depending on your source networking configurations for each NIC, you can specify **Subnet**, **Private IP address**, **Public IP**, **Network security group**, or **Load balancer** to attach to each NIC under test failover settings in **Compute and Network** before doing a disaster recovery drill.

## Run a test failover

This example shows how to use a Recovery Services vault to do a VM test failover.

1. Select a vault and go to **Protected items > Replicated items** and select a VM.
2. In **Test Failover**, select a recovery point to use for the failover:
  - **Latest**: Processes all the data in Site Recovery and provides the lowest RTO (Recovery Time Objective).
  - **Latest processed**: Fails the VM over to the latest recovery point that was processed by Site Recovery. The time stamp is shown. With this option, no time is spent processing data, so it provides a low RTO.
  - **Latest app-consistent**: This option fails over all VMs to the latest app-consistent recovery point. The time stamp is shown.
  - **Custom**: Fail over to particular recovery point. Custom is only available when you fail over a single VM, and not for failover with a recovery plan.

3. Select the target Azure virtual network that Azure VMs in the secondary region will connect to after the failover.

**NOTE**

If the test failover settings are pre-configured for the replicated item, the dropdown menu to select an Azure virtual network isn't visible.

4. To start the failover, select **OK**. To track the progress from the vault, go to **Monitoring > Site Recovery jobs** and select the **Test Failover** job.
5. After the failover finishes, the replica Azure VM appears in the Azure portal's **Virtual Machines**. Make sure that the VM is running, sized appropriately, and connected to the appropriate network.
6. To delete the VMs that were created during the test failover, select **Cleanup test failover** on the replicated item or the recovery plan. In **Notes**, record and save any observations associated with the test failover.

## Next steps

[Run a production failover](#)

# Fail over and reprotect Azure VMs between regions

8/5/2019 • 2 minutes to read • [Edit Online](#)

This tutorial describes how to fail over an Azure virtual machine (VM) to a secondary Azure region with the [Azure Site Recovery](#) service. After you've failed over, you reprotect the VM. In this tutorial, you learn how to:

- Fail over the Azure VM
- Reprotect the secondary Azure VM, so that it replicates to the primary region.

## NOTE

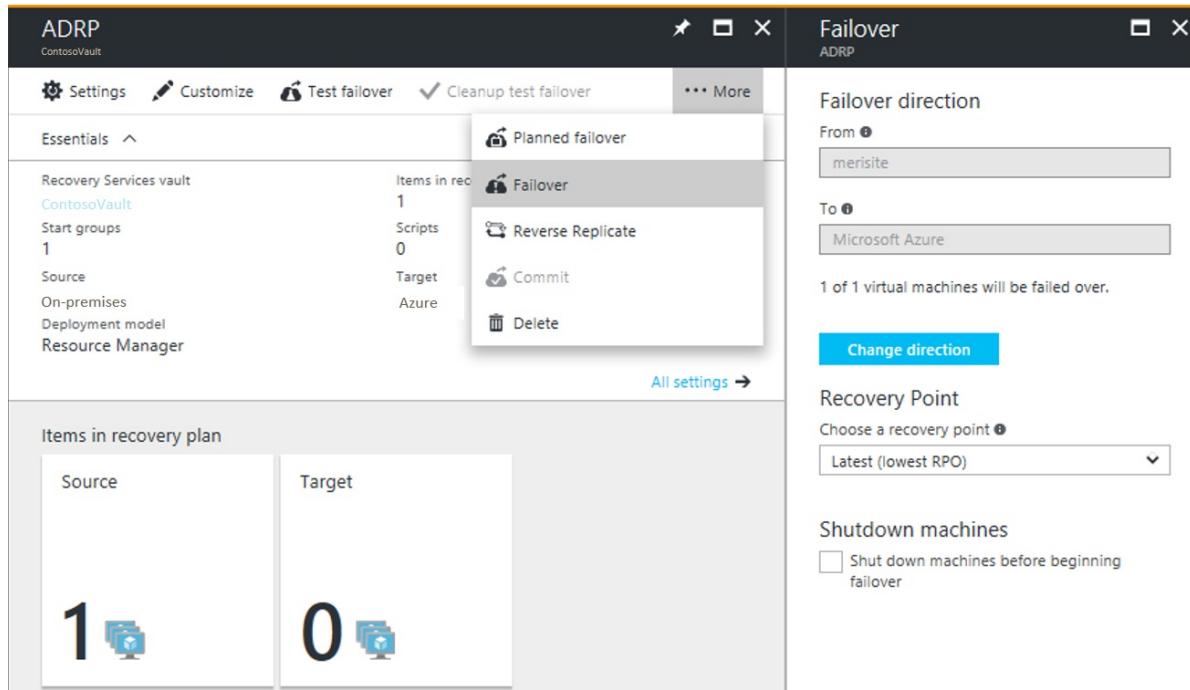
This tutorial contains the simplest path with default settings and minimum customization. For more complex scenarios, use the articles under 'How To' for Azure VMs.

## Prerequisites

- Before you start, review [frequently asked questions](#) about failover.
- Make sure that you've completed a [disaster recovery drill](#) to check everything is working as expected.
- Verify the VM properties before you run the test failover. The VM must comply with [Azure requirements](#).

## Run a failover to the secondary region

### 1. In **Replicated items**, select the VM that you want to fail over > **Failover**



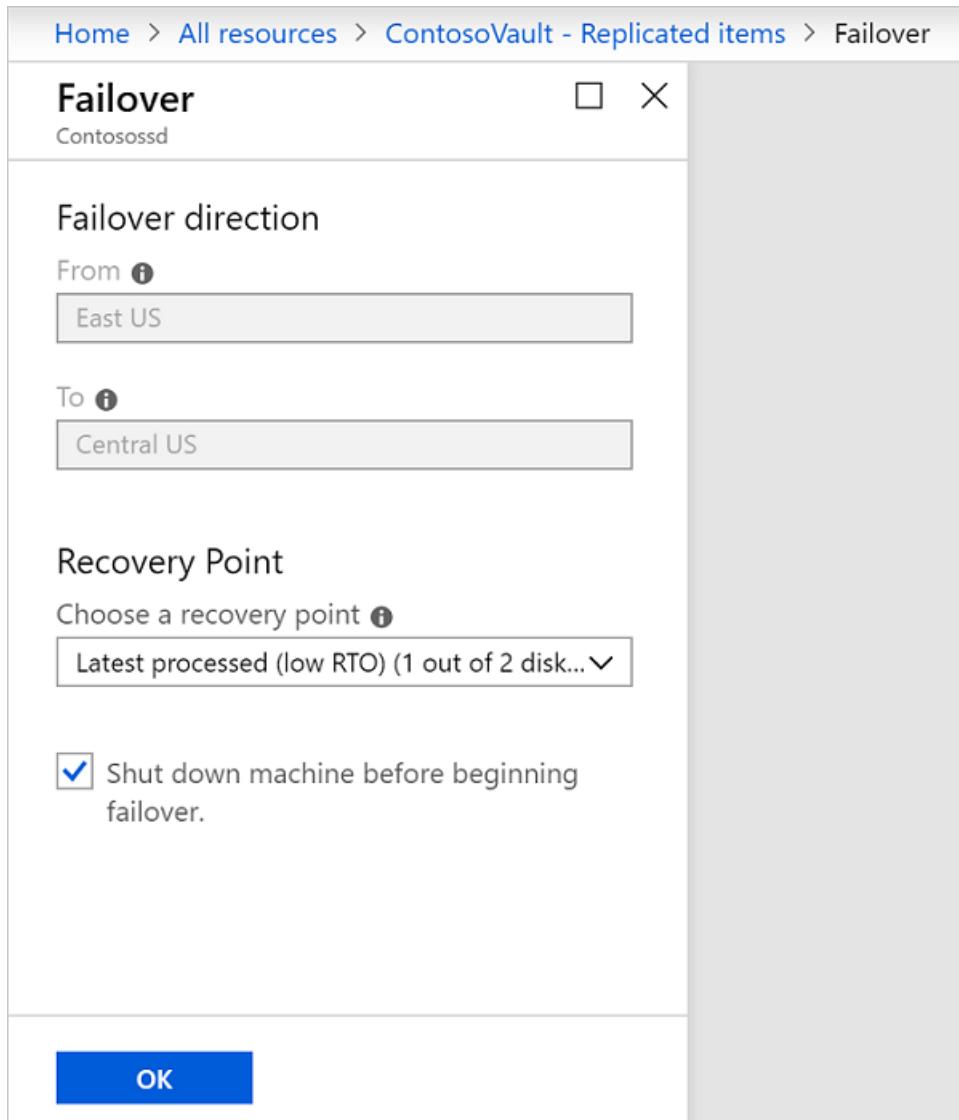
### 2. In **Failover**, select a **Recovery Point** to fail over to. You can use one of the following options:

- **Latest** (default): Processes all the data in the Site Recovery service and provides the lowest Recovery Point Objective (RPO).
- **Latest processed**: Reverts the virtual machine to the latest recovery point that has been processed by Site Recovery service.
- **Custom**: Fails over to a particular recovery point. This option is useful for performing a test failover.

3. Select **Shut down machine before beginning failover** if you want Site Recovery to attempt to do a shutdown of source VMs before triggering the failover. Shutdown helps to ensure no data loss. Failover continues even if shutdown fails. Site Recovery does not clean up the source after failover.
4. Follow the failover progress on the **Jobs** page.
5. After the failover, validate the virtual machine by logging in to it. If you want to go another recovery point for the virtual machine, then you can use **Change recovery point** option.
6. Once you are satisfied with the failed over virtual machine, you can **Commit** the failover. Committing deletes all the recovery points available with the service. You won't now be able to change the recovery point.

**NOTE**

When you fail over a VM to which you add a disk after you enabled replication for the VM, replication points will show the disks that are available for recovery. For example, if a VM has a single disk and you add a new one, replication points that were created before you added the disk will show that the replication point consists of "1 of 2 disks".

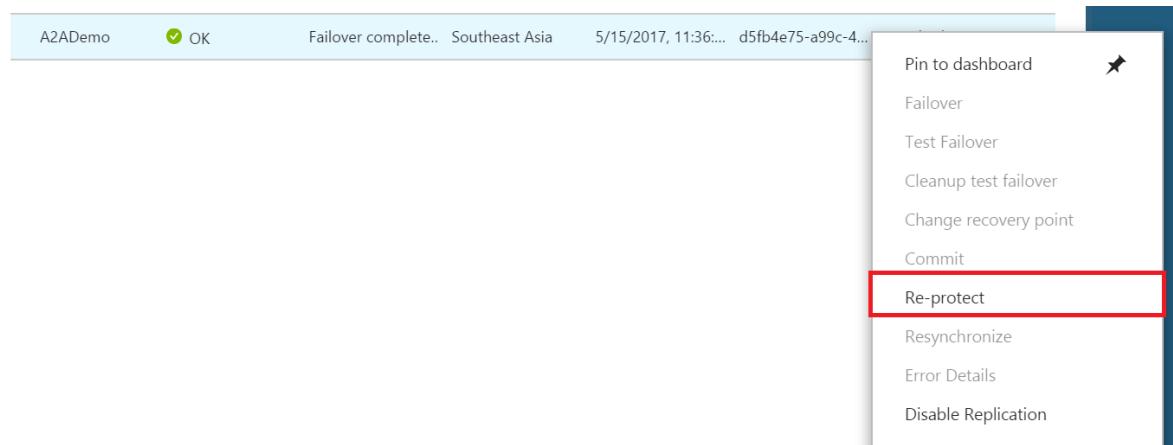


## Reprotect the secondary VM

After failover of the VM, you need to reprotect it so that it replicates back to the primary region.

1. Make sure that the VM is in the **Failover committed** state, and check that the primary region is available, and you're able to create and access new resources in it.

2. In **Vault > Replicated items**, right-click the VM that's been failed over, and then select **Re-Protect**.



3. Verify that the direction of protection, secondary to primary region, is already selected.
4. Review the **Resource group, Network, Storage, and Availability sets** information. Any resources marked as new are created as part of the reprotect operation.
5. Click **OK** to trigger a reprotect job. This job seeds the target site with the latest data. Then, it replicates the deltas to the primary region. The VM is now in a protected state.

## Next steps

- After reprotecting, [learn how to](#) fail back to the primary region when it's available.
- [Learn more](#) about the reprotection flow.

# Understanding Azure virtual machine usage

1/14/2020 • 7 minutes to read • [Edit Online](#)

By analyzing your Azure usage data, powerful consumption insights can be gained – insights that can enable better cost management and allocation throughout your organization. This document provides a deep dive into your Azure Compute consumption details. For more details on general Azure usage, navigate to [Understanding your bill](#).

## Download your usage details

To begin, [download your usage details](#). The table below provides the definition and example values of usage for Virtual Machines deployed via the Azure Resource Manager. This document does not contain detailed information for VMs deployed via our classic model.

| FIELDS             | MEANING                                                                                                                                                                                                                                                                                                                                                                                                | EXAMPLE VALUES                         |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Usage Date         | The date when the resource was used.                                                                                                                                                                                                                                                                                                                                                                   | "11/23/2017"                           |
| Meter ID           | Identifies the top-level service for which this usage belongs to.                                                                                                                                                                                                                                                                                                                                      | "Virtual Machines"                     |
| Meter Sub-Category | The billed meter identifier. <ul style="list-style-type: none"><li>For Compute Hour usage, there is a meter for each VM Size + OS (Windows, Non-Windows) + Region.</li><li>For Premium software usage, there is a meter for each software type. Most premium software images have different meters for each core size. For more information, visit the <a href="#">Compute Pricing Page</a>.</li></ul> | "2005544f-659d-49c9-9094-8e0aea1be3a5" |
| Meter Name         | This is specific for each service in Azure. For compute, it is always "Compute Hours".                                                                                                                                                                                                                                                                                                                 | "Compute Hours"                        |
| Meter Region       | Identifies the location of the datacenter for certain services that are priced based on datacenter location.                                                                                                                                                                                                                                                                                           | "JA East"                              |
| Unit               | Identifies the unit that the service is charged in. Compute resources are billed per hour.                                                                                                                                                                                                                                                                                                             | "Hours"                                |
| Consumed           | The amount of the resource that has been consumed for that day. For Compute, we bill for each minute the VM ran for a given hour (up to 6 decimals of accuracy).                                                                                                                                                                                                                                       | "1", "0.5"                             |

| FIELDS            | MEANING                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | EXAMPLE VALUES                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource Location | Identifies the datacenter where the resource is running.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | "JA East"                                                                                                                                                                                                                                                                                                                                                                                                   |
| Consumed Service  | The Azure platform service that you used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | "Microsoft.Compute"                                                                                                                                                                                                                                                                                                                                                                                         |
| Resource Group    | The resource group in which the deployed resource is running in. For more information, see <a href="#">Azure Resource Manager overview</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "MyRG"                                                                                                                                                                                                                                                                                                                                                                                                      |
| Instance ID       | The identifier for the resource. The identifier contains the name you specify for the resource when it was created. For VMs, the Instance ID will contain the SubscriptionId, ResourceGroupName, and VMName (or scale set name for scale set usage).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <pre>"/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/ resourceGroups/MyRG/providers/Microsoft.Compute/virtualMachines/MyVM1"</pre> <p>or</p> <pre>"/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/ resourceGroups/MyRG/providers/Microsoft.Compute/virtualMachineScaleSets/ MyVMSS1"</pre>                                                                                                            |
| Tags              | Tag you assign to the resource. Use tags to group billing records. Learn how to <a href="#">tag your Virtual Machines</a> . This is available for Resource Manager VMs only.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <pre>"</pre> <pre>{"myDepartment":"RD","myUser":"myName"}</pre>                                                                                                                                                                                                                                                                                                                                             |
| Additional Info   | <p>Service-specific metadata. For VMs, we populate the following in the additional info field:</p> <ul style="list-style-type: none"> <li>• Image Type- specific image that you ran. Find the full list of supported strings below under Image Types.</li> <li>• Service Type: the size that you deployed.</li> <li>• VMName: name of your VM. This is only populated for scale set VMs. If you need your VM Name for scale set VMs, you can find that in the Instance ID string above.</li> <li>• UsageType: This specifies the type of usage this represents. <ul style="list-style-type: none"> <li>◦ ComputeHR is the Compute Hour usage for the underlying VM, like Standard_D1_v2.</li> <li>◦ ComputeHR_SW is the premium software charge if the VM is using premium software, like Microsoft R Server.</li> </ul> </li> </ul> | <p>Virtual Machines</p> <pre>{"ImageType":"Canonical","ServiceType":"Standard_DS1_v2","VMName":"","UsageType":"ComputeHR"}</pre> <p>Virtual Machine Scale Sets</p> <pre>{"ImageType":"Canonical","ServiceType":"Standard_DS1_v2","VMName":"myVM1","UsageType":"ComputeHR"}</pre> <p>Premium Software</p> <pre>{"ImageType":"","ServiceType":"Standard_DS1_v2","VMName":"","UsageType":"ComputeHR_SW"}</pre> |

## Image Type

For some images in the Azure gallery, the image type is populated in the Additional Info field. This enables users to understand and track what they have deployed on their Virtual Machine. The values that are populated in this field based on the image you have deployed are the following:

- BitRock
- Canonical
- FreeBSD
- Open Logic
- Oracle
- SLES for SAP
- SQL Server 14 Preview on Windows Server 2012 R2 Preview
- SUSE
- SUSE Premium
- StorSimple Cloud Appliance
- Red Hat
- Red Hat for SAP Business Applications
- Red Hat for SAP HANA
- Windows Client BYOL
- Windows Server BYOL
- Windows Server Preview

## Service Type

The service type field in the Additional Info field corresponds to the exact VM size you deployed. Premium storage VMs (SSD-based) and non-premium storage VMs (HDD-based) are priced the same. If you deploy an SSD-based size, like Standard\_DS2\_v2, you see the non-SSD size ('Standard\_D2\_v2 VM') in the Meter Sub-Category column and the SSD-size ('Standard\_DS2\_v2') in the Additional Info field.

## Region Names

The region name populated in the Resource Location field in the usage details varies from the region name used in the Azure Resource Manager. Here is a mapping between the region values:

| RESOURCE MANAGER REGION NAME | RESOURCE LOCATION IN USAGE DETAILS |
|------------------------------|------------------------------------|
| australiaeast                | AU East                            |
| australiasoutheast           | AU Southeast                       |
| brazilsouth                  | BR South                           |
| CanadaCentral                | CA Central                         |
| CanadaEast                   | CA East                            |
| CentralIndia                 | IN Central                         |
| centralus                    | Central US                         |

| RESOURCE MANAGER REGION NAME | RESOURCE LOCATION IN USAGE DETAILS |
|------------------------------|------------------------------------|
| chinaeast                    | China East                         |
| chinanorth                   | China North                        |
| eastasia                     | East Asia                          |
| eastus                       | East US                            |
| eastus2                      | East US 2                          |
| GermanyCentral               | DE Central                         |
| GermanyNortheast             | DE Northeast                       |
| japaneast                    | JA East                            |
| japanwest                    | JA West                            |
| KoreaCentral                 | KR Central                         |
| KoreaSouth                   | KR South                           |
| northcentralus               | North Central US                   |
| northeurope                  | North Europe                       |
| southcentralus               | South Central US                   |
| southeastasia                | Southeast Asia                     |
| SouthIndia                   | IN South                           |
| UKNorth                      | US North                           |
| uksouth                      | UK South                           |
| UKSouth2                     | UK South 2                         |
| ukwest                       | UK West                            |
| USDoDCentral                 | US DoD Central                     |
| USDoDEast                    | US DoD East                        |
| USGovArizona                 | USGov Arizona                      |
| usgoviowa                    | USGov Iowa                         |
| USGovTexas                   | USGov Texas                        |

| RESOURCE MANAGER REGION NAME | RESOURCE LOCATION IN USAGE DETAILS |
|------------------------------|------------------------------------|
| usgovvirginia                | USGov Virginia                     |
| westcentralus                | US West Central                    |
| westeurope                   | West Europe                        |
| WestIndia                    | IN West                            |
| westus                       | West US                            |
| westus2                      | US West 2                          |

## Virtual machine usage FAQ

### What resources are charged when deploying a VM?

VMs acquire costs for the VM itself, any premium software running on the VM, the storage account\managed disk associated with the VM, and the networking bandwidth transfers from the VM.

### How can I tell if a VM is using Azure Hybrid Benefit in the Usage CSV?

If you deploy using the [Azure Hybrid Benefit](#), you are charged the Non-Windows VM rate since you are bringing your own license to the cloud. In your bill, you can distinguish which Resource Manager VMs are running Azure Hybrid Benefit because they have either "Windows\_Server BYOL" or "Windows\_Client BYOL" in the ImageType column.

### How are Basic vs. Standard VM Types differentiated in the Usage CSV?

Both Basic and Standard A-Series VMs are offered. If you deploy a Basic VM, in the Meter Sub Category, it has the string "Basic." If you deploy a Standard A-Series VM, then the VM size appears as "A1 VM" since Standard is the default. To learn more about the differences between Basic and Standard, see the [Pricing Page](#).

### What are ExtraSmall, Small, Medium, Large, and ExtraLarge sizes?

ExtraSmall - ExtraLarge are the legacy names for Standard\_A0 – Standard\_A4. In classic VM usage records, you might see this convention used if you have deployed these sizes.

### What is the difference between Meter Region and Resource Location?

The Meter Region is associated with the meter. For some Azure services who use one price for all regions, the Meter Region field could be blank. However, since VMs have dedicated prices per region for Virtual Machines, this field is populated. Similarly, the Resource Location for Virtual Machines is the location where the VM is deployed. The Azure region in both fields are the same, although they might have a different string convention for the region name.

### Why is the ImageType value blank in the Additional Info field?

The ImageType field is only populated for a subset of images. If you did not deploy one of the images above, the ImageType is blank.

### Why is the VMName blank in the Additional Info?

The VMName is only populated in the Additional Info field for VMs in a scale set. The InstanceID field contains the VM name for non-scale set VMs.

### What does ComputeHR mean in the UsageType field in the Additional Info?

ComputeHR stands for Compute Hour which represents the usage event for the underlying infrastructure cost. If the UsageType is ComputeHR\_SW, the usage event represents the premium software charge for the VM.

## **How do I know if I am charged for premium software?**

When exploring which VM Image best fits your needs, be sure to check out the [Azure Marketplace](#). The image has the software plan rate. If you see "Free" for the rate, there is no additional cost for the software.

## **What is the difference between Microsoft.ClassicCompute and Microsoft.Compute in the Consumed service?**

Microsoft.ClassicCompute represents classic resources deployed via the Azure Service Manager. If you deploy via the Resource Manager, then Microsoft.Compute is populated in the consumed service. Learn more about the [Azure Deployment models](#).

## **Why is the InstanceID field blank for my Virtual Machine usage?**

If you deploy via the classic deployment model, the InstanceID string is not available.

## **Why are the tags for my VMs not flowing to the usage details?**

Tags only flow to you the Usage CSV for Resource Manager VMs only. Classic resource tags are not available in the usage details.

## **How can the consumed quantity be more than 24 hours one day?**

In the Classic model, billing for resources is aggregated at the Cloud Service level. If you have more than one VM in a Cloud Service that uses the same billing meter, your usage is aggregated together. VMs deployed via Resource Manager are billed at the VM level, so this aggregation will not apply.

## **Why is pricing not available for DS/FS/GS/LS sizes on the pricing page?**

Premium storage capable VMs are billed at the same rate as non-premium storage capable VMs. Only your storage costs differ. Visit the [storage pricing page](#) for more information.

## **Next steps**

To learn more about your usage details, see [Understand your bill for Microsoft Azure](#).

# Common PowerShell commands for creating and managing Azure Virtual Machines

1/14/2020 • 2 minutes to read • [Edit Online](#)

This article covers some of the Azure PowerShell commands that you can use to create and manage virtual machines in your Azure subscription. For more detailed help with specific command-line switches and options, you can use the **Get-Help command**.

These variables might be useful for you if running more than one of the commands in this article:

- \$location - The location of the virtual machine. You can use [Get-AzLocation](#) to find a geographical region that works for you.
- \$myResourceGroup - The name of the resource group that contains the virtual machine.
- \$myVM - The name of the virtual machine.

## Create a VM - simplified

| TASK                            | COMMAND                                                                                                                                                                                                                                                                                                               |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a simple VM              | <code>New-AzVM -Name \$myVM</code><br><br>New-AzVM has a set of <i>simplified</i> parameters, where all that is required is a single name. The value for <code>-Name</code> will be used as the name for all of the resources required for creating a new VM. You can specify more, but this is all that is required. |
| Create a VM from a custom image | <code>New-AzVm -ResourceGroupName \$myResourceGroup -Name \$myVM ImageName "myImage" -Location \$location</code><br><br>You need to have already created your own <a href="#">managed image</a> . You can use an image to make multiple, identical VMs.                                                               |

## Create a VM configuration

| TASK                       | COMMAND                                                                                                                                                                                                                                                                                                                |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a VM configuration  | <code>\$vm = New-AzVMConfig -VMName \$myVM -VMSize "Standard_D1_v1"</code><br><br>The VM configuration is used to define or update settings for the VM. The configuration is initialized with the name of the VM and its <a href="#">size</a> .                                                                        |
| Add configuration settings | <code>\$vm = Set-AzVMOperatingSystem -VM \$vm -Windows -ComputerName \$myVM -Credential \$cred -ProvisionVMAgent -EnableAutoUpdate</code><br><br>Operating system settings including <a href="#">credentials</a> are added to the configuration object that you previously created using <code>New-AzVMConfig</code> . |

| TASK                     | COMMAND                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add a network interface  | <pre>\$vm = Add-AzVMNetworkInterface -VM \$vm -Id \$nic.Id</pre> <p>A VM must have a <a href="#">network interface</a> to communicate in a virtual network. You can also use <a href="#">Get-AzNetworkInterface</a> to retrieve an existing network interface object.</p>                                                                                                          |
| Specify a platform image | <pre>\$vm = Set-AzVMSourceImage -VM \$vm -PublisherName "publisher_name" -Offer "publisher_offer" -Skus "product_sku" -Version "latest"</pre> <p><a href="#">Image information</a> is added to the configuration object that you previously created using New-AzVMConfig. The object returned from this command is only used when you set the OS disk to use a platform image.</p> |
| Create a VM              | <pre>New-AzVM -ResourceGroupName \$myResourceGroup -Location \$location -VM \$vm</pre> <p>All resources are created in a <a href="#">resource group</a>. Before you run this command, run New-AzVMConfig, Set-AzVMOperatingSystem, Set-AzVMSourceImage, Add-AzVMNetworkInterface, and Set-AzVMOSDisk.</p>                                                                          |
| Update a VM              | <pre>Update-AzVM -ResourceGroupName \$myResourceGroup -VM \$vm</pre> <p>Get the current VM configuration using Get-AzVM, change configuration settings on the VM object, and then run this command.</p>                                                                                                                                                                            |

## Get information about VMs

| TASK                         | COMMAND                                                                                                                                                         |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| List VMs in a subscription   | <a href="#">Get-AzVM</a>                                                                                                                                        |
| List VMs in a resource group | <pre>Get-AzVM -ResourceGroupName \$myResourceGroup</pre> <p>To get a list of resource groups in your subscription, use <a href="#">Get-AzResourceGroup</a>.</p> |
| Get information about a VM   | <pre>Get-AzVM -ResourceGroupName \$myResourceGroup -Name \$myVM</pre>                                                                                           |

## Manage VMs

| TASK       | COMMAND                                                                      |
|------------|------------------------------------------------------------------------------|
| Start a VM | <a href="#">Start-AzVM</a> -ResourceGroupName \$myResourceGroup -Name \$myVM |
| Stop a VM  | <a href="#">Stop-AzVM</a> -ResourceGroupName \$myResourceGroup -Name \$myVM  |

| TASK                 | COMMAND                                                                     |
|----------------------|-----------------------------------------------------------------------------|
| Restart a running VM | <code>Restart-AzVM -ResourceGroupName \$myResourceGroup -Name \$myVM</code> |
| Delete a VM          | <code>Remove-AzVM -ResourceGroupName \$myResourceGroup -Name \$myVM</code>  |

## Next steps

- See the basic steps for creating a virtual machine in [Create a Windows VM using Resource Manager and PowerShell](#).

# Resize a Windows VM

1/14/2020 • 2 minutes to read • [Edit Online](#)

This article shows you how to move a VM to a different [VM size](#).

After you create a virtual machine (VM), you can scale the VM up or down by changing the VM size. In some cases, you must deallocate the VM first. This can happen if the new size is not available on the hardware cluster that is currently hosting the VM.

If your VM uses Premium Storage, make sure that you choose an **s** version of the size to get Premium Storage support. For example, choose Standard\_E4s\_v3 instead of Standard\_E4\_v3.

## Use the portal

1. Open the [Azure portal](#).
2. Open the page for the virtual machine.
3. In the left menu, select **Size**.
4. Pick a new size from the list of available sizes and then select **Resize**.

If the virtual machine is currently running, changing its size will cause it to be restarted. Stopping the virtual machine may reveal additional sizes.

## Use PowerShell to resize a VM not in an availability set

Set some variables. Replace the values with your own information.

```
$resourceGroup = "myResourceGroup"
$vmName = "myVM"
```

List the VM sizes that are available on the hardware cluster where the VM is hosted.

```
Get-AzVMSize -ResourceGroupName $resourceGroup -VMName $vmName
```

If the size you want is listed, run the following commands to resize the VM. If the desired size is not listed, go on to step 3.

```
$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName
$vm.HardwareProfile.VmSize = "<newVMsize>"
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
```

If the size you want is not listed, run the following commands to deallocate the VM, resize it, and restart the VM. Replace **<newVMsize>** with the size you want.

```
Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName
$vm.HardwareProfile.VmSize = "<newVMSize>"
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
```

## WARNING

Deallocating the VM releases any dynamic IP addresses assigned to the VM. The OS and data disks are not affected.

## Use PowerShell to resize a VM in an availability set

If the new size for a VM in an availability set is not available on the hardware cluster currently hosting the VM, then all VMs in the availability set will need to be deallocated to resize the VM. You also might need to update the size of other VMs in the availability set after one VM has been resized. To resize a VM in an availability set, perform the following steps.

```
$resourceGroup = "myResourceGroup"
$vmName = "myVM"
```

List the VM sizes that are available on the hardware cluster where the VM is hosted.

```
Get-AzVMSize -ResourceGroupName $resourceGroup -VMName $vmName
```

If the desired size is listed, run the following commands to resize the VM. If it is not listed, go to the next section.

```
$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName
$vm.HardwareProfile.VmSize = "<newVmSize>"
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
```

If the size you want is not listed, continue with the following steps to deallocate all VMs in the availability set, resize VMs, and restart them.

Stop all VMs in the availability set.

```
$as = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup
$vmIDs = $as.VirtualMachinesReferences
foreach ($vmID in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
}
```

Resize and restart the VMs in the availability set.

```
$newSize = "<newVmSize>"
$as = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup
$vmIDs = $as.VirtualMachinesReferences
foreach ($vmID in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 $vm = Get-AzVM -ResourceGroupName $resourceGroup -Name $vmName
 $vm.HardwareProfile.VmSize = $newSize
 Update-AzVM -ResourceGroupName $resourceGroup -VM $vm
 Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
}
```

## Next steps

For additional scalability, run multiple VM instances and scale out. For more information, see [Automatically scale](#)

Windows machines in a Virtual Machine Scale Set.

# Change the OS disk used by an Azure VM using PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

If you have an existing VM, but you want to swap the disk for a backup disk or another OS disk, you can use Azure PowerShell to swap the OS disks. You don't have to delete and recreate the VM. You can even use a managed disk in another resource group, as long as it isn't already in use.

The VM does need to be stopped\deallocated, then the resource ID of the managed disk can be replaced with the resource ID of a different managed disk.

Make sure that the VM size and storage type are compatible with the disk you want to attach. For example, if the disk you want to use is in Premium Storage, then the VM needs to be capable of Premium Storage (like a DS-series size). Both disks must also be the same size.

Get a list of disks in a resource group using [Get-AzDisk](#)

```
Get-AzDisk -ResourceGroupName myResourceGroup | Format-Table -Property Name
```

When you have the name of the disk that you would like to use, set that as the OS disk for the VM. This example stop\deallocates the VM named *myVM* and assigns the disk named *newDisk* as the new OS disk.

```
Get the VM
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM

Make sure the VM is stopped\deallocated
Stop-AzVM -ResourceGroupName myResourceGroup -Name $vm.Name -Force

Get the new disk that you want to swap in
$disk = Get-AzDisk -ResourceGroupName myResourceGroup -Name newDisk

Set the VM configuration to point to the new disk
Set-AzVMDisk -VM $vm -ManagedDiskId $disk.Id -Name $disk.Name

Update the VM with the new OS disk
Update-AzVM -ResourceGroupName myResourceGroup -VM $vm

Start the VM
Start-AzVM -Name $vm.Name -ResourceGroupName myResourceGroup
```

## Next steps

To create a copy of a disk, see [Snapshot a disk](#).

# How to tag a Windows virtual machine in Azure

2/25/2020 • 4 minutes to read • [Edit Online](#)

This article describes different ways to tag a Windows virtual machine in Azure through the Resource Manager deployment model. Tags are user-defined key/value pairs which can be placed directly on a resource or a resource group. Azure currently supports up to 50 tags per resource and resource group. Tags may be placed on a resource at the time of creation or added to an existing resource. Please note that tags are supported for resources created via the Resource Manager deployment model only. If you want to tag a Linux virtual machine, see [How to tag a Linux virtual machine in Azure](#).

## Tagging a Virtual Machine through Templates

First, let's look at tagging through templates. [This template](#) places tags on the following resources: Compute (Virtual Machine), Storage (Storage Account), and Network (Public IP Address, Virtual Network, and Network Interface). This template is for a Windows VM but can be adapted for Linux VMs.

Click the **Deploy to Azure** button from the [template link](#). This will navigate to the [Azure portal](#) where you can deploy this template.

### Simple deployment of a VM with Tags

 [Deploy to Azure](#)

 [Visualize](#)

This template includes the following tags: *Department*, *Application*, and *Created By*. You can add/edit these tags directly in the template if you would like different tag names.

```
"apiVersion": "2015-05-01-preview",
"type": "Microsoft.Compute/virtualMachines",
"name": "[variables('vmName')]",
"location": "[variables('location')]",
"tags": {
 "Department": "[parameters('departmentName')]",
 "Application": "[parameters('applicationName')]",
 "Created By": "[parameters('createdBy')]"
},
```

As you can see, the tags are defined as key/value pairs, separated by a colon (:). The tags must be defined in this format:

```
"tags": {
 "Key1" : "Value1",
 "Key2" : "Value2"
}
```

Save the template file after you finish editing it with the tags of your choice.

Next, in the **Edit Parameters** section, you can fill out the values for your tags.

DEPARTMENTNAME (string) ⓘ

APPLICATIONNAME (string) ⓘ

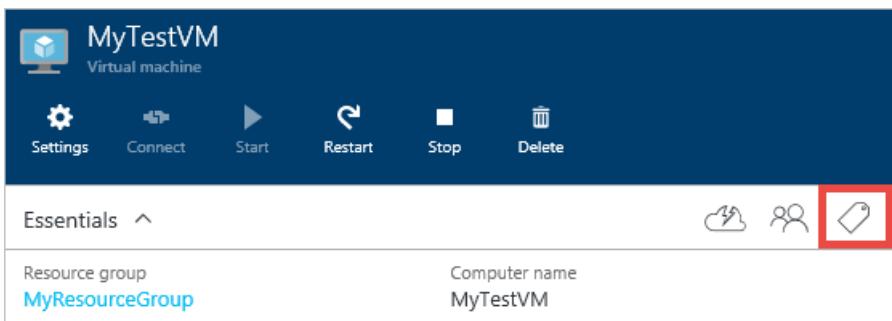
CREATEDBY (string) ⓘ

Click **Create** to deploy this template with your tag values.

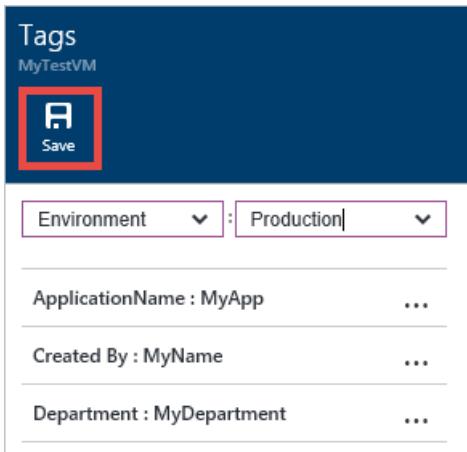
## Tagging through the Portal

After creating your resources with tags, you can view, add, and delete tags in the portal.

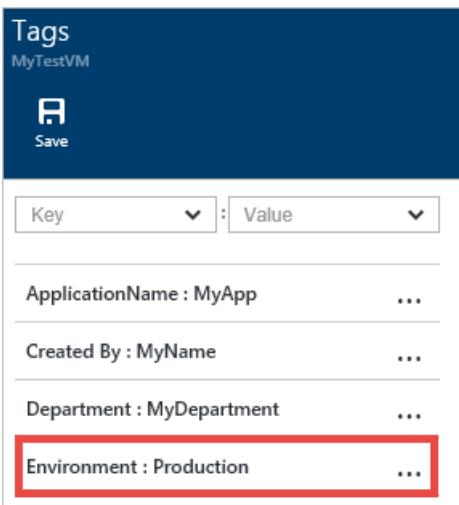
Select the tags icon to view your tags:



Add a new tag through the portal by defining your own Key/Value pair, and save it.



Your new tag should now appear in the list of tags for your resource.



## Tagging with PowerShell

To create, add, and delete tags through PowerShell, you first need to set up your [PowerShell environment with Azure Resource Manager](#). Once you have completed the setup, you can place tags on Compute, Network, and Storage resources at creation or after the resource is created via PowerShell. This article will concentrate on viewing/editing tags placed on Virtual Machines.

First, navigate to a Virtual Machine through the `Get-AzVM` cmdlet.

```
PS C:\> Get-AzVM -ResourceGroupName "MyResourceGroup" -Name "MyTestVM"
```

If your Virtual Machine already contains tags, you will then see all the tags on your resource:

```
Tags : {
 "Application": "MyApp1",
 "Created By": "MyName",
 "Department": "MyDepartment",
 "Environment": "Production"
}
```

If you would like to add tags through PowerShell, you can use the `Set-AzResource` command. Note when updating tags through PowerShell, tags are updated as a whole. So if you are adding one tag to a resource that already has tags, you will need to include all the tags that you want to be placed on the resource. Below is an example of how to add additional tags to a resource through PowerShell Cmdlets.

This first cmdlet sets all of the tags placed on *MyTestVM* to the `$tags` variable, using the `Get-AzResource` and `Tags` property.

```
PS C:\> $tags = (Get-AzResource -ResourceGroupName MyResourceGroup -Name MyTestVM).Tags
```

The second command displays the tags for the given variable.

```
PS C:\> $tags

Key Value
---- ----
Department MyDepartment
Application MyApp1
Created By MyName
Environment Production
```

The third command adds an additional tag to the `$tags` variable. Note the use of the `+=` to append the new key/value pair to the `$tags` list.

```
PS C:\> $tags += @{$Location="MyLocation"}
```

The fourth command sets all of the tags defined in the `$tags` variable to the given resource. In this case, it is `MyTestVM`.

```
PS C:\> Set-AzResource -ResourceGroupName MyResourceGroup -Name MyTestVM -ResourceType "Microsoft.Compute/VirtualMachines" -Tag $tags
```

The fifth command displays all of the tags on the resource. As you can see, `Location` is now defined as a tag with `MyLocation` as the value.

```
PS C:\> (Get-AzResource -ResourceGroupName MyResourceGroup -Name MyTestVM).Tags
```

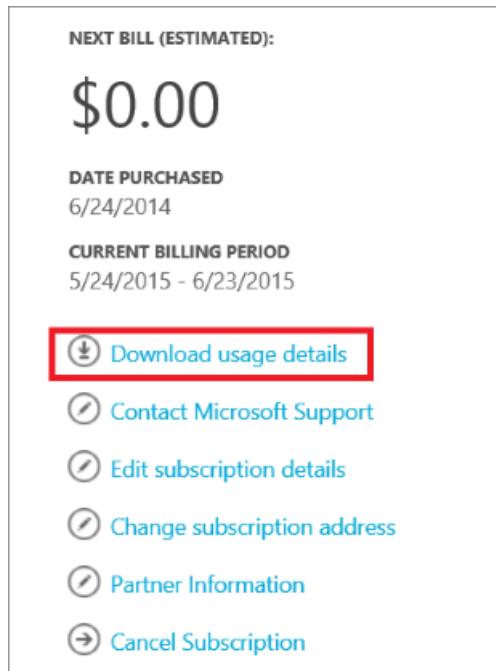
| Key         | Value        |
|-------------|--------------|
| Department  | MyDepartment |
| Application | MyApp1       |
| Created By  | MyName       |
| Environment | Production   |
| Location    | MyLocation   |

To learn more about tagging through PowerShell, check out the [Azure Resource Cmdlets](#).

## Viewing your tags in the usage details

Tags placed on Compute, Network, and Storage resources in the Resource Manager deployment model will be populated in your usage details in the [billing portal](#).

Click on **Download usage details** to view the usage details in your subscription.



Select your billing statement and the **Version 2** usage details:

Click here to [Understand Your Bill](#).

Current period

7/24/2014 - 8/23/2014

[View Current Statement](#)

[Download Usage](#) ▾

Version 2 - Preview

[Download Usage](#)

Version 1

From the usage details, you can see all of the tags in the **Tags** column:

| Consumed Service    | Resource Group    | Instance Id                                                                                                                                       | Tags                                                                                                                                                       |
|---------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "Microsoft.Compute" | "MYRESOURCEGROUP" | "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/MYRESOURCEGROUP/providers/Microsoft.Compute/virtualMachines/MyWindowsVM"      | "[{"Department":"MyDepartment","Application":"MyApp1","Created By":"MyName","Type":"Virtual Machine","Environment":"Production","Location":"MyLocation"}]" |
| "Microsoft.Storage" | "myresourcegroup" | "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myresourcegroup/providers/Microsoft.Storage/storageAccounts/mystorageaccount" | "[{"Application":"MyApp1","Created By":"MyName","Department":"MyDepartment","Type":"Storage Account"}]"                                                    |
| "Microsoft.Network" | "MyResourceGroup" | "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/MyResourceGroup/providers/Microsoft.Network/publicIPAddresses/myPublicIP"     | "[{"Department":"MyDepartment","Application":"MyApp1","Created By":"MyName","Type":"Public IP"}]"                                                          |

By analyzing these tags along with usage, organizations will be able to gain new insights into their consumption data.

## Next steps

- To learn more about tagging your Azure resources, see [Azure Resource Manager Overview](#) and [Using Tags to organize your Azure Resources](#).
- To see how tags can help you manage your use of Azure resources, see [Understanding your Azure Bill](#) and [Gain insights into your Microsoft Azure resource consumption](#).

# Time sync for Windows VMs in Azure

11/26/2019 • 7 minutes to read • [Edit Online](#)

Time sync is important for security and event correlation. Sometimes it is used for distributed transactions implementation. Time accuracy between multiple computer systems is achieved through synchronization. Synchronization can be affected by multiple things, including reboots and network traffic between the time source and the computer fetching the time.

Azure is now backed by infrastructure running Windows Server 2016. Windows Server 2016 has improved algorithms used to correct time and condition the local clock to synchronize with UTC. Windows Server 2016 also improved the VMCTimeSync service that governs how VMs sync with the host for accurate time. Improvements include more accurate initial time on VM start or VM restore and interrupt latency correction for samples provided to Windows Time (W32time).

## NOTE

For a quick overview of Windows Time service, take a look at this [high-level overview video](#).

For more information, see [Accurate time for Windows Server 2016](#).

## Overview

Accuracy for a computer clock is gauged on how close the computer clock is to the Coordinated Universal Time (UTC) time standard. UTC is defined by a multinational sample of precise atomic clocks that can only be off by one second in 300 years. But, reading UTC directly requires specialized hardware. Instead, time servers are synced to UTC and are accessed from other computers to provide scalability and robustness. Every computer has time synchronization service running that knows what time servers to use and periodically checks if computer clock needs to be corrected and adjusts time if needed.

Azure hosts are synchronized to internal Microsoft time servers that take their time from Microsoft-owned Stratum 1 devices, with GPS antennas. Virtual machines in Azure can either depend on their host to pass the accurate time (*host time*) on to the VM or the VM can directly get time from a time server, or a combination of both.

Virtual machine interactions with the host can also affect the clock. During [memory preserving maintenance](#), VMs are paused for up to 30 seconds. For example, before maintenance begins the VM clock shows 10:00:00 AM and lasts 28 seconds. After the VM resumes, the clock on the VM would still show 10:00:00 AM, which would be 28 seconds off. To correct for this, the VMCTimeSync service monitors what is happening on the host and prompts for changes to happen on the VMs to compensate.

The VMCTimeSync service operates in either sample or sync mode and will only influence the clock forward. In sample mode, which requires W32time to be running, the VMCTimeSync service polls the host every 5 seconds and provides time samples to W32time. Approximately every 30 seconds, the W32time service takes the latest time sample and uses it to influence the guest's clock. Sync mode activates if a guest has been resumed or if a guest's clock drifts more than 5 seconds behind the host's clock. In cases where the W32time service is properly running, the latter case should never happen.

Without time synchronization working, the clock on the VM would accumulate errors. When there is only one VM, the effect might not be significant unless the workload requires highly accurate timekeeping. But in most cases, we have multiple, interconnected VMs that use time to track transactions and the time needs to be consistent throughout the entire deployment. When time between VMs is different, you could see the following effects:

- Authentication will fail. Security protocols like Kerberos or certificate-dependent technology rely on time being consistent across the systems.
- It's very hard to figure out what has happened in a system if logs (or other data) don't agree on time. The same event would look like it occurred at different times, making correlation difficult.
- If clock is off, the billing could be calculated incorrectly.

The best results for Windows deployments are achieved by using Windows Server 2016 as the guest operating system, which ensures you can use the latest improvements in time synchronization.

## Configuration options

There are three options for configuring time sync for your Windows VMs hosted in Azure:

- Host time and time.windows.com. This is the default configuration used in Azure Marketplace images.
- Host-only.
- Use another, external time server with or without using host time.

### Use the default

By default Windows OS VM images are configured for w32time to sync from two sources:

- The NtpClient provider, which gets information from time.windows.com.
- The VMICTimeSync service, used to communicate the host time to the VMs and make corrections after the VM is paused for maintenance. Azure hosts use Microsoft-owned Stratum 1 devices to keep accurate time.

w32time would prefer the time provider in the following order of priority: stratum level, root delay, root dispersion, time offset. In most cases, w32time would prefer time.windows.com to the host because time.windows.com reports lower stratum.

For domain joined machines the domain itself establishes time sync hierarchy, but the forest root still needs to take time from somewhere and the following considerations would still hold true.

### Host-only

Because time.windows.com is a public NTP server, syncing time with it requires sending traffic over the internet, varying packet delays can negatively affect quality of the time sync. Removing time.windows.com by switching to host-only sync can sometimes improve your time sync results.

Switching to host-only time sync makes sense if you experience time sync issues using the default configuration. Try out the host-only sync to see if that would improve the time sync on VM.

### External time server

If you have specific time sync requirements, there is also an option of using external time servers. External time servers can provide specific time, which can be useful for test scenarios, ensuring time uniformity with machines hosted in non-Microsoft datacenters, or handling leap seconds in a special way.

You can combine external servers with the VMICTimeSync service and VMICTimeProvider to provide results similar to the default configuration.

## Check your configuration

Check if the NtpClient time provider is configured to use explicit NTP servers (NTP) or domain time sync (NT5DS).

```
w32tm /dumpreg /subkey:Parameters | findstr /i "type"
```

If the VM is using NTP, you will see the following output:

| Value Name | Value Type | Value Data |
|------------|------------|------------|
| Type       | REG_SZ     | NTP        |

To see what time server the NtpClient time provider is using, at an elevated command prompt type:

```
w32tm /dumpreg /subkey:Parameters | findstr /i "ntpserver"
```

If the VM is using the default, the output will look like this:

|           |        |                      |
|-----------|--------|----------------------|
| NtpServer | REG_SZ | time.windows.com,0x8 |
|-----------|--------|----------------------|

To see what time provider is being used currently.

```
w32tm /query /source
```

Here is the output you could see and what it would mean:

- **time.windows.com** - in the default configuration, w32time would get time from time.windows.com. The time sync quality depends on internet connectivity to it and is affected by packet delays. This is the usual output from the default setup.
- **VM IC Time Synchronization Provider** - the VM is syncing time from the host. This usually is the result if you opt-in for host-only time sync or the NtpServer is not available at the moment.
- *Your domain server* - the current machine is in a domain and the domain defines the time sync hierarchy.
- *Some other server* - w32time was explicitly configured to get the time from that another server. Time sync quality depends on this time server quality.
- **Local CMOS Clock** - clock is unsynchronized. You can get this output if w32time hasn't had enough time to start after a reboot or when all the configured time sources are not available.

## Opt-in for host-only time sync

Azure is constantly working on improving time sync on hosts and can guarantee that all the time sync infrastructure is collocated in Microsoft-owned datacenters. If you have time sync issues with the default setup that prefers to use time.windows.com as the primary time source, you can use the following commands to opt-in to host-only time sync.

Mark the VMIC provider as enabled.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\VMICTimeProvider /v Enabled /t REG_DWORD /d 1 /f
```

Mark the NTPClient provider as disabled.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\NtpClient /v Enabled /t REG_DWORD /d 0 /f
```

Restart the w32time Service.

```
net stop w32time && net start w32time
```

## Windows Server 2012 and R2 VMs

Windows Server 2012 and Windows Server 2012 R2 have different default settings for time sync. The w32time by default is configured in a way that prefers low overhead of the service over to precise time.

If you want to move your Windows Server 2012 and 2012 R2 deployments to use the newer defaults that prefer precise time, you can apply the following settings.

Update the w32time poll and update intervals to match Windows Server 2016 settings.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config /v MinPollInterval /t REG_DWORD /d 6 /f
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config /v MaxPollInterval /t REG_DWORD /d 10 /f
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config /v UpdateInterval /t REG_DWORD /d 100 /f
w32tm /config /update
```

For w32time to be able to use the new poll intervals, the NtpServers be marked as using them. If servers are annotated with 0x1 bitflag mask, that would override this mechanism and w32time would use SpecialPollInterval instead. Make sure that specified NTP servers are either using 0x8 flag or no flag at all:

Check what flags are being used for the used NTP servers.

```
w32tm /dumpreg /subkey:Parameters | findstr /i "ntpserver"
```

## Next steps

Below are links to more details about the time sync:

- [Windows Time Service Tools and Settings](#)
- [Windows Server 2016 Improvements](#)
- [Accurate Time for Windows Server 2016](#)
- [Support boundary to configure the Windows Time service for high-accuracy environments](#)

# Run scripts in your Windows VM

6/28/2019 • 2 minutes to read • [Edit Online](#)

To automate tasks or troubleshoot issues, you may need to run commands in a VM. The following article gives a brief overview of the features that are available to run scripts and commands within your VMs.

## Custom Script Extension

The [Custom Script Extension](#) is primarily used for post deployment configuration and software installation.

- Download and run scripts in Azure virtual machines.
- Can be run using Azure Resource Manager templates, Azure CLI, REST API, PowerShell, or Azure portal.
- Script files can be downloaded from Azure storage or GitHub, or provided from your PC when run from the Azure portal.
- Run PowerShell script in Windows machines and Bash script in Linux machines.
- Useful for post deployment configuration, software installation, and other configuration or management tasks.

## Run command

The [Run Command](#) feature enables virtual machine and application management and troubleshooting using scripts, and is available even when the machine is not reachable, for example if the guest firewall doesn't have the RDP or SSH port open.

- Run scripts in Azure virtual machines.
- Can be run using [Azure portal](#), [REST API](#), [Azure CLI](#), or [PowerShell](#)
- Quickly run a script and view output and repeat as needed in the Azure portal.
- Script can be typed directly or you can run one of the built-in scripts.
- Run PowerShell script in Windows machines and Bash script in Linux machines.
- Useful for virtual machine and application management and for running scripts in virtual machines that are unreachable.

## Hybrid Runbook Worker

The [Hybrid Runbook Worker](#) provides general machine, application, and environment management with user's custom scripts stored in an Automation account.

- Run scripts in Azure and non-Azure machines.
- Can be run using Azure portal, Azure CLI, REST API, PowerShell, webhook.
- Scripts stored and managed in an Automation Account.
- Run PowerShell, PowerShell Workflow, Python, or Graphical runbooks
- No time limit on script run time.
- Multiple scripts can run concurrently.
- Full script output is returned and stored.
- Job history available for 90 days.
- Scripts can run as Local System or with user-supplied credentials.
- Requires [manual installation](#)

## Serial console

The [Serial console](#) provides direct access to a VM, similar to having a keyboard connected to the VM.

- Run commands in Azure virtual machines.
- Can be run using a text-based console to the machine in the Azure portal.
- Login to the machine with a local user account.
- Useful when access to the virtual machine is needed regardless of the machine's network or operating system state.

## Next steps

Learn more about the different features that are available to run scripts and commands within your VMs.

- [Custom Script Extension](#)
- [Run Command](#)
- [Hybrid Runbook Worker](#)
- [Serial console](#)

# Custom Script Extension for Windows

2/28/2020 • 11 minutes to read • [Edit Online](#)

The Custom Script Extension downloads and executes scripts on Azure virtual machines. This extension is useful for post deployment configuration, software installation, or any other configuration or management tasks. Scripts can be downloaded from Azure storage or GitHub, or provided to the Azure portal at extension run time. The Custom Script Extension integrates with Azure Resource Manager templates, and can be run using the Azure CLI, PowerShell, Azure portal, or the Azure Virtual Machine REST API.

This document details how to use the Custom Script Extension using the Azure PowerShell module, Azure Resource Manager templates, and details troubleshooting steps on Windows systems.

## Prerequisites

### NOTE

Do not use Custom Script Extension to run Update-AzVM with the same VM as its parameter, since it will wait on itself.

### Operating System

The Custom Script Extension for Windows will run on the extension supported extension OSs, for more information, see this [Azure Extension supported operating systems](#).

### Script Location

You can configure the extension to use your Azure Blob storage credentials to access Azure Blob storage. The script location can be anywhere, as long as the VM can route to that end point, such as GitHub or an internal file server.

### Internet Connectivity

If you need to download a script externally such as from GitHub or Azure Storage, then additional firewall and Network Security Group ports need to be opened. For example, if your script is located in Azure Storage, you can allow access using Azure NSG Service Tags for [Storage](#).

If your script is on a local server, then you may still need additional firewall and Network Security Group ports need to be opened.

### Tips and Tricks

- The highest failure rate for this extension is because of syntax errors in the script, test the script runs without error, and also put in additional logging into the script to make it easier to find where it failed.
- Write scripts that are idempotent. This ensures that if they run again accidentally, it will not cause system changes.
- Ensure the scripts don't require user input when they run.
- There's 90 minutes allowed for the script to run, anything longer will result in a failed provision of the extension.
- Don't put reboots inside the script, this action will cause issues with other extensions that are being installed. Post reboot, the extension won't continue after the restart.
- If you have a script that will cause a reboot, then install applications and run scripts, you can schedule the reboot using a Windows Scheduled Task, or use tools such as DSC, Chef, or Puppet extensions.
- The extension will only run a script once, if you want to run a script on every boot, then you need to use the extension to create a Windows Scheduled Task.
- If you want to schedule when a script will run, you should use the extension to create a Windows Scheduled Task.
- When the script is running, you will only see a 'transitioning' extension status from the Azure portal or CLI. If you want more frequent status updates of a running script, you'll need to create your own solution.
- Custom Script extension does not natively support proxy servers, however you can use a file transfer tool that supports proxy servers within your script, such as [Curl](#)
- Be aware of non-default directory locations that your scripts or commands may rely on, have logic to handle this situation.
- Custom Script Extension will run under the LocalSystem Account

## Extension schema

The Custom Script Extension configuration specifies things like script location and the command to be run. You can store this configuration in configuration files, specify it on the command line, or specify it in an Azure Resource Manager template.

You can store sensitive data in a protected configuration, which is encrypted and only decrypted inside the virtual machine. The protected configuration is useful when the execution command includes secrets such as a password.

These items should be treated as sensitive data and specified in the extensions protected setting configuration. Azure VM extension protected setting data is encrypted, and only decrypted on the target virtual machine.

```
{
 "apiVersion": "2018-06-01",
 "type": "Microsoft.Compute/virtualMachines/extensions",
 "name": "virtualMachineName/config-app",
 "location": "[resourceGroup().location]",
 "dependsOn": [
 "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'), copyIndex())]",
 "[variables('musicstoresqlName')]"
],
 "tags": {
 "displayName": "config-app"
 },
 "properties": {
 "publisher": "Microsoft.Compute",
 "type": "CustomScriptExtension",
 "typeHandlerVersion": "1.10",
 "autoUpgradeMinorVersion": true,
 "settings": {
 "fileUris": [
 "script location"
],
 "timestamp":123456789
 },
 "protectedSettings": {
 "commandToExecute": "myExecutionCommand",
 "storageAccountName": "myStorageAccountName",
 "storageAccountKey": "myStorageAccountKey",
 "managedIdentity" : {}
 }
 }
}
```

#### NOTE

managedIdentity property **must not** be used in conjunction with storageAccountName or storageAccountKey properties

#### NOTE

Only one version of an extension can be installed on a VM at a point in time, specifying custom script twice in the same Resource Manager template for the same VM will fail.

#### NOTE

We can use this schema inside the VirtualMachine resource or as a standalone resource. The name of the resource has to be in this format "virtualMachineName/extensionName", if this extension is used as a standalone resource in the ARM template.

### Property values

| NAME                      | VALUE / EXAMPLE                                                                                                                                                                                                                                                                                   | DATA TYPE      |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| apiVersion                | 2015-06-15                                                                                                                                                                                                                                                                                        | date           |
| publisher                 | Microsoft.Compute                                                                                                                                                                                                                                                                                 | string         |
| type                      | CustomScriptExtension                                                                                                                                                                                                                                                                             | string         |
| typeHandlerVersion        | 1.10                                                                                                                                                                                                                                                                                              | int            |
| fileUris (e.g.)           | <a href="https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1">https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1</a> | array          |
| timestamp (e.g.)          | 123456789                                                                                                                                                                                                                                                                                         | 32-bit integer |
| commandToExecute (e.g.)   | powershell -ExecutionPolicy Unrestricted -File configure-music-app.ps1                                                                                                                                                                                                                            | string         |
| storageAccountName (e.g.) | examplestorageacct                                                                                                                                                                                                                                                                                | string         |
| storageAccountKey (e.g.)  | TmJK/1N3AbAZ3q/+hOXoi/l73zOqsaxXDHqa9Y83/v5UpXQp2DQIBuv2Tfp60cE/OaHsJmQZ7teQfczQj8hg==                                                                                                                                                                                                            | string         |
| managedIdentity (e.g.)    | { } or { "clientId": "31b403aa-c364-4240-a7ff-d85fb6cd7232" } or { "objectId": "12dd289c-0583-46e5-b9b4-115d5c19ef4b" }                                                                                                                                                                           | json object    |

#### NOTE

These property names are case-sensitive. To avoid deployment problems, use the names as shown here.

### Property value details

- commandToExecute : (**required**, string) the entry point script to execute. Use this field instead if your command contains secrets such as passwords, or your fileUris

are sensitive.

- `fileUris` : (optional, string array) the URLs for file(s) to be downloaded.
- `timestamp` (optional, 32-bit integer) use this field only to trigger a rerun of the script by changing value of this field. Any integer value is acceptable; it must only be different than the previous value.
- `storageAccountName` : (optional, string) the name of storage account. If you specify storage credentials, all `fileUris` must be URLs for Azure Blobs.
- `storageAccountKey` : (optional, string) the access key of storage account
- `managedIdentity` : (optional, json object) the [managed identity](#) for downloading file(s)
  - `clientId` : (optional, string) the client ID of the managed identity
  - `objectId` : (optional, string) the object ID of the managed identity

The following values can be set in either public or protected settings, the extension will reject any configuration where the values below are set in both public and protected settings.

- `commandToExecute`

Using public settings maybe useful for debugging, but it's recommended that you use protected settings.

Public settings are sent in clear text to the VM where the script will be executed. Protected settings are encrypted using a key known only to the Azure and the VM. The settings are saved to the VM as they were sent, that is, if the settings were encrypted they're saved encrypted on the VM. The certificate used to decrypt the encrypted values is stored on the VM, and used to decrypt settings (if necessary) at runtime.

#### Property: `managedIdentity`

CustomScript (version 1.10 onwards) supports [managed identity](#) for downloading file(s) from URLs provided in the "fileUris" setting. It allows CustomScript to access Azure Storage private blobs or containers without the user having to pass secrets like SAS tokens or storage account keys.

To use this feature, the user must add a [system-assigned](#) or [user-assigned](#) identity to the VM or VMSS where CustomScript is expected to run, and [grant the managed identity access to the Azure Storage container or blob](#).

To use the system-assigned identity on the target VM/VMSS, set "managedidentity" field to an empty json object.

Example:

```
{
 "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
 "commandToExecute": "powershell.exe script1.ps1",
 "managedIdentity" : {}
}
```

To use the user-assigned identity on the target VM/VMSS, configure "managedidentity" field with the client ID or the object ID of the managed identity.

Examples:

```
{
 "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
 "commandToExecute": "powershell.exe script1.ps1",
 "managedIdentity" : { "clientId": "31b403aa-c364-4240-a7ff-d85fb6cd7232" }
}
```

```
{
 "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
 "commandToExecute": "powershell.exe script1.ps1",
 "managedIdentity" : { "objectId": "12dd289c-0583-46e5-b9b4-115d5c19ef4b" }
}
```

#### NOTE

managedidentity property **must not** be used in conjunction with storageAccountName or storageAccountKey properties

## Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. The JSON schema, which is detailed in the previous section can be used in an Azure Resource Manager template to run the Custom Script Extension during deployment. The following samples show how to use the Custom Script extension:

- [Tutorial: Deploy virtual machine extensions with Azure Resource Manager templates](#)
- [Deploy Two Tier Application on Windows and Azure SQL DB](#)

## PowerShell deployment

The `Set-AzVMCustomScriptExtension` command can be used to add the Custom Script extension to an existing virtual machine. For more information, see [Set-AzVMCustomScriptExtension](#).

```
Set-AzVMCustomScriptExtension -ResourceGroupName <resourceGroupName> `
 -VMName <vmName> `
 -Location myLocation `
 -FileUri <fileUrl> `
 -Run "myScript.ps1" `
 -Name DemoScriptExtension
```

## Additional examples

### Using multiple scripts

In this example, you have three scripts that are used to build your server. The **commandToExecute** calls the first script, then you have options on how the others are called. For example, you can have a master script that controls the execution, with the right error handling, logging, and state management. The scripts are downloaded to the local machine for running. For example in `1_Add_Tools.ps1` you would call `2_Add_Features.ps1` by adding `.\2_Add_Features.ps1` to the script, and repeat this process for the other scripts you define in `$settings`.

```
$fileUri = @("https://xxxxxxxx.blob.core.windows.net/buildServer1/1_Add_Tools.ps1",
"https://xxxxxxxx.blob.core.windows.net/buildServer1/2_Add_Features.ps1",
"https://xxxxxxxx.blob.core.windows.net/buildServer1/3_CompleteInstall.ps1")

$settings = @{$"fileUris" = $fileUri};

$storageAcctName = "xxxxxxxx"
$storageKey = "1234ABCD"
$protectedSettings = @{$"storageAccountName" = $storageAcctName; "storageAccountKey" = $storageKey; "commandToExecute" = "powershell -ExecutionPolicy Unrestricted -File 1_Add_Tools.ps1"};

#run command
Set-AzVMExtension -ResourceGroupName <resourceGroupName> `
 -Location <locationName> `
 -VMName <vmName> `
 -Name "buildserver1" `
 -Publisher "Microsoft.Compute" `
 -ExtensionType "CustomScriptExtension" `
 -TypeHandlerVersion "1.10" `
 -Settings $settings `
 -ProtectedSettings $protectedSettings `
```

### Running scripts from a local share

In this example, you may want to use a local SMB server for your script location. By doing this, you don't need to provide any other settings, except **commandToExecute**.

```
$protectedSettings = @{$"commandToExecute" = "powershell -ExecutionPolicy Unrestricted -File \\filesrv\build\serverUpdate1.ps1"};

Set-AzVMExtension -ResourceGroupName <resourceGroupName> `
 -Location <locationName> `
 -VMName <vmName> `
 -Name "serverUpdate" `
 -Publisher "Microsoft.Compute" `
 -ExtensionType "CustomScriptExtension" `
 -TypeHandlerVersion "1.10" `
 -ProtectedSettings $protectedSettings
```

### How to run custom script more than once with CLI

If you want to run the custom script extension more than once, you can only do this action under these conditions:

- The extension **Name** parameter is the same as the previous deployment of the extension.
- Update the configuration otherwise the command won't be re-executed. You can add in a dynamic property into the command, such as a timestamp.

Alternatively, you can set the **ForceUpdateTag** property to **true**.

### Using Invoke-WebRequest

If you are using **Invoke-WebRequest** in your script, you must specify the parameter `-UseBasicParsing` or else you will receive the following error when checking the detailed status:

```
The response content cannot be parsed because the Internet Explorer engine is not available, or Internet Explorer's first-launch configuration is not complete. Specify the UseBasicParsing parameter and try again.
```

## Virtual Machine Scale Sets

To deploy the Custom Script Extension on a Scale Set, see [Add-AzVmssExtension](#)

## Classic VMs

### IMPORTANT

Classic VMs will be retired on March 1, 2023.

If you use IaaS resources from ASM, please complete your migration by March 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.

For more information, see [Migrate your IaaS resources to Azure Resource Manager by March 1, 2023](#).

To deploy the Custom Script Extension on classic VMs, you can use the Azure portal or the Classic Azure PowerShell cmdlets.

### Azure portal

Navigate to your Classic VM resource. Select **Extensions** under **Settings**.

Click **+ Add** and in the list of resources choose **Custom Script Extension**.

On the **Install extension** page, select the local PowerShell file, and fill out any arguments and click **Ok**.

## PowerShell

Use the [Set-AzureVMCustomScriptExtension](#) cmdlet can be used to add the Custom Script extension to an existing virtual machine.

```
define your file URI
$fileUri = 'https://xxxxxxxx.blob.core.windows.net/scripts/Create-File.ps1'

create vm object
$vm = Get-AzureVM -Name <vmName> -ServiceName <cloudServiceName>

set extension
Set-AzureVMCustomScriptExtension -VM $vm -FileUri $fileUri -Run 'Create-File.ps1'

update vm
$vm | Update-AzureVM
```

## Troubleshoot and support

### Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure PowerShell module. To see the deployment state of extensions for a given VM, run the following command:

```
Get-AzVMExtension -ResourceGroupName <resourceGroupName> -VMName <vmName> -Name myExtensionName
```

Extension output is logged to files found under the following folder on the target virtual machine.

```
C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.CustomScriptExtension
```

The specified files are downloaded into the following folder on the target virtual machine.

```
C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.*\Downloads\<n>
```

where `<n>` is a decimal integer, which may change between executions of the extension. The `1.*` value matches the actual, current `typeHandlerVersion` value of the extension. For example, the actual directory could be `C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.8\Downloads\2`.

When executing the `commandToExecute` command, the extension sets this directory (for example, `...\Downloads\2`) as the current working directory. This process enables the use of relative paths to locate the files downloaded via the `fileURIs` property. See the table below for examples.

Since the absolute download path may vary over time, it's better to opt for relative script/file paths in the `commandToExecute` string, whenever possible. For example:

```
"commandToExecute": "powershell.exe . . . -File \".\scripts\myscript.ps1\""
```

Path information after the first URI segment is kept for files downloaded via the `fileURIs` property list. As shown in the table below, downloaded files are mapped into download subdirectories to reflect the structure of the `fileURIs` values.

### Examples of Downloaded Files

| URI IN FILEURIS                                                         | RELATIVE DOWNLOADED LOCATION        | ABSOLUTE DOWNLOADED LOCATION <sup>1</sup>                                  |
|-------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------|
| <code>https://someAcct.blob.core.windows.net/aContainer/script</code>   | <code>./scripts/myscript.ps1</code> | <code>C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1</code> |
| <code>https://someAcct.blob.core.windows.net/aContainer/topLevel</code> | <code>./topLevel.ps1</code>         | <code>C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1</code> |

<sup>1</sup> The absolute directory paths change over the lifetime of the VM, but not within a single execution of the CustomScript extension.

### Support

If you need more help at any point in this article, you can contact the Azure experts on the [MSDN Azure and Stack Overflow forums](#). You can also file an Azure support incident. Go to the [Azure support site](#) and select Get support. For information about using Azure Support, read the [Microsoft Azure support FAQ](#).

# Run PowerShell scripts in your Windows VM by using Run Command

11/7/2019 • 4 minutes to read • [Edit Online](#)

The Run Command feature uses the virtual machine (VM) agent to run PowerShell scripts within an Azure Windows VM. You can use these scripts for general machine or application management. They can help you to quickly diagnose and remediate VM access and network issues and get the VM back to a good state.

## Benefits

You can access your virtual machines in multiple ways. Run Command can run scripts on your virtual machines remotely by using the VM agent. You use Run Command through the Azure portal, [REST API](#), or [PowerShell](#) for Windows VMs.

This capability is useful in all scenarios where you want to run a script within a virtual machine. It's one of the only ways to troubleshoot and remediate a virtual machine that doesn't have the RDP or SSH port open because of improper network or administrative user configuration.

## Restrictions

The following restrictions apply when you're using Run Command:

- Output is limited to the last 4,096 bytes.
- The minimum time to run a script is about 20 seconds.
- Scripts run as System on Windows.
- One script at a time can run.
- Scripts that prompt for information (interactive mode) are not supported.
- You can't cancel a running script.
- The maximum time a script can run is 90 minutes. After that, it will time out.
- Outbound connectivity from the VM is required to return the results of the script.

### NOTE

To function correctly, Run Command requires connectivity (port 443) to Azure public IP addresses. If the extension doesn't have access to these endpoints, the scripts might run successfully but not return the results. If you're blocking traffic on the virtual machine, you can use [service tags](#) to allow traffic to Azure public IP addresses by using the `AzureCloud` tag.

## Available commands

This table shows the list of commands available for Windows VMs. You can use the **RunPowerShellScript** command to run any custom script that you want. When you're using the Azure CLI or PowerShell to run a command, the value that you provide for the `--command-id` or `-CommandId` parameter must be one of the following listed values. When you specify a value that is not an available command, you receive this error:

The entity was not found in this Azure location

| NAME                       | DESCRIPTION                                                                                                                                                         |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RunPowerShellScript</b> | Runs a PowerShell script.                                                                                                                                           |
| <b>EnableRemotePS</b>      | Configures the machine to enable remote PowerShell.                                                                                                                 |
| <b>EnableAdminAccount</b>  | Checks if the local administrator account is disabled, and if so enables it.                                                                                        |
| <b>IPConfig</b>            | Shows detailed information for the IP address, subnet mask, and default gateway for each adapter bound to TCP/IP.                                                   |
| <b>RDPSettings</b>         | Checks registry settings and domain policy settings. Suggests policy actions if the machine is part of a domain or modifies the settings to default values.         |
| <b>ResetRDPCert</b>        | Removes the SSL certificate tied to the RDP listener and restores the RDP listener security to default. Use this script if you see any issues with the certificate. |
| <b>SetRDPPort</b>          | Sets the default or user-specified port number for Remote Desktop connections. Enables firewall rules for inbound access to the port.                               |

## Azure CLI

The following example uses the [az vm run-command](#) command to run a shell script on an Azure Windows VM.

```
script.ps1
param(
[string]$arg1,
[string]$arg2
)
Write-Host This is a sample script with parameters $arg1 and $arg2

az vm run-command invoke --command-id RunPowerShellScript --name win-vm -g my-resource-group \
--scripts @script.ps1 --parameters "arg1=somefoo" "arg2=somebar"
```

## Azure portal

Go to a VM in the [Azure portal](#) and select **Run command** under **OPERATIONS**. You see a list of the available commands to run on the VM.

 WindowsVM1 - Run command

Virtual machine

Search (Ctrl+ /)

Disaster recovery

Update management

Inventory

Change tracking

Run command

Monitoring

Alerts

Metrics

Diagnostics settings

Advisor recommendations

Run Command uses the VM agent to let you run a script inside this virtual machine. This can be helpful for troubleshooting and recovery, and for general machine and application maintenance. Select a command below to see details.

| NAME                | DESCRIPTION                                 |
|---------------------|---------------------------------------------|
| RunPowerShellScript | Executes a PowerShell script                |
| EnableAdminAccount  | Enable administrator account                |
| EnableRemotePS      | Enable remote PowerShell                    |
| IPConfig            | List IP configuration                       |
| RDPSettings         | Verify RDP Listener Settings                |
| ResetRDPCert        | Restore RDP Authentication mode to defaults |
| SetRDPPort          | Set Remote Desktop port                     |

Learn more  
Run Command  
Provide feedback

Choose a command to run. Some of the commands might have optional or required input parameters. For those commands, the parameters are presented as text fields for you to provide the input values. For each command, you can view the script that's being run by expanding **View script**. **RunPowerShellScript** is different from the other commands, because it allows you to provide your own custom script.

**NOTE**

The built-in commands are not editable.

After you choose the command, select **Run** to run the script. After the script finishes, it returns the output and any errors in the output window. The following screenshot shows an example output from running the **RDPSettings** command.

Run Command Script

RDPSettings

**i** Script execution complete

Details  
Checks registry settings and domain policy settings. Suggests policy actions if machine is part of a domain or modifies the settings to default values.

View script

Parameters  
No parameters

**Run**

Output

```
Not domain joined
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\PortNumber: 3389
HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\fDenyTSConnections:
HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\KeepAliveEnable: 1
HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\KeepAliveInterval: 1
HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\KeepAliveTimeout: 1
HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\fDisableAutoReconnect: 0
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\fInheritReconnectSame: 1
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\fReconnectSame: 0
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\fInheritMaxSessionTime: 1
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\fInheritMaxDisconnectionTime: 1
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\MaxDisconnectionTime: 0
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\MaxConnectionTime: 0
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\fInheritMaxIdleTime: 1
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\MaxIdleTime: 0
HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp\MaxInstanceCount: 4294967295
```

## PowerShell

The following example uses the [Invoke-AzVMRunCommand](#) cmdlet to run a PowerShell script on an Azure VM. The cmdlet expects the script referenced in the `-ScriptPath` parameter to be local to where the cmdlet is being run.

```
Invoke-AzVMRunCommand -ResourceGroupName '<myResourceGroup>' -Name '<myVMName>' -CommandId 'RunPowerShellScript' -ScriptPath '<pathToScript>' -Parameter @{"arg1" = "var1";"arg2" = "var2"}
```

## Limiting access to Run Command

Listing the run commands or showing the details of a command requires the `Microsoft.Compute/locations/runCommands/read` permission at the subscription level. The built-in [Reader](#) role and higher levels have this permission.

Running a command requires the `Microsoft.Compute/virtualMachines/runCommand/action` permission at the subscription level. The [Virtual Machine Contributor](#) role and higher levels have this permission.

You can use one of the [built-in roles](#) or create a [custom role](#) to use Run Command.

## Next steps

To learn about other ways to run scripts and commands remotely in your VM, see [Run scripts in your Windows VM](#).

# Use the D: drive as a data drive on a Windows VM

11/13/2019 • 2 minutes to read • [Edit Online](#)

If your application needs to use the D drive to store data, follow these instructions to use a different drive letter for the temporary disk. Never use the temporary disk to store data that you need to keep.

If you resize or **Stop (Deallocate)** a virtual machine, this may trigger placement of the virtual machine to a new hypervisor. A planned or unplanned maintenance event may also trigger this placement. In this scenario, the temporary disk will be reassigned to the first available drive letter. If you have an application that specifically requires the D: drive, you need to follow these steps to temporarily move the pagefile.sys, attach a new data disk and assign it the letter D and then move the pagefile.sys back to the temporary drive. Once complete, Azure will not take back the D: if the VM moves to a different hypervisor.

For more information about how Azure uses the temporary disk, see [Understanding the temporary drive on Microsoft Azure Virtual Machines](#)

## Attach the data disk

First, you'll need to attach the data disk to the virtual machine. To do this using the portal, see [How to attach a managed data disk in the Azure portal](#).

## Temporarily move pagefile.sys to C drive

1. Connect to the virtual machine.
2. Right-click the **Start** menu and select **System**.
3. In the left-hand menu, select **Advanced system settings**.
4. In the **Performance** section, select **Settings**.
5. Select the **Advanced** tab.
6. In the **Virtual memory** section, select **Change**.
7. Select the **C** drive and then click **System managed size** and then click **Set**.
8. Select the **D** drive and then click **No paging file** and then click **Set**.
9. Click **Apply**. You will get a warning that the computer needs to be restarted for the changes to take affect.
10. Restart the virtual machine.

## Change the drive letters

1. Once the VM restarts, log back on to the VM.
2. Click the **Start** menu and type **diskmgmt.msc** and hit Enter. Disk Management will start.
3. Right-click on **D**, the Temporary Storage drive, and select **Change Drive Letter and Paths**.
4. Under Drive letter, select a new drive such as **T** and then click **OK**.
5. Right-click on the data disk, and select **Change Drive Letter and Paths**.
6. Under Drive letter, select drive **D** and then click **OK**.

## Move pagefile.sys back to the temporary storage drive

1. Right-click the **Start** menu and select **System**
2. In the left-hand menu, select **Advanced system settings**.
3. In the **Performance** section, select **Settings**.

4. Select the **Advanced** tab.
5. In the **Virtual memory** section, select **Change**.
6. Select the OS drive **C** and click **No paging file** and then click **Set**.
7. Select the temporary storage drive **T** and then click **System managed size** and then click **Set**.
8. Click **Apply**. You will get a warning that the computer needs to be restarted for the changes to take affect.
9. Restart the virtual machine.

## Next steps

- You can increase the storage available to your virtual machine by [attaching an additional data disk](#).

# Change the availability set for a VM

2/2/2020 • 2 minutes to read • [Edit Online](#)

The following steps describe how to change the availability set of a VM using Azure PowerShell. A VM can only be added to an availability set when it is created. To change the availability set, you need to delete and then recreate the virtual machine.

This article applies to both Linux and Windows VMs.

This article was last tested on 2/12/2019 using the [Azure Cloud Shell](#) and the [Az PowerShell module](#) version 1.2.0.

## Change the availability set

The following script provides an example of gathering the required information, deleting the original VM and then recreating it in a new availability set.

```
Set variables
$resourceGroup = "myResourceGroup"
$vmName = "myVM"
$newAvailSetName = "myAvailabilitySet"

Get the details of the VM to be moved to the Availability Set
$originalVM = Get-AzVM `-
 -ResourceGroupName $resourceGroup `-
 -Name $vmName

Create new availability set if it does not exist
$availSet = Get-AzAvailabilitySet `-
 -ResourceGroupName $resourceGroup `-
 -Name $newAvailSetName `-
 -ErrorAction Ignore
if (-Not $availSet) {
 $availSet = New-AzAvailabilitySet `-
 -Location $originalVM.Location `-
 -Name $newAvailSetName `-
 -ResourceGroupName $resourceGroup `-
 -PlatformFaultDomainCount 2 `-
 -PlatformUpdateDomainCount 2 `-
 -Sku Aligned
}

Remove the original VM
Remove-AzVM -ResourceGroupName $resourceGroup -Name $vmName

Create the basic configuration for the replacement VM.
$newVM = New-AzVMConfig `-
 -VMName $originalVM.Name `-
 -VMSize $originalVM.HardwareProfile.VmSize `-
 -AvailabilitySetId $availSet.Id

For a Linux VM, change the last parameter from -Windows to -Linux
Set-AzVMOSDisk `-
 -VM $newVM -CreateOption Attach `-
 -ManagedDiskId $originalVM.StorageProfile.OsDisk.ManagedDisk.Id `-
 -Name $originalVM.StorageProfile.OsDisk.Name `-
 -Windows

Add Data Disks
foreach ($disk in $originalVM.StorageProfile.DataDisks) {
 Add-AzVMDataDisk -VM $newVM `-
```

```

 -Name $disk.Name `
 -ManagedDiskId $disk.ManagedDisk.Id `
 -Caching $disk.Caching `
 -Lun $disk.Lun `
 -DiskSizeInGB $disk.DiskSizeGB `
 -CreateOption Attach
}

Add NIC(s) and keep the same NIC as primary
foreach ($nic in $originalVM.NetworkProfile.NetworkInterfaces) {
if ($nic.Primary -eq "True")
{
 Add-AzVMNetworkInterface `
 -VM $newVM `
 -Id $nic.Id -Primary
}
else
{
 Add-AzVMNetworkInterface `
 -VM $newVM `
 -Id $nic.Id
}
}

Recreate the VM
New-AzVM `
 -ResourceGroupName $resourceGroup `
 -Location $originalVM.Location `
 -VM $newVM `
 -DisableBginfoExtension

```

## Next steps

Add additional storage to your VM by adding an additional [data disk](#).

# Download the template for a VM

11/13/2019 • 2 minutes to read • [Edit Online](#)

When you create a VM in Azure using the portal or PowerShell, a Resource Manager template is automatically created for you. You can use this template to quickly duplicate a deployment. The template contains information about all of the resources in a resource group. For a virtual machine, this means the template contains everything that is created in support of the VM in that resource group, including the networking resources.

## Download the template using the portal

1. Log in to the [Azure portal](#).
2. On the left menu, select **Virtual Machines**.
3. Select the virtual machine from the list.
4. Select **Export template**.
5. Select **Download** from the menu at the top and save the .zip file to your local computer.
6. Open the .zip file and extract the files to a folder. The .zip file contains:
  - parameters.json
  - template.json

The template.json file is the template.

## Download the template using PowerShell

You can also download the .json template file using the `Export-AzResourceGroup` cmdlet. You can use the `-path` parameter to provide the filename and path for the .json file. This example shows how to download the template for the resource group named **myResourceGroup** to the **C:\users\public\downloads** folder on your local computer.

```
Export-AzResourceGroup -ResourceGroupName "myResourceGroup" -Path "C:\users\public\downloads"
```

## Next steps

To learn more about deploying resources using templates, see [Resource Manager template walkthrough](#).

# Azure Virtual Machine Agent overview

1/17/2020 • 3 minutes to read • [Edit Online](#)

The Microsoft Azure Virtual Machine Agent (VM Agent) is a secure, lightweight process that manages virtual machine (VM) interaction with the Azure Fabric Controller. The VM Agent has a primary role in enabling and executing Azure virtual machine extensions. VM Extensions enable post-deployment configuration of VM, such as installing and configuring software. VM extensions also enable recovery features such as resetting the administrative password of a VM. Without the Azure VM Agent, VM extensions cannot be run.

This article details installation and detection of the Azure Virtual Machine Agent.

## Install the VM Agent

### Azure Marketplace image

The Azure VM Agent is installed by default on any Windows VM deployed from an Azure Marketplace image. When you deploy an Azure Marketplace image from the portal, PowerShell, Command Line Interface, or an Azure Resource Manager template, the Azure VM Agent is also installed.

The Windows Guest Agent Package is broken into two parts:

- Provisioning Agent (PA)
- Windows Guest Agent (WinGA)

To boot a VM you must have the PA installed on the VM, however the WinGA does not need to be installed. At VM deploy time, you can select not to install the WinGA. The following example shows how to select the *provisionVmAgent* option with an Azure Resource Manager template:

```
"resources": [{}
 "name": "[parameters('virtualMachineName')]",
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2016-04-30-preview",
 "location": "[parameters('location')]",
 "dependsOn": "[concat('Microsoft.Network/networkInterfaces/', parameters('networkInterfaceName'))]",
 "properties": {
 "osProfile": {
 "computerName": "[parameters('virtualMachineName')]",
 "adminUsername": "[parameters('adminUsername')]",
 "adminPassword": "[parameters('adminPassword')]",
 "windowsConfiguration": {
 "provisionVmAgent": "false"
 }
 }
 }
]
```

If you do not have the Agents installed, you cannot use some Azure services, such as Azure Backup or Azure Security. These services require an extension to be installed. If you have deployed a VM without the WinGA, you can install the latest version of the agent later.

### Manual installation

The Windows VM agent can be manually installed with a Windows installer package. Manual installation may be necessary when you create a custom VM image that is deployed to Azure. To manually install the Windows VM Agent, [download the VM Agent installer](#). The VM Agent is supported on Windows Server 2008 R2 and later.

#### NOTE

It is important to update the AllowExtensionOperations option after manually installing the VM Agent on a VM that was deployed from image without ProvisionVMAgent enable.

```
$vm.OSProfile.AllowExtensionOperations = $true
$vm | Update-AzVM
```

#### Prerequisites

- The Windows VM Agent needs at least Windows Server 2008 R2 (64-bits) to run, with the .Net Framework 4.0. See [Minimum version support for virtual machine agents in Azure](#)
- Ensure your VM has access to IP address 168.63.129.16. For more information see [What is IP address 168.63.129.16](#).

## Detect the VM Agent

#### PowerShell

The Azure Resource Manager PowerShell module can be used to retrieve information about Azure VMs. To see information about a VM, such as the provisioning state for the Azure VM Agent, use [Get-AzVM](#):

```
Get-AzVM
```

The following condensed example output shows the *ProvisionVMAgent* property nested inside *OSProfile*. This property can be used to determine if the VM agent has been deployed to the VM:

```
OSProfile :
ComputerName : myVM
AdminUsername : myUserName
WindowsConfiguration :
 ProvisionVMAgent : True
 EnableAutomaticUpdates : True
```

The following script can be used to return a concise list of VM names and the state of the VM Agent:

```
$vms = Get-AzVM

foreach ($vm in $vms) {
 $agent = $vm | Select -ExpandProperty OSProfile | Select -ExpandProperty WindowsConfiguration | Select
 ProvisionVMAgent
 Write-Host $vm.Name $agent.ProvisionVMAgent
}
```

#### Manual Detection

When logged in to a Windows VM, Task Manager can be used to examine running processes. To check for the Azure VM Agent, open Task Manager, click the *Details* tab, and look for a process name

**WindowsAzureGuestAgent.exe**. The presence of this process indicates that the VM agent is installed.

## Upgrade the VM Agent

The Azure VM Agent for Windows is automatically upgraded. As new VMs are deployed to Azure, they receive the latest VM agent at VM provision time. Custom VM images should be manually updated to include the new VM

agent at image creation time.

## Windows Guest Agent Automatic Logs Collection

Windows Guest Agent has a feature to automatically collect some logs. This feature is controlled by the CollectGuestLogs.exe process. It exists for both PaaS Cloud Services and IaaS Virtual Machines and its goal is to quickly & automatically collect some diagnostics logs from a VM - so they can be used for offline analysis. The collected logs are Event Logs, OS Logs, Azure Logs and some registry keys. It produces a ZIP file that is transferred to the VM's Host. This ZIP file can then be looked at by Engineering Teams and Support professionals to investigate issues on request of the customer owning the VM.

## Next steps

For more information about VM extensions, see [Azure virtual machine extensions and features overview](#).

# Guidance for mitigating speculative execution side-channel vulnerabilities in Azure

11/13/2019 • 7 minutes to read • [Edit Online](#)

**Last document update:** 12 November 2019 10:00 AM PST.

The disclosure of a [new class of CPU vulnerabilities](#) known as speculative execution side-channel attacks has resulted in questions from customers seeking more clarity.

Microsoft has deployed mitigations across all our cloud services. The infrastructure that runs Azure and isolates customer workloads from each other is protected. This means that a potential attacker using the same infrastructure can't attack your application using these vulnerabilities.

Azure is using [memory preserving maintenance](#) whenever possible, to minimize customer impact and eliminate the need for reboots. Azure will continue utilizing these methods when making systemwide updates to the host and protect our customers.

More information about how security is integrated into every aspect of Azure is available on the [Azure Security Documentation](#) site.

## NOTE

Since this document was first published, multiple variants of this vulnerability class have been disclosed. Microsoft continues to be heavily invested in protecting our customers and providing guidance. This page will be updated as we continue to release further fixes.

On November 12, 2019, [Intel published](#) a technical advisory around Intel® Transactional Synchronization Extensions (Intel® TSX) Transaction Asynchronous Abort (TAA) vulnerability that is assigned [CVE-2019-11135](#). This vulnerability affects Intel® Core® processors and Intel® Xeon® processors. Microsoft Azure has released operating system updates and is deploying new microcode, as it is made available by Intel, throughout our fleet to protect our customers against these new vulnerabilities. Azure is closely working with Intel to test and validate the new microcode prior to its official release on the platform.

**Customers that are running untrusted code within their VM** need to take action to protect against these vulnerabilities by reading below for additional guidance on all speculative execution side-channel vulnerabilities (Microsoft Advisories ADV 180002, 180018, and 190013).

Other customers should evaluate these vulnerabilities from a Defense in Depth perspective and consider the security and performance implications of their chosen configuration.

## Keeping your operating systems up-to-date

While an OS update is not required to isolate your applications running on Azure from other Azure customers, it is always a best practice to keep your software up-to-date. The latest Security Rollups for Windows contain mitigations for several speculative execution side channel vulnerabilities. Similarly, Linux distributions have released multiple updates to address these vulnerabilities. Here are our recommended actions to update your operating system:

| OFFERING             | RECOMMENDED ACTION                                                                |
|----------------------|-----------------------------------------------------------------------------------|
| Azure Cloud Services | Enable <a href="#">auto update</a> or ensure you are running the newest Guest OS. |

| OFFERING                       | RECOMMENDED ACTION                                                                                                           |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Azure Linux Virtual Machines   | Install updates from your operating system provider. For more information, see <a href="#">Linux</a> later in this document. |
| Azure Windows Virtual Machines | Install the latest security rollup.                                                                                          |
| Other Azure PaaS Services      | There is no action needed for customers using these services. Azure automatically keeps your OS versions up-to-date.         |

## Additional guidance if you are running untrusted code

Customers who allow untrusted users to execute arbitrary code may wish to implement some additional security features inside their Azure Virtual Machines or Cloud Services. These features protect against the intra-process disclosure vectors that several speculative execution vulnerabilities describe.

Example scenarios where additional security features are recommended:

- You allow code that you do not trust to run inside your VM.
  - *For example, you allow one of your customers to upload a binary or script that you then execute within your application.*
- You allow users that you do not trust to log into your VM using low privileged accounts.
  - *For example, you allow a low-privileged user to log into one of your VMs using remote desktop or SSH.*
- You allow untrusted users access to virtual machines implemented via nested virtualization.
  - *For example, you control the Hyper-V host, but allocate the VMs to untrusted users.*

Customers who do not implement a scenario involving untrusted code do not need to enable these additional security features.

## Enabling additional security

You can enable additional security features inside your VM or Cloud Service if you are running untrusted code. In parallel, ensure your operating system is up-to-date to enable security features inside your VM or Cloud Service

### Windows

Your target operating system must be up-to-date to enable these additional security features. While numerous speculative execution side channel mitigations are enabled by default, the additional features described here must be enabled manually and may cause a performance impact.

**Step 1: Disable hyper-threading on the VM** - Customers running untrusted code on a hyper-threaded VM will need to disable hyper-threading or move to a non-hyper-threaded VM size. Reference [this doc](#) for a list of hyper-threaded VM sizes (where ratio of vCPU to Core is 2:1). To check if your VM has hyper-threading enabled, please refer to the below script using the Windows command line from within the VM.

Type `wmic` to enter the interactive interface. Then type the below to view the amount of physical and logical processors on the VM.

```
CPU Get NumberOfCores,NumberOfLogicalProcessors /Format:List
```

If the number of logical processors is greater than physical processors (cores), then hyper-threading is enabled. If you are running a hyper-threaded VM, please [contact Azure Support](#) to get hyper-threading disabled. Once hyper-

threading is disabled, **support will require a full VM reboot**. Please refer to [Core count](#) to understand why your VM core count decreased.

**Step 2:** In parallel to Step 1, follow the instructions in [KB4072698](#) to verify protections are enabled using the [SpeculationControl](#) PowerShell module.

**NOTE**

If you previously downloaded this module, you will need to install the newest version.

The output of the PowerShell script should have the below values to validate enabled protections against these vulnerabilities:

```
Windows OS support for branch target injection mitigation is enabled: True
Windows OS support for kernel VA shadow is enabled: True
Windows OS support for speculative store bypass disable is enabled system-wide: False
Windows OS support for L1 terminal fault mitigation is enabled: True
Windows OS support for MDS mitigation is enabled: True
Windows OS support for TAA mitigation is enabled: True
```

If the output shows `MDS mitigation is enabled: False`, please [contact Azure Support](#) for available mitigation options.

**Step 3:** To enable Kernel Virtual Address Shadowing (KVAS) and Branch Target Injection (BTI) OS support, follow the instructions in [KB4072698](#) to enable protections using the [Session Manager](#) registry keys. A reboot is required.

**Step 4:** For deployments that are using [nested virtualization](#) (D3 and E3 only): These instructions apply inside the VM you are using as a Hyper-V host.

1. Follow the instructions in [KB4072698](#) to enable protections using the [MinVmVersionForCpuBasedMitigations](#) registry keys.
2. Set the hypervisor scheduler type to `Core` by following the instructions [here](#).

## Linux

Enabling the set of additional security features inside requires that the target operating system be fully up-to-date. Some mitigations will be enabled by default. The following section describes the features which are off by default and/or reliant on hardware support (microcode). Enabling these features may cause a performance impact. Reference your operating system provider's documentation for further instructions

**Step 1: Disable hyper-threading on the VM** - Customers running untrusted code on a hyper-threaded VM will need to disable hyper-threading or move to a non-hyper-threaded VM. Reference [this doc](#) for a list of hyper-threaded VM sizes (where ratio of vCPU to Core is 2:1). To check if you are running a hyper-threaded VM, run the `lscpu` command in the Linux VM.

If `Thread(s) per core = 2`, then hyper-threading has been enabled.

If `Thread(s) per core = 1`, then hyper-threading has been disabled.

Sample output for a VM with hyper-threading enabled:

|                      |                |
|----------------------|----------------|
| CPU Architecture:    | x86_64         |
| CPU op-mode(s):      | 32-bit, 64-bit |
| Byte Order:          | Little Endian  |
| CPU(s):              | 8              |
| On-line CPU(s) list: | 0-7            |
| Thread(s) per core:  | 2              |
| Core(s) per socket:  | 4              |
| Socket(s):           | 1              |
| NUMA node(s):        | 1              |

If you are running a hyper-threaded VM, please [contact Azure Support](#) to get hyper-threading disabled. Once hyper-threading is disabled, **support will require a full VM reboot**. Please refer to [Core count](#) to understand why your VM core count decreased.

**Step 2:** To mitigate against any of the below speculative execution side-channel vulnerabilities, refer to your operating system provider's documentation:

- [Redhat and CentOS](#)
- [SUSE](#)
- [Ubuntu](#)

#### Core count

When a hyper-threaded VM is created, Azure allocates 2 threads per core - these are called vCPUs. When hyper-threading is disabled, Azure removes a thread and surfaces up single threaded cores (physical cores). The ratio of vCPU to CPU is 2:1, so once hyper-threading is disabled, the CPU count in the VM will appear to have decreased by half. For example, a D8\_v3 VM is a hyper-threaded VM running on 8 vCPUs (2 threads per core x 4 cores).

When hyper-threading is disabled, CPUs will drop to 4 physical cores with 1 thread per core.

## Next steps

This article provides guidance to the below speculative execution side-channel attacks that affect many modern processors:

#### Spectre Meltdown:

- CVE-2017-5715 - Branch Target Injection (BTI)
- CVE-2017-5754 - Kernel Page Table Isolation (KPTI)
- CVE-2018-3639 – Speculative Store Bypass (KPTI)
- [CVE-2019-1125](#) – Windows Kernel Information – variant of Spectre Variant 1

#### L1 Terminal Fault (L1TF):

- CVE-2018-3615 - Intel Software Guard Extensions (Intel SGX)
- CVE-2018-3620 - Operating Systems (OS) and System Management Mode (SMM)
- CVE-2018-3646 – impacts Virtual Machine Manager (VMM)

#### Microarchitectural Data Sampling:

- CVE-2019-11091 - Microarchitectural Data Sampling Uncacheable Memory (MDSUM)
- CVE-2018-12126 - Microarchitectural Store Buffer Data Sampling (MSBDS)
- CVE-2018-12127 - Microarchitectural Load Port Data Sampling (MLPDS)
- CVE-2018-12130 - Microarchitectural Fill Buffer Data Sampling (MFBDS)

Transactional Synchronization Extensions (Intel® TSX) Transaction Asynchronous Abort:

- [CVE-2019-11135](#) – TSX Transaction Asynchronous Abort (TAA)



# Azure Instance Metadata service

2/25/2020 • 20 minutes to read • [Edit Online](#)

The Azure Instance Metadata Service (IMDS) provides information about currently running virtual machine instances and can be used to manage and configure your virtual machines. Information provided includes the SKU, network configuration, and upcoming maintenance events. For a complete list of the data that is available, see [metadata APIs](#).

Azure's Instance Metadata Service is a REST Endpoint accessible to all IaaS VMs created via the [Azure Resource Manager](#). The endpoint is available at a well-known non-routable IP address (`169.254.169.254`) that can be accessed only from within the VM.

## IMPORTANT

This service is **generally available** in all Azure Regions. It regularly receives updates to expose new information about virtual machine instances. This page reflects the up-to-date [metadata APIs](#) available.

## Service availability

The service is available in generally available Azure regions. Not all API version may be available in all Azure Regions.

| REGIONS                                      | AVAILABILITY?       | SUPPORTED VERSIONS                                                                                                                                         |
|----------------------------------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| All Generally Available Global Azure Regions | Generally Available | 2017-04-02, 2017-08-01, 2017-12-01, 2018-02-01, 2018-04-02, 2018-10-01, 2019-02-01, 2019-03-11, 2019-04-30, 2019-06-01, 2019-06-04, 2019-08-01, 2019-08-15 |
| Azure Government                             | Generally Available | 2017-04-02, 2017-08-01, 2017-12-01, 2018-02-01, 2018-04-02, 2018-10-01, 2019-02-01, 2019-03-11, 2019-04-30, 2019-06-01, 2019-06-04, 2019-08-01, 2019-08-15 |
| Azure China 21Vianet                         | Generally Available | 2017-04-02, 2017-08-01, 2017-12-01, 2018-02-01, 2018-04-02, 2018-10-01, 2019-02-01, 2019-03-11, 2019-04-30, 2019-06-01, 2019-06-04, 2019-08-01, 2019-08-15 |
| Azure Germany                                | Generally Available | 2017-04-02, 2017-08-01, 2017-12-01, 2018-02-01, 2018-04-02, 2018-10-01, 2019-02-01, 2019-03-11, 2019-04-30, 2019-06-01, 2019-06-04, 2019-08-01, 2019-08-15 |

The version 2019-11-01 is currently getting deployed and may not be available in all regions.

This table is updated when there are service updates and/or new supported versions are available.

To try out the Instance Metadata Service, create a VM from [Azure Resource Manager](#) or the [Azure portal](#) in the

above regions and follow the examples below. Further examples of how to query IMDS can be found at [Azure Instance Metadata Samples](#)

## Usage

### Versioning

The Instance Metadata Service is versioned, and specifying the API version in the HTTP request is mandatory.

You can see the newest versions listed in this [availability table](#).

As newer versions are added, older versions can still be accessed for compatibility if your scripts have dependencies on specific data formats.

When no version is specified, an error is returned with a list of the newest supported versions.

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance"
```

### Response

```
{
 "error": "Bad request. api-version was not specified in the request. For more information refer to
aka.ms/azureimds",
 "newest-versions": [
 "2018-10-01",
 "2018-04-02",
 "2018-02-01"
]
}
```

### Using headers

When you query the Instance Metadata Service, you must provide the header `Metadata: true` to ensure the request was not unintentionally redirected.

### Retrieving metadata

Instance metadata is available for running VMs created/managed using [Azure Resource Manager](#). Access all data categories for a virtual machine instance using the following request:

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2017-08-01"
```

#### NOTE

All instance metadata queries are case-sensitive.

### Data output

By default, the Instance Metadata Service returns data in JSON format (`Content-Type: application/json`).

However, different APIs return data in different formats if requested. The following table is a reference of other data formats APIs may support.

| API              | DEFAULT DATA FORMAT | OTHER FORMATS |
|------------------|---------------------|---------------|
| /instance        | json                | text          |
| /scheduledevents | json                | none          |
| /attested        | json                | none          |

To access a non-default response format, specify the requested format as a query string parameter in the request. For example:

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2017-08-01&format=text"
```

#### NOTE

For leaf nodes the `format=json` doesn't work. For these queries `format=text` needs to be explicitly specified if the default format is json.

## Security

The Instance Metadata Service endpoint is accessible only from within the running virtual machine instance on a non-routable IP address. In addition, any request with a `X-Forwarded-For` header is rejected by the service.

Requests must also contain a `Metadata: true` header to ensure that the actual request was directly intended and not a part of unintentional redirection.

## Error

If there is a data element not found or a malformed request, the Instance Metadata Service returns standard HTTP errors. For example:

| HTTP STATUS CODE       | REASON                                                                                     |
|------------------------|--------------------------------------------------------------------------------------------|
| 200 OK                 |                                                                                            |
| 400 Bad Request        | Missing <code>Metadata: true</code> header or missing the format when querying a leaf node |
| 404 Not Found          | The requested element doesn't exist                                                        |
| 405 Method Not Allowed | Only <code>GET</code> requests are supported                                               |
| 410 Gone               | Retry after some time for a max of 70 seconds                                              |
| 429 Too Many Requests  | The API currently supports a maximum of 5 queries per second                               |
| 500 Service Error      | Retry after some time                                                                      |

## Examples

#### NOTE

All API responses are JSON strings. All following example responses are pretty-printed for readability.

## Retrieving network information

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/network?api-version=2017-08-01"
```

### Response

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "interface": [
 {
 "ipv4": {
 "ipAddress": [
 {
 "privateIpAddress": "10.1.0.4",
 "publicIpAddress": "X.X.X.X"
 }
],
 "subnet": [
 {
 "address": "10.1.0.0",
 "prefix": "24"
 }
]
 },
 "ipv6": {
 "ipAddress": []
 },
 "macAddress": "000D3AF806EC"
 }
]
}
```

## Retrieving public IP address

```
curl -H Metadata:true
"http://169.254.169.254/metadata/instance/network/interface/0/ipv4/ipAddress/0/publicIpAddress?api-
version=2017-08-01&format=text"
```

## Retrieving all metadata for an instance

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2019-06-01"
```

### Response

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "compute": {
 "azEnvironment": "AzurePublicCloud",
```

```
"customData": "",
"location": "centralus",
"name": "negasonic",
"offer": "lampstack",
"osType": "Linux",
"placementGroupId": "",
"plan": {
 "name": "5-6",
 "product": "lampstack",
 "publisher": "bitnami"
},
"platformFaultDomain": "0",
"platformUpdateDomain": "0",
"provider": "Microsoft.Compute",
"publicKeys": [],
"publisher": "bitnami",
"resourceGroupName": "myrg",
"resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx/resourceGroups/myrg/providers/Microsoft.Compute/virtualMachines/negasonic",
"sku": "5-6",
"storageProfile": {
 "dataDisks": [
 {
 "caching": "None",
 "createOption": "Empty",
 "diskSizeGB": "1024",
 "image": {
 "uri": ""
 },
 "lun": "0",
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampledataldiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampledataldiskname",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
],
 "imageReference": {
 "id": "",
 "offer": "UbuntuServer",
 "publisher": "Canonical",
 "sku": "16.04.0-LTS",
 "version": "latest"
 },
 "osDisk": {
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "diskSizeGB": "30",
 "diffDiskSettings": {
 "option": "Local"
 },
 "encryptionSettings": {
 "enabled": "false"
 },
 "image": {
 "uri": ""
 },
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampleosdiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampleosdiskname",
 "osType": "Linux"
 }
}
```

```

 "osType": "Linux",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
}
},
"subscriptionId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
"tags": "Department:IT;Environment:Test;Role:WebRole",
"version": "7.1.1902271506",
"vmId": "13f56399-bd52-4150-9748-7190aae1ff21",
"vmScaleSetName": "",
"vmSize": "Standard_A1_v2",
"zone": "1"
},
"network": {
 "interface": [
 {
 "ipv4": {
 "ipAddress": [
 {
 "privateIpAddress": "10.1.2.5",
 "publicIpAddress": "X.X.X.X"
 }
],
 "subnet": [
 {
 "address": "10.1.2.0",
 "prefix": "24"
 }
]
 },
 "ipv6": {
 "ipAddress": []
 },
 "macAddress": "000D3A36DDDE"
 }
]
}
}

```

## Retrieving metadata in Windows Virtual Machine

### Request

Instance metadata can be retrieved in Windows via the `curl` program:

```
curl -H @{'Metadata'='true'} http://169.254.169.254/metadata/instance?api-version=2019-06-01 | select -ExpandProperty Content
```

Or through the `Invoke-RestMethod` PowerShell cmdlet:

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -URI http://169.254.169.254/metadata/instance?api-version=2019-06-01 -Method get
```

### Response

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
```

```
"compute": {
 "azEnvironment": "AzurePublicCloud",
 "customData": "",
 "location": "centralus",
 "name": "negasonic",
 "offer": "lampstack",
 "osType": "Linux",
 "placementGroupId": "",
 "plan": {
 "name": "5-6",
 "product": "lampstack",
 "publisher": "bitnami"
 },
 "platformFaultDomain": "0",
 "platformUpdateDomain": "0",
 "provider": "Microsoft.Compute",
 "publicKeys": [],
 "publisher": "bitnami",
 "resourceGroupName": "myrg",
 "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxx-
xxxxxxxx/resourceGroups/myrg/providers/Microsoft.Compute/virtualMachines/negasonic",
 "sku": "5-6",
 "storageProfile": {
 "dataDisks": [
 {
 "caching": "None",
 "createOption": "Empty",
 "diskSizeGB": "1024",
 "image": {
 "uri": ""
 },
 "lun": "0",
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampledatadiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampledatadiskname",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
],
 "imageReference": {
 "id": "",
 "offer": "UbuntuServer",
 "publisher": "Canonical",
 "sku": "16.04.0-LTS",
 "version": "latest"
 },
 "osDisk": {
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "diskSizeGB": "30",
 "diffDiskSettings": {
 "option": "Local"
 },
 "encryptionSettings": {
 "enabled": "false"
 },
 "image": {
 "uri": ""
 },
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampleosdiskname",
 "storageAccountType": "Standard_LRS"
 }
 }
 }
}
```

```

 },
 "name": "exampleosdiskname",
 "osType": "Linux",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 },
},
"subscriptionId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
"tags": "Department:IT;Environment:Test;Role:WebRole",
"version": "7.1.1902271506",
"vmId": "13f56399-bd52-4150-9748-7190aae1ff21",
"vmScaleSetName": "",
"vmSize": "Standard_A1_v2",
"zone": "1"
},
"network": {
 "interface": [
 {
 "ipv4": {
 "ipAddress": [
 {
 "privateIpAddress": "10.0.1.4",
 "publicIpAddress": "X.X.X.X"
 }
],
 "subnet": [
 {
 "address": "10.0.1.0",
 "prefix": "24"
 }
]
 },
 "ipv6": {
 "ipAddress": []
 },
 "macAddress": "002248020E1E"
 }
]
}
}

```

## Metadata APIs

The following APIs are available through the metadata endpoint:

| DATA            | DESCRIPTION                                                                            | VERSION INTRODUCED |
|-----------------|----------------------------------------------------------------------------------------|--------------------|
| attested        | See <a href="#">Attested Data</a>                                                      | 2018-10-01         |
| identity        | Managed identities for Azure resources.<br>See <a href="#">acquire an access token</a> | 2018-02-01         |
| instance        | See <a href="#">Instance API</a>                                                       | 2017-04-02         |
| scheduledevents | See <a href="#">Scheduled Events</a>                                                   | 2017-08-01         |

### Instance API

The following Compute categories are available through the Instance API:

**NOTE**

Through the metadata endpoint, the following categories are accessed through instance/compute

| DATA                 | DESCRIPTION                                                                                               | VERSION INTRODUCED |
|----------------------|-----------------------------------------------------------------------------------------------------------|--------------------|
| azEnvironment        | Azure Environment where the VM is running in                                                              | 2018-10-01         |
| customData           | This feature is currently disabled, and we will update this documentation when it becomes available       | 2019-02-01         |
| location             | Azure Region the VM is running in                                                                         | 2017-04-02         |
| name                 | Name of the VM                                                                                            | 2017-04-02         |
| offer                | Offer information for the VM image and is only present for images deployed from Azure image gallery       | 2017-04-02         |
| osType               | Linux or Windows                                                                                          | 2017-04-02         |
| placementGroupId     | <a href="#">Placement Group</a> of your virtual machine scale set                                         | 2017-08-01         |
| plan                 | <a href="#">Plan</a> containing name, product, and publisher for a VM if it is an Azure Marketplace Image | 2018-04-02         |
| platformUpdateDomain | <a href="#">Update domain</a> the VM is running in                                                        | 2017-04-02         |
| platformFaultDomain  | <a href="#">Fault domain</a> the VM is running in                                                         | 2017-04-02         |
| provider             | Provider of the VM                                                                                        | 2018-10-01         |
| publicKeys           | <a href="#">Collection of Public Keys</a> assigned to the VM and paths                                    | 2018-04-02         |
| publisher            | Publisher of the VM image                                                                                 | 2017-04-02         |
| resourceGroupName    | <a href="#">Resource group</a> for your Virtual Machine                                                   | 2017-08-01         |
| resourceId           | The <a href="#">fully qualified</a> ID of the resource                                                    | 2019-03-11         |
| sku                  | Specific SKU for the VM image                                                                             | 2017-04-02         |
| storageProfile       | See <a href="#">Storage Profile</a>                                                                       | 2019-06-01         |
| subscriptionId       | Azure subscription for the Virtual Machine                                                                | 2017-08-01         |

| DATA           | DESCRIPTION                                                      | VERSION INTRODUCED |
|----------------|------------------------------------------------------------------|--------------------|
| tags           | Tags for your Virtual Machine                                    | 2017-08-01         |
| tagsList       | Tags formatted as a JSON array for easier programmatic parsing   | 2019-06-04         |
| version        | Version of the VM image                                          | 2017-04-02         |
| vmlid          | Unique identifier for the VM                                     | 2017-04-02         |
| vmScaleSetName | Virtual machine scale set Name of your virtual machine scale set | 2017-12-01         |
| vmSize         | VM size                                                          | 2017-04-02         |
| zone           | Availability Zone of your virtual machine                        | 2017-12-01         |

The following Network categories are available through the Instance API:

**NOTE**

Through the metadata endpoint, the following categories are accessed through instance/network/interface

| DATA                  | DESCRIPTION                   | VERSION INTRODUCED |
|-----------------------|-------------------------------|--------------------|
| ipv4/privateIpAddress | Local IPv4 address of the VM  | 2017-04-02         |
| ipv4/publicIpAddress  | Public IPv4 address of the VM | 2017-04-02         |
| subnet/address        | Subnet address of the VM      | 2017-04-02         |
| subnet/prefix         | Subnet prefix, example 24     | 2017-04-02         |
| ipv6/ipAddress        | Local IPv6 address of the VM  | 2017-04-02         |
| macAddress            | VM mac address                | 2017-04-02         |

## Attested Data

Part of the scenario served by Instance Metadata Service is to provide guarantees that the data provided is coming from Azure. We sign part of this information so that marketplace images can be sure that it's their image running on Azure.

### Example Attested Data

**NOTE**

All API responses are JSON strings. The following example responses are pretty-printed for readability.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/attested/document?api-version=2018-10-01&nonce=1234567890"
```

Api-version is a mandatory field. Refer to the [service availability section](#) for supported API versions.Nonce is an optional 10-digit string. If not provided, IMDS returns the current UTC timestamp in its place. Due to IMDS's caching mechanism, a previously cached nonce value may be returned.

## Response

### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "encoding": "pkcs7", "signature": "MIIEGyJKoZIhvcNAQcCoIIEAzCCA/8CAQExDzANBgkqhkiG9w0BAQsFADCBugYJKoZIhvcNAQcBoIGsB1GpeyJub25jZSI6IjEyMzQ1NjY3NjYiLCJwbGFuIjp7Im5hbWUiOiiLCJwcm9kdWN0IjoiIiwicHVibGlzaGvIjoiIn0sInRpbdWtDfGftcCI6IejjcmVhdGVkT24i0iIxMs8yMC8x0CAyMjowNzozOSAtMDAwMCIsImV4cGlyZXNPbiI6IjExLzIwLzE4IDIy0jA40jI0IC0wMDAwIn0sInZtSWQi0iIifaCCaj8wggi7MIBpKADAgECAhBnxW5Kh8ds1eBA0E2mIBj0MA0GCSqGSIB3DQEBAUAMCsxKTAnBgNVBAMTIHRLc3RzdWjkB21haw4ubW0YWRhdGEuYXp1cmUuY29tMB4XDTE4MTEyMDIxNTc1N1oXDTE4MTIyMDIxNTc1N1owKzEpMCCGA1UEAxMgdGVzdhN1YmRvbWFpbis5tZXRhZGF0YS5henVyZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAML/tBo86ENWPzmXZ0kPKx5dY5QZ150mA81ommszE71x2sCLonzv4/Unk4H+jMMWRwIea2CuQ5RhdWAhvKq6if4okKnt66fxm+YT vz9z0CTfCLmlT+nsdf0AsG1xZppEapC0cd9vDGNCKyE8ayI1pliaeOnFjG0lwvMY04uWz2MdAgMBAAGjYDBeMFwGA1UdAQRVMFOAENyKHLa04Ut4Mp7TkJFFyhLTArMSkwJwYDVQQDEyB0ZXN0c3ViZG9tYWluLm1ldGFkYXRhLmF6dXJlLmNvbYIQZ8VuSoFhbJRAQNBnpiAsdDANBgkqhkiG9w0BAQFQAObgQCLSM6aX5Bs1KHCJp4VQtzXPzXF71rVKCocHy3N9PTJQ9Fpnd+bYw2vSpQHg/Aig82WuDFpPreJvr7Pa938mZq9WuOGjQKK2FYDTg6fXD8pkPdygh1X5boGWAMMr7bFkup+lsT+n2tRw2wbNkn01tQ0wICtqy2VqzWwLi45RBwTGB6DCB5QIBATA/MCsxKTAnBgNVBAMTIHRLc3RzdWjkB21haw4ubW0YWRhdGEuYXp1cmUuY29tAhBnxW5Kh8ds1eBA0E2mIBj0MA0GCSqGSIB3DQEBCwUAMA0GCSqGSIB3DQEBAQUABIGA1d1BM/yYIqqv8SDE4kjQo3U1/IKAVR8ETKcve5BAdGSNkTUooUGVniTXeuvdj5Nkmaz0aKzp9fEtByqqPOyw/n1XaZg0044HDGiPUJ90xVYmfek6p9RpJBu6kiKhnnYTeluk5u75phe5ZbfBhuPhXmYAdjc7Nmw97nx8NnprQ="}
```

The signature blob is a [pkcs7](#) signed version of document. It contains the certificate used for signing along with the VM details like vmlid, sku, nonce, subscriptionId, timeStamp for creation and expiry of the document and the plan information about the image. The plan information is only populated for Azure Market place images. The certificate can be extracted from the response and used to validate that the response is valid and is coming from Azure.

## Retrieving attested metadata in Windows Virtual Machine

### Request

Instance metadata can be retrieved in Windows via the PowerShell utility `curl`:

```
curl -H @{"Metadata"='true'} "http://169.254.169.254/metadata/attested/document?api-version=2018-10-01&nonce=1234567890" | select -ExpandProperty Content
```

Or through the `Invoke-RestMethod` cmdlet:

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -URI "http://169.254.169.254/metadata/attested/document?api-version=2018-10-01&nonce=1234567890" -Method get
```

Api-version is a mandatory field. Refer to the service availability section for supported API versions.Nonce is an optional 10-digit string. If not provided, IMDS returns the current UTC timestamp in its place. Due to IMDS's caching mechanism, a previously cached nonce value may be returned.

## Response

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "encoding": "pkcs7", "signature": "MIEEgYJKoZIhvcNAQcCoIIAzCCA/8CAQExDzANBgkqhkiG9w0BAQsFADCBugYJKoZIhvcNAQcBoIGsB1GpeyJub25jZSI6IjEyMzQ1NjY3NjYiLCJwbGFuIjp7Im5hbWUiOiiLCJwcm9kdWN0IjoiIiwcHvibGlzaGVyIjoiIn0sInRpbtWTdGFTcCI6eyJjcmVhdGVkT24i0iIxMS8xOCAYmjowNzozOSAtMDAwMCIsImV4cGlyZXNPbiI6IjExLzIwLzE4IDIy0jA40jI0IC0wMDAwIn0sInZtSWQiOiiifaCCAj8wggi7MIBpKADAgECAhBnxW5Kh8ds1EBA0E2mIBJ0MA0GCSqGSIB3DQEBAUAMCsxKTAnBgNVBAMTIHR1c3RzdWJkb21haw4ubW0YWRhdGEuYXp1cmUuY29tMB4XDTE4MTExNTc1N1oXDE4MTIyMDIxNTc1N1lowKzEpMccGA1UEAxMgdGVzdHN1YmRvbWFpbisTZXRhZGF0YS5henVyZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAML/tBo86ENWPzmXZ0kPKX5dY5QZ150mA81ommszE71x2sClonzv4/UWk4H+jMMWRwIea2CuQ5RhdWAhvKq6if4okKNt66fxm+YTVz9z0CTfCLmLT+nsdf0AsG1xZppEapC0Cd9vD6NCKyE8aYI1pliaeOnFjG0WvMY04uWz2MdAgMBAAGjYDBeMFwGA1UdAQRMFOAENnYkHLa04Ut4Mpt7TkJFFyhLTArMSkwJwYDVQQDEyB0ZXN0c3ViZG9tYWluLm1ldGFkYXRhLmF6dXJ1LmNvbYIQZ8VuSoFhbJRAQNBnpiaSdDANBgkqhkiG9w0BAQQFAAOBgQCLSM6aX5Bs1KHCJp4VQtxZPzXF71rVKCocHy3N9PTJQ9Fpnd+bYw2vSpQHg/Aig82WuDFpPreJvr7Pa938mZq9HUOGjQKK2FYDTg6fXD8pkPdygh1X5boGWAMMrF7bFkup+lsT+n2tRw2wbNkn01tQ0WIctqy2VqzWwLi45RBwTGB6DCB5QIBATA/MCsxKTAnBgNVBAMTIHR1c3RzdWJkb21haW4ubW0YWRhdGEuYXp1cmUuY29tAhBnxW5Kh8ds1EBA0E2mIBJ0MA0GCSqGSIB3DQEBCwUAMA0GCSqGSIB3DQEBAQUABIGA1d1BM/yYIqqv8SDE4kjQo3U1/IKAJR8ETKcve5BAdGSNKTUooUGVniTXeuvdj5NKmaz0aKZp9fEtByqqPOyw/n1XaZg0044HDGiPUJ90xVYmfefK6p9RpJBu6kiKhnnYTeluk5u75phe5ZbMZfBhuPhXmYAdjc7Nmw97nx8Nnpr="}
}
```

The signature blob is a [pkcs7](#) signed version of document. It contains the certificate used for signing along with the VM details like vmId, sku, nonce, subscriptionId, timeStamp for creation and expiry of the document and the plan information about the image. The plan information is only populated for Azure Market place images. The certificate can be extracted from the response and used to validate that the response is valid and is coming from Azure.

## Example scenarios for usage

### Tracking VM running on Azure

As a service provider, you may require to track the number of VMs running your software or have agents that need to track uniqueness of the VM. To be able to get a unique ID for a VM, use the `vmId` field from Instance Metadata Service.

#### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/vmId?api-version=2017-08-01&format=text"
```

#### Response

```
5c08b38e-4d57-4c23-ac45-aca61037f084
```

### Placement of containers, data-partitions based fault/update domain

For certain scenarios, placement of different data replicas is of prime importance. For example, [HDFS replica placement](#) or container placement via an [orchestrator](#) may you require to know the `platformFaultDomain` and `platformUpdateDomain` the VM is running on. You can also use [Availability Zones](#) for the instances to make these decisions. You can query this data directly via the Instance Metadata Service.

#### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/platformFaultDomain?api-version=2017-08-01&format=text"
```

#### Response

## Getting more information about the VM during support case

As a service provider, you may get a support call where you would like to know more information about the VM. Asking the customer to share the compute metadata can provide basic information for the support professional to know about the kind of VM on Azure.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute?api-version=2019-06-01"
```

### Response

#### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "azEnvironment": "AzurePublicCloud",
 "customData": "",
 "location": "centralus",
 "name": "negasonic",
 "offer": "lampstack",
 "osType": "Linux",
 "placementGroupId": "",
 "plan": {
 "name": "5-6",
 "product": "lampstack",
 "publisher": "bitnami"
 },
 "platformFaultDomain": "0",
 "platformUpdateDomain": "0",
 "provider": "Microsoft.Compute",
 "publicKeys": [],
 "publisher": "bitnami",
 "resourceGroupName": "myrg",
 "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxx-
xxxxxxxx/resourceGroups/myrg/providers/Microsoft.Compute/virtualMachines/negasonic",
 "sku": "5-6",
 "storageProfile": {
 "dataDisks": [
 {
 "caching": "None",
 "createOption": "Empty",
 "diskSizeGB": "1024",
 "image": {
 "uri": ""
 },
 "lun": "0",
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampledataldiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampledataldiskname",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
]
 }
}
```

```

],
 "imageReference": {
 "id": "",
 "offer": "UbuntuServer",
 "publisher": "Canonical",
 "sku": "16.04.0-LTS",
 "version": "latest"
 },
 "osDisk": {
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "diskSizeGB": "30",
 "diffDiskSettings": {
 "option": "Local"
 },
 "encryptionSettings": {
 "enabled": "false"
 },
 "image": {
 "uri": ""
 },
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-23/providers/Microsoft.Compute/disks/exampleosdiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampleosdiskname",
 "osType": "Linux",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
 },
 "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
 "tags": "Department:IT;Environment:Test;Role:WebRole",
 "version": "7.1.1902271506",
 "vmId": "13f56399-bd52-4150-9748-7190aae1ff21",
 "vmScaleSetName": "",
 "vmSize": "Standard_A1_v2",
 "zone": "1"
}

```

## Getting Azure Environment where the VM is running

Azure has various sovereign clouds like [Azure Government](#). Sometimes you need the Azure Environment to make some runtime decisions. The following sample shows you how you can achieve this behavior.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/azEnvironment?api-version=2018-10-01&format=text"
```

### Response

```
AzurePublicCloud
```

The regions and the values of the Azure Environment are listed below.

| REGIONS                                      | AZURE ENVIRONMENT |
|----------------------------------------------|-------------------|
| All Generally Available Global Azure Regions | AzurePublicCloud  |

| REGIONS              | AZURE ENVIRONMENT      |
|----------------------|------------------------|
| Azure Government     | AzureUSGovernmentCloud |
| Azure China 21Vianet | AzureChinaCloud        |
| Azure Germany        | AzureGermanCloud       |

## Getting the tags for the VM

Tags may have been applied to your Azure VM to logically organize them into a taxonomy. The tags assigned to a VM can be retrieved by using the request below.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/tags?api-version=2018-10-01&format=text"
```

### Response

```
Department:IT;Environment:Test;Role:WebRole
```

The `tags` field is a string with the tags delimited by semicolons. This can be a problem if semicolons are used in the tags themselves. If a parser is written to programmatically extract the tags, you should rely on the `tagsList` field which is a JSON array with no delimiters, and consequently, easier to parse.

### Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/tagsList?api-version=2019-06-04&format=JSON"
```

### Response

```
[
 {
 "name": "Department",
 "value": "IT"
 },
 {
 "name": "Environment",
 "value": "Test"
 },
 {
 "name": "Role",
 "value": "WebRole"
 }
]
```

## Validating that the VM is running in Azure

Marketplace vendors want to ensure that their software is licensed to run only in Azure. If someone copies the VHD out to on-premises, then they should have the ability to detect that. By calling into Instance Metadata Service, Marketplace vendors can get signed data that guarantees response only from Azure.

## NOTE

Requires jq to be installed.

## Request

```
Get the signature
curl --silent -H Metadata:True http://169.254.169.254/metadata/attested/document?api-version=2019-04-30 |
jq -r '.["signature"]' > signature
Decode the signature
base64 -d signature > decodedsignature
#Get PKCS7 format
openssl pkcs7 -in decodedsignature -inform DER -out sign.pk7
Get Public key out of pkc7
openssl pkcs7 -in decodedsignature -inform DER -print_certs -out signer.pem
#Get the intermediate certificate
wget -q -O intermediate.cer "$(openssl x509 -in signer.pem -text -noout | grep " CA Issuers -" | awk -FURI:
'{print $2}')"
openssl x509 -inform der -in intermediate.cer -out intermediate.pem
#Verify the contents
openssl smime -verify -in sign.pk7 -inform pem -noverify
```

## Response

```
Verification successful
{
 "nonce": "20181128-001617",
 "plan": {
 "name": "",
 "product": "",
 "publisher": ""
 },
 "timeStamp": {
 "createdOn": "11/28/18 00:16:17 -0000",
 "expiresOn": "11/28/18 06:16:17 -0000"
 },
 "vmId": "d3e0e374-fda6-4649-bbc9-7f20dc379f34",
 "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
 "sku": "RS3-Pro"
}
```

| DATA                | DESCRIPTION                                                                                                                    |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------|
| nonce               | User supplied optional string with the request. If no nonce was supplied in the request, the current UTC timestamp is returned |
| plan                | <a href="#">Plan</a> for a VM in it's an Azure Marketplace Image, contains name, product, and publisher                        |
| timestamp/createdOn | The UTC timestamp at which the first signed document was created                                                               |
| timestamp/expiresOn | The UTC timestamp at which the signed document expires                                                                         |
| vmid                | <a href="#">Unique identifier</a> for the VM                                                                                   |

| DATA           | DESCRIPTION                                                             |
|----------------|-------------------------------------------------------------------------|
| subscriptionId | Azure subscription for the Virtual Machine, introduced in<br>2019-04-30 |
| sku            | Specific SKU for the VM image, introduced in 2019-11-01                 |

#### Verifying the signature

Once you get the signature above, you can verify that the signature is from Microsoft. Also you can verify the intermediate certificate and the certificate chain. Lastly, you can verify the subscription ID is correct.

#### NOTE

The certificate for Public cloud and sovereign cloud will be different.

| CLOUD                                        | CERTIFICATE                  |
|----------------------------------------------|------------------------------|
| All Generally Available Global Azure Regions | *.metadata.azure.com         |
| Azure Government                             | *.metadata.azure.us          |
| Azure China 21Vianet                         | *.metadata.azure.cn          |
| Azure Germany                                | *.metadata.microsoftazure.de |

There is a known issue around the certificate used for signing. The certificates may not have an exact match of `metadata.azure.com` for public cloud. Hence the certification validation should allow a common name from any `.metadata.azure.com` subdomain.

```
Verify the subject name for the main certificate
openssl x509 -noout -subject -in signer.pem
Verify the issuer for the main certificate
openssl x509 -noout -issuer -in signer.pem
#Validate the subject name for intermediate certificate
openssl x509 -noout -subject -in intermediate.pem
Verify the issuer for the intermediate certificate
openssl x509 -noout -issuer -in intermediate.pem
Verify the certificate chain
openssl verify -verbose -CAfile /etc/ssl/certs/Baltimore_CyberTrust_Root.pem -untrusted intermediate.pem
signer.pem
```

In cases where the intermediate certificate cannot be downloaded due to network constraints during validation, the intermediate certificate can be pinned. However, Azure will roll over the certificates as per standard PKI practice. The pinned certificates would need to be updated when roll over happens. Whenever a change to update the intermediate certificate is planned, the Azure blog will be updated and Azure customers will be notified. The intermediate certificates can be found [here](#). The intermediate certificates for each of the regions can be different.

#### Failover Clustering in Windows Server

For certain scenarios, when querying Instance Metadata Service with Failover Clustering, it is necessary to add a route to the routing table.

1. Open command prompt with administrator privileges.
2. Run the following command and note the address of the Interface for Network Destination (`0.0.0.0`) in the

## IPv4 Route Table.

```
route print
```

### NOTE

The following example output from a Windows Server VM with Failover Cluster enabled contains only the IPv4 Route Table for simplicity.

### IPv4 Route Table

| Active Routes:  |                 |         |               |           |        |
|-----------------|-----------------|---------|---------------|-----------|--------|
| Network         | Destination     | Netmask | Gateway       | Interface | Metric |
|                 | 0.0.0.0         | 0.0.0.0 | 10.0.1.1      | 10.0.1.10 | 266    |
| 10.0.1.0        | 255.255.255.192 | On-link | 10.0.1.10     | 266       |        |
| 10.0.1.10       | 255.255.255.255 | On-link | 10.0.1.10     | 266       |        |
| 10.0.1.15       | 255.255.255.255 | On-link | 10.0.1.10     | 266       |        |
| 10.0.1.63       | 255.255.255.255 | On-link | 10.0.1.10     | 266       |        |
| 127.0.0.0       | 255.0.0.0       | On-link | 127.0.0.1     | 331       |        |
| 127.0.0.1       | 255.255.255.255 | On-link | 127.0.0.1     | 331       |        |
| 127.255.255.255 | 255.255.255.255 | On-link | 127.0.0.1     | 331       |        |
| 169.254.0.0     | 255.255.0.0     | On-link | 169.254.1.156 | 271       |        |
| 169.254.1.156   | 255.255.255.255 | On-link | 169.254.1.156 | 271       |        |
| 169.254.255.255 | 255.255.255.255 | On-link | 169.254.1.156 | 271       |        |
| 224.0.0.0       | 240.0.0.0       | On-link | 127.0.0.1     | 331       |        |
| 224.0.0.0       | 240.0.0.0       | On-link | 169.254.1.156 | 271       |        |
| 224.0.0.0       | 240.0.0.0       | On-link | 10.0.1.10     | 266       |        |
| 255.255.255.255 | 255.255.255.255 | On-link | 127.0.0.1     | 331       |        |
| 255.255.255.255 | 255.255.255.255 | On-link | 169.254.1.156 | 271       |        |
| 255.255.255.255 | 255.255.255.255 | On-link | 10.0.1.10     | 266       |        |

- Run the following command and use the address of the Interface for Network Destination ( `0.0.0.0` ) which is ( `10.0.1.10` ) in this example.

```
route add 169.254.169.254/32 10.0.1.10 metric 1 -p
```

### Storage profile

Instance Metadata Service can provide details about the storage disks associated with the VM. This data can be found at the instance/compute/storageProfile endpoint.

The storage profile of a VM is divided into three categories - image reference, OS disk, and data disks.

The image reference object contains the following information about the OS image:

| DATA      | DESCRIPTION                                  |
|-----------|----------------------------------------------|
| id        | Resource ID                                  |
| offer     | Offer of the platform or marketplace image   |
| publisher | Image publisher                              |
| sku       | Image sku                                    |
| version   | Version of the platform or marketplace image |

The OS disk object contains the following information about the OS disk used by the VM:

| DATA                    | DESCRIPTION                                            |
|-------------------------|--------------------------------------------------------|
| caching                 | Caching requirements                                   |
| createOption            | Information about how the VM was created               |
| diffDiskSettings        | Ephemeral disk settings                                |
| diskSizeGB              | Size of the disk in GB                                 |
| image                   | Source user image virtual hard disk                    |
| lun                     | Logical unit number of the disk                        |
| managedDisk             | Managed disk parameters                                |
| name                    | Disk name                                              |
| vhd                     | Virtual hard disk                                      |
| writeAcceleratorEnabled | Whether or not writeAccelerator is enabled on the disk |

The data disks array contains a list of data disks attached to the VM. Each data disk object contains the following information:

| DATA                    | DESCRIPTION                                            |
|-------------------------|--------------------------------------------------------|
| caching                 | Caching requirements                                   |
| createOption            | Information about how the VM was created               |
| diffDiskSettings        | Ephemeral disk settings                                |
| diskSizeGB              | Size of the disk in GB                                 |
| encryptionSettings      | Encryption settings for the disk                       |
| image                   | Source user image virtual hard disk                    |
| managedDisk             | Managed disk parameters                                |
| name                    | Disk name                                              |
| osType                  | Type of OS included in the disk                        |
| vhd                     | Virtual hard disk                                      |
| writeAcceleratorEnabled | Whether or not writeAccelerator is enabled on the disk |

The following is an example of how to query the VM's storage information.

## Request

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance/compute/storageProfile?api-version=2019-06-01"
```

## Response

### NOTE

The response is a JSON string. The following example response is pretty-printed for readability.

```
{
 "dataDisks": [
 {
 "caching": "None",
 "createOption": "Empty",
 "diskSizeGB": "1024",
 "image": {
 "uri": ""
 },
 "lun": "0",
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-23/providers/Microsoft.Compute/disks/exampledatadiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampledatadiskname",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
],
 "imageReference": {
 "id": "",
 "offer": "UbuntuServer",
 "publisher": "Canonical",
 "sku": "16.04.0-LTS",
 "version": "latest"
 },
 "osDisk": {
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "diskSizeGB": "30",
 "diffDiskSettings": {
 "option": "Local"
 },
 "encryptionSettings": {
 "enabled": "false"
 },
 "image": {
 "uri": ""
 },
 "managedDisk": {
 "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourceGroups/macikgo-test-may-23/providers/Microsoft.Compute/disks/exampleosdiskname",
 "storageAccountType": "Standard_LRS"
 },
 "name": "exampleosdiskname",
 "osType": "Linux",
 "vhd": {
 "uri": ""
 },
 "writeAcceleratorEnabled": "false"
 }
}
}
```

## Examples of calling metadata service using different languages inside the VM

| LANGUAGE | EXAMPLE                                                                                                                                         |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Ruby     | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.rb">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.rb</a> |

| LANGUAGE     | EXAMPLE                                                                                                                                                           |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Go           | <a href="https://github.com/Microsoft/azureimds/blob/master/imdssample.go">https://github.com/Microsoft/azureimds/blob/master/imdssample.go</a>                   |
| Python       | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.py">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.py</a>                   |
| C++          | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample-windows.cpp">https://github.com/Microsoft/azureimds/blob/master/IMDSSample-windows.cpp</a> |
| C#           | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.cs">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.cs</a>                   |
| JavaScript   | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.js">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.js</a>                   |
| PowerShell   | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.ps1">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.ps1</a>                 |
| Bash         | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.sh">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.sh</a>                   |
| Perl         | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.pl">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.pl</a>                   |
| Java         | <a href="https://github.com/Microsoft/azureimds/blob/master/imdssample.java">https://github.com/Microsoft/azureimds/blob/master/imdssample.java</a>               |
| Visual Basic | <a href="https://github.com/Microsoft/azureimds/blob/master/IMDSSample.vb">https://github.com/Microsoft/azureimds/blob/master/IMDSSample.vb</a>                   |
| Puppet       | <a href="https://github.com/keirans/azuremetadata">https://github.com/keirans/azuremetadata</a>                                                                   |

## FAQ

- I am getting the error `400 Bad Request, Required metadata header not specified`. What does this mean?
  - The Instance Metadata Service requires the header `Metadata: true` to be passed in the request. Passing this header in the REST call allows access to the Instance Metadata Service.
- Why am I not getting compute information for my VM?
  - Currently the Instance Metadata Service only supports instances created with Azure Resource Manager. In the future, support for Cloud Service VMs might be added.
- I created my Virtual Machine through Azure Resource Manager a while back. Why am I not see compute metadata information?
  - For any VMs created after Sep 2016, add a [Tag](#) to start seeing compute metadata. For older VMs (created before Sep 2016), add/remove extensions or data disks to the VM to refresh metadata.
- I am not seeing all data populated for new version
  - For any VMs created after Sep 2016, add a [Tag](#) to start seeing compute metadata. For older VMs (created before Sep 2016), add/remove extensions or data disks to the VM to refresh metadata.
- Why am I getting the error `500 Internal Server Error`?

- Retry your request based on exponential back off system. If the issue persists contact Azure support.
6. Where do I share additional questions/comments?
    - Send your comments on <https://feedback.azure.com>.
  7. Would this work for Virtual Machine Scale Set Instance?
    - Yes Metadata service is available for Scale Set Instances.
  8. How do I get support for the service?
    - To get support for the service, create a support issue in Azure portal for the VM where you are not able to get metadata response after long retries.
  9. I get request timed out for my call to the service?
    - Metadata calls must be made from the primary IP address assigned to the primary network card of the VM, in addition in case you have changed your routes there must be a route for 169.254.0.0/16 address out of your network card.
  10. I updated my tags in virtual machine scale set but they don't appear in the instances unlike VMs?
    - Currently for ScaleSets tags only show to the VM on a reboot/reimage/or a disk change to the instance.

# Problem

NEW SUPPORT REQUEST



\* Severity ⓘ

C - Minimal impact



\* Problem type

Management



\* Category

✓ Choose a category

Backup

Cannot stop, start, or restart a VM

Capacity issues that are related to SAP HANA large instances

Instance Metadata Service

Manage an Exchange Server

Manage an instance of SQL Server

Manage encrypted disks, keys or secrets, or permissions

Manage or use RDS in Azure

Manage or use a VPN

Manage or use a cluster in Azure

Manage or use a virtual network

Manage or use endpoints

Unable to delete a virtual machine

Virtual machine restarts

When did the problem start?

Choose a date



Enter a local time

File upload ⓘ

Select a file



Share diagnostic information ⓘ

[Learn more about the information we collect](#)

**Next**

## Next steps

- Learn more about [Scheduled Events](#)

# How to enable nested virtualization in an Azure VM

11/13/2019 • 6 minutes to read • [Edit Online](#)

Nested virtualization is supported in several Azure virtual machine families. This capability provides great flexibility in supporting scenarios such as development, testing, training, and demonstration environments.

This article steps through enabling Hyper-V on an Azure VM and configuring Internet connectivity to that guest virtual machine.

## Create a nesting capable Azure VM

Create a new Windows Server 2016 Azure VM. For quick reference, all v3 virtual machines support nested virtualization. For a complete list of virtual machine sizes that support nesting, check out the [Azure Compute Unit article](#).

Remember to choose a VM size large enough to support the demands of a guest virtual machine. In this example, we are using a D3\_v3 size Azure VM.

You can view the regional availability of Dv3 or Ev3 series virtual machines [here](#).

### NOTE

For detailed instructions on creating a new virtual machine, see [Create and Manage Windows VMs with the Azure PowerShell module](#)

## Connect to your Azure VM

Create a remote desktop connection to the virtual machine.

1. Click the **Connect** button on the virtual machine properties. A Remote Desktop Protocol file (.rdp file) is created and downloaded.
2. To connect to your VM, open the downloaded RDP file. If prompted, click **Connect**. On a Mac, you need an RDP client such as this [Remote Desktop Client](#) from the Mac App Store.
3. Enter the user name and password you specified when creating the virtual machine, then click **Ok**.
4. You may receive a certificate warning during the sign-in process. Click **Yes** or **Continue** to proceed with the connection.

## Enable the Hyper-V feature on the Azure VM

You can configure these settings manually or we have provided a PowerShell script to automate the configuration.

### Option 1: Use a PowerShell script to configure nested virtualization

A PowerShell script to enable nested virtualization on a Windows Server 2016 host is available on [GitHub](#). The script checks pre-requisites and then configures nested virtualization on the Azure VM. A restart of the Azure VM is necessary to complete the configuration. This script may work in other environments but is not guaranteed.

Check out the Azure blog post with a live video demonstration on nested virtualization running on Azure!

<https://aka.ms/AzureNVblog>.

### Option 2: Configure nested virtualization manually

1. On the Azure VM, open PowerShell as an Administrator.
2. Enable the Hyper-V feature and Management Tools.

```
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart
```

**WARNING**

This command restarts the Azure VM. You will lose your RDP connection during the restart process.

3. After the Azure VM restarts, reconnect to your VM using RDP.

## Set up internet connectivity for the guest virtual machine

Create a new virtual network adapter for the guest virtual machine and configure a NAT Gateway to enable Internet connectivity.

### Create a NAT virtual network switch

1. On the Azure VM, open PowerShell as an Administrator.
2. Create an internal switch.

```
New-VMSwitch -Name "InternalNAT" -SwitchType Internal
```

3. View the properties of the switch and note the ifIndex for the new adapter.

```
Get-NetAdapter
```

| Administrator: Windows PowerShell |                                      |         |        |                   |           |
|-----------------------------------|--------------------------------------|---------|--------|-------------------|-----------|
| PS C:\Users\Nimda> Get-NetAdapter |                                      |         |        |                   |           |
| Name                              | InterfaceDescription                 | ifIndex | Status | MacAddress        | LinkSpeed |
| vEthernet (InternalNAT)           | Hyper-V Virtual Ethernet Adapter     | 13      | Up     | 00-15-5D-01-04-00 | 10 Gbps   |
| Ethernet 3                        | Microsoft Hyper-V Network Adapter #3 | 4       | Up     | 00-0D-3A-F9-AC-5A | 40 Gbps   |

**NOTE**

Take note of the "ifIndex" for the virtual switch you just created.

4. Create an IP address for the NAT Gateway.

In order to configure the gateway, you need some information about your network:

- IPAddress - The NAT Gateway IP specifies the IPv4 or IPv6 address to use as the default gateway address for the virtual network subnet. The generic form is a.b.c.1 (for example, "192.168.0.1"). While the final position doesn't have to be .1, it usually is (based on prefix length). Typically you should use an RFC 1918 private network address space.
- PrefixLength - The subnet prefix length defines the local subnet size (subnet mask). The subnet prefix length will be an integer value between 0 and 32. 0 would map the entire internet, 32 would only allow one mapped IP. Common values range from 24 to 12 depending on how many IPs need to be attached to the

NAT. A common PrefixLength is 24 -- this is a subnet mask of 255.255.255.0.

- InterfaceIndex - **ifIndex** is the interface index of the virtual switch created in the previous step.

```
New-NetIPAddress -IPAddress 192.168.0.1 -PrefixLength 24 -InterfaceIndex 13
```

## Create the NAT network

In order to configure the gateway, you will need to provide information about the network and NAT Gateway:

- Name - This is the name of the NAT network.
- InternalIPInterfaceAddressPrefix - The NAT subnet prefix describes both the NAT Gateway IP prefix from above as well as the NAT Subnet Prefix Length from above. The generic form will be a.b.c.0/NAT Subnet Prefix Length.

In PowerShell, create a new NAT network.

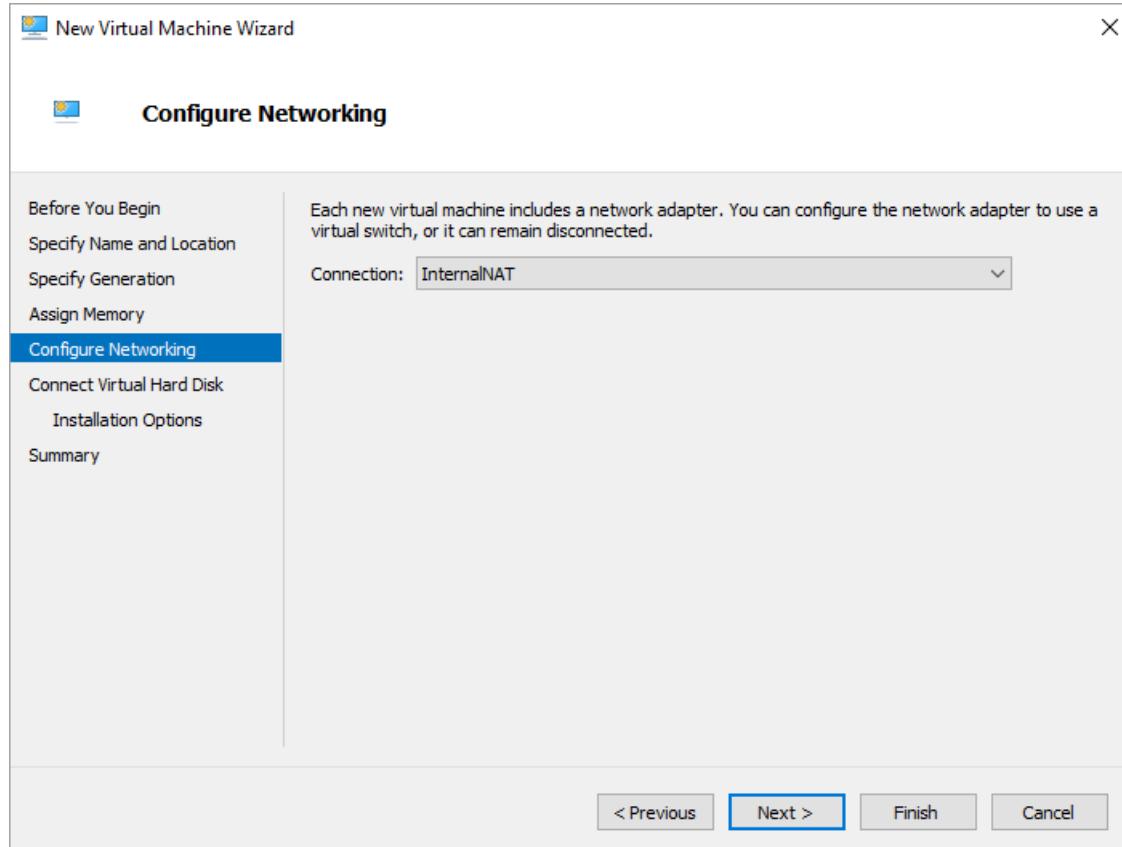
```
New-NetNat -Name "InternalNat" -InternalIPInterfaceAddressPrefix 192.168.0.0/24
```

## Create the guest virtual machine

### IMPORTANT

The Azure guest agent is not supported on nested VMs, and may cause issues on both the host and nested VMs. Don't install the Azure agent on nested VMs, and don't use an image for creating the nested VMs that already has the Azure guest agent installed.

1. Open Hyper-V Manager and create a new virtual machine. Configure the virtual machine to use the new Internal network you created.



2. Install an operating system on the guest virtual machine.

#### **NOTE**

You need installation media for an operating system to install on the VM. In this case we are using Windows 10 Enterprise.

## Assign an IP address to the guest virtual machine

You can assign an IP address to the guest virtual machine either by manually setting a static IP address on the guest virtual machine or configuring DHCP on the Azure VM to assign the IP address dynamically.

### **Option 1: Configure DHCP to dynamically assign an IP address to the guest virtual machine**

Follow the steps below to configure DHCP on the host virtual machine for dynamic address assignment.

#### **Install DHCP Server on the Azure VM**

1. Open Server Manager. On the Dashboard, click **Add roles and features**. The Add Roles and Features Wizard appears.
2. In wizard, click **Next** until the Server Roles page.
3. Click to select the **DHCP Server** checkbox, click **Add Features**, and then click **Next** until you complete the wizard.
4. Click **Install**.

#### **Configure a new DHCP scope**

1. Open DHCP Manager.
2. In the navigation pane, expand the server name, right-click **IPv4**, and click **New Scope**. The New Scope Wizard appears, click **Next**.
3. Enter a Name and Description for the scope and click **Next**.
4. Define an IP Range for your DCHP Server (for example, 192.168.0.100 to 192.168.0.200).
5. Click **Next** until the Default Gateway page. Enter the IP Address you created earlier (for example, 192.168.0.1) as the Default Gateway, then click **Add**.
6. Click **Next** until the wizard completes, leaving all default values, then click **Finish**.

### **Option 2: Manually set a static IP address on the guest virtual machine**

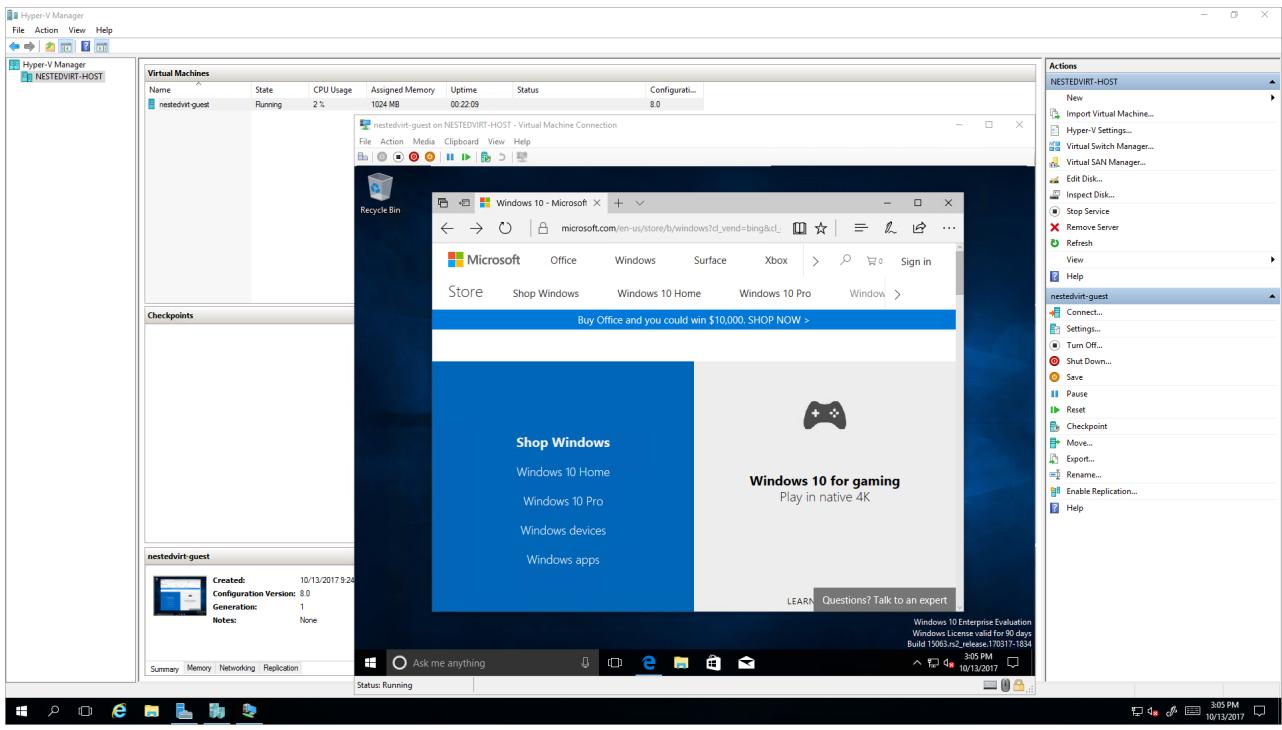
If you did not configure DHCP to dynamically assign an IP address to the guest virtual machine, follow these steps to set a static IP address.

1. On the Azure VM, open PowerShell as an Administrator.
2. Right-click the guest virtual machine and click Connect.
3. Sign in to the guest virtual machine.
4. On the guest virtual machine, open the Network and Sharing Center.
5. Configure the network adapter for an address within the range of the NAT network you created in the previous section.

In this example you will use an address in the 192.168.0.0/24 range.

## Test connectivity in guest virtual machine

In the guest virtual machine, open your browser and navigate to a web page.



## Set up intranet connectivity for the guest virtual machine

For instructions on how to enable transparent connectivity between Guest VMs and Azure VMs, please reference [this document](#).

# Handling planned maintenance notifications

2/10/2020 • 7 minutes to read • [Edit Online](#)

Azure periodically performs updates to improve the reliability, performance, and security of the host infrastructure for virtual machines. Updates are changes like patching the hosting environment or upgrading and decommissioning hardware. A majority of these updates are completed without any impact to the hosted virtual machines. However, there are cases where updates do have an impact:

- If the maintenance does not require a reboot, Azure uses in-place migration to pause the VM while the host is updated. These types maintenance operations are applied fault domain by fault domain. Progress is stopped if any warning health signals are received.
- If maintenance requires a reboot, you get a notice of when the maintenance is planned. You are given a time window of about 35 days where you can start the maintenance yourself, when it works for you.

Planned maintenance that requires a reboot is scheduled in waves. Each wave has different scope (regions).

- A wave starts with a notification to customers. By default, notification is sent to the Service Administrator and Co-Administrators. You can add more recipients and messaging options like email, SMS, and webhooks, using [Activity Log Alerts](#).
- Once a notification goes out, a *self-service window* is made available. During this window, you can query which of your virtual machines are affected and start maintenance based on your own scheduling needs. The self-service window is typically about 35 days.
- After the self-service window, a *scheduled maintenance window* begins. At some point during this window, Azure schedules and applies the required maintenance to your virtual machine.

The goal in having two windows is to give you enough time to start maintenance and reboot your virtual machine while knowing when Azure will automatically start maintenance.

You can use the Azure portal, PowerShell, REST API, and CLI to query for the maintenance windows for your VMs and start self-service maintenance.

## Should you start maintenance using during the self-service window?

The following guidelines should help you decide whether to use this capability and start maintenance at your own time.

### NOTE

Self-service maintenance might not be available for all of your VMs. To determine if proactive redeploy is available for your VM, look for the **Start now** in the maintenance status. Self-service maintenance is currently not available for Cloud Services (Web/Worker Role) and Service Fabric.

Self-service maintenance is not recommended for deployments using **availability sets**. Availability sets are already only updated one update domain at a time.

- Let Azure trigger the maintenance. For maintenance that requires reboot, maintenance will be done update domain by update domain. The update domains do not necessarily receive the maintenance sequentially, and that there is a 30-minute pause between update domains.
- If a temporary loss of some capacity (1 update domain) is a concern, you can add instances during the maintenance period.

- For maintenance that does not require reboot, updates are applied at the fault domain level.

**Don't** use self-service maintenance in the following scenarios:

- If you shut down your VMs frequently, either manually, using DevTest Labs, using auto-shutdown, or following a schedule, it could revert the maintenance status and therefore cause additional downtime.
- On short-lived VMs that you know will be deleted before the end of the maintenance wave.
- For workloads with a large state stored in the local (ephemeral) disk that is desired to be maintained upon update.
- For cases where you resize your VM often, as it could revert the maintenance status.
- If you have adopted scheduled events that enable proactive failover or graceful shutdown of your workload, 15 minutes before start of maintenance shutdown

**Use** self-service maintenance, if you are planning to run your VM uninterrupted during the scheduled maintenance phase and none of the counter-indications mentioned above are applicable.

It is best to use self-service maintenance in the following cases:

- You need to communicate an exact maintenance window to your management or end-customer.
- You need to complete the maintenance by a given date.
- You need to control the sequence of maintenance, for example, multi-tier application to guarantee safe recovery.
- More than 30 minutes of VM recovery time is needed between two update domains (UDs). To control the time between update domains, you must trigger maintenance on your VMs one update domain (UD) at a time.

## FAQ

### **Q: Why do you need to reboot my virtual machines now?**

**A:** While the majority of updates and upgrades to the Azure platform do not impact virtual machine's availability, there are cases where we can't avoid rebooting virtual machines hosted in Azure. We have accumulated several changes that require us to restart our servers that will result in virtual machines reboot.

### **Q: If I follow your recommendations for High Availability by using an Availability Set, am I safe?**

**A:** Virtual machines deployed in an availability set or virtual machine scale sets have the notion of Update Domains (UD). When performing maintenance, Azure honors the UD constraint and will not reboot virtual machines from different UD (within the same availability set). Azure also waits for at least 30 minutes before moving to the next group of virtual machines.

For more information about high availability, see [Availability for virtual machines in Azure](#).

### **Q: How do I get notified about planned maintenance?**

**A:** A planned maintenance wave starts by setting a schedule to one or more Azure regions. Soon after, an email notification is sent to the subscription Admins (one email per subscription). Additional channels and recipients for this notification could be configured using Activity Log Alerts. In case you deploy a virtual machine to a region where planned maintenance is already scheduled, you will not receive the notification but rather need to check the maintenance state of the VM.

### **Q: I don't see any indication of planned maintenance in the portal, Powershell, or CLI. What is wrong?**

**A:** Information related to planned maintenance is available during a planned maintenance wave only for the VMs that are going to be impacted by it. In other words, if you see no data, it could be that the maintenance wave has already completed (or not started) or that your virtual machine is already hosted in an updated server.

### **Q: Is there a way to know exactly when my virtual machine will be impacted?**

**A:** When setting the schedule, we define a time window of several days. However, the exact sequencing of servers (and VMs) within this window is unknown. Customers who would like to know the exact time for their VMs can use [scheduled events](#) and query from within the virtual machine and receive a 15-minute notification before a VM reboot.

**Q: How long will it take you to reboot my virtual machine?**

**A:** Depending on the size of your VM, reboot may take up to several minutes during the self-service maintenance window. During the Azure initiated reboots in the scheduled maintenance window, the reboot will typically take about 25 minutes. Note that in case you use Cloud Services (Web/Worker Role), Virtual Machine Scale Sets, or availability sets, you will be given 30 minutes between each group of VMs (UD) during the scheduled maintenance window.

**Q: What is the experience in the case of Virtual Machine Scale Sets?**

**A:** Planned maintenance is now available for Virtual Machine Scale Sets. For instructions on how to initiate self-service maintenance refer [planned maintenance for virtual machine scale sets](#) document.

**Q: What is the experience in the case of Cloud Services (Web/Worker Role) and Service Fabric?**

**A:** While these platforms are impacted by planned maintenance, customers using these platforms are considered safe given that only VMs in a single Upgrade Domain (UD) will be impacted at any given time. Self-service maintenance is currently not available for Cloud Services (Web/Worker Role) and Service Fabric.

**Q: I don't see any maintenance information on my VMs. What went wrong?**

**A:** There are several reasons why you're not seeing any maintenance information on your VMs:

1. You are using a subscription marked as Microsoft internal.
2. Your VMs are not scheduled for maintenance. It could be that the maintenance wave has ended, canceled, or modified so that your VMs are no longer impacted by it.
3. You don't have the **Maintenance** column added to your VM list view. While we have added this column to the default view, customers who configured to see non-default columns must manually add the **Maintenance** column to their VM list view.

**Q: My VM is scheduled for maintenance for the second time. Why?**

**A:** There are several use cases where you will see your VM scheduled for maintenance after you have already completed your maintenance-redeploy:

1. We have canceled the maintenance wave and restarted it with a different payload. It could be that we've detected faulted payload and we simply need to deploy an additional payload.
2. Your VM was *service healed* to another node due to a hardware fault.
3. You have selected to stop (deallocate) and restart the VM.
4. You have **auto shutdown** turned on for the VM.

## Next steps

You can handle planned maintenance using the [Azure CLI](#), [Azure PowerShell](#) or [portal](#).

# Handling planned maintenance notifications using the Azure CLI

2/28/2020 • 2 minutes to read • [Edit Online](#)

**This article applies to virtual machines running both Linux and Windows.**

You can use the CLI to see when VMs are scheduled for [maintenance](#). Planned maintenance information is available from `az vm get-instance-view`.

Maintenance information is returned only if there is maintenance planned.

```
az vm get-instance-view -n myVM -g myResourceGroup --query instanceView.maintenanceRedeployStatus
```

## Start maintenance

The following call will start maintenance on a VM if `IsCustomerInitiatedMaintenanceAllowed` is set to true.

```
az vm perform-maintenance -g myResourceGroup -n myVM
```

## Classic deployments

### IMPORTANT

Classic VMs will be retired on March 1, 2023.

If you use IaaS resources from ASM, please complete your migration by March 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.

For more information, see [Migrate your IaaS resources to Azure Resource Manager by March 1, 2023](#).

If you still have legacy VMs that were deployed using the classic deployment model, you can use the Azure classic CLI to query for VMs and initiate maintenance.

Make sure you are in the correct mode to work with classic VM by typing:

```
azure config mode asm
```

To get the maintenance status of a VM named *myVM*, type:

```
azure vm show myVM
```

To start maintenance on your classic VM named *myVM* in the *myService* service and *myDeployment* deployment, type:

```
azure compute virtual-machine initiate-maintenance --service-name myService --name myDeployment --virtual-machine-name myVM
```

## Next steps

You can also handle planned maintenance using the [Azure PowerShell](#) or [portal](#).

# Handling planned maintenance notifications using the portal

2/10/2020 • 2 minutes to read • [Edit Online](#)

**This article applies to virtual machines running both Linux and Windows.**

Once a [planned maintenance](#) wave is scheduled, you can check for a list of virtual machines that are impacted.

You can use the Azure portal and look for VMs scheduled for maintenance.

1. Sign in to the [Azure portal](#).
2. In the left navigation, click **Virtual Machines**.
3. In the Virtual Machines pane, select **Edit columns** button to open the list of available columns.
4. Select and add the following columns:

**Maintenance status:** Shows the maintenance status for the VM. The following are the potential values:

| VALUE           | DESCRIPTION                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Start now       | The VM is in the self-service maintenance window that lets you initiate the maintenance yourself. See below on how to start maintenance on your VM.                                                                |
| Scheduled       | The VM is scheduled for maintenance with no option for you to initiate maintenance. You can learn of the maintenance window by selecting the Maintenance - Scheduled window in this view or by clicking on the VM. |
| Already updated | Your VM is already updated and no further action is required at this time.                                                                                                                                         |
| Retry later     | You have initiated maintenance with no success. You will be able to use the self-service maintenance option at a later time.                                                                                       |
| Retry now       | You can retry a previously unsuccessful self-initiated maintenance.                                                                                                                                                |
| -               | Your VM is not part of a planned maintenance wave.                                                                                                                                                                 |

**Maintenance - Self-service window:** Shows the time window when you can self-start maintenance on your VMs.

**Maintenance - Scheduled window:** Shows the time window when Azure will maintain your VM in order to complete maintenance.

## Notification and alerts in the portal

Azure communicates a schedule for planned maintenance by sending an email to the subscription owner and co-owners group. You can add additional recipients and channels to this communication by creating Azure activity log alerts. For more information, see [Create activity log alerts on service notifications](#).

Make sure you set the **Event type** as **Planned maintenance**, and **Services** as **Virtual Machine Scale Sets** and/or **Virtual Machines**.

## Start Maintenance on your VM from the portal

While looking at the VM details, you will be able to see more maintenance-related details.

At the top of the VM details view, a new notification ribbon will be added if your VM is included in a planned maintenance wave. In addition, a new option is added to start maintenance when possible.

Click on the maintenance notification to see the maintenance page with more details on the planned maintenance. From there, you will be able to **start maintenance** on your VM.

Once you start maintenance, your virtual machine will be maintained and the maintenance status will be updated to reflect the result within few minutes.

If you missed the self-service window, you will still be able to see the window when your VM will be maintained by Azure.

## Next steps

You can also handle planned maintenance using the [Azure CLI](#) or [PowerShell](#).

# Handling planned maintenance using PowerShell

2/28/2020 • 2 minutes to read • [Edit Online](#)

**This article applies to virtual machines running both Linux and Windows.**

You can use Azure PowerShell to see when VMs are scheduled for [maintenance](#). Planned maintenance information is available from the [Get-AzVM](#) cmdlet when you use the `-Status` parameter.

Maintenance information is returned only if there is maintenance planned. If no maintenance is scheduled that impacts the VM, the cmdlet does not return any maintenance information.

```
Get-AzVM -ResourceGroupName myResourceGroup -Name myVM -Status
```

The following properties are returned under `MaintenanceRedeployStatus`:

| VALUE                                              | DESCRIPTION                                                                                       |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------|
| <code>IsCustomerInitiatedMaintenanceAllowed</code> | Indicates whether you can start maintenance on the VM at this time                                |
| <code>PreMaintenanceWindowStartTime</code>         | The beginning of the maintenance self-service window when you can initiate maintenance on your VM |
| <code>PreMaintenanceWindowEndTime</code>           | The end of the maintenance self-service window when you can initiate maintenance on your VM       |
| <code>MaintenanceWindowStartTime</code>            | The beginning of the maintenance scheduled in which Azure initiates maintenance on your VM        |
| <code>MaintenanceWindowEndTime</code>              | The end of the maintenance scheduled window in which Azure initiates maintenance on your VM       |
| <code>LastOperationResultCode</code>               | The result of the last attempt to initiate maintenance on the VM                                  |

You can also get the maintenance status for all VMs in a resource group by using [Get-AzVM](#) and not specifying a VM.

```
Get-AzVM -ResourceGroupName myResourceGroup -Status
```

The following PowerShell example takes your subscription ID and returns a list of VMs that are scheduled for maintenance.

```

function MaintenanceIterator
{
 Select-AzSubscription -SubscriptionId $args[0]

 $rgList= Get-AzResourceGroup

 for ($rgIdx=0; $rgIdx -lt $rgList.Length ; $rgIdx++)
 {
 $rg = $rgList[$rgIdx]
 $vmList = Get-AzVM -ResourceGroupName $rg.ResourceGroupName
 for ($vmIdx=0; $vmIdx -lt $vmList.Length ; $vmIdx++)
 {
 $vm = $vmList[$vmIdx]
 $vmDetails = Get-AzVM -ResourceGroupName $rg.ResourceGroupName -Name $vm.Name -Status
 if ($vmDetails.MaintenanceRedeployStatus)
 {
 Write-Output "VM: $($vmDetails.Name) IsCustomerInitiatedMaintenanceAllowed: $($vmDetails.MaintenanceRedeployStatus.IsCustomerInitiatedMaintenanceAllowed)
($($vmDetails.MaintenanceRedeployStatus.LastOperationMessage))"
 }
 }
 }
}

```

## Start maintenance on your VM using PowerShell

Using information from the function in the previous section, the following starts maintenance on a VM if **IsCustomerInitiatedMaintenanceAllowed** is set to true.

```
Restart-AzVM -PerformMaintenance -name $vm.Name -ResourceGroupName $rg.ResourceGroupName
```

## Classic deployments

### IMPORTANT

Classic VMs will be retired on March 1, 2023.

If you use IaaS resources from ASM, please complete your migration by March 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.

For more information, see [Migrate your IaaS resources to Azure Resource Manager by March 1, 2023](#).

If you still have legacy VMs that were deployed using the classic deployment model, you can use PowerShell to query for VMs and initiate maintenance.

To get the maintenance status of a VM, type:

```
Get-AzureVM -ServiceName <Service name> -Name <VM name>
```

To start maintenance on your classic VM, type:

```
Restart-AzureVM -InitiateMaintenance -ServiceName <service name> -Name <VM name>
```

## Next steps

You can also handle planned maintenance using the [Azure CLI](#) or [portal](#).

# Preview: Control updates with Maintenance Control and Azure PowerShell

2/14/2020 • 5 minutes to read • [Edit Online](#)

Manage platform updates, that don't require a reboot, using maintenance control. Azure frequently updates its infrastructure to improve reliability, performance, security or launch new features. Most updates are transparent to users. Some sensitive workloads, like gaming, media streaming, and financial transactions, can't tolerate even few seconds of a VM freezing or disconnecting for maintenance. Maintenance control gives you the option to wait on platform updates and apply them within a 35-day rolling window.

Maintenance control lets you decide when to apply updates to your isolated VMs.

With maintenance control, you can:

- Batch updates into one update package.
- Wait up to 35 days to apply updates.
- Automate platform updates for your maintenance window using Azure Functions.
- Maintenance configurations work across subscriptions and resource groups.

## IMPORTANT

Maintenance Control is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Limitations

- VMs must be on a [dedicated host](#), or be created using an [isolated VM size](#).
- After 35 days, an update will automatically be applied.
- User must have **Resource Contributor** access.

## Enable the PowerShell module

Make sure `PowerShellGet` is up to date.

```
Install-Module -Name PowerShellGet -Repository PSGallery -Force
```

The Az.Maintenance PowerShell cmdlets are in preview, so you need to install the module with the `AllowPrerelease` parameter in Cloud Shell or your local PowerShell installation.

```
Install-Module -Name Az.Maintenance -AllowPrerelease
```

If you are installing locally, make sure you open your PowerShell prompt as an administrator.

You may also be asked to confirm that you want to install from an *untrusted repository*. Type `y` or select **Yes to All** to install the module.

## Create a maintenance configuration

Create a resource group as a container for your configuration. In this example, a resource group named *myMaintenanceRG* is created in *eastus*. If you already have a resource group that you want to use, you can skip this part and replace the resource group name with your own in the rest of the examples.

```
New-AzResourceGroup `
 -Location eastus `
 -Name myMaintenanceRG
```

Use [New-AzMaintenanceConfiguration](#) to create a maintenance configuration. This example creates a maintenance configuration named *myConfig* scoped to the host.

```
$config = New-AzMaintenanceConfiguration `
 -ResourceGroup myMaintenanceRG `
 -Name myConfig `
 -MaintenanceScope host `
 -Location eastus
```

Using `-MaintenanceScope host` ensures that the maintenance configuration is used for controlling updates to the host.

If you try to create a configuration with the same name, but in a different location, you will get an error. Configuration names must be unique to your subscription.

You can query for available maintenance configurations using [Get-AzMaintenanceConfiguration](#).

```
Get-AzMaintenanceConfiguration | Format-Table -Property Name,Id
```

## Assign the configuration

Use [New-AzConfigurationAssignment](#) to assign the configuration to your isolated VM or Azure Dedicated Host.

### Isolated VM

Apply the configuration to a VM using the ID of the configuration. Specify `-ResourceType VirtualMachines` and supply the name of the VM for `-ResourceName`, and the resource group of the VM for `-ResourceGroupName`.

```
New-AzConfigurationAssignment `
 -ResourceGroupName myResourceGroup `
 -Location eastus `
 -ResourceName myVM `
 -ResourceType VirtualMachines `
 -ProviderName Microsoft.Compute `
 -ConfigurationAssignmentName $config.Name `
 -MaintenanceConfigurationId $config.Id
```

### Dedicated host

To apply a configuration to a dedicated host, you also need to include `-ResourceType hosts`, `-ResourceParentName` with the name of the host group, and `-ResourceParentType hostGroups`.

```
New-AzConfigurationAssignment `
-ResourceGroupName myResourceGroup `
-Location eastus `
-ResourceName myHost `
-ResourceType hosts `
-ResourceParentName myHostGroup `
-ResourceParentType hostGroups `
-ProviderName Microsoft.Compute `
-ConfigurationAssignmentName $config.Name `
-MaintenanceConfigurationId $config.Id
```

## Check for pending updates

Use [Get-AzMaintenanceUpdate](#) to see if there are pending updates. Use `-subscription` to specify the Azure subscription of the VM if it is different from the one that you are logged into.

If there are no updates to show, this command will return nothing. Otherwise, it will return a `PSApplyUpdate` object:

```
{
 "maintenanceScope": "Host",
 "impactType": "Freeze",
 "status": "Pending",
 "impactDurationInSec": 9,
 "notBefore": "2020-02-21T16:47:44.8728029Z",
 "properties": {
 "resourceId": "/subscriptions/39c6cced-4d6c-4dd5-af86-57499cd3f846/resourcegroups/Ignite2019/providers/Microsoft.Compute/virtualMachines/MCDemo3"
 }
}
```

### Isolated VM

Check for pending updates for an isolated VM. In this example, the output is formatted as a table for readability.

```
Get-AzMaintenanceUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myVM `
-ResourceType VirtualMachines `
-ProviderName Microsoft.Compute | Format-Table
```

### Dedicated host

To check for pending updates for a dedicated host. In this example, the output is formatted as a table for readability. Replace the values for the resources with your own.

```
Get-AzMaintenanceUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myHost `
-ResourceType hosts `
-ResourceParentName myHostGroup `
-ResourceParentType hostGroups `
-ProviderName Microsoft.Compute | Format-Table
```

## Apply updates

Use [New-AzApplyUpdate](#) to apply pending updates.

### Isolated VM

Create a request to apply updates to an isolated VM.

```
New-AzApplyUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myVM `
-ResourceType VirtualMachines `
-ProviderName Microsoft.Compute
```

On success, this command will return a `PSApplyUpdate` object. You can use the Name attribute in the `Get-AzApplyUpdate` command to check the update status. See [Check update status](#).

## Dedicated host

Apply updates to a dedicated host.

```
New-AzApplyUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myHost `
-ResourceType hosts `
-ResourceParentName myHostGroup `
-ResourceParentType hostGroups `
-ProviderName Microsoft.Compute
```

## Check update status

Use `Get-AzApplyUpdate` to check on the status of an update. The commands shown below show the status of the latest update by using `default` for the `-ApplyUpdateName` parameter. You can substitute the name of the update (returned by the `New-AzApplyUpdate` command) to get the status of a specific update.

```
Status : Completed
ResourceId : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShants/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-iso
LastUpdateTime : 1/1/2020 12:00:00 AM
Id : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShants/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-iso/providers/Microsoft.Maintenance/applyUpdates/default
Name : default
Type : Microsoft.Maintenance/applyUpdates
```

`LastUpdateTime` will be the time when the update got complete, either initiated by you or by the platform in case self-maintenance window was not used. If there has never been an update applied through maintenance control it will show default value.

## Isolated VM

Check for updates to a specific virtual machine.

```
Get-AzApplyUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myVM `
-ResourceType VirtualMachines `
-ProviderName Microsoft.Compute `
-ApplyUpdateName default
```

## Dedicated host

Check for updates to a dedicated host.

```
Get-AzApplyUpdate `
-ResourceGroupName myResourceGroup `
-ResourceName myHost `
-ResourceType hosts `
-ResourceParentName myHostGroup `
-ResourceParentType hostGroups `
-ProviderName Microsoft.Compute `
-ApplyUpdateName myUpdateName
```

## Remove a maintenance configuration

Use [Remove-AzMaintenanceConfiguration](#) to delete a maintenance configuration.

```
Remove-AzMaintenanceConfiguration `
-ResourceGroupName myResourceGroup `
-Name $config.Name
```

## Next steps

To learn more, see [Maintenance and updates](#).

# Preview: Control updates with Maintenance Control and the Azure CLI

2/14/2020 • 5 minutes to read • [Edit Online](#)

Manage platform updates, that don't require a reboot, using maintenance control. Azure frequently updates its infrastructure to improve reliability, performance, security or launch new features. Most updates are transparent to users. Some sensitive workloads, like gaming, media streaming, and financial transactions, can't tolerate even few seconds of a VM freezing or disconnecting for maintenance. Maintenance control gives you the option to wait on platform updates and apply them within a 35-day rolling window.

Maintenance control lets you decide when to apply updates to your isolated VMs and Azure Dedicated Hosts.

With maintenance control, you can:

- Batch updates into one update package.
- Wait up to 35 days to apply updates.
- Automate platform updates for your maintenance window using Azure Functions.
- Maintenance configurations work across subscriptions and resource groups.

## IMPORTANT

Maintenance Control is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Limitations

- VMs must be on a [dedicated host](#), or be created using an [isolated VM size](#).
- After 35 days, an update will automatically be applied.
- User must have **Resource Contributor** access.

## Install the maintenance extension

If you choose to install the [Azure CLI](#) locally, you need version 2.0.76 or later.

Install the `maintenance` preview CLI extension locally or in Cloud Shell.

```
az extension add -n maintenance
```

## Create a maintenance configuration

Use `az maintenance configuration create` to create a maintenance configuration. This example creates a maintenance configuration named *myConfig* scoped to the host.

```
az group create \
--location eastus \
--name myMaintenanceRG
az maintenance configuration create \
-g myMaintenanceRG \
--name myConfig \
--maintenanceScope host \
--location eastus
```

Copy the configuration ID from the output to use later.

Using `--maintenanceScope host` ensures that the maintenance config is used for controlling updates to the host.

If you try to create a configuration with the same name, but in a different location, you will get an error.

Configuration names must be unique to your subscription.

You can query for available maintenance configurations using `az maintenance configuration list`.

```
az maintenance configuration list --query "[].{Name:name, ID:id}" -o table
```

## Assign the configuration

Use `az maintenance assignment create` to assign the configuration to your isolated VM or Azure Dedicated Host.

### Isolated VM

Apply the configuration to a VM using the ID of the configuration. Specify `--resource-type virtualMachines` and supply the name of the VM for `--resource-name`, and the resource group for the VM in `--resource-group`, and the location of the VM for `--location`.

```
az maintenance assignment create \
--resource-group myMaintenanceRG \
--location eastus \
--resource-name myVM \
--resource-type virtualMachines \
--provider-name Microsoft.Compute \
--configuration-assignment-name myConfig \
--maintenance-configuration-id "/subscriptions/1111abcd-1a11-1a2b-1a12-123456789abc/resourcegroups/myMaintenanceRG/providers/Microsoft.Maintenance/maintenanceConfigurations/myConfig"
```

### Dedicated host

To apply a configuration to a dedicated host, you need to include `--resource-type hosts`, `--resource-parent-name` with the name of the host group, and `--resource-parent-type hostGroups`.

The parameter `--resource-id` is the ID of the host. You can use [az vm host get-instance-view](#) to get the ID of your dedicated host.

```
az maintenance assignment create \
-g myDHRG \
--resource-name myHost \
--resource-type hosts \
--provider-name Microsoft.Compute \
--configuration-assignment-name myConfig \
--maintenance-configuration-id "/subscriptions/1111abcd-1a11-1a2b-1a12-
123456789abc/resourcegroups/myDHRG/providers/Microsoft.Maintenance/maintenanceConfigurations/myConf
ig" \
-l eastus \
--resource-parent-name myHostGroup \
--resource-parent-type hostGroups
```

## Check configuration

You can verify that the configuration was applied correctly, or check to see what configuration is currently applied using `az maintenance assignment list`.

### Isolated VM

```
az maintenance assignment list \
--provider-name Microsoft.Compute \
--resource-group myMaintenanceRG \
--resource-name myVM \
--resource-type virtualMachines \
--query "[].{resource:resourceGroup, configName:name}" \
--output table
```

### Dedicated host

```
az maintenance assignment list \
--resource-group myDHRG \
--resource-name myHost \
--resource-type hosts \
--provider-name Microsoft.Compute \
--resource-parent-name myHostGroup \
--resource-parent-type hostGroups \
--query "[].{ResourceGroup:resourceGroup,configName:name}" \
-o table
```

## Check for pending updates

Use `az maintenance update list` to see if there are pending updates. Update --subscription to be the ID for the subscription that contains the VM.

If there are no updates, the command will return an error message, which will contain the text:

```
Resource not found...StatusCode: 404
```

If there are updates, only one will be returned, even if there are multiple updates pending. The data for this update will be returned in an object:

```
[
 {
 "impactDurationInSec": 9,
 "impactType": "Freeze",
 "maintenanceScope": "Host",
 "notBefore": "2020-03-03T07:23:04.905538+00:00",
 "resourceId": "/subscriptions/9120c5ff-e78e-4bd0-b29f-
75c19cadd078/resourcegroups/DemoRG/providers/Microsoft.Compute/hostGroups/demoHostGroup/hosts/myHost",
 "status": "Pending"
 }
]
```

## Isolated VM

Check for pending updates for an isolated VM. In this example, the output is formatted as a table for readability.

```
az maintenance update list \
-g myMaintenanceRg \
--resource-name myVM \
--resource-type virtualMachines \
--provider-name Microsoft.Compute \
-o table
```

## Dedicated host

To check for pending updates for a dedicated host. In this example, the output is formatted as a table for readability. Replace the values for the resources with your own.

```
az maintenance update list \
--subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
-g myHostResourceGroup \
--resource-name myHost \
--resource-type hosts \
--provider-name Microsoft.Compute \
--resource-parentname myHostGroup \
--resource-parent-type hostGroups \
-o table
```

# Apply updates

Use `az maintenance apply update` to apply pending updates. On success, this command will return JSON containing the details of the update.

## Isolated VM

Create a request to apply updates to an isolated VM.

```
az maintenance applyupdate create \
--subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
--resource-group myMaintenanceRG \
--resource-name myVM \
--resource-type virtualMachines \
--provider-name Microsoft.Compute
```

## Dedicated host

Apply updates to a dedicated host.

```
az maintenance applyupdate create \
--subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
--resource-group myHostResourceGroup \
--resource-name myHost \
--resource-type hosts \
--provider-name Microsoft.Compute \
--resource-parent-name myHostGroup \
--resource-parent-type hostGroups
```

## Check the status of applying updates

You can check on the progress of the updates using `az maintenance applyupdate get`.

You can use `default` as the update name to see results for the last update, or replace `myUpdateName` with the name of the update that was returned when you ran `az maintenance applyupdate create`.

```
Status : Completed
ResourceId : /subscriptions/12ae7457-4a34-465c-94c1-
 17c058c2bd25/resourcegroups/TestShants/providers/Microsoft.Com-
 pute/virtualMachines/DXT-test-04-iso
LastUpdateTime : 1/1/2020 12:00:00 AM
Id : /subscriptions/12ae7457-4a34-465c-94c1-
 17c058c2bd25/resourcegroups/TestShants/providers/Microsoft.Com-
 pute/virtualMachines/DXT-test-04-iso/providers/Microsoft.Mainten-
 ance/applyUpdates/default
Name : default
Type : Microsoft.Maintenance/applyUpdates
```

`LastUpdateTime` will be the time when the update got complete, either initiated by you or by the platform in case self-maintenance window was not used. If there has never been an update applied through maintenance control it will show default value.

### Isolated VM

```
az maintenance applyupdate get \
--resource-group myMaintenanceRG \
--resource-name myVM \
--resource-type virtualMachines \
--provider-name Microsoft.Compute \
--apply-update-name default
```

### Dedicated host

```
az maintenance applyupdate get \
--subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
--resource-group myMaintenanceRG \
--resource-name myHost \
--resource-type hosts \
--provider-name Microsoft.Compute \
--resource-parent-name myHostGroup \
--resource-parent-type hostGroups \
--apply-update-name myUpdateName \
--query "{LastUpdate:lastUpdateTime, Name:name, ResourceGroup:resourceGroup, Status:status}" \
--output table
```

## Delete a maintenance configuration

Use `az maintenance configuration delete` to delete a maintenance configuration. Deleting the configuration removes the maintenance control from the associated resources.

```
az maintenance configuration delete \
--subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
-g myResourceGroup \
--name myConfig
```

## Next steps

To learn more, see [Maintenance and updates](#).

# Azure Metadata Service: Scheduled Events for Windows VMs

2/28/2020 • 7 minutes to read • [Edit Online](#)

Scheduled Events is an Azure Metadata Service that gives your application time to prepare for virtual machine maintenance. It provides information about upcoming maintenance events (e.g. reboot) so your application can prepare for them and limit disruption. It is available for all Azure Virtual Machine types including PaaS and IaaS on both Windows and Linux.

For information about Scheduled Events on Linux, see [Scheduled Events for Linux VMs](#).

## NOTE

Scheduled Events is generally available in all Azure Regions. See [Version and Region Availability](#) for latest release information.

## Why Scheduled Events?

Many applications can benefit from time to prepare for virtual machine maintenance. The time can be used to perform application specific tasks that improve availability, reliability, and serviceability including:

- Checkpoint and restore
- Connection draining
- Primary replica failover
- Removal from load balancer pool
- Event logging
- Graceful shutdown

Using Scheduled Events your application can discover when maintenance will occur and trigger tasks to limit its impact. Enabling scheduled events gives your virtual machine a minimum amount of time before the maintenance activity is performed. See the Event Scheduling section below for details.

Scheduled Events provides events in the following use cases:

- [Platform initiated maintenance](#) (for example, VM reboot, live migration or memory preserving updates for host)
- Degraded hardware
- User initiated maintenance (e.g. user restarts or redeploys a VM)
- [Spot VM](#) and [Spot scale set](#) instance evictions

## The Basics

Azure Metadata service exposes information about running Virtual Machines using a REST Endpoint accessible from within the VM. The information is available via a non-routable IP so that it is not exposed outside the VM.

### Endpoint Discovery

For VNET enabled VMs, the metadata service is available from a static non-routable IP, `169.254.169.254`. The full endpoint for the latest version of Scheduled Events is:

```
http://169.254.169.254/metadata/scheduledevents?api-version=2019-01-01
```

If the Virtual Machine is not created within a Virtual Network, the default cases for cloud services and classic VMs, additional logic is required to discover the IP address to use. Refer to this sample to learn how to [discover the host endpoint](#).

## Version and Region Availability

The Scheduled Events Service is versioned. Versions are mandatory and the current version is `2019-01-01`.

| VERSION    | RELEASE TYPE         | REGIONS | RELEASE NOTES                                                                                                                                                               |
|------------|----------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2019-01-01 | General Availability | All     | <ul style="list-style-type: none"><li>Added support for virtual machine scale sets EventType 'Terminate'</li></ul>                                                          |
| 2017-11-01 | General Availability | All     | <ul style="list-style-type: none"><li>Added support for Spot VM eviction EventType 'Preempt'</li></ul>                                                                      |
| 2017-08-01 | General Availability | All     | <ul style="list-style-type: none"><li>Removed prepended underscore from resource names for IaaS VMs</li><li>Metadata Header requirement enforced for all requests</li></ul> |
| 2017-03-01 | Preview              | All     | <ul style="list-style-type: none"><li>Initial release</li></ul>                                                                                                             |

### NOTE

Previous preview releases of scheduled events supported `{latest}` as the api-version. This format is no longer supported and will be deprecated in the future.

## Enabling and Disabling Scheduled Events

Scheduled Events is enabled for your service the first time you make a request for events. You should expect a delayed response in your first call of up to two minutes. You should query the endpoint periodically to detect upcoming maintenance events as well as the status of maintenance activities that are being performed.

Scheduled Events is disabled for your service if it does not make a request for 24 hours.

## User Initiated Maintenance

User initiated virtual machine maintenance via the Azure portal, API, CLI, or PowerShell results in a scheduled event. This allows you to test the maintenance preparation logic in your application and allows your application to prepare for user initiated maintenance.

Restarting a virtual machine schedules an event with type `Reboot`. Redeploying a virtual machine schedules an event with type `Redeploy`.

## Using the API

### Headers

When you query the Metadata Service, you must provide the header `Metadata:true` to ensure the request was not unintentionally redirected. The `Metadata:true` header is required for all scheduled events requests. Failure to include the header in the request will result in a Bad Request response from the Metadata Service.

### Query for events

You can query for Scheduled Events simply by making the following call:

## PowerShell

```
curl http://169.254.169.254/metadata/scheduledevents?api-version=2019-01-01 -H @{"Metadata"="true"}
```

A response contains an array of scheduled events. An empty array means that there are currently no events scheduled. In the case where there are scheduled events, the response contains an array of events:

```
{
 "DocumentIncarnation": {IncarnationID},
 "Events": [
 {
 "EventId": {eventID},
 "EventType": "Reboot" | "Redeploy" | "Freeze" | "Preempt" | "Terminate",
 "ResourceType": "VirtualMachine",
 "Resources": [{resourceName}],
 "EventStatus": "Scheduled" | "Started",
 "NotBefore": {timeInUTC},
 }
]
}
```

The DocumentIncarnation is an ETag and provides an easy way to inspect if the Events payload has changed since the last query.

## Event Properties

| PROPERTY     | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EventId      | Globally unique identifier for this event.<br><br>Example: <ul style="list-style-type: none"><li>• 602d9444-d2cd-49c7-8624-8643e7171297</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| EventType    | Impact this event causes.<br><br>Values: <ul style="list-style-type: none"><li>• <code>Freeze</code> : The Virtual Machine is scheduled to pause for a few seconds. CPU and network connectivity may be suspended, but there is no impact on memory or open files.</li><li>• <code>Reboot</code> : The Virtual Machine is scheduled for reboot (non-persistent memory is lost).</li><li>• <code>Redeploy</code> : The Virtual Machine is scheduled to move to another node (ephemeral disks are lost).</li><li>• <code>Preempt</code> : The Spot Virtual Machine is being deleted (ephemeral disks are lost).</li><li>• <code>Terminate</code> : The Virtual Machine is scheduled to be deleted.</li></ul> |
| ResourceType | Type of resource this event impacts.<br><br>Values: <ul style="list-style-type: none"><li>• <code>VirtualMachine</code></li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| PROPERTY     | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resources    | <p>List of resources this event impacts. This is guaranteed to contain machines from at most one <a href="#">Update Domain</a>, but may not contain all machines in the UD.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>• ["FrontEnd_IN_0", "BackEnd_IN_0"]</li> </ul>                                                                                                                                       |
| Event Status | <p>Status of this event.</p> <p>Values:</p> <ul style="list-style-type: none"> <li>• <code>Scheduled</code> : This event is scheduled to start after the time specified in the <code>NotBefore</code> property.</li> <li>• <code>Started</code> : This event has started.</li> </ul> <p>No <code>Completed</code> or similar status is ever provided; the event will no longer be returned when the event is completed.</p> |
| NotBefore    | <p>Time after which this event may start.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>• Mon, 19 Sep 2016 18:29:47 GMT</li> </ul>                                                                                                                                                                                                                                                                             |

## Event Scheduling

Each event is scheduled a minimum amount of time in the future based on event type. This time is reflected in an event's `NotBefore` property.

| EVENT TYPE | MINIMUM NOTICE                                      |
|------------|-----------------------------------------------------|
| Freeze     | 15 minutes                                          |
| Reboot     | 15 minutes                                          |
| Redeploy   | 10 minutes                                          |
| Preempt    | 30 seconds                                          |
| Terminate  | <a href="#">User Configurable</a> : 5 to 15 minutes |

## Event Scope

Scheduled events are delivered to:

- Standalone Virtual Machines
- All Virtual Machines in a Cloud Service
- All Virtual Machines in an Availability Set
- All Virtual Machines in a Scale Set Placement Group.

As a result, you should check the `Resources` field in the event to identify which VMs are going to be impacted.

## Starting an event

Once you have learned of an upcoming event and completed your logic for graceful shutdown, you can approve the outstanding event by making a `POST` call to the metadata service with the `EventId`. This indicates to Azure that it can shorten the minimum notification time (when possible).

The following is the json expected in the `POST` request body. The request should contain a list of `StartRequests`. Each `StartRequest` contains the `EventId` for the event you want to expedite:

```
{
 "StartRequests" : [
 {
 "EventId": {EventId}
 }
]
}
```

#### PowerShell

```
curl -H @{"Metadata"="true"} -Method POST -Body '{"StartRequests": [{"EventId": "f020ba2e-3bc0-4c40-a10b-86575a9eabd5"}]}' -Uri http://169.254.169.254/metadata/scheduledevents?api-version=2019-01-01
```

#### NOTE

Acknowledging an event allows the event to proceed for all `Resources` in the event, not just the virtual machine that acknowledges the event. You may therefore choose to elect a leader to coordinate the acknowledgement, which may be as simple as the first machine in the `Resources` field.

## PowerShell sample

The following sample queries the metadata service for scheduled events and approves each outstanding event.

```

How to get scheduled events
function Get-ScheduledEvents($uri)
{
 $scheduledEvents = Invoke-RestMethod -Headers @{"Metadata"="true"} -URI $uri -Method get
 $json = ConvertTo-Json $scheduledEvents
 Write-Host "Received following events: `n" $json
 return $scheduledEvents
}

How to approve a scheduled event
function Approve-ScheduledEvent($eventId, $uri)
{
 # Create the Scheduled Events Approval Document
 $startRequests = [array]@{"EventId" = $eventId}
 $scheduledEventsApproval = @{"StartRequests" = $startRequests}

 # Convert to JSON string
 $approvalString = ConvertTo-Json $scheduledEventsApproval

 Write-Host "Approving with the following: `n" $approvalString

 # Post approval string to scheduled events endpoint
 Invoke-RestMethod -Uri $uri -Headers @{"Metadata"="true"} -Method POST -Body $approvalString
}

function Handle-ScheduledEvents($scheduledEvents)
{
 # Add logic for handling events here
}

#####
Sample Scheduled Events Interaction
#####

Set up the scheduled events URI for a VNET-enabled VM
$localhostIP = "169.254.169.254"
$scheduledEventURI = 'http://{0}/metadata/scheduledevents?api-version=2019-01-01' -f $localhostIP

Get events
$scheduledEvents = Get-ScheduledEvents $scheduledEventURI

Handle events however is best for your service
Handle-ScheduledEvents $scheduledEvents

Approve events when ready (optional)
foreach($event in $scheduledEvents.Events)
{
 Write-Host "Current Event: `n" $event
 $entry = Read-Host "`nApprove event? Y/N"
 if($entry -eq "Y" -or $entry -eq "y")
 {
 Approve-ScheduledEvent $event.EventId $scheduledEventURI
 }
}

```

## Next steps

- Watch a [Scheduled Events Demo](#) on Azure Friday.
- Review the Scheduled Events code samples in the [Azure Instance Metadata Scheduled Events GitHub Repository](#)
- Read more about the APIs available in the [Instance Metadata service](#).
- Learn about [planned maintenance for Windows virtual machines in Azure](#).

# Monitoring Scheduled Events

11/13/2019 • 7 minutes to read • [Edit Online](#)

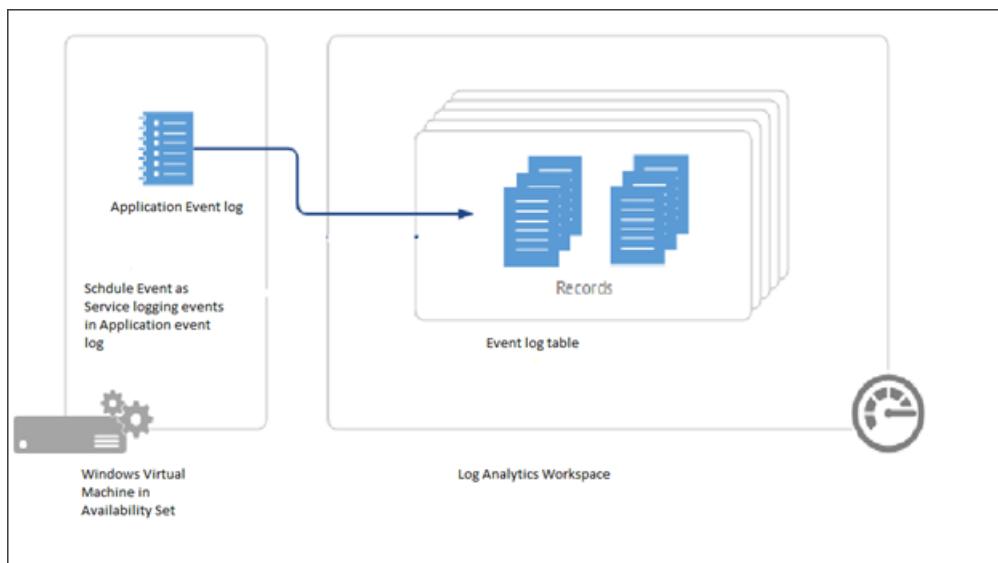
Updates are applied to different parts of Azure every day, to keep the services running on them secure, and up-to-date. In addition to planned updates, unplanned events may also occur. For example, if any hardware degradation or fault is detected, Azure services may need to perform unplanned maintenance. Using live migration, memory preserving updates and generally keeping a strict bar on the impact of updates, in most cases these events are almost transparent to customers, and they have no impact or at most cause a few seconds of virtual machine freeze. However, for some applications, even a few seconds of virtual machine freeze could cause an impact. Knowing in advance about upcoming Azure maintenance is important, to ensure the best experience for those applications. [Scheduled Events service](#) provides you a programmatic interface to be notified about upcoming maintenance, and enables you to gracefully handle the maintenance.

In this article, we will show how you can use scheduled events to be notified about maintenance events that could be affecting your VMs and build some basic automation that can help with monitoring and analysis.

## Routing scheduled events to Log Analytics

Scheduled Events is available as part of the [Azure Instance Metadata Service](#), which is available on every Azure virtual machine. Customers can write automation to query the endpoint of their virtual machines to find scheduled maintenance notifications and perform mitigations, like saving the state and taking the virtual machine out of rotation. We recommend building automation to record the Scheduled Events so you can have an auditing log of Azure maintenance events.

In this article, we will walk you through how to capture maintenance Scheduled Events to Log Analytics. Then, we will trigger some basic notification actions, like sending an email to your team and getting a historical view of all events that have affected your virtual machines. For the event aggregation and automation we will use [Log Analytics](#), but you can use any monitoring solution to collect these logs and trigger automation.



## Prerequisites

For this example, you will need to create a [Windows Virtual Machine in an Availability Set](#). Scheduled Events provide notifications about changes that can affect any of the virtual machines in your availability set, Cloud Service, Virtual Machine Scale Set or standalone VMs. We will be running a [service](#) that polls for scheduled events on one of the VMs that will act as a collector, to get events for all of the other VMs in the availability set.

Don't delete the group resource group at the end of the tutorial.

You will also need to [create a Log Analytics workspace](#) that we will use to aggregate information from the VMs in the availability set.

## Set up the environment

You should now have 2 initial VMs in an availability set. Now we need to create a 3rd VM, called myCollectorVM, in the same availability set.

```
New-AzVm `
 -ResourceGroupName "myResourceGroupAvailability" `
 -Name "myCollectorVM" `
 -Location "East US" `
 -VirtualNetworkName "myVnet" `
 -SubnetName "mySubnet" `
 -SecurityGroupName "myNetworkSecurityGroup" `
 -OpenPorts 3389 `
 -PublicIpAddressName "myPublicIpAddress3" `
 -AvailabilitySetName "myAvailabilitySet" `
 -Credential $cred
```

Download the installation .zip file of the project from [GitHub](#).

Connect to **myCollectorVM** and copy the .zip file to the virtual machine and extract all of the files. On your VM, open a PowerShell prompt. Move your prompt into the folder containing `SchService.ps1`, for example:

```
PS C:\Users\azureuser\AzureScheduledEventsService-master\AzureScheduledEventsService-master\Powershell> , and set up the service.
```

```
.\SchService.ps1 -Setup
```

Start the service.

```
.\SchService.ps1 -Start
```

The service will now start polling every 10 seconds for any scheduled events and approve the events to expedite the maintenance. Freeze, Reboot, Redeploy, and Preempt are the events captured by Schedule events. Note that you can extend the script to trigger some mitigations prior to approving the event.

Validate the service status and make sure it is running.

```
.\SchService.ps1 -status
```

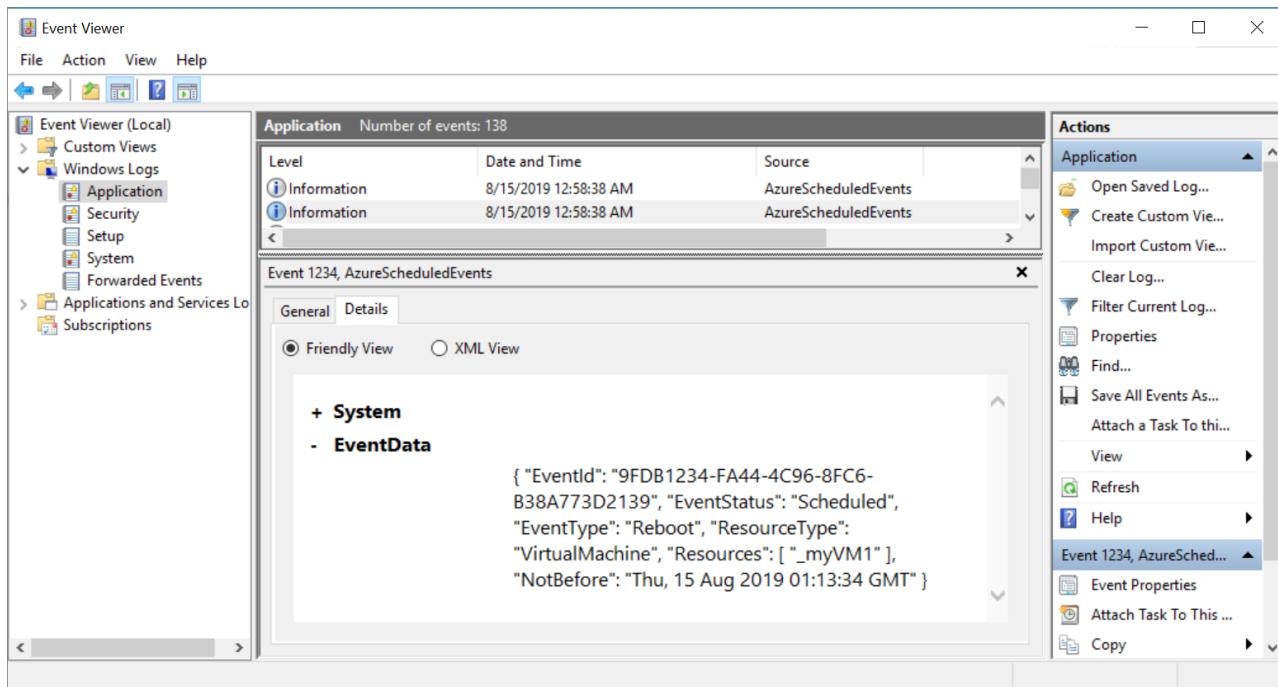
This should return `Running`.

The service will now start polling every 10 seconds for any scheduled events and approve the events to expedite the maintenance. Freeze, Reboot, Redeploy and Preempt are the events captured by Schedule events. You can extend the script to trigger some mitigations prior to approving the event.

When any of the above events are captured by Schedule Event service, it will get logged in the Application Event Log Event Status, Event Type, Resources (Virtual machine names) and NotBefore (minimum notice period). You can locate the events with ID 1234 in the Application Event Log.

Once the service is set up and started, it will log events in the Windows Application logs. To verify this works, restart one of the virtual machines in the availability set and you should see an event being logged in Event viewer

in Windows Logs > Application log showing the VM restarted.



When events are captured by the Schedule Event service, it will get logged in the application even log with Event Status, Event Type, Resources (VM name) and NotBefore (minimum notice period). You can locate the events with ID 1234 in the Application Event Log.

#### NOTE

In this example, the virtual machines were are in an availability set, which enabled us to designate a single virtual machine as the collector to listen and route scheduled events to our log analytics works space. If you have standalone virtual machines, you can run the service on every virtual machine, and then connect them individually to your log analytics workspace.

For our set up, we chose Windows, but you can design a similar solution on Linux.

At any point you can stop/remove the Scheduled Event Service by using the switches `-stop` and `-remove`.

## Connect to the workspace

We now want to connect a Log Analytics Workspace to the collector VM. The Log Analytics workspace acts as a repository and we will configure event log collection to capture the application logs from the collector VM.

To route the Scheduled Events to the Events Log, which will be saved as Application log by our service, you will need to connect your virtual machine to your Log Analytics workspace.

1. Open the page for the workspace you created.
2. Under **Connect to a data source** select **Azure virtual machines (VMs)**.

## Get started with Log Analytics

Log Analytics collects data from a variety of sources and uses a powerful query language to give you insights into the operation of your applications and resources. Use Azure Monitor to access the complete set of tools for monitoring all of your Azure resources

### 1 Connect a data source

Select one or more data sources to connect to the workspace

Azure virtual machines (VMs)

Windows, Linux and other sources

Azure Activity logs

3. Search for and select **myCollectorVM**.

4. On the new page for **myCollectorVM**, select **Connect**.

This will install the [Microsoft Monitoring agent](#) in your virtual machine. It will take a few minutes to connect your VM to the workspace and install the extension.

## Configure the workspace

1. Open the page for your workspace and select **Advanced settings**.

2. Select **Data** from the left menu, then select **Windows Event Logs**.

3. In **Collect from the following event logs**, start typing *application* and then select **Application** from the list.

The screenshot shows the 'Advanced settings' page for a Log Analytics workspace. On the left, there's a sidebar with icons for Home, Overview, CollectorWorkspace, Advanced settings, Refresh, Logs, Save, and Discard. The main area has a title 'Advanced settings' and a subtitle 'collectorworkspace'. Below this are sections for 'Connected Sources', 'Data' (which is selected), and 'Computer Groups'. Under 'Data', there's a list of log types: Windows Event Logs, Windows Performance Counters, Linux Performance Counters, IIS Logs, Custom Fields, Custom Logs, and Syslog. The 'Windows Event Logs' section is expanded, showing a list of event logs. A search bar at the top right says 'Collect events from the following event logs' with 'application' typed in. A dropdown menu shows 'Application' selected, with a list of log names like Microsoft-Windows-AppHost/ApplicationTracing, Microsoft-Windows-Application Server-Applications/Admin, etc.

4. Leave **ERROR**, **WARNING**, and **INFORMATION** selected and then select **Save** to save the settings.

### NOTE

There will be some delay, and it may take up to 10 minutes before the log is available.

## Creating an alert rule with Azure Monitor

Once the events are pushed to Log Analytics, you can run the following [query](#) to look for the schedule Events.

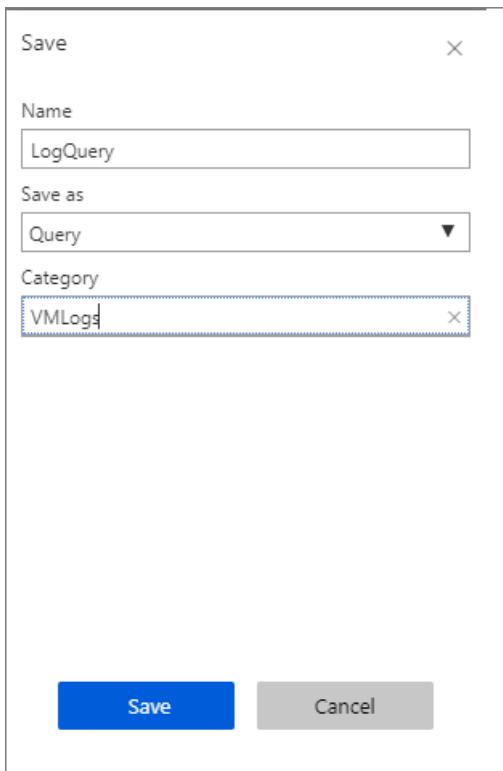
1. At the top of the page, select **Logs** and paste the following into the text box:

```

Event
| where EventLog == "Application" and Source contains "AzureScheduledEvents" and RenderedDescription
contains "Scheduled" and RenderedDescription contains "EventStatus"
| project TimeGenerated, RenderedDescription
| extend ReqJson= parse_json(RenderedDescription)
| extend EventId = ReqJson["EventId"]
,EventStatus = ReqJson["EventStatus"]
,EventType = ReqJson["EventType"]
,NotBefore = ReqJson["NotBefore"]
,ResourceType = ReqJson["ResourceType"]
,Resources = ReqJson["Resources"]
| project-away RenderedDescription,ReqJson

```

2. Select **Save**, and then type *logQuery* for the name, leave **Query** as the type, type *VMLogs* as the **Category**, and then select **Save**.



3. Select **New alert rule**.
4. In the **Create rule** page, leave `collectorworkspace` as the **Resource**.
5. Under **Condition**, select the entry *Whenever the customer log search is*. The **Configure signal logic** page will open.
6. Under **Threshold value**, enter *0* and then select **Done**.
7. Under **Actions**, select **Create action group**. The **Add action group** page will open.
8. In **Action group name**, type *myActionGroup*.
9. In **Short name**, type **myActionGroup**.
10. In **Resource group**, select **myResourceGroupAvailability**.
11. Under Actions, in **ACTION NAME** type **Email**, and then select **Email/SMS/Push/Voice**. The **Email/SMS/Push/Voice** page will open.
12. Select **Email**, type in your e-mail address, then select **OK**.
13. In the **Add action group** page, select **OK**.

14. In the **Create rule** page, under **ALERT DETAILS**, type *myAlert* for the **Alert rule name**, and then type *Email alert rule* for the **Description**.
15. When you are finished, select **Create alert rule**.
16. Restart one of the VMs in the availability set. Within a few minutes, you should get an e-mail that the alert has been triggered.

To manage your alert rules, go to the resource group, select **Alerts** from the left menu, and then select **Manage alert rules** from the top of the page.

## Next steps

To learn more, see the [Scheduled events service](#) page on GitHub.

# What is Azure Monitor for VMs (preview)?

2/27/2020 • 2 minutes to read • [Edit Online](#)

Azure Monitor for VMs monitors your Azure virtual machines (VM) and virtual machine scale sets at scale. It analyzes the performance and health of your Windows and Linux VMs, and monitors their processes and dependencies on other resources and external processes.

It includes support for monitoring performance and application dependencies for VMs that are hosted on-premises or in another cloud provider. The following key features deliver in-depth insight:

- **Pre-defined trending performance charts:** Display core performance metrics from the guest VM operating system.
- **Dependency map:** Displays the interconnected components with the VM from various resource groups and subscriptions.

## NOTE

We recently [announced changes](#) we are making to the Health feature based on the feedback we have received from our public preview customers. Given the number of changes we will be making, we are going to stop offering the Health feature for new customers. Existing customers can continue to use the health feature. For more details, please refer to our [General Availability FAQ](#).

Integration with Azure Monitor logs delivers powerful aggregation and filtering, and it can analyze data trends over time. Such comprehensive workload monitoring can't be achieved with Azure Monitor or Service Map alone.

You can view this data in a single VM from the virtual machine directly, or you can use Azure Monitor to deliver an aggregated view of your VMs where the view supports Azure resource-context or workspace-context modes. For more information, see [access modes overview](#).

The screenshot displays two main views of the Azure Monitor for VMs interface. On the left, the 'Monitor - Virtual Machines (preview)' dashboard shows a summary of VM health and performance. It includes sections for Overview, Activity log, Metrics, Logs, Service Health, Workbooks, Applications, and Virtual Machines (preview). The 'Virtual Machines (preview)' section is expanded, showing options like List VMs and Manage Coverage. On the right, the 'SKUbuntu1804 - Insights (preview)' page provides detailed insights for a specific VM. This page includes a summary card for the VM, a 'Get Started' button, and tabs for Performance, Map, and Health (preview). The 'Performance' tab shows a chart for 'SKUbuntu1804' with a time range of 'Last 30 minutes'. The 'Map' tab displays a dependency graph where the VM is connected to four other servers via ports 53, 32526, 443, and 80. The 'Health' tab shows a summary of the VM's health status. The overall interface is designed to provide a holistic view of VM performance and dependencies.

Azure Monitor for VMs can deliver predictable performance and availability of vital applications. It identifies

performance bottlenecks and network issues. Azure Monitor for VMs can also help you understand whether an issue is related to other dependencies.

## Data usage

When you deploy Azure Monitor for VMs, the data that's collected by your VMs is ingested and stored in Azure Monitor. Performance and dependency data collected are stored in a Log Analytics workspace. Based on the pricing that's published on the [Azure Monitor pricing page](#), Azure Monitor for VMs is billed for:

- The data that's ingested and stored.
- The alert rules that are created.
- The notifications that are sent.

The log size varies by the string lengths of performance counters, and it can increase with the number of logical disks and network adapters allocated to the VM. If you already have a workspace and are collecting these counters, no duplicate charges are applied. If you're already using Service Map, the only change you'll see is the additional connection data that's sent to Azure Monitor.

## Next steps

To understand the requirements and methods that help you monitor your virtual machines, review [Deploy Azure Monitor for VMs](#).

# Create, view, and manage metric alerts using Azure Monitor

2/27/2020 • 5 minutes to read • [Edit Online](#)

Metric alerts in Azure Monitor provide a way to get notified when one of your metrics crosses a threshold. Metric alerts work on a range of multi-dimensional platform metrics, custom metrics, Application Insights standard and custom metrics. In this article, we will describe how to create, view, and manage metric alert rules through Azure portal and Azure CLI. You can also create metric alert rules using Azure Resource Manager templates, which are described in [a separate article](#).

You can learn more about how metric alerts work from [metric alerts overview](#).

## Create with Azure portal

The following procedure describes how to create a metric alert rule in Azure portal:

1. In [Azure portal](#), click on **Monitor**. The Monitor blade consolidates all your monitoring settings and data in one view.
2. Click **Alerts** then click **+ New alert rule**.

### TIP

Most resource blades also have **Alerts** in their resource menu under **Monitoring**, you could create alerts from there as well.

3. Click **Select target**, in the context pane that loads, select a target resource that you want to alert on. Use **Subscription** and **Resource type** drop-downs to find the resource you want to monitor. You can also use the search bar to find your resource.
4. If the selected resource has metrics you can create alerts on, **Available signals** on the bottom right will include metrics. You can view the full list of resource types supported for metric alerts in this [article](#).
5. Once you have selected a target resource, click on **Add condition**.
6. You will see a list of signals supported for the resource, select the metric you want to create an alert on.
7. You will see a chart for the metric for the last six hours. Use the **Chart period** dropdown to select to see longer history for the metric.
8. If the metric has dimensions, you will see a dimensions table presented. Select one or more values per dimension.
  - The displayed dimension values are based on metric data from the last three days.
  - If the dimension value you're looking for isn't displayed, click "+" to add a custom value.
  - You can also **Select \*** for any of the dimensions. **Select \*** will dynamically scale the selection to all current and future values for a dimension.
9. The metric alert rule will evaluate the condition for all combinations of values selected. [Learn more about how alerting on multi-dimensional metrics works](#).
9. Select the **Threshold** type, **Operator**, and **Aggregation type**. This will determine the logic that the metric alert rule will evaluate.

- If you are using a **Static** threshold, continue to define a **Threshold value**. The metric chart can help determine what might be a reasonable threshold.
- If you are using a **Dynamic** threshold, continue to define the **Threshold sensitivity**. The metric chart will display the calculated thresholds based on recent data. [Learn more about Dynamic Thresholds condition type and sensitivity options](#).

10. Optionally, refine the condition by adjusting **Aggregation granularity** and **Frequency of evaluation**.
11. Click **Done**.
12. Optionally, add another criteria if you want to monitor a complex alert rule. Currently users can have alert rules with Dynamic Thresholds criteria as a single criterion.
13. Fill in **Alert details** like **Alert rule name**, **Description**, and **Severity**.
14. Add an action group to the alert either by selecting an existing action group or creating a new action group.
15. Click **Done** to save the metric alert rule.

**NOTE**

Metric alert rules created through portal are created in the same resource group as the target resource.

## View and manage with Azure portal

You can view and manage metric alert rules using the Manage Rules blade under Alerts. The procedure below shows you how to view your metric alert rules and edit one of them.

1. In Azure portal, navigate to **Monitor**
2. Click on **Alerts** and **Manage rules**
3. In the **Manage rules** blade, you can view all your alert rules across subscriptions. You can further filter the rules using **Resource group**, **Resource type**, and **Resource**. If you want to see only metric alerts, select **Signal type** as Metrics.

**TIP**

In the **Manage rules** blade, you can select multiple alert rules and enable/disable them. This might be useful when certain target resources need to be put under maintenance

4. Click on the name of the metric alert rule you want to edit
5. In the Edit Rule, click on the **Alert criteria** you want to edit. You can change the metric, threshold condition and other fields as required

**NOTE**

You can't edit the **Target resource** and **Alert Rule Name** after the metric alert is created.

6. Click **Done** to save your edits.

## With Azure CLI

The previous sections described how to create, view, and manage metric alert rules using Azure portal. This section will describe how to do the same using cross-platform [Azure CLI](#). Quickest way to start using Azure CLI is through

Azure Cloud Shell. For this article, we will use Cloud Shell.

1. Go to Azure portal, click on **Cloud Shell**.
2. At the prompt, you can use commands with `--help` option to learn more about the command and how to use it. For example, the following command shows you the list of commands available for creating, viewing, and managing metric alerts

```
az monitor metrics alert --help
```

3. You can create a simple metric alert rule that monitors if average Percentage CPU on a VM is greater than 90

```
az monitor metrics alert create -n {nameofthealert} -g {ResourceGroup} --scopes {VirtualMachineResourceID} --condition "avg Percentage CPU > 90" --description {descriptionofthealert}
```

4. You can view all the metric alerts in a resource group using the following command

```
az monitor metrics alert list -g {ResourceGroup}
```

5. You can see the details of a particular metric alert rule using the name or the resource ID of the rule.

```
az monitor metrics alert show -g {ResourceGroup} -n {AlertRuleName}
```

```
az monitor metrics alert show --ids {RuleResourceId}
```

6. You can disable a metric alert rule using the following command.

```
az monitor metrics alert update -g {ResourceGroup} -n {AlertRuleName} --enabled false
```

7. You can delete a metric alert rule using the following command.

```
az monitor metrics alert delete -g {ResourceGroup} -n {AlertRuleName}
```

## Next steps

- [Create metric alerts using Azure Resource Manager Templates](#).
- [Understand how metric alerts work](#).
- [Understand how metric alerts with Dynamic Thresholds condition work](#).
- [Understand the web hook schema for metric alerts](#)

# Create, view, and manage log alerts using Azure Monitor

2/27/2020 • 11 minutes to read • [Edit Online](#)

## Overview

This article shows you how to set up log alerts using the alerts interface inside Azure portal. Definition of an alert rule is in three parts:

- Target: Specific Azure resource, which is to be monitored
- Criteria: Specific condition or logic that when seen in Signal, should trigger action
- Action: Specific call sent to a receiver of a notification - email, SMS, webhook etc.

The term **Log Alerts** to describe alerts where signal is log query in a [Log Analytics workspace](#) or [Application Insights](#). Learn more about functionality, terminology, and types from [Log alerts - Overview](#).

### NOTE

Popular log data from [a Log Analytics workspace](#) is now also available on the metric platform in Azure Monitor. For details view, [Metric Alert for Logs](#)

## Managing log alerts from the Azure portal

Detailed next is step-by-step guide to using log alerts using the Azure portal interface.

### Create a log alert rule with the Azure portal

1. In the [portal](#), select **Monitor** and under the MONITOR section - choose **Alerts**.

The screenshot shows the 'Monitor - Alerts' page. At the top, there's a breadcrumb navigation: 'Home > Monitor - Alerts'. Below it is a search bar labeled 'Search (Ctrl+I)'. The main navigation menu has several items: 'Overview' (with a circular icon), 'Activity log' (with a blue square icon), 'Alerts' (with a yellow speech bubble icon, highlighted in blue), 'Metrics' (with a blue chart icon), 'Logs' (with a blue grid icon), 'Service Health' (with a blue heart icon), and 'Workbooks (preview)' (with a blue document icon). The 'Alerts' item is currently active.

2. Select the **New Alert Rule** button to create a new alert in Azure.



3. The Create Alert section is shown with the three parts consisting of: *Define alert condition*, *Define alert details*, and *Define action group*.

## Create rule

Rules management

|                                                                                                                 |                                                                                                                                     |                  |
|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------|
|                                | <b>* RESOURCE</b><br>Select the target(s) that you wish to monitor<br><a href="#">Select</a>                                        | <b>HIERARCHY</b> |
|                                | <b>* CONDITION</b><br>No condition defined, click on 'Add condition' to select a signal and define its logic<br><a href="#">Add</a> |                  |
|                                | <b>ACTIONS</b><br>No configured actions<br><a href="#">Add</a>                                                                      |                  |
| <b>ALERT DETAILS</b>                                                                                            |                                                                                                                                     |                  |
| <b>* Alert rule name</b> <a href="#">?</a><br>Specify alert rule name. Sample: 'Percentage CPU greater than 70' |                                                                                                                                     |                  |
| Description<br>Specify alert description here...                                                                |                                                                                                                                     |                  |
| Enable rule upon creation<br><input checked="" type="radio"/> Yes <input type="radio"/> No                      |                                                                                                                                     |                  |

[Create alert rule](#)

4. Define the alert condition by using the **Select Resource** link and specifying the target by selecting a resource. Filter by choosing the *Subscription*, *Resource Type*, and required *Resource*.

**NOTE**

For creating a log alert - verify the **log** signal is available for the selected resource before you proceed.

## Select a resource



For metric and log based alert rules please select a specific target, for activity log alert rules you can select a subscription, a resource type or a resource group.

\* Filter by subscription i

Filter by resource type i

**RESOURCE**

▼ **Fabrikam Enterprise**

▼ **FabrikamIT**



**logs**

Selection preview

Available signal(s) : Log, Metric, Activity Log

**Fabrikam Enterprise** >

**FabrikamIT** > **logs**

**Done**

5. **Log Alerts:** Ensure **Resource Type** is an analytics source like *Log Analytics* or *Application Insights* and signal type as **Log**, then once appropriate **resource** is chosen, click **Done**. Next use the **Add criteria** button to view list of signal options available for the resource and from the signal list **Custom log search** option for chosen log monitor service like *Log Analytics* or *Application Insights*.

## Configure signal logic

A signal can be of the form metric, a log search query or an activity log. Based on selected target(s), the list of supported signals is shown below. Select one to setup the alert condition.

### All signals (57)

| SIGNAL NAME                 |                      |                                                                                                    | MONITOR SERVICE | SIGNAL TYPE |
|-----------------------------|----------------------|----------------------------------------------------------------------------------------------------|-----------------|-------------|
| Custom log search           | Application Insights |  Log              |                 |             |
| PageView_Revenue            | Application Insights |  Log(Saved Query) |                 |             |
| All Administrative opera... | Administrative       |  Activity Log     |                 |             |
| Application Insights ana... | Administrative       |  Activity Log     |                 |             |
| Application Insights API... | Administrative       |  Activity Log     |                 |             |
| Application insights co...  | Administrative       |  Activity Log     |                 |             |
| Application insights co...  | Administrative       |  Activity Log     |                 |             |
| Application insights co...  | Administrative       |  Activity Log     |                 |             |
| Application Insights exp... | Administrative       |  Activity Log    |                 |             |
| Application Insights Co...  | Administrative       |  Activity Log   |                 |             |
| Subscription migration...   | Administrative       |  Activity Log   |                 |             |
| Migrate subscription to...  | Administrative       |  Activity Log   |                 |             |
| Rollback subscription to... | Administrative       |  Activity Log   |                 |             |
| All Security operations     | Security             |  Activity Log   |                 |             |
| Application Insights ana... | Security             |  Activity Log   |                 |             |
| Application Insights API... | Security             |  Activity Log   |                 |             |
| Application insights co...  | Security             |  Activity Log   |                 |             |
| Application insights co...  | Security             |  Activity Log   |                 |             |
| Application insights co...  | Security             |  Activity Log   |                 |             |
| Application Insights exp... | Security             |  Activity Log   |                 |             |

Done

#### NOTE

Alerts lists can import analytics query as signal type - **Log (Saved Query)**, as seen in above illustration. So users can perfect your query in Analytics and then save them for future use in alerts - more details on using saving query available at [using log query in Azure Monitor](#) or [shared query in application insights analytics](#).

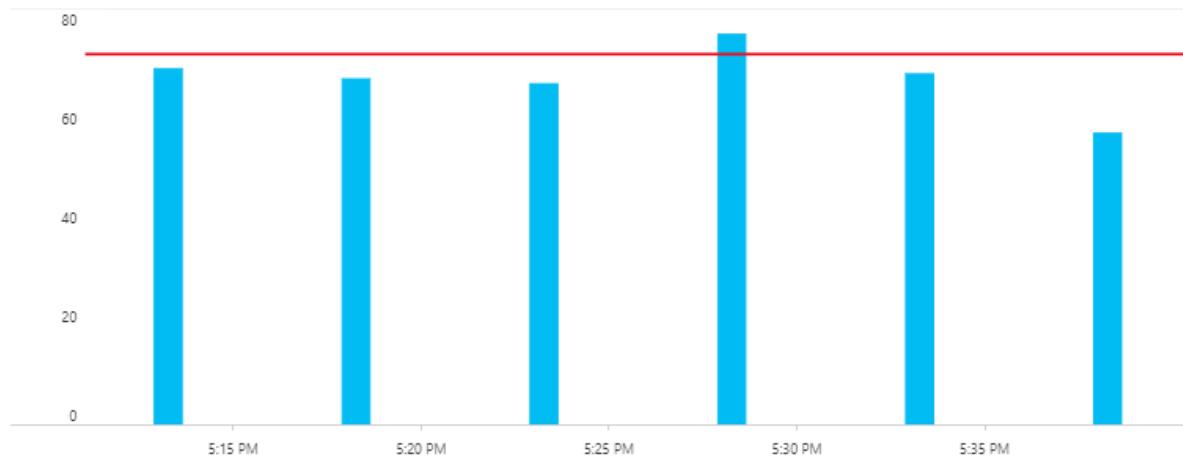
6. **Log Alerts:** Once selected, query for alerting can be stated in **Search Query** field; if the query syntax is incorrect the field displays error in RED. If the query syntax is correct - For reference historic data of the stated query is shown as a graph with option to tweak the time window from last six hours to last week.

## Configure signal logic

X

[-> Back to signal selection](#)

### Custom log search



\* Search query [?](#)

|                                            |                                     |
|--------------------------------------------|-------------------------------------|
| Event<br>  where EventLevelName == "Error" | <input checked="" type="checkbox"/> |
|--------------------------------------------|-------------------------------------|

[View result of query in Azure Monitor - Logs](#)

Query to be executed : Event | where EventLevelName == "Error" | count  
For time window : 7/19/2019, 5:28:16 PM - 7/19/2019, 5:38:16 PM

### Alert logic

|                                                 |                                            |                                                                               |
|-------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------|
| Based on <a href="#">?</a><br>Number of results | Operator <a href="#">?</a><br>Greater than | * Threshold value <a href="#">?</a><br>75 <input checked="" type="checkbox"/> |
|-------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------|

### Condition preview

Whenever the custom log search is greater than 75 count

### Evaluated based on

|                                               |                                               |
|-----------------------------------------------|-----------------------------------------------|
| * Period (in minutes) <a href="#">?</a><br>10 | Frequency (in minutes) <a href="#">?</a><br>5 |
|-----------------------------------------------|-----------------------------------------------|

[Done](#)

#### NOTE

Historical data visualization can only be shown if the query results have time details. If your query results in summarized data or specific column values - same is shown as a singular plot. For Metric Measurement type of Log Alerts using Application Insights or [switched to new API](#), you can specify which specific variable to group the data by using the **Aggregate on** option; as illustrated in below:

Configure signal logic

<- Back to signal selection

Custom log search

Pivoted on performanceBucket=<250... ▾

| Time    | Count |
|---------|-------|
| 4:50 PM | 70    |
| 4:55 PM | 80    |
| 5 PM    | 65    |
| 5:05 PM | 45    |
| 5:10 PM | 105   |

\* Search query ⓘ  
requests | summarize AggregatedValue=sum(itemCount) by bin(timestamp, 5min), performanceBucket

View result of query in Azure Monitor - Logs ⓘ

Query to be executed : requests | summarize AggregatedValue=sum(itemCount) by bin\_at(timestamp, 5min, now()), performanceBucket  
For time window : 7/19/2019, 4:43:06 PM - 7/19/2019, 5:13:06 PM

Alert logic

Based on ⓘ Metric measurement Aggregate value ⓘ Greater than \* Threshold value ⓘ 100

Trigger Alert Based On Consecutive breaches 1

Aggregate on ⓘ performanceBucket

Condition preview  
Whenever the custom log search is greater than 100 count

Evaluated based on

\* Period (in minutes) ⓘ 30 Frequency (in minutes) ⓘ 5

Done

7. **Log Alerts:** With the visualization in place, **Alert Logic** can be selected from shown options of Condition, Aggregation and finally Threshold. Finally specify in the logic, the time to assess for the specified condition, using **Period** option. Along with how often Alert should run by selecting **Frequency**. **Log Alerts** can be based on:

- **Number of Records:** An alert is created if the count of records returned by the query is either greater than or less than the value provided.
  - **Metric Measurement:** An alert is created if each *aggregate value* in the results exceeds the threshold value provided and it is *grouped by* chosen value. The number of breaches for an alert is the number of times the threshold is exceeded in the chosen time period. You can specify Total breaches for any combination of breaches across the results set or Consecutive breaches to require that the breaches must occur in consecutive samples.
8. As the second step, define a name for your alert in the **Alert rule name** field along with a **Description** detailing specifics for the alert and **Severity** value from the options provided. These details are reused in all alert emails, notifications, or push done by Azure Monitor. Additionally, user can choose to immediately activate the alert rule on creation by appropriately toggling **Enable rule upon creation** option.

For **Log Alerts** only, some additional functionality is available in Alert details:

- **SUPPRESS ALERTS:** When you turn on suppression for the alert rule, actions for the rule are disabled for a defined length of time after creating a new alert. The rule is still running and creates alert records provided the criteria is met. Allowing you time to correct the problem without running duplicate actions.

**ALERT DETAILS**

\* Alert rule name

Description

\* Severity

Enable rule upon creation

Suppress Alerts ?

**TIP**

Specify an suppress alert value greater than frequency of alert to ensure notifications are stopped without overlap

9. As the third and final step, specify if any **Action Group** needs to be triggered for the alert rule when alert condition is met. You can choose any existing Action Group with alert or create a new Action Group. According to selected Action Group, when alert is trigger Azure will: send email(s), send SMS(s), call Webhook(s), remediate using Azure Runbooks, push to your ITSM tool, etc. Learn more about [Action Groups](#).

#### NOTE

Refer to the [Azure subscription service limits](#) for limits on Runbook payloads triggered for log alerts via Azure action groups

For **Log Alerts** some additional functionality is available to override the default Actions:

- **Email Notification:** Overrides *e-mail subject* in the email, sent via Action Group; if one or more email actions exist in the said Action Group. You cannot modify the body of the mail and this field is **not** for email address.

- **Include custom Json payload:** Overrides the webhook JSON used by Action Groups; if one or more webhook actions exist in the said Action Group. User can specify format of JSON to be used for all webhooks configured in associated Action Group; for more information on webhook formats, see [webhook action for Log Alerts](#). View Webhook option is provided to check format using sample JSON data.

ACTIONS

No configured actions

Add

**Customize Actions**

Email subject ⓘ

Include custom Json payload for webhook ⓘ

10. If all fields are valid and with green tick the **create alert rule** button can be clicked and an alert is created in Azure Monitor - Alerts. All alerts can be viewed from the alerts Dashboard.

Home > ACMETelco-Portal - Logs (Analytics) > Create rule

Create rule

Rules management

**\* RESOURCE**

ACMETelco-Portal

**HIERARCHY**

Contoso Corp > ACMETelco

Select

**\* CONDITION**

Monthly cost in USD (Estimated) ⓘ

Whenever the Custom log search is Greater than 100 count

\$ 1.50

Total \$ 1.50

Add

**INFO** We currently support configuring only two metrics signals or one log search signal or one activity log signal per alert rule. An alert will be triggered when the conditions for all the above configured criteria are met

**ACTIONS**

View configured actions

Add

**Customize Actions**

Email subject ⓘ

Include custom Json payload for webhook ⓘ

**ALERT DETAILS**

\* Alert rule name ⓘ

Late Requests

Description

Checking if requests which are very late are exceeding threshold

\* Severity ⓘ

Sev 3

Enable rule upon creation

Yes  No

Suppress Alerts ⓘ

Create alert rule

Within a few minutes, the alert is active and triggers as previously described.

Users can also finalize their analytics query in [log analytics](#) and then push it to create an alert via 'Set Alert' button - then following instructions from Step 6 onwards in the above tutorial.

Run Time range: Set in query

Save Copy link Export New alert rule Pin

```
// List all computer heartbeats from the last hour
Heartbeat
| where TimeGenerated > ago(1h)
```

## View & manage log alerts in Azure portal

1. In the [portal](#), select **Monitor** and under the MONITOR section - choose **Alerts**.

2. The **Alerts Dashboard** is displayed - wherein all Azure Alerts (including log alerts) are displayed in a singular board; including every instance of when your log alert rule has fired. To learn more, see [Alert Management](#).

## NOTE

Log alert rules comprise of custom query-based logic provided by users and hence without a resolved state. Due to which every time the conditions specified in the log alert rule are met, it is fired.

3. Select the **Manage rules** button on the top bar, to navigate to the rule management section - where all alert rules created are listed; including alerts that have been disabled.



# Managing log alerts using Azure Resource Template

Log alerts in Azure Monitor are associated with resource type `Microsoft.Insights/scheduledQueryRules/`. For more information on this resource type, see [Azure Monitor - Scheduled Query Rules API reference](#). Log alerts for Application Insights or Log Analytics, can be created using [Scheduled Query Rules API](#).

## NOTE

Log alerts for Log Analytics can also be managed using legacy [Log Analytics Alert API](#) and legacy templates of [Log Analytics saved searches and alerts](#) as well. For more information on using the new ScheduledQueryRules API detailed here by default, see [Switch to new API for Log Analytics Alerts](#).

## Sample Log alert creation using Azure Resource Template

The following is the structure for [Scheduled Query Rules creation](#) based resource template using standard log search query of [number of results type log alert](#), with sample data set as variables.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {},
 "variables": {
 "alertLocation": "southcentralus",
 "alertName": "samplelogalert",
 "alertDescription": "Sample log search alert",
 "alertStatus": "true",
 "alertSource": {
 "Query": "requests",
 "SourceId": "/subscriptions/a123d7efg-123c-1234-5678-a12bc3defgh4/resourceGroups/myRG/providers/microsoft.insights/components/sampleAIapplication",
 "Type": "ResultCount"
 },
 "alertSchedule": {
 "Frequency": 15,
 "Time": 60
 },
 "alertActions": {
 "SeverityLevel": "4"
 },
 "alertTrigger": {
 "Operator": "GreaterThan",
 "Threshold": "1"
 },
 "actionGrp": {
 "ActionGroup": "/subscriptions/a123d7efg-123c-1234-5678-
```

```

a12bc3defgh4/resourceGroups/myRG/providers/microsoft.insights/actiongroups/sampleAG",
 "Subject": "Customized Email Header",
 "Webhook": "{ \"alertname\": \"#alertrulename\", \"IncludeSearchResults\":true }"
}
},
"resources": [
{
 "name": "[variables('alertName')]",
 "type": "Microsoft.Insights/scheduledQueryRules",
 "apiVersion": "2018-04-16",
 "location": "[variables('alertLocation')]",
 "properties": {
 "description": "[variables('alertDescription')]",
 "enabled": "[variables('alertStatus')]",
 "source": {
 "query": "[variables('alertSource').Query]",
 "dataSourceId": "[variables('alertSource').SourceId]",
 "queryType": "[variables('alertSource').Type]"
 },
 "schedule": {
 "frequencyInMinutes": "[variables('alertSchedule').Frequency]",
 "timeWindowInMinutes": "[variables('alertSchedule').Time]"
 },
 "action": {
 "odata.type": "Microsoft.WindowsAzure.Management.Monitoring.Alerts.Models.Microsoft.AppInsights.Nexus.DataContracts.Resources.ScheduledQueryRules.AlertingAction",
 "severity": "[variables('alertActions').SeverityLevel]",
 "aznsAction": {
 "actionGroup": "[array(variables('actionGrp').ActionGroup)]",
 "emailSubject": "[variables('actionGrp').Subject]",
 "customWebhookPayload": "[variables('actionGrp').Webhook]"
 },
 "trigger": {
 "thresholdOperator": "[variables('alertTrigger').Operator]",
 "threshold": "[variables('alertTrigger').Threshold]"
 }
 }
 }
}
]
}

```

The sample json above can be saved as (say) sampleScheduledQueryRule.json for the purpose of this walk through and can be deployed using [Azure Resource Manager in Azure portal](#).

### Log alert with cross-resource query using Azure Resource Template

The following is the structure for [Scheduled Query Rules creation](#) based resource template using [cross-resource log search query](#) of [metric measurement type log alert](#), with sample data set as variables.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {},
 "variables": {
 "alertLocation": "Region Name for your Application Insights App or Log Analytics Workspace",
 "alertName": "sample log alert",
 "alertDescr": "Sample log search alert",
 "alertStatus": "true",
 "alertSource": {
 "Query": "union workspace(\"servicews\").Update, app('serviceapp').requests | summarize AggregatedValue = count() by bin(TimeGenerated,1h), Classification",
 "Resource1": "/subscriptions/a123d7efg-123c-1234-5678-a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.OperationalInsights/workspaces/servicews",
 "Resource2": "/subscriptions/a123d7efg-123c-1234-5678-
```

```

a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.insights/components/serviceapp",
 "SourceId": "/subscriptions/a123d7efg-123c-1234-5678-
a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.OperationalInsights/workspaces/servicews",
 "Type":"ResultCount"
 },
 "alertSchedule":{
 "Frequency": 15,
 "Time": 60
 },
 "alertActions":{
 "SeverityLevel": "4",
 "SuppressTimeinMin": 20
 },
 "alertTrigger":{
 "Operator":"GreaterThan",
 "Threshold":"1"
 },
 "metricMeasurement": {
 "thresholdOperator": "Equal",
 "threshold": "1",
 "metricTriggerType": "Consecutive",
 "metricColumn": "Classification"
 },
 "actionGrp":{
 "ActionGroup": "/subscriptions/a123d7efg-123c-1234-5678-
a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.insights/actiongroups/sampleAG",
 "Subject": "Customized Email Header",
 "Webhook": "{ \"alertname\": \"#alertrulename\", \"IncludeSearchResults\":true }"
 }
 },
 "resources": [
 {
 "name": "[variables('alertName')]",
 "type": "Microsoft.Insights/scheduledQueryRules",
 "apiVersion": "2018-04-16",
 "location": "[variables('alertLocation')]",
 "properties": {
 "description": "[variables('alertDescr')]",
 "enabled": "[variables('alertStatus')]",
 "source": {
 "query": "[variables('alertSource').Query]",
 "authorizedResources": "[concat(array(variables('alertSource').Resource1),
array(variables('alertSource').Resource2))]",
 "dataSourceId": "[variables('alertSource').SourceId]",
 "queryType": "[variables('alertSource').Type]"
 },
 "schedule": {
 "frequencyInMinutes": "[variables('alertSchedule').Frequency]",
 "timeWindowInMinutes": "[variables('alertSchedule').Time]"
 },
 "action": {
 "odata.type": "Microsoft.WindowsAzure.Management.Monitoring.Alerts.Models.Microsoft.AppInsights.Nexus.DataContracts.Resources.ScheduledQueryRules.AlertingAction",
 "severity": "[variables('alertActions').SeverityLevel]",
 "throttlingInMin": "[variables('alertActions').SuppressTimeinMin]",
 "aznsAction": {
 "actionGroup": "[array(variables('actionGrp').ActionGroup)]",
 "emailSubject": "[variables('actionGrp').Subject]",
 "customWebhookPayload": "[variables('actionGrp').Webhook]"
 },
 "trigger": {
 "thresholdOperator": "[variables('alertTrigger').Operator]",
 "threshold": "[variables('alertTrigger').Threshold]",
 "metricTrigger": {
 "thresholdOperator": "[variables('metricMeasurement').thresholdOperator]",
 "threshold": "[variables('metricMeasurement').threshold]",
 "metricColumn": "[variables('metricMeasurement').metricColumn]",
 "metricTriggerType": "[variables('metricMeasurement').metricTriggerType]"
 }
 }
 }
 }
 }
]
}

```

```
 }
 }
}
}
```

#### IMPORTANT

When using cross-resource query in log alert, the usage of [authorizedResources](#) is mandatory and user must have access to the list of resources stated

The sample json above can be saved as (say) sampleScheduledQueryRule.json for the purpose of this walk through and can be deployed using [Azure Resource Manager in Azure portal](#).

## Managing log alerts using PowerShell

#### NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

Azure Monitor - [Scheduled Query Rules API](#) is a REST API and fully compatible with Azure Resource Manager REST API. And PowerShell cmdlets listed below are available to leverage the [Scheduled Query Rules API](#).

1. [New-AzScheduledQueryRule](#) : Powershell cmdlet to create a new log alert rule.
2. [Set-AzScheduledQueryRule](#) : Powershell cmdlet to update an existing log alert rule.
3. [New-AzScheduledQueryRuleSource](#) : Powershell cmdlet to create or update object specifying source parameters for a log alert. Used as input by [New-AzScheduledQueryRule](#) and [Set-AzScheduledQueryRule](#) cmdlet.
4. [New-AzScheduledQueryRuleSchedule](#): Powershell cmdlet to create or update object specifying schedule parameters for a log alert. Used as input by [New-AzScheduledQueryRule](#) and [Set-AzScheduledQueryRule](#) cmdlet.
5. [New-AzScheduledQueryRuleAlertingAction](#) : Powershell cmdlet to create or update object specifying action parameters for a log alert. Used as input by [New-AzScheduledQueryRule](#) and [Set-AzScheduledQueryRule](#) cmdlet.
6. [New-AzScheduledQueryRuleAznsActionGroup](#) : Powershell cmdlet to create or update object specifying action groups parameters for a log alert. Used as input by [New-AzScheduledQueryRuleAlertingAction](#) cmdlet.
7. [New-AzScheduledQueryRuleTriggerCondition](#) : Powershell cmdlet to create or update object specifying trigger condition parameters for log alert. Used as input by [New-AzScheduledQueryRuleAlertingAction](#) cmdlet.
8. [New-AzScheduledQueryRuleLogMetricTrigger](#) : Powershell cmdlet to create or update object specifying metric trigger condition parameters for [metric measurement type log alert](#). Used as input by [New-AzScheduledQueryRuleTriggerCondition](#) cmdlet.
9. [Get-AzScheduledQueryRule](#) : Powershell cmdlet to list existing log alert rules or a specific log alert rule
10. [Update-AzScheduledQueryRule](#) : Powershell cmdlet to enable or disable log alert rule
11. [Remove-AzScheduledQueryRule](#): Powershell cmdlet to delete an existing log alert rule

#### NOTE

ScheduledQueryRules PowerShell cmdlets can only manage rules created cmdlet itself or using Azure Monitor - [Scheduled Query Rules API](#). Log alert rules created using legacy [Log Analytics Alert API](#) and legacy templates of [Log Analytics saved searches and alerts](#) can be managed using ScheduledQueryRules PowerShell cmdlets only after user [switches API preference for Log Analytics Alerts](#).

Illustrated next are the steps for creation of a sample log alert rule using the scheduledQueryRules PowerShell cmdlets.

```
$source = New-AzScheduledQueryRuleSource -Query 'Heartbeat | summarize AggregatedValue = count() by bin(TimeGenerated, 5m), _ResourceId' -DataSourceId "/subscriptions/a123d7efg-123c-1234-5678-a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.OperationalInsights/workspaces/servicews"

$schedule = New-AzScheduledQueryRuleSchedule -FrequencyInMinutes 15 -TimeWindowInMinutes 30

$metricTrigger = New-AzScheduledQueryRuleLogMetricTrigger -ThresholdOperator "GreaterThan" -Threshold 2 - MetricTriggerType "Consecutive" -MetricColumn "_ResourceId"

$triggerCondition = New-AzScheduledQueryRuleTriggerCondition -ThresholdOperator "LessThan" -Threshold 5 - MetricTrigger $metricTrigger

$aznsActionGroup = New-AzScheduledQueryRuleAznsActionGroup -ActionGroup "/subscriptions/a123d7efg-123c-1234-5678-a12bc3defgh4/resourceGroups/contosoRG/providers/microsoft.insights/actiongroups/sampleAG" -EmailSubject "Custom email subject" -CustomWebhookPayload "{ `\"alert`:`#${alertrulename}`, `\"IncludeSearchResults`:true }"

$alertingAction = New-AzScheduledQueryRuleAlertingAction -AznsAction $aznsActionGroup -Severity "3" -Trigger $triggerCondition

New-AzScheduledQueryRule -ResourceGroupName "contosoRG" -Location "Region Name for your Application Insights App or Log Analytics Workspace" -Action $alertingAction -Enabled $true -Description "Alert description" - Schedule $schedule -Source $source -Name "Alert Name"
```

## Managing log alerts using CLI or API

Azure Monitor - [Scheduled Query Rules API](#) is a REST API and fully compatible with Azure Resource Manager REST API. Hence it can be used via Powershell using Resource Manager commands for Azure CLI.

#### NOTE

Log alerts for Log Analytics can also be managed using legacy [Log Analytics Alert API](#) and legacy templates of [Log Analytics saved searches and alerts](#) as well. For more information on using the new ScheduledQueryRules API detailed here by default, see [Switch to new API for Log Analytics Alerts](#).

Log alerts currently do not have dedicated CLI commands currently; but as illustrated below can be used via Azure Resource Manager CLI command for sample Resource Template shown earlier (sampleScheduledQueryRule.json) in the Resource Template section:

```
az group deployment create --resource-group contosoRG --template-file sampleScheduledQueryRule.json
```

On successful operation, 201 will be returned to state new alert rule creation or 200 will be returned if an existing alert rule was modified.

## Next steps

- Learn about [Log Alerts in Azure Alerts](#)

- Understand [Webhook actions for log alerts](#)
- Learn more about [Application Insights](#)
- Learn more about [log queries](#).

# Shared Image Gallery overview

11/13/2019 • 17 minutes to read • [Edit Online](#)

Shared Image Gallery is a service that helps you build structure and organization around your managed images.

Shared Image Galleries provide:

- Managed global replication of images.
- Versioning and grouping of images for easier management.
- Highly available images with Zone Redundant Storage (ZRS) accounts in regions that support Availability Zones. ZRS offers better resilience against zonal failures.
- Sharing across subscriptions, and even between Active Directory (AD) tenants, using RBAC.
- Scaling your deployments with image replicas in each region.

Using a Shared Image Gallery you can share your images to different users, service principals, or AD groups within your organization. Shared images can be replicated to multiple regions, for quicker scaling of your deployments.

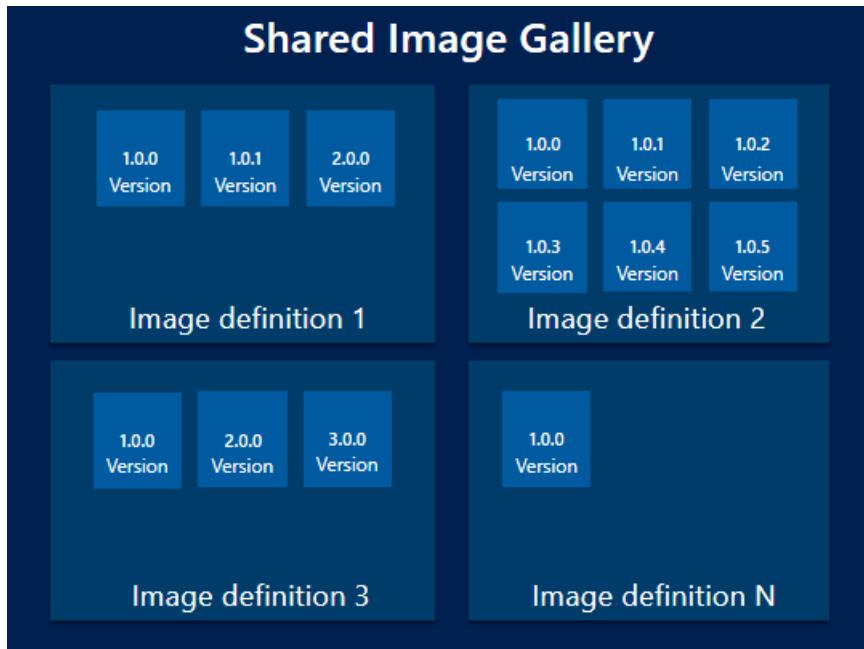
A managed image is a copy of either a full VM (including any attached data disks) or just the OS disk, depending on how you create the image. When you create a VM from the image, a copy of the VHDs in the image are used to create the disks for the new VM. The managed image remains in storage and can be used over and over again to create new VMs.

If you have a large number of managed images that you need to maintain and would like to make them available throughout your company, you can use a Shared Image Gallery as a repository that makes it easy to share your images.

The Shared Image Gallery feature has multiple resource types:

| RESOURCE                | DESCRIPTION                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Managed image</b>    | A basic image that can be used alone or used to create an <b>image version</b> in an image gallery. Managed images are created from <b>generalized</b> VMs. A managed image is a special type of VHD that can be used to make multiple VMs and can now be used to create shared image versions.                                              |
| <b>Snapshot</b>         | A copy of a VHD that can be used to make an <b>image version</b> . Snapshots can be taken from a <b>specialized</b> VM (one that hasn't been generalized) then used alone or with snapshots of data disks, to create a specialized image version.                                                                                            |
| <b>Image gallery</b>    | Like the Azure Marketplace, an <b>image gallery</b> is a repository for managing and sharing images, but you control who has access.                                                                                                                                                                                                         |
| <b>Image definition</b> | Images are defined within a gallery and carry information about the image and requirements for using it within your organization. You can include information like whether the image is generalized or specialized, the operating system, minimum and maximum memory requirements, and release notes. It is a definition of a type of image. |

| RESOURCE             | DESCRIPTION                                                                                                                                                                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Image version</b> | An <b>image version</b> is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an <b>image version</b> to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times. |



## Image definitions

Image definitions are a logical grouping for versions of an image. The image definition holds information about why the image was created, what OS it is for, and information about using the image. An image definition is like a plan for all of the details around creating a specific image. You don't deploy a VM from an image definition, but from the image version created from the definition.

There are three parameters for each image definition that are used in combination - **Publisher**, **Offer** and **SKU**. These are used to find a specific image definition. You can have image versions that share one or two, but not all three values. For example, here are three image definitions and their values:

| IMAGE DEFINITION | PUBLISHER | OFFER   | SKU      |
|------------------|-----------|---------|----------|
| myImage1         | Contoso   | Finance | Backend  |
| myImage2         | Contoso   | Finance | Frontend |
| myImage3         | Testing   | Finance | Frontend |

All three of these have unique sets of values. The format is similar to how you can currently specify publisher, offer, and SKU for [Azure Marketplace images](#) in Azure PowerShell to get the latest version of a Marketplace image. Each image definition needs to have a unique set of these values.

The following are other parameters that can be set on your image definition so that you can more easily track your resources:

- Operating system state - You can set the OS state to [generalized or specialized](#).

- Operating system - can be either Windows or Linux.
- Description - use description to give more detailed information on why the image definition exists. For example, you might have an image definition for your front-end server that has the application pre-installed.
- Eula - can be used to point to an end-user license agreement specific to the image definition.
- Privacy Statement and Release notes - store release notes and privacy statements in Azure storage and provide a URL for accessing them as part of the image definition.
- End-of-life date - attach an end-of-life date to your image definition to be able to use automation to delete old image definitions.
- Tag - you can add tags when you create your image definition. For more information about tags, see [Using tags to organize your resources](#)
- Minimum and maximum vCPU and memory recommendations - if your image has vCPU and memory recommendations, you can attach that information to your image definition.
- Disallowed disk types - you can provide information about the storage needs for your VM. For example, if the image isn't suited for standard HDD disks, you add them to the disallow list.

## Generalized and specialized images

There are two operating system states supported by Shared Image Gallery. Typically images require that the VM used to create the image has been generalized before taking the image. Generalizing is a process that removes machine and user specific information from the VM. For Windows, the Sysprep tool is used. For Linux, you can use `waagent -deprovision` or `-deprovision+user` parameters.

Specialized VMs have not been through a process to remove machine specific information and accounts. Also, VMs created from specialized images do not have an `osProfile` associated with them. This means that specialized images will have some limitations.

- Accounts that could be used to log into the VM can also be used on any VM created using the specialized image that is created from that VM.
- VMs will have the **Computer name** of the VM the image was taken from. You should change the computer name to avoid collisions.
- The `osProfile` is how some sensitive information is passed to the VM, using `secrets`. This may cause issues using KeyVault, WinRM and other functionality that uses `secrets` in the `osProfile`. In some cases, you can use managed service identities (MSI) to work around these limitations.

### IMPORTANT

Specialized images are currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

**Known preview limitations** VMs can only be created from specialized images using the portal or API. There is no CLI or PowerShell support for the preview.

## Regional Support

Source regions are listed in the table below. All public regions can be target regions, but to replicate to Australia Central and Australia Central 2 you need to have your subscription whitelisted. To request whitelisting, go to:

<https://azure.microsoft.com/global-infrastructure/australia/contact/>

| SOURCE REGIONS    |            |             |             |
|-------------------|------------|-------------|-------------|
| Australia Central | China East | South India | West Europe |

| SOURCE REGIONS      |                |                  |                 |
|---------------------|----------------|------------------|-----------------|
| Australia Central 2 | China East 2   | Southeast Asia   | UK South        |
| Australia East      | China North    | Japan East       | UK West         |
| Australia Southeast | China North 2  | Japan West       | US DoD Central  |
| Brazil South        | East Asia      | Korea Central    | US DoD East     |
| Canada Central      | East US        | Korea South      | US Gov Arizona  |
| Canada East         | East US 2      | North Central US | US Gov Texas    |
| Central India       | East US 2 EUAP | North Europe     | US Gov Virginia |
| Central US          | France Central | South Central US | West India      |
| Central US EUAP     | France South   | West Central US  | West US         |
|                     |                |                  | West US 2       |

## Limits

There are limits, per subscription, for deploying resources using Shared Image Galleries:

- 100 shared image galleries, per subscription, per region
- 1,000 image definitions, per subscription, per region
- 10,000 image versions, per subscription, per region
- Any disk attached to the image must be less than or equal to 1TB in size

For more information, see [Check resource usage against limits](#) for examples on how to check your current usage.

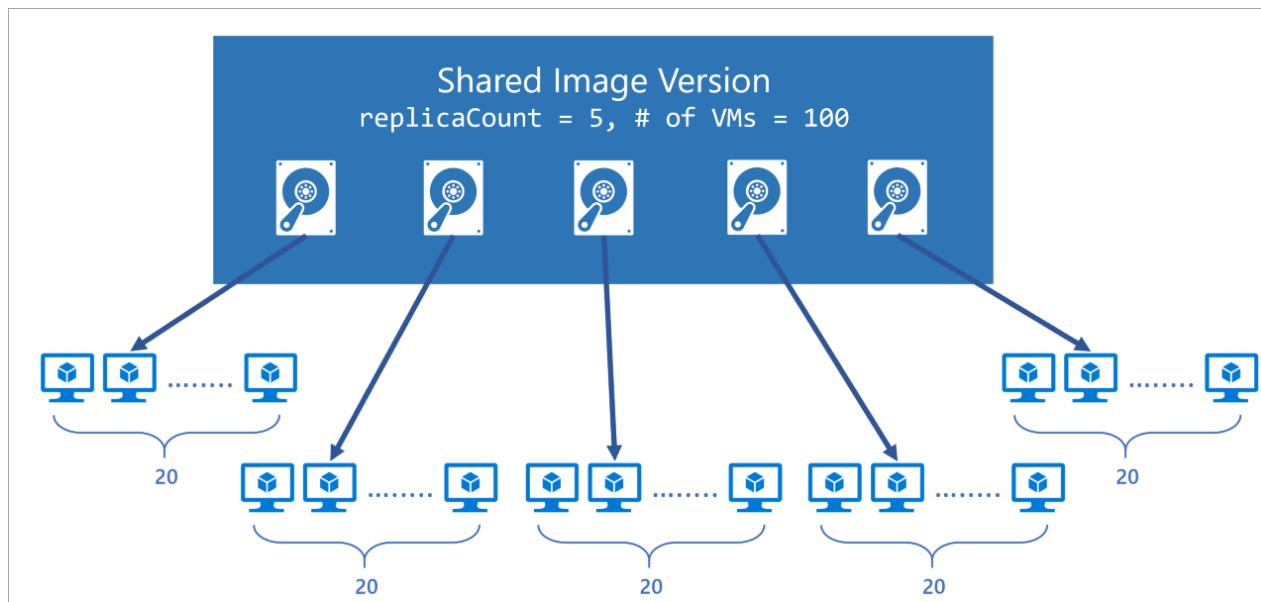
## Scaling

Shared Image Gallery allows you to specify the number of replicas you want Azure to keep of the images. This helps in multi-VM deployment scenarios as the VM deployments can be spread to different replicas reducing the chance of instance creation processing being throttled due to overloading of a single replica.

With Shared Image Gallery, you can now deploy up to a 1,000 VM instances in a virtual machine scale set (up from 600 with managed images). Image replicas provide for better deployment performance, reliability and consistency. You can set a different replica count in each target region, based on the scale needs for the region. Since each replica is a deep copy of your image, this helps scale your deployments linearly with each extra replica. While we understand no two images or regions are the same, here's our general guideline on how to use replicas in a region:

- For non-Virtual Machine Scale Set (VMSS) Deployments - For every 20 VMs that you create concurrently, we recommend you keep one replica. For example, if you are creating 120 VMs concurrently using the same image in a region, we suggest you keep at least 6 replicas of your image.
- For Virtual Machine Scale Set (VMSS) deployments - For every scale set deployment with up to 600 instances, we recommend you keep at least one replica. For example, if you are creating 5 scale sets concurrently, each with 600 VM instances using the same image in a single region, we suggest you keep at least 5 replicas of your image.

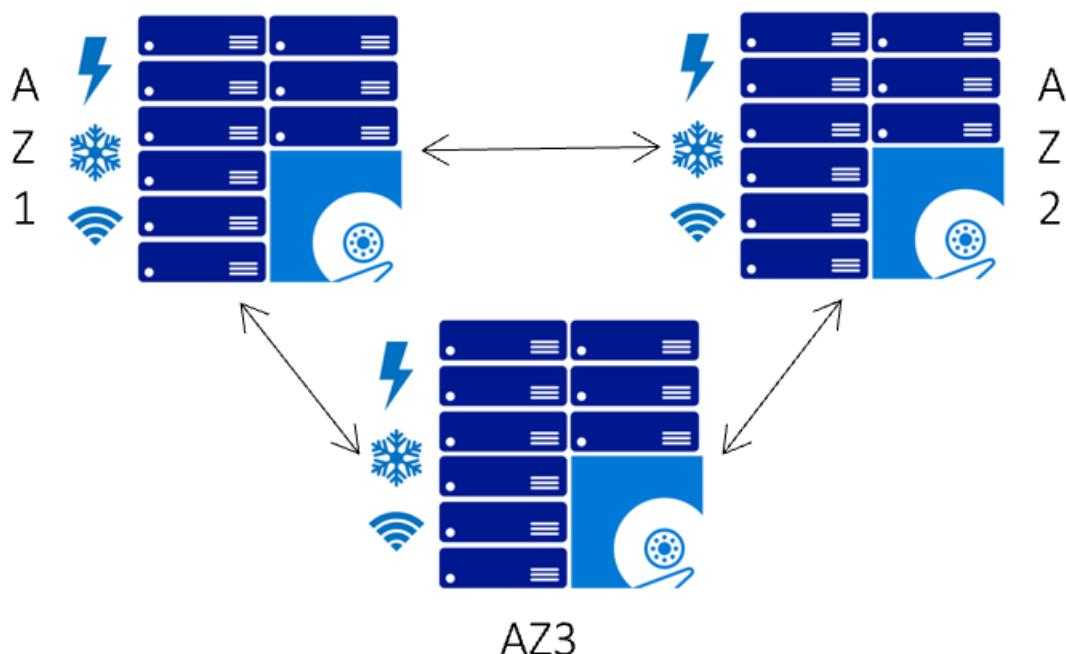
We always recommend you to overprovision the number of replicas due to factors like image size, content and OS type.



## Make your images highly available

Azure Zone Redundant Storage (ZRS) provides resilience against an Availability Zone failure in the region. With the general availability of Shared Image Gallery, you can choose to store your images in ZRS accounts in regions with Availability Zones.

You can also choose the account type for each of the target regions. The default storage account type is Standard\_LRS, but you can choose Standard\_ZRS for regions with Availability Zones. Check the regional availability of ZRS [here](#).

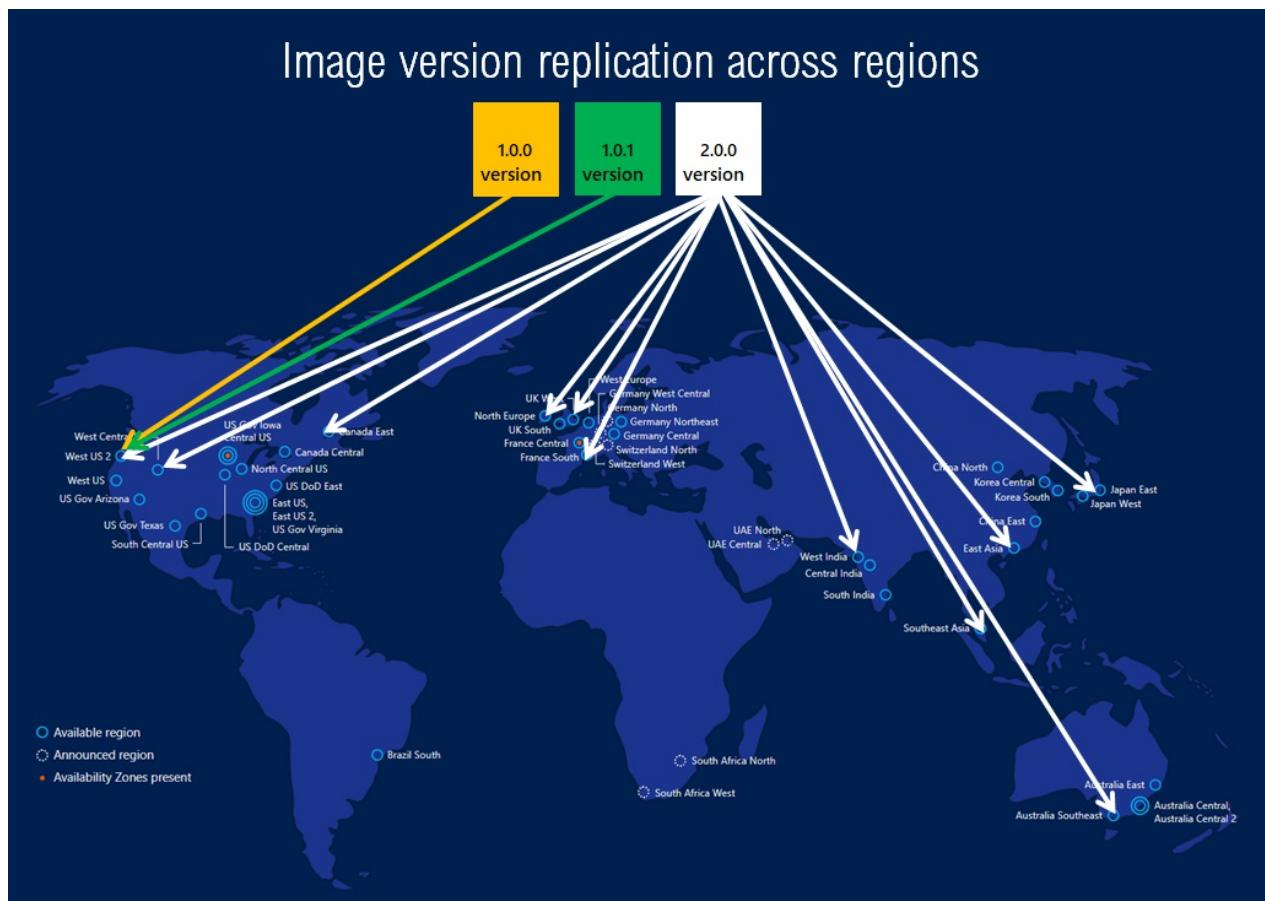


## Replication

Shared Image Gallery also allows you to replicate your images to other Azure regions automatically. Each Shared

Image version can be replicated to different regions depending on what makes sense for your organization. One example is to always replicate the latest image in multi-regions while all older versions are only available in 1 region. This can help save on storage costs for Shared Image versions.

The regions a Shared Image version is replicated to can be updated after creation time. The time it takes to replicate to different regions depends on the amount of data being copied and the number of regions the version is replicated to. This can take a few hours in some cases. While the replication is happening, you can view the status of replication per region. Once the image replication is complete in a region, you can then deploy a VM or scale-set using that image version in the region.



## Access

As the Shared Image Gallery, Image Definition, and Image version are all resources, they can be shared using the built-in native Azure RBAC controls. Using RBAC you can share these resources to other users, service principals, and groups. You can even share access to individuals outside of the tenant they were created within. Once a user has access to the Shared Image version, they can deploy a VM or a Virtual Machine Scale Set. Here is the sharing matrix that helps understand what the user gets access to:

| SHARED WITH USER     | SHARED IMAGE GALLERY | IMAGE DEFINITION | IMAGE VERSION |
|----------------------|----------------------|------------------|---------------|
| Shared Image Gallery | Yes                  | Yes              | Yes           |
| Image Definition     | No                   | Yes              | Yes           |

We recommend sharing at the Gallery level for the best experience. We do not recommend sharing individual image versions. For more information about RBAC, see [Manage access to Azure resources using RBAC](#).

Images can also be shared, at scale, even across tenants using a multi-tenant app registration. For more information about sharing images across tenants, see [Share gallery VM images across Azure tenants](#).

## Billing

There is no extra charge for using the Shared Image Gallery service. You will be charged for the following resources:

- Storage costs of storing the Shared Image versions. Cost depends on the number of replicas of the image version and the number of regions the version is replicated to. For example, if you have 2 images and both are replicated to 3 regions, then you will be charged for 6 managed disks based on their size. For more information, see [Managed Disks pricing](#).
- Network egress charges for replication of the first image version from the source region to the replicated regions. Subsequent replicas are handled within the region, so there are no additional charges.

## Updating resources

Once created, you can make some changes to the image gallery resources. These are limited to:

Shared image gallery:

- Description

Image definition:

- Recommended vCPUs
- Recommended memory
- Description
- End of life date

Image version:

- Regional replica count
- Target regions
- Exclude from latest
- End of life date

## SDK support

The following SDKs support creating Shared Image Galleries:

- [.NET](#)
- [Java](#)
- [Node.js](#)
- [Python](#)
- [Go](#)

## Templates

You can create Shared Image Gallery resource using templates. There are several Azure Quickstart Templates available:

- [Create a Shared Image Gallery](#)
- [Create an Image Definition in a Shared Image Gallery](#)
- [Create an Image Version in a Shared Image Gallery](#)
- [Create a VM from Image Version](#)

# Frequently asked questions

- How can I list all the Shared Image Gallery resources across subscriptions?
- Can I move my existing image to the shared image gallery?
- Can I create an image version from a specialized disk?
- Can I move the Shared Image Gallery resource to a different subscription after it has been created?
- Can I replicate my image versions across clouds such as Azure China 21Vianet or Azure Germany or Azure Government Cloud?
- Can I replicate my image versions across subscriptions?
- Can I share image versions across Azure AD tenants?
- How long does it take to replicate image versions across the target regions?
- What is the difference between source region and target region?
- How do I specify the source region while creating the image version?
- How do I specify the number of image version replicas to be created in each region?
- Can I create the shared image gallery in a different location than the one for the image definition and image version?
- What are the charges for using the Shared Image Gallery?
- What API version should I use to create Shared Image Gallery and Image Definition and Image Version?
- What API version should I use to create Shared VM or Virtual Machine Scale Set out of the Image Version?

## How can I list all the Shared Image Gallery resources across subscriptions?

To list all the Shared Image Gallery resources across subscriptions that you have access to on the Azure portal, follow the steps below:

1. Open the [Azure portal](#).
2. Go to **All Resources**.
3. Select all the subscriptions under which you'd like to list all the resources.
4. Look for resources of type **Private gallery**.

To see the image definitions and image versions, you should also select **Show hidden types**.

To list all the Shared Image Gallery resources across subscriptions that you have permissions to, use the following command in the Azure CLI:

```
az account list -otsv --query "[].id" | xargs -n 1 az sig list --subscription
```

## Can I move my existing image to the shared image gallery?

Yes. There are 3 scenarios based on the types of images you may have.

Scenario 1: If you have a managed image in the same subscription as your SIG, then you can create an image definition and image version from it.

Scenario 2: If you have an unmanaged image in the same subscription as your SIG, you can create a managed image from it, and then create an image definition and image version from it.

Scenario 3: If you have a VHD in your local file system, then you need to upload the VHD to a managed image, then you can create an image definition and image version from it.

- If the VHD is of a Windows VM, see [Upload a VHD](#).
- If the VHD is for a Linux VM, see [Upload a VHD](#)

## Can I create an image version from a specialized disk?

Yes, support for specialized disks as images is in preview. You can only create a VM from a specialized image using the portal ([Windows](#) or [Linux](#)) and API. There is no PowerShell support for the preview.

### **Can I move the Shared Image Gallery resource to a different subscription after it has been created?**

No, you cannot move the shared image gallery resource to a different subscription. However, you will be able to replicate the image versions in the gallery to other regions as required.

### **Can I replicate my image versions across clouds such as Azure China 21Vianet or Azure Germany or Azure Government Cloud?**

No, you cannot replicate image versions across clouds.

### **Can I replicate my image versions across subscriptions?**

No, you may replicate the image versions across regions in a subscription and use it in other subscriptions through RBAC.

### **Can I share image versions across Azure AD tenants?**

Yes, you can use RBAC to share to individuals across tenants. But, to share at scale, see "Share gallery images across Azure tenants" using [PowerShell](#) or [CLI](#).

### **How long does it take to replicate image versions across the target regions?**

The image version replication time is entirely dependent on the size of the image and the number of regions it is being replicated to. However, as a best practice, it is recommended that you keep the image small, and the source and target regions close for best results. You can check the status of the replication using the -ReplicationStatus flag.

### **What is the difference between source region and target region?**

Source region is the region in which your image version will be created, and target regions are the regions in which a copy of your image version will be stored. For each image version, you can only have one source region. Also, make sure that you pass the source region location as one of the target regions when you create an image version.

### **How do I specify the source region while creating the image version?**

While creating an image version, you can use the **--location** tag in CLI and the **-Location** tag in PowerShell to specify the source region. Please ensure the managed image that you are using as the base image to create the image version is in the same location as the location in which you intend to create the image version. Also, make sure that you pass the source region location as one of the target regions when you create an image version.

### **How do I specify the number of image version replicas to be created in each region?**

There are two ways you can specify the number of image version replicas to be created in each region:

1. The regional replica count which specifies the number of replicas you want to create per region.
2. The common replica count which is the default per region count in case regional replica count is not specified.

To specify the regional replica count, pass the location along with the number of replicas you want to create in that region: "South Central US=2".

If regional replica count is not specified with each location, then the default number of replicas will be the common replica count that you specified.

To specify the common replica count in CLI, use the **--replica-count** argument in the `az sig image-version create` command.

### **Can I create the shared image gallery in a different location than the one for the image definition and image version?**

Yes, it is possible. But, as a best practice, we encourage you to keep the resource group, shared image gallery, image definition, and image version in the same location.

## **What are the charges for using the Shared Image Gallery?**

There are no charges for using the Shared Image Gallery service, except the storage charges for storing the image versions and network egress charges for replicating the image versions from source region to target regions.

## **What API version should I use to create Shared Image Gallery and Image Definition and Image Version?**

To work with shared image galleries, image definitions, and image versions, we recommend you use API version 2018-06-01. Zone Redundant Storage (ZRS) requires version 2019-03-01 or later.

## **What API version should I use to create Shared VM or Virtual Machine Scale Set out of the Image Version?**

For VM and Virtual Machine Scale Set deployments using an image version, we recommend you use API version 2018-04-01 or higher.

## Next steps

Learn how to [deploy shared images using Azure PowerShell](#).

# Create a shared image gallery with Azure PowerShell

11/13/2019 • 8 minutes to read • [Edit Online](#)

A [Shared Image Gallery](#) simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap deployment tasks like preloading applications, application configurations, and other OS configurations.

The Shared Image Gallery lets you share your custom VM images with others in your organization, within or across regions, within an AAD tenant. Choose which images you want to share, which regions you want to make them available in, and who you want to share them with. You can create multiple galleries so that you can logically group shared images.

The gallery is a top-level resource that provides full role-based access control (RBAC). Images can be versioned, and you can choose to replicate each image version to a different set of Azure regions. The gallery only works with Managed Images.

The Shared Image Gallery feature has multiple resource types. We will be using or building these in this article:

| RESOURCE                | DESCRIPTION                                                                                                                                                                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Managed image</b>    | This is a basic image that can be used alone or used to create an <b>image version</b> in an image gallery. Managed images are created from generalized VMs. A managed image is a special type of VHD that can be used to make multiple VMs and can now be used to create shared image versions.                                        |
| <b>Image gallery</b>    | Like the Azure Marketplace, an <b>image gallery</b> is a repository for managing and sharing images, but you control who has access.                                                                                                                                                                                                    |
| <b>Image definition</b> | Images are defined within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It is a definition of a type of image.                                                             |
| <b>Image version</b>    | An <b>image version</b> is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an <b>image version</b> to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times. |

For every 20 VMs that you create concurrently, we recommend you keep one replica. For example, if you are creating 120 VMs concurrently using the same image in a region, we suggest you keep at least 6 replicas of your image. For more information, see [Scaling](#).

## Before you begin

To complete the example in this article, you must have an existing managed image. You can follow [Tutorial: Create a custom image of an Azure VM with Azure PowerShell](#) to create one if needed. If the managed image contains a data disk, the data disk size cannot be more than 1 TB.

When working through this article, replace the resource group and VM names where needed.

## 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.

## Get the managed image

You can see a list of images that are available in a resource group using [Get-AzImage](#). Once you know the image name and what resource group it is in, you can use `Get-AzImage` again to get the image object and store it in a variable to use later. This example gets an image named *myImage* from the "myResourceGroup" resource group and assigns it to the variable `$managedImage`.

```
$managedImage = Get-AzImage `
-ImageName myImage `
-ResourceGroupName myResourceGroup
```

## Create an image gallery

An image gallery is the primary resource used for enabling image sharing. Allowed characters for Gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name cannot contain dashes. Gallery names must be unique within your subscription.

Create an image gallery using [New-AzGallery](#). The following example creates a gallery named *myGallery* in the *myGalleryRG* resource group.

```
$resourceGroup = New-AzResourceGroup `
-Name 'myGalleryRG' `
-Location 'West Central US'
$gallery = New-AzGallery `
-GalleryName 'myGallery' `
-ResourceGroupName $resourceGroup.ResourceGroupName `
-Location $resourceGroup.Location `
-Description 'Shared Image Gallery for my organization'
```

## Create an image definition

Image definitions create a logical grouping for images. They are used to manage information about the image versions that are created within them. Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes and periods. For more information about the values you can specify for an image definition, see [Image definitions](#).

Create the image definition using [New-AzGalleryImageDefinition](#). In this example, the gallery image is named *myGalleryImage*.

```
$galleryImage = New-AzGalleryImageDefinition `
 -GalleryName $gallery.Name `
 -ResourceGroupName $resourceGroup.ResourceGroupName `
 -Location $gallery.Location `
 -Name 'myImageDefinition' `
 -OsState generalized `
 -OsType Windows `
 -Publisher 'myPublisher' `
 -Offer 'myOffer' `
 -Sku 'mySKU'
```

## Create an image version

Create an image version from a managed image using [New-AzGalleryImageVersion](#).

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*.

In this example, the image version is *1.0.0* and it's replicated to both *West Central US* and *South Central US* datacenters. When choosing target regions for replication, remember that you also have to include the *source* region as a target for replication.

```
$region1 = @{Name='South Central US';ReplicaCount=1}
$region2 = @{Name='West Central US';ReplicaCount=2}
$targetRegions = @($region1,$region2)
$job = $imageVersion = New-AzGalleryImageVersion `
 -GalleryImageDefinitionName $galleryImage.Name `
 -GalleryImageVersionName '1.0.0' `
 -GalleryName $gallery.Name `
 -ResourceGroupName $resourceGroup.ResourceGroupName `
 -Location $resourceGroup.Location `
 -TargetRegion $targetRegions `
 -Source $managedImage.Id.ToString() `
 -PublishingProfileEndOfLifeDate '2020-01-01' `
 -asJob
```

It can take a while to replicate the image to all of the target regions, so we have created a job so we can track the progress. To see the progress of the job, type `$job.State`.

```
$job.State
```

### NOTE

You need to wait for the image version to completely finish being built and replicated before you can use the same managed image to create another image version.

You can also store your image version in [Zone Redundant Storage](#) by adding `-StorageAccountType Standard_ZRS` when you create the image version.

## Share the gallery

We recommend that you share access at the image gallery level. Use an email address and the [Get-AzADUser](#) cmdlet to get the object ID for the user, then use [New-AzRoleAssignment](#) to give them access to the gallery. Replace the example email, alinne\_montes@contoso.com in this example, with your own information.

```

Get the object ID for the user
$user = Get-AzADUser -StartsWith alinne_montes@contoso.com
Grant access to the user for our gallery
New-AzRoleAssignment `
 -ObjectId $user.Id `
 -RoleDefinitionName Reader `
 -ResourceName $gallery.Name `
 -ResourceType Microsoft.Compute/galleries `
 -ResourceGroupName $resourceGroup.ResourceGroupName

```

## Create VMs from an image

Once the image version is complete, you can create one or more new VMs. Using the [New-AzVM](#) cmdlet.

This example creates a VM named *myVMfromImage*, in the *myResourceGroup* in the *South Central US* datacenter.

```

$resourceGroup = "myResourceGroup"
.setLocation = "South Central US"
$vmName = "myVMfromImage"

Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

Network pieces
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location $location `
 -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location $location `
 -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp `
 -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `
 -DestinationPortRange 3389 -Access Allow
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location `
 -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location $location `
 -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

Create a virtual machine configuration using $imageVersion.Id to specify the shared image
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1_v2 | `
Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `
Set-AzVMSourceImage -Id $imageVersion.Id | `
Add-AzVMNetworkInterface -Id $nic.Id

Create a virtual machine
New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

```

## Shared image management

Here are some examples of common management tasks and how to complete them using PowerShell.

List all galleries by name.

```

$galleries = Get-AzResource -ResourceType Microsoft.Compute/galleries
$galleries.Name

```

List all image definitions by name.

```
$imageDefinitions = Get-AzResource -ResourceType Microsoft.Compute/galleries/images
$imageDefinitions.Name
```

List all image versions by name.

```
$imageVersions = Get-AzResource -ResourceType Microsoft.Compute/galleries/images/versions
$imageVersions.Name
```

Delete an image version. This example deletes the image version named *1.0.0*.

```
Remove-AzGalleryImageVersion `
-GalleryImageDefinitionName myImageDefinition `
-GalleryName myGallery `
-Name 1.0.0 `
-ResourceGroupName myGalleryRG
```

## Update resources

There are some limitations on what can be updated. The following items can be updated:

Shared image gallery:

- Description

Image definition:

- Recommended vCPUs
- Recommended memory
- Description
- End of life date

Image version:

- Regional replica count
- Target regions
- Exclusion from latest
- End of life date

If you plan on adding replica regions, do not delete the source managed image. The source managed image is needed for replicating the image version to additional regions.

To update the description of a gallery, use [Update-AzGallery](#).

```
Update-AzGallery `
-Name $gallery.Name `
-ResourceGroupName $resourceGroup.Name
```

This example shows how to use [Update-AzGalleryImageDefinition](#) to update the end-of-life date for our image definition.

```
Update-AzGalleryImageDefinition `
 -GalleryName $gallery.Name `
 -Name $galleryImage.Name `
 -ResourceGroupName $resourceGroup.Name `
 -EndOfLifeDate 01/01/2030
```

This example shows how to use [Update-AzGalleryImageVersion](#) to exclude this image version from being used as the *latest* image.

```
Update-AzGalleryImageVersion `
 -GalleryImageDefinitionName $galleryImage.Name `
 -GalleryName $gallery.Name `
 -Name $galleryVersion.Name `
 -ResourceGroupName $resourceGroup.Name `
 -PublishingProfileExcludeFromLatest
```

## Clean up resources

When deleting resources, you need to start with last item in the nested resources - the image version. Once versions are deleted, you can delete the image definition. You can't delete the gallery until all resources beneath it have been deleted.

```
$resourceGroup = "myResourceGroup"
$gallery = "myGallery"
$imageDefinition = "myImageDefinition"
$imageVersion = "myImageVersion"

Remove-AzGalleryImageVersion `
 -GalleryImageDefinitionName $imageDefinition `
 -GalleryName $gallery `
 -Name $imageVersion `
 -ResourceGroupName $resourceGroup

Remove-AzGalleryImageDefinition `
 -ResourceGroupName $resourceGroup `
 -GalleryName $gallery `
 -GalleryImageDefinitionName $imageDefinition

Remove-AzGallery `
 -Name $gallery `
 -ResourceGroupName $resourceGroup

Remove-AzResourceGroup -Name $resourceGroup
```

## Next steps

[Azure Image Builder \(preview\)](#) can help automate image version creation, you can even use it to update and [create a new image version from an existing image version](#).

You can also create Shared Image Gallery resource using templates. There are several Azure Quickstart Templates available:

- [Create a Shared Image Gallery](#)
- [Create an Image Definition in a Shared Image Gallery](#)
- [Create an Image Version in a Shared Image Gallery](#)
- [Create a VM from Image Version](#)

For more information about Shared Image Galleries, see the [Overview](#). If you run into issues, see [Troubleshooting shared image galleries](#).

# Create an Azure Shared Image Gallery using the portal

12/9/2019 • 10 minutes to read • [Edit Online](#)

A [Shared Image Gallery](#) simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap deployment tasks like preloading applications, application configurations, and other OS configurations.

The Shared Image Gallery lets you share your custom VM images with others in your organization, within or across regions, within an AAD tenant. Choose which images you want to share, which regions you want to make them available in, and who you want to share them with. You can create multiple galleries so that you can logically group shared images.

The gallery is a top-level resource that provides full role-based access control (RBAC). Images can be versioned, and you can choose to replicate each image version to a different set of Azure regions. The gallery only works with Managed Images.

The Shared Image Gallery feature has multiple resource types. We will be using or building these in this article:

| RESOURCE                | DESCRIPTION                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Managed image</b>    | A basic image that can be used alone or used to create an <b>image version</b> in an image gallery. Managed images are created from <a href="#">generalized</a> VMs. A managed image is a special type of VHD that can be used to make multiple VMs and can now be used to create shared image versions.                                     |
| <b>Snapshot</b>         | A copy of a VHD that can be used to make an <b>image version</b> . Snapshots can be taken from a <a href="#">specialized</a> VM (one that hasn't been generalized) then used alone or with snapshots of data disks, to create a specialized image version.                                                                                   |
| <b>Image gallery</b>    | Like the Azure Marketplace, an <b>image gallery</b> is a repository for managing and sharing images, but you control who has access.                                                                                                                                                                                                         |
| <b>Image definition</b> | Images are defined within a gallery and carry information about the image and requirements for using it within your organization. You can include information like whether the image is generalized or specialized, the operating system, minimum and maximum memory requirements, and release notes. It is a definition of a type of image. |
| <b>Image version</b>    | An <b>image version</b> is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an <b>image version</b> to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.      |

## IMPORTANT

Specialized images are currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

**Known preview limitations** VMs can only be created from specialized images using the portal or API. There is no CLI or PowerShell support for the preview.

## Before you begin

To complete the example in this article, you must have an existing managed image of a generalized VM, or a snapshot of a specialized VM. You can follow [Tutorial: Create a custom image of an Azure VM with Azure PowerShell](#) to create a managed image, or [Create a snapshot](#) for a specialized VM. For both managed images and snapshots, the data disk size cannot be more than 1 TB.

When working through this article, replace the resource group and VM names where needed.

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

### NOTE

If you registered to use Shared Image Galleries during the preview, you might need to re-register the `Microsoft.Compute` provider. Open [Cloud Shell](#) and type: `az provider register -n Microsoft.Compute`

## Create an image gallery

An image gallery is the primary resource used for enabling image sharing. Allowed characters for Gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name cannot contain dashes. Gallery names must be unique within your subscription.

The following example creates a gallery named *myGallery* in the *myGalleryRG* resource group.

1. Select **Create a resource** in the upper left-hand corner of the Azure portal.
2. Use the type **Shared image gallery** in the search box and select **Shared image gallery** in the results.
3. In the **Shared image gallery** page, click **Create**.
4. Select the correct subscription.
5. In **Resource group**, select **Create new** and type *myGalleryRG* for the name.
6. In **Name**, type *myGallery* for the name of the gallery.
7. Leave the default for **Region**.
8. You can type a short description of the gallery, like *My image gallery for testing*. and then click **Review + create**.
9. After validation passes, select **Create**.
10. When the deployment is finished, select **Go to resource**.

## Create an image definition

Image definitions create a logical grouping for images. They are used to manage information about the image versions that are created within them. Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes and periods. For more information about the values you can specify for an image definition, see

## Image definitions.

Create the gallery image definition inside of your gallery. In this example, the gallery image is named *myImageDefinition*.

1. On the page for your new image gallery, select **Add a new image definition** from the top of the page.
2. For **Image definition name**, type *myImageDefinition*.
3. For **Operating system**, select the correct option based on your source VM.
4. For **VM generation**, select the option based on your source VM. In most cases, this will be *Gen 1*. For more information, see [Support for generation 2 VMs](#).
5. For **Operating system state**, select the option based on your source VM. For more information, see [Generalized and specialized](#).
6. For **Publisher**, type *myPublisher*.
7. For **Offer**, type *myOffer*.
8. For **SKU**, type *mySKU*.
9. When finished, select **Review + create**.
10. After the image definition passes validation, select **Create**.
11. When the deployment is finished, select **Go to resource**.

## Create an image version

Create an image version from a managed image. In this example, the image version is *1.0.0* and it's replicated to both *West Central US* and *South Central US* datacenters. When choosing target regions for replication, remember that you also have to include the *source* region as a target for replication.

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*.

The steps for creating an image version are slightly different, depending on whether the source is a generalized image or a snapshot of a specialized VM.

### Option: Generalized

1. In the page for your image definition, select **Add version** from the top of the page.
2. In **Region**, select the region where your managed image is stored. Image versions need to be created in the same region as the managed image they are created from.
3. For **Name**, type *1.0.0*. The image version name should follow *major.minor.patch* format using integers.
4. In **Source image**, select your source managed image from the drop-down.
5. In **Exclude from latest**, leave the default value of *No*.
6. For **End of life date**, select a date from the calendar that is a couple of months in the future.
7. In **Replication**, leave the **Default replica count** as *1*. You need to replicate to the source region, so leave the first replica as the default and then pick a second replica region to be *East US*.
8. When you are done, select **Review + create**. Azure will validate the configuration.
9. When image version passes validation, select **Create**.
10. When the deployment is finished, select **Go to resource**.

It can take a while to replicate the image to all of the target regions.

### Option: Specialized

1. In the page for your image definition, select **Add version** from the top of the page.
2. In **Region**, select the region where your snapshot is stored. Image versions need to be created in the same region as the source they are created from.
3. For **Name**, type *1.0.0*. The image version name should follow *major.minor.patch* format using integers.

4. In **OS disk snapshot**, select the snapshot from your source VM from the drop-down. If your source VM had a data disk that you would like to include, select the correct **LUN** number from the drop-down, and then select the snapshot of the data disk for **Data disk snapshot**.
5. In **Exclude from latest**, leave the default value of *No*.
6. For **End of life date**, select a date from the calendar that is a couple of months in the future.
7. In **Replication**, leave the **Default replica count** as 1. You need to replicate to the source region, so leave the first replica as the default and then pick a second replica region to be *East US*.
8. When you are done, select **Review + create**. Azure will validate the configuration.
9. When image version passes validation, select **Create**.
10. When the deployment is finished, select **Go to resource**.

## Share the gallery

We recommend that you share access at the image gallery level. The following walks you through sharing the gallery that you just created.

1. Open the [Azure portal](#).
2. In the menu at the left, select **Resource groups**.
3. In the list of resource groups, select **myGalleryRG**. The blade for your resource group will open.
4. In the menu on the left of the **myGalleryRG** page, select **Access control (IAM)**.
5. Under **Add a role assignment**, select **Add**. The **Add a role assignment** pane will open.
6. Under **Role**, select **Reader**.
7. Under **assign access to**, leave the default of **Azure AD user, group, or service principal**.
8. Under **Select**, type in the email address of the person that you would like to invite.
9. If the user is outside of your organization, you will see the message **This user will be sent an email that enables them to collaborate with Microsoft**. Select the user with the email address and then click **Save**.

If the user is outside of your organization, they will get an email invitation to join the organization. The user needs to accept the invitation, then they will be able to see the gallery and all of the image definitions and versions in their list of resources.

## Create VMs

Now you can create one or more new VMs. This example creates a VM named *myVM*, in the *myResourceGroup*, in the *East US* datacenter.

1. Go to your image definition. You can use the resource filter to show all image definitions available.
2. On the page for your image definition, select **Create VM** from the menu at the top of the page.
3. For **Resource group**, select **Create new** and type *myResourceGroup* for the name.
4. In **Virtual machine name**, type *myVM*.
5. For **Region**, select *East US*.
6. For **Availability options**, leave the default of *No infrastructure redundancy required*.
7. The value for **Image** is automatically filled with the **latest** image version if you started from the page for the image definition.
8. For **Size**, choose a VM size from the list of available sizes and then choose **Select**.
9. Under **Administrator account**, if the image was generalized, you need to provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the [defined complexity requirements](#). If your image was specialized, the username and password fields will greyed out because the username and password for the source VM are used.
10. If you want to allow remote access to the VM, under **Public inbound ports**, choose **Allow selected ports** and then select **RDP (3389)** from the drop-down. If you don't want to allow remote access to the VM, leave **None**

selected for **Public inbound ports**.

11. When you are finished, select the **Review + create** button at the bottom of the page.
12. After the VM passes validation, select **Create** at the bottom of the page to start the deployment.

## Clean up resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select **Delete**, then confirm the name of the resource group to delete.

If you want to delete individual resources, you need to delete them in reverse order. For example, to delete an image definition, you need to delete all of the image versions created from that image.

## Next steps

You can also create Shared Image Gallery resource using templates. There are several Azure Quickstart Templates available:

- [Create a Shared Image Gallery](#)
- [Create an Image Definition in a Shared Image Gallery](#)
- [Create an Image Version in a Shared Image Gallery](#)
- [Create a VM from Image Version](#)

For more information about Shared Image Galleries, see the [Overview](#). If you run into issues, see [Troubleshooting shared image galleries](#).

# Create a shared image gallery with the Azure CLI

11/13/2019 • 7 minutes to read • [Edit Online](#)

A [Shared Image Gallery](#) simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap configurations such as preloading applications, application configurations, and other OS configurations.

The Shared Image Gallery lets you share your custom VM images with others in your organization, within or across regions, within an AAD tenant. Choose which images you want to share, which regions you want to make them available in, and who you want to share them with. You can create multiple galleries so that you can logically group shared images.

The gallery is a top-level resource that provides full role-based access control (RBAC). Images can be versioned, and you can choose to replicate each image version to a different set of Azure regions. The gallery only works with Managed Images.

The Shared Image Gallery feature has multiple resource types. We will be using or building these in this article:

| RESOURCE                | DESCRIPTION                                                                                                                                                                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Managed image</b>    | This is a basic image that can be used alone or used to create an <b>image version</b> in an image gallery. Managed images are created from generalized VMs. A managed image is a special type of VHD that can be used to make multiple VMs and can now be used to create shared image versions.                                        |
| <b>Image gallery</b>    | Like the Azure Marketplace, an <b>image gallery</b> is a repository for managing and sharing images, but you control who has access.                                                                                                                                                                                                    |
| <b>Image definition</b> | Images are defined within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It is a definition of a type of image.                                                             |
| <b>Image version</b>    | An <b>image version</b> is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an <b>image version</b> to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times. |

## Before you begin

To complete the example in this article, you must have an existing managed image of a generalized VM. For more information, see [Tutorial: Create a custom image of an Azure VM with the Azure CLI](#). If the managed image contains a data disk, the data disk size cannot be more than 1 TB.

## 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/bash>. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

If you prefer to install and use the CLI locally, see [Install Azure CLI](#).

## Create an image gallery

An image gallery is the primary resource used for enabling image sharing. Allowed characters for Gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name cannot contain dashes. Gallery names must be unique within your subscription.

Create an image gallery using [az sig create](#). The following example creates a gallery named *myGallery* in *myGalleryRG*.

```
az group create --name myGalleryRG --location WestCentralUS
az sig create --resource-group myGalleryRG --gallery-name myGallery
```

## Create an image definition

Image definitions create a logical grouping for images. They are used to manage information about the image versions that are created within them. Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes, and periods. For more information about the values you can specify for an image definition, see [Image definitions](#).

Create an initial image definition in the gallery using [az sig image-definition create](#).

```
az sig image-definition create \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition \
--publisher myPublisher \
--offer myOffer \
--sku 16.04-LTS \
--os-type Linux
```

## Create an image version

Create versions of the image as needed using [az image gallery create-image-version](#). You will need to pass in the ID of the managed image to use as a baseline for creating the image version. You can use [az image list](#) to get information about images that are in a resource group.

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*.

In this example, the version of our image is *1.0.0* and we are going to create 2 replicas in the *West Central US* region, 1 replica in the *South Central US* region and 1 replica in the *East US 2* region using zone-redundant storage.

```
az sig image-version create \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition \
--gallery-image-version 1.0.0 \
--target-regions "WestCentralUS" "SouthCentralUS=1" "EastUS2=1=Standard_ZRS" \
--replica-count 2 \
--managed-image "/subscriptions/<subscription
ID>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/myImage"
```

#### NOTE

You need to wait for the image version to completely finish being built and replicated before you can use the same managed image to create another image version.

You can also store all of your image version replicas in [Zone Redundant Storage](#) by adding

```
--storage-account-type standard_zrs
```

## Share the gallery

We recommend that you share with other users at the gallery level. To get the object ID of your gallery, use [az sig show](#).

```
az sig show \
 --resource-group myGalleryRG \
 --gallery-name myGallery \
 --query id
```

Use the object ID as a scope, along with an email address and [az role assignment create](#) to give a user access to the shared image gallery.

```
az role assignment create --role "Reader" --assignee <email address> --scope <gallery ID>
```

## Create a VM

Create a VM from the latest image version using [az vm create](#).

```
az vm create\
 --resource-group myGalleryRG \
 --name myVM \
 --image "/subscriptions/<subscription ID where the gallery is located>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myGallery/images/myImageDefinition" \
 --generate-ssh-keys
```

You can also use a specific version by using the image version ID for the `--image` parameter. For example, to use image version 1.0.0 type:

```
--image "/subscriptions/<subscription ID where the gallery is located>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myGallery/images/myImageDefinition/versions/1.0.0"
```

## Using RBAC to share images

You can share images across subscriptions using Role-Based Access Control (RBAC). Any user that has read permissions to an image version, even across subscriptions, will be able to deploy a Virtual Machine using the image version.

For more information about how to share resources using RBAC, see [Manage access using RBAC and Azure CLI](#).

## List information

Get the location, status and other information about the available image galleries using [az sig list](#).

```
az sig list -o table
```

List the image definitions in a gallery, including information about OS type and status, using [az sig image-definition list](#).

```
az sig image-definition list --resource-group myGalleryRG --gallery-name myGallery -o table
```

List the shared image versions in a gallery, using [az sig image-version list](#).

```
az sig image-version list --resource-group myGalleryRG --gallery-name myGallery --gallery-image-definition myImageDefinition -o table
```

Get the ID of an image version using [az sig image-version show](#).

```
az sig image-version show \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition \
--gallery-image-version 1.0.0 \
--query "id"
```

## Update resources

There are some limitations on what can be updated. The following items can be updated:

Shared image gallery:

- Description

Image definition:

- Recommended vCPUs
- Recommended memory
- Description
- End of life date

Image version:

- Regional replica count
- Target regions
- Exclusion from latest
- End of life date

If you plan on adding replica regions, do not delete the source managed image. The source managed image is needed for replicating the image version to additional regions.

Update the description of a gallery using ([az sig update](#).

```
az sig update \
--gallery-name myGallery \
--resource-group myGalleryRG \
--set description="My updated description."
```

Update the description of an image definition using [az sig image-definition update](#).

```
az sig image-definition update \
--gallery-name myGallery\
--resource-group myGalleryRG \
--gallery-image-definition myImageDefinition \
--set description="My updated description."
```

Update an image version to add a region to replicate to using [az sig image-version update](#). This change will take a while as the image gets replicated to the new region.

```
az sig image-version update \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition \
--gallery-image-version 1.0.0 \
--add publishingProfile.targetRegions name=eastus
```

## Delete resources

You have to delete resources in reverse order, by deleting the image version first. After you delete all of the image versions, you can delete the image definition. After you delete all image definitions, you can delete the gallery.

Delete an image version using [az sig image-version delete](#).

```
az sig image-version delete \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition \
--gallery-image-version 1.0.0
```

Delete an image definition using [az sig image-definition delete](#).

```
az sig image-definition delete \
--resource-group myGalleryRG \
--gallery-name myGallery \
--gallery-image-definition myImageDefinition
```

Delete an image gallery using [az sig delete](#).

```
az sig delete \
--resource-group myGalleryRG \
--gallery-name myGallery
```

## Next steps

[Azure Image Builder \(preview\)](#) can help automate image version creation, you can even use it to update and [create a new image version from an existing image version](#).

You can also create Shared Image Gallery resources using templates. There are several Azure Quickstart Templates available:

- [Create a Shared Image Gallery](#)
- [Create an Image Definition in a Shared Image Gallery](#)
- [Create an Image Version in a Shared Image Gallery](#)
- [Create a VM from Image Version](#)

For more information about Shared Image Galleries, see the [Overview](#). If you run into issues, see [Troubleshooting](#)

[shared image galleries.](#)

# Share gallery VM images across Azure tenants

11/13/2019 • 4 minutes to read • [Edit Online](#)

Shared Image Galleries let you share images using RBAC. You can use RBAC to share images within your tenant, and even to individuals outside of your tenant. For more information about this simple sharing option, see the [Share the gallery](#).

But, if you want to share images outside of your Azure tenant, at scale, you should create an app registration to facilitate sharing. Using an app registration can enable more complex sharing scenarios, like:

- Managing shared images when one company acquires another, and the Azure infrastructure is spread across separate tenants.
- Azure Partners manage Azure infrastructure on behalf of their customers. Customization of images is done within the partners tenant, but the infrastructure deployments will happen in the customer's tenant.

## Create the app registration

Create an application registration that will be used by both tenants to share the image gallery resources.

1. Open the [App registrations \(preview\)](#) in the Azure portal.
2. Select **New registration** from the menu at the top of the page.
3. In **Name**, type *myGalleryApp*.
4. In **Supported account types**, select **Accounts in any organizational directory and personal Microsoft accounts**.
5. In **Redirect URI**, type <https://www.microsoft.com> and then select **Register**. After the app registration has been created, the overview page will open.
6. On the overview page, copy the **Application (client) ID** and save for use later.
7. Select **Certificates & secrets**, and then select **New client secret**.
8. In **Description**, type *Shared image gallery cross-tenant app secret*.
9. In **Expires**, leave the default of **In 1 year** and then select **Add**.
10. Copy the value of the secret and save it to a safe place. You cannot retrieve it after you leave the page.

Give the app registration permission to use the shared image gallery.

1. In the Azure portal, select the Shared Image Gallery that you want to share with another tenant.
2. Select **Access control (IAM)**, and under **Add role assignment** select **Add**.
3. Under **Role**, select **Reader**.
4. Under **Assign access to**, leave this as **Azure AD user, group, or service principal**.
5. Under **Select**, type *myGalleryApp* and select it when it shows up in the list. When you are done, select **Save**.

## Give Tenant 2 access

Give Tenant 2 access to the application by requesting a sign-in using a browser. Replace <*Tenant2 ID*> with the tenant ID for the tenant that you would like to share your image gallery with. Replace <*Application (client) ID*> with the application ID of the app registration you created. When done making the replacements, paste the URL into a browser and follow the sign-in prompts to sign into Tenant 2.

```
https://login.microsoftonline.com/<Tenant 2 ID>/oauth2/authorize?client_id=<Application (client) ID>&response_type=code&redirect_uri=https%3A%2F%2Fwww.microsoft.com%2F
```

In the [Azure portal](#) sign in as Tenant 2 and give the app registration access to the resource group where you want to create the VM.

1. Select the resource group and then select **Access control (IAM)**. Under **Add role assignment** select **Add**.
2. Under **Role**, type **Contributor**.
3. Under **Assign access to**, leave this as **Azure AD user, group, or service principal**.
4. Under **Select** type *myGalleryApp* then select it when it shows up in the list. When you are done, select **Save**.

#### NOTE

You need to wait for the image version to completely finish being built and replicated before you can use the same managed image to create another image version.

#### IMPORTANT

You cannot use the portal to deploy a VM from an image in another azure tenant. To create a VM from an image shared between tenants, you must use the [Azure CLI](#) or Powershell.

## Create a VM using PowerShell

Log into both tenants using the application ID, secret and tenant ID.

```
$applicationId = '<App ID>'
$secret = <Secret> | ConvertTo-SecureString -AsPlainText -Force
$tenant1 = "<Tenant 1 ID>"
$tenant2 = "<Tenant 2 ID>"
$cred = New-Object -TypeName PSCredential -ArgumentList $applicationId, $secret
Clear-AzContext
Connect-AzAccount -ServicePrincipal -Credential $cred -Tenant "<Tenant 1 ID>"
Connect-AzAccount -ServicePrincipal -Credential $cred -Tenant "<Tenant 2 ID>"
```

Create the VM in the resource group that has permission on the app registration. Replace the information in this example with your own.

```

$resourceGroup = "myResourceGroup"
$location = "South Central US"
$vmName = "myVMfromImage"

Set a variable for the image version in Tenant 1 using the full image ID of the shared image version
$image = "/subscriptions/<Tenant 1 subscription>/resourceGroups/<Resource
group>/providers/Microsoft.Compute/galleries/<Gallery>/images/<Image definition>/versions/<version>"

Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

Networking pieces
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location $location `

-Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location $location `

-Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp `

-Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `

-DestinationPortRange 3389 -Access Allow
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location `

-Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location $location `

-SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

Create a virtual machine configuration using the $image variable to specify the shared image
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1_v2 | `

Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `

Set-AzVMSourceImage -Id $image | `

Add-AzVMNetworkInterface -Id $nic.Id

Create a virtual machine
New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

```

## Next steps

You can also create shared image gallery resources using the [Azure portal](#).

# Troubleshooting shared image galleries

11/13/2019 • 3 minutes to read • [Edit Online](#)

If you run into issues while performing any operations on shared image galleries, image definitions, and image versions, run the failing command again in debug mode. Debug mode is activated by passing the **-debug** switch with CLI and the **-Debug** switch with PowerShell. Once you've located the error, follow this document to troubleshoot the errors.

## Unable to create a shared image gallery

Possible causes:

*The gallery name is invalid.*

Allowed characters for Gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name cannot contain dashes. Change the gallery name and try again.

*The gallery name is not unique within your subscription.*

Pick another gallery name and try again.

## Unable to create an image definition

Possible causes:

*image definition name is invalid.*

Allowed characters for image definition are uppercase or lowercase letters, digits, dots, dashes, and periods. Change the image definition name and try again.

*The mandatory properties for creating an image definition are not populated.*

The properties such as name, publisher, offer, sku, and OS type are mandatory. Verify if all the properties are being passed.

Make sure that the **OSType**, either Linux or Windows, of the image definition is the same as the source managed image that you are using to create the image version.

## Unable to create an image version

Possible causes:

*Image version name is invalid.*

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*. Change the image version name and try again.

*Source managed image from which the image version is being created is not found.*

Check if the source image exists and is in the same region as the image version.

*The managed image isn't done being provisioned.*

Make sure the provisioning state of the source managed image is **Succeeded**.

*The target region list does not include the source region.*

The target region list must include the source region of the image version. Make sure you have included the source region in the list of target regions where you want Azure to replicate your image version to.

*Replication to all the target regions not completed.*

Use the **--expand ReplicationStatus** flag to check if the replication to all the specified target regions has been completed. If not, wait up to 6 hours for the job to complete. If it fails, run the command again to create and replicate the image version. If there are a lot of target regions the image version is being replicated to, consider doing the replication in phases.

## Unable to create a VM or a scale set

Possible causes:

*The user trying to create a VM or virtual machine scale set doesn't have the read access to the image version.*

Contact the subscription owner and ask them to give read access to the image version or the parent resources (like the shared image gallery or image definition) through [Role Based Access Control](#) (RBAC).

*The image version is not found.*

Verify that the region you are trying to create a VM or virtual machine scale in is included in the list of target regions of the image version. If the region is already in the list of target regions, then verify if the replication job has been completed. You can use the **-ReplicationStatus** flag to check if the replication to all the specified target regions has been completed.

*The VM or virtual machine scale set creation takes a long time.*

Verify that the **OSType** of the image version that you are trying to create the VM or virtual machine scale set from has the same **OSType** of the source managed image that you used to create the image version.

## Unable to share resources

The sharing of shared image gallery, image definition, and image version resources across subscriptions is enabled using [Role-Based Access Control](#) (RBAC).

## Replication is slow

Use the **--expand ReplicationStatus** flag to check if the replication to all the specified target regions has been completed. If not, wait for up to 6 hours for the job to complete. If it fails, trigger the command again to create and replicate the image version. If there are a lot of target regions the image version is being replicated to, consider doing the replication in phases.

## Azure limits and quotas

[Azure limits and quotas](#) apply to all shared image gallery, image definition, and image version resources. Make sure you are within the limits for your subscriptions.

## Next steps

Learn more about [shared image galleries](#).

# Preview: Azure Image Builder overview

7/9/2019 • 4 minutes to read • [Edit Online](#)

Standardized virtual machine (VM) images allow organizations to migrate to the cloud and ensure consistency in the deployments. Images typically include predefined security and configuration settings and necessary software. Setting up your own imaging pipeline requires time, infrastructure and setup, but with Azure VM Image Builder, just provide a simple configuration describing your image, submit it to the service, and the image is built, and distributed.

The Azure VM Image Builder (Azure Image Builder) lets you start with a Windows or Linux-based Azure Marketplace image, existing custom images or Red Hat Enterprise Linux (RHEL) ISO and begin to add your own customizations. Because the Image Builder is built on [HashiCorp Packer](#), you can also import your existing Packer shell provisioner scripts. You can also specify where you would like your images hosted, in the [Azure Shared Image Gallery](#), as a managed image or a VHD.

## IMPORTANT

Azure Image Builder is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Preview features

For the preview, these features are supported:

- Creation of golden baseline images, that includes your minimum security and corporate configurations, and allow departments to customize it further for their needs.
- Patching of existing images, Image Builder will allow you to continually patch existing custom images.
- Integration with the Azure Shared Image Gallery, allows you to distribute, version, and scale images globally, and gives you an image management system.
- Integration with existing image build pipelines, just call Image Builder from your pipeline, or use the simple Preview Image Builder Azure DevOps Task.
- Migrate an existing image customization pipeline to Azure. Use your existing scripts, commands, and processes to customize images.
- Use Red Hat Bring Your Own Subscription support. Create Red Hat Enterprise images for use with your eligible, unused Red Hat subscriptions.
- Creation of images in VHD format.

## Regions

The Azure Image Builder Service will be available for preview in these regions. Images can be distributed outside of these regions.

- East US
- East US 2
- West Central US
- West US
- West US 2

# OS support

AIB will support Azure Marketplace base OS images:

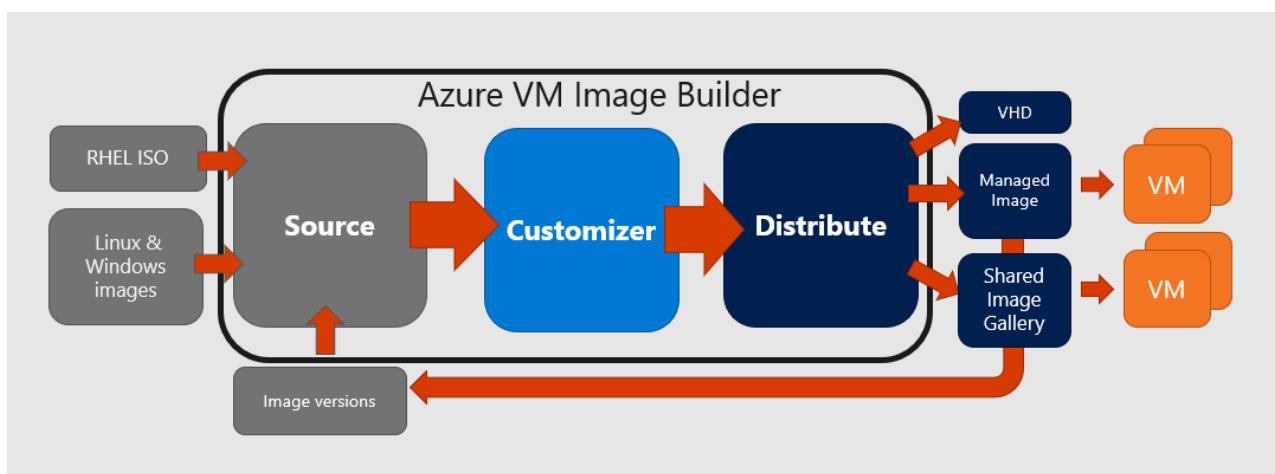
- Ubuntu 18.04
- Ubuntu 16.04
- RHEL 7.6, 7.7
- CentOS 7.6, 7.7
- SLES 12 SP4
- SLES 15, SLES 15 SP1
- Windows 10 RS5 Enterprise/Professional/Enterprise for Virtual Desktop (EVD)
- Windows 2016
- Windows 2019

AIB will support RHEL ISO's, as a source for:

- RHEL 7.3
- RHEL 7.4
- RHEL 7.5

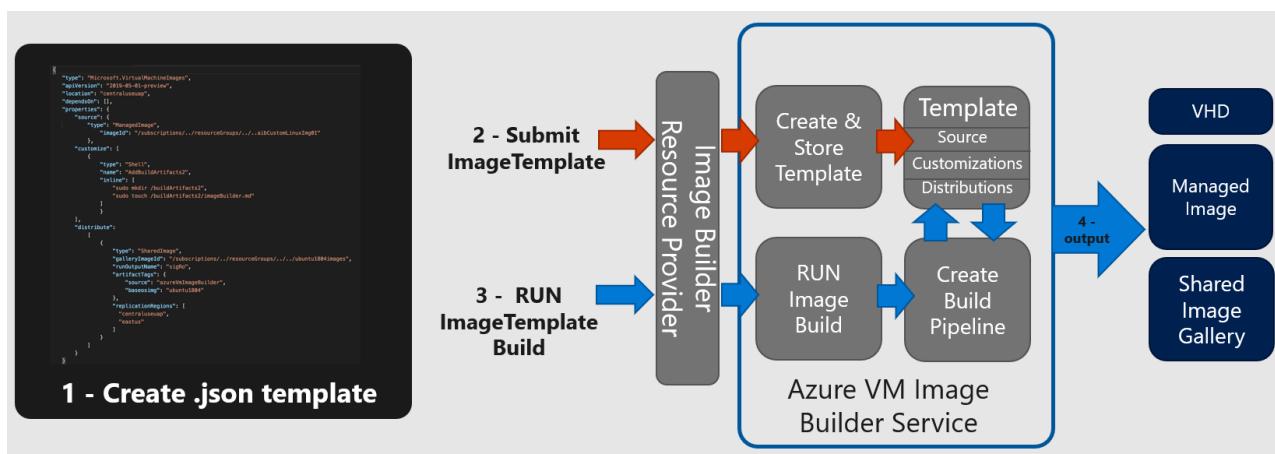
RHEL 7.6 ISOs are not supported, but are being tested.

## How it works



The Azure Image Builder is a fully managed Azure service that is accessible by an Azure resource provider. The Azure Image Builder process has three main parts: source, customize and distribute, these are represented in a template. The diagram below shows the components, with some of their properties.

### Image Builder process



1. Create the Image Template as a json file. This json file contains information about the image source, customizations, and distribution. There are multiple examples in the [Azure Image Builder GitHub repository](#).
2. Submit it to the service, this will create an Image Template artifact in the resource group you specify. In the background, Image Builder will download the source image or ISO, and scripts as needed. These are stored in a separate resource group that is automatically created in your subscription, in the format: IT\_<DestinationResourceGroup>\_<TemplateName>.
3. Once the Image Template is created, you can then build the image. In the background Image Builder uses the template and source files to create a VM (default size: Standard\_D1\_v2), network, public IP, NSG, and storage in the IT\_<DestinationResourceGroup>\_<TemplateName> resource group.
4. As part of the image creation, Image builder distributes the image it according to the template, then deletes the additional resources in the IT\_<DestinationResourceGroup>\_<TemplateName> resource group that was created for the process.

## Permissions

To allow Azure VM Image Builder to distribute images to either the managed images or to a Shared Image Gallery, you will need to provide 'Contributor' permissions for the service "Azure Virtual Machine Image Builder" (app ID: cf32a0cc-373c-47c9-9156-0db11f6a6dfc) on the resource groups.

If you are using an existing custom managed image or image version, then the Azure Image Builder will need a minimum of 'Reader' access to those resource groups.

You can assign access using the Azure CLI:

```
az role assignment create \
--assignee cf32a0cc-373c-47c9-9156-0db11f6a6dfc \
--role Contributor \
--scope /subscriptions/$subscriptionID/resourceGroups/<distributeResourceGroupName>
```

You can assign access using the PowerShell:

```
New-AzRoleAssignment -ObjectId ef511139-6170-438e-a6e1-763dc31bdf74 -Scope
/subscriptions/$subscriptionID/resourceGroups/<distributeResourceGroupName> -RoleDefinitionName Contributor
```

If the service account is not found, that may mean that the subscription where you are adding the role assignment has not yet registered for the resource provider.

## Costs

You will incur some compute, networking and storage costs when creating, building and storing images with Azure Image Builder. These costs are similar to the costs incurred in manually creating custom images. For the resources, you will be charged at your Azure rates.

During the image creation process, files are downloaded and stored in the `IT_<DestinationResourceGroup>_<TemplateName>` resource group, which will incur a small storage costs. If you do not want to keep these, delete the **Image Template** after the image build.

Image Builder creates a VM using a D1v2 VM size, and the storage, and networking needed for the VM. These resources will last for the duration of the build process, and will be deleted once Image Builder has finished creating the image.

Azure Image Builder will distribute the image to your chosen regions, which might incur network egress charges.

## Next steps

To try out the Azure Image Builder, see the articles for building [Linux](#) or [Windows](#) images.

# Preview: Create a Windows VM with Azure Image Builder

8/6/2019 • 4 minutes to read • [Edit Online](#)

This article is to show you how you can create a customized Windows image using the Azure VM Image Builder. The example in this article uses [customizers](#) for customizing the image:

- PowerShell (ScriptUri) - download and run a [PowerShell script](#).
- Windows Restart - restarts the VM.
- PowerShell (inline) - run a specific command. In this example, it creates a directory on the VM using `mkdir c:\\buildActions`.
- File - copy a file from GitHub onto the VM. This example copies `index.md` to `c:\\buildArtifacts\\index.html` on the VM.

You can also specify a `buildTimeoutInMinutes`. The default is 240 minutes, and you can increase a build time to allow for longer running builds.

We will be using a sample json template to configure the image. The json file we are using is here: [helloImageTemplateWin.json](#).

## IMPORTANT

Azure Image Builder is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Register the features

To use Azure Image Builder during the preview, you need to register the new feature.

```
az feature register --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview
```

Check the status of the feature registration.

```
az feature show --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview | grep state
```

Check your registration.

```
az provider show -n Microsoft.VirtualMachineImages | grep registrationState
az provider show -n Microsoft.Storage | grep registrationState
```

If they do not say registered, run the following:

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Storage
```

## Set variables

We will be using some pieces of information repeatedly, so we will create some variables to store that information.

```
Resource group name - we are using myImageBuilderRG in this example
imageResourceGroup=myWinImgBuilderRG
Region location
location=WestUS2
Name for the image
imageName=myWinBuilderImage
Run output name
runOutputName=aibWindows
name of the image to be created
imageName=aibWinImage
```

Create a variable for your subscription ID. You can get this using `az account show | grep id`.

```
subscriptionID=<Your subscription ID>
```

## Create a resource group

This resource group is used to store the image configuration template artifact and the image.

```
az group create -n $imageResourceGroup -l $location
```

## Set permissions on the resource group

Give Image Builder 'contributor' permission to create the image in the resource group. Without this, the image build will fail.

The `--assignee` value is the app registration ID for the Image Builder service.

```
az role assignment create \
--assignee cf32a0cc-373c-47c9-9156-0db11f6a6dfc \
--role Contributor \
--scope /subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
```

## Download the image configuration template example

A parameterized image configuration template has been created for you to try. Download the example json file and configure it with the variables you set previously.

```
curl
https://raw.githubusercontent.com/danielsollondon/azvmimagebuilder/master/quickstarts/0_Creating_a_Custom_Windows_Managed_Image/helloImageTemplateWin.json -o helloImageTemplateWin.json

sed -i -e "s/<subscriptionID>/$subscriptionID/g" helloImageTemplateWin.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" helloImageTemplateWin.json
sed -i -e "s/<region>/\$location/g" helloImageTemplateWin.json
sed -i -e "s/<imageName>/\$imageName/g" helloImageTemplateWin.json
sed -i -e "s/<runOutputName>/\$runOutputName/g" helloImageTemplateWin.json
```

You can modify this example, in the terminal using a text editor like `vi`.

```
vi helloImageTemplateLinux.json
```

#### NOTE

For the source image, you must always [specify a version](#), you cannot use `latest`. If you add or change the resource group where the image is distributed to, you must make the [permissions are set](#) on the resource group.

## Create the image

Submit the image configuration to the VM Image Builder service

```
az resource create \
--resource-group $imageResourceGroup \
--properties @helloImageTemplateWin.json \
--is-full-object \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n helloImageTemplateWin01
```

When complete, this will return a success message back to the console, and create an

[Image Builder Configuration Template](#) in the `$imageResourceGroup`. You can see this resource in the resource group in the Azure portal, if you enable 'Show hidden types'.

In the background, Image Builder will also create a staging resource group in your subscription. This resource group is used for the image build. It will be in this format: `IT_<DestinationResourceGroup>_<TemplateName>`

#### NOTE

You must not delete the staging resource group directly. First delete the image template artifact, this will cause the staging resource group to be deleted.

If the service reports a failure during the image configuration template submission:

- Review these [troubleshooting](#) steps.
- You will need to delete the template, using the following snippet, before you retry submission.

```
az resource delete \
--resource-group $imageResourceGroup \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n helloImageTemplateLinux01
```

## Start the image build

Start the image building process using `az resource invoke-action`.

```
az resource invoke-action \
--resource-group $imageResourceGroup \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n helloImageTemplateWin01 \
--action Run
```

Wait until the build is complete. This can take about 15 minutes.

If you encounter any errors, please review these [troubleshooting](#) steps.

## Create the VM

Create the VM using the image you built. Replace <password> with your own password for the `aibuser` on the VM.

```
az vm create \
--resource-group $imageResourceGroup \
--name aibImgWinVm00 \
--admin-username aibuser \
--admin-password <password> \
--image $imageName \
--location $location
```

## Verify the customization

Create a Remote Desktop connection to the VM using the username and password you set when you created the VM. Inside the VM, open a cmd prompt and type:

```
dir c:\
```

You should see these two directories created during image customization:

- buildActions
- buildArtifacts

## Clean up

When you are done, delete the resources.

### Delete the image builder template

```
az resource delete \
--resource-group $imageResourceGroup \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n helloImageTemplateWin01
```

### Delete the image resource group

```
az group delete -n $imageResourceGroup
```

## Next steps

To learn more about the components of the .json file used in this article, see [Image builder template reference](#).

# Preview: Create a Windows image and distribute it to a Shared Image Gallery

1/17/2020 • 6 minutes to read • [Edit Online](#)

This article is to show you how you can use the Azure Image Builder, and Azure PowerShell, to create an image version in a [Shared Image Gallery](#), then distribute the image globally. You can also do this using the [Azure CLI](#).

We will be using a .json template to configure the image. The .json file we are using is here:

[armTemplateWinSIG.json](#). We will be downloading and editing a local version of the template, so this article is written using local PowerShell session.

To distribute the image to a Shared Image Gallery, the template uses [sharedImage](#) as the value for the [distribute](#) section of the template.

Azure Image Builder automatically runs sysprep to generalize the image, this is a generic sysprep command, which you can [override](#) if needed.

Be aware how many times you layer customizations. You can run the Sysprep command up to 8 times on a single Windows image. After running Sysprep 8 times, you must recreate your Windows image. For more information, see [Limits on how many times you can run Sysprep](#).

## IMPORTANT

Azure Image Builder is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Register the features

To use Azure Image Builder during the preview, you need to register the new feature.

```
Register-AzProviderFeature -FeatureName VirtualMachineTemplatePreview -ProviderNamespace
Microsoft.VirtualMachineImages
```

Check the status of the feature registration.

```
Get-AzProviderFeature -FeatureName VirtualMachineTemplatePreview -ProviderNamespace
Microsoft.VirtualMachineImages
```

Wait until [RegistrationState](#) is [Registered](#) before moving to the next step.

Check your provider registrations. Make sure each returns [Registered](#).

```
Get-AzResourceProvider -ProviderNamespace Microsoft.VirtualMachineImages | Format-table -Property
ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.Storage | Format-table -Property
ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.Compute | Format-table -Property
ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.KeyVault | Format-table -Property
ResourceTypes,RegistrationState
```

If they do not return `Registered`, use the following to register the providers:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.VirtualMachineImages
Register-AzResourceProvider -ProviderNamespace Microsoft.Storage
Register-AzResourceProvider -ProviderNamespace Microsoft.Compute
Register-AzResourceProvider -ProviderNamespace Microsoft.KeyVault
```

## Create variables

We will be using some pieces of information repeatedly, so we will create some variables to store that information. Replace the values for the variables, like `username` and `vmpassword`, with your own information.

```
Get existing context
$currentAzContext = Get-AzContext

Get your current subscription ID.
$subscriptionID=$currentAzContext.Subscription.Id

Destination image resource group
$imageResourceGroup="aibwinsig"

Location
$location="westus"

Image distribution metadata reference name
$runOutputName="aibCustWinManImg02ro"

Image template name
$imageTemplateName="helloImageTemplateWin02ps"

Distribution properties object name (runOutput).
This gives you the properties of the managed image on completion.
$runOutputName="winclientR01"
```

## Create the resource group

Create a resource group and give Azure Image Builder permission to create resources in that resource group.

```
New-AzResourceGroup `
-Name $imageResourceGroup `
-Location $location
New-AzRoleAssignment `
-ObjectId ef511139-6170-438e-a6e1-763dc31bdf74 `
-Scope /subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup `
-RoleDefinitionName Contributor
```

## Create the Shared Image Gallery

To use Image Builder with a shared image gallery, you need to have an existing image gallery and image definition. Image Builder will not create the image gallery and image definition for you.

If you don't already have a gallery and image definition to use, start by creating them. First, create an image gallery.

```
Image gallery name
$sigGalleryName= "myIBSIG"

Image definition name
$imageDefName = "winSvrimage"

additional replication region
$replRegion2="eastus"

Create the gallery
New-AzGallery `

 -GalleryName $sigGalleryName `

 -ResourceGroupName $imageResourceGroup `

 -Location $location

Create the image definition
New-AzGalleryImageDefinition `

 -GalleryName $sigGalleryName `

 -ResourceGroupName $imageResourceGroup `

 -Location $location `

 -Name $imageDefName `

 -OsState generalized `

 -OsType Windows `

 -Publisher 'myCompany' `

 -Offer 'WindowsServer' `

 -Sku 'WinSrv2019'
```

## Download and configure the template

Download the .json template and configure it with your variables.

```
$templateFilePath = "armTemplateWinSIG.json"

Invoke-WebRequest `

 -Uri
"https://raw.githubusercontent.com/danielsollondon/azvmimagebuilder/master/quickstarts/1_Creating_a_Custom _Win_Shared_Image_Gallery_Image/armTemplateWinSIG.json" `

 -OutFile $templateFilePath `

 -UseBasicParsing

(Get-Content -path $templateFilePath -Raw) `

 -replace '<subscriptionID>',$subscriptionID | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<rgName>',$imageResourceGroup | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<runOutputName>',$runOutputName | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<imageDefName>',$imageDefName | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<sharedImageGalName>',$sigGalleryName | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<region1>',$location | Set-Content -Path $templateFilePath

(Get-Content -path $templateFilePath -Raw) `

 -replace '<region2>',$replRegion2 | Set-Content -Path $templateFilePath
```

## Create the image version

Your template must be submitted to the service, this will download any dependent artifacts, like scripts, and store them in the staging Resource Group, prefixed with *IT\_*.

```
New-AzResourceGroupDeployment `
-ResourceGroupName $imageResourceGroup `
-TemplateFile $templateFilePath `
-api-version "2019-05-01-preview" `
-imageTemplateName $imageTemplateName `
-svclocation $location
```

To build the image you need to invoke 'Run' on the template.

```
Invoke-AzResourceAction `
-ResourceName $imageTemplateName `
-ResourceGroupName $imageResourceGroup `
-ResourceType Microsoft.VirtualMachineImages/imageTemplates `
-ApiVersion "2019-05-01-preview" `
-Action Run
```

Creating the image and replicating it to both regions can take a while. Wait until this part is finished before moving on to creating a VM.

For information on options for automating getting the image build status, see the [Readme](#) for this template on GitHub.

## Create the VM

Create a VM from the image version that was created by Azure Image Builder.

Get the image version you created.

```
$imageVersion = Get-AzGalleryImageVersion `
-ResourceGroupName $imageResourceGroup `
-GalleryName $sigGalleryName `
-GalleryImageDefinitionName $imageDefName
```

Create the VM in the second region that were the image was replicated.

```

$vmResourceGroup = "myResourceGroup"
$vmName = "myVMfromImage"

Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

Create a resource group
New-AzResourceGroup -Name $vmResourceGroup -Location $replRegion2

Network pieces
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $vmResourceGroup -Location $replRegion2 `
 -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $vmResourceGroup -Location $replRegion2 `
 -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp `
 -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `
 -DestinationPortRange 3389 -Access Allow
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $vmResourceGroup -Location $replRegion2 `
 -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $vmResourceGroup -Location $replRegion2 `
 -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

Create a virtual machine configuration using $imageVersion.Id to specify the shared image
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1_v2 | `
Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `
Set-AzVMSourceImage -Id $imageVersion.Id | `
Add-AzVMNetworkInterface -Id $nic.Id

Create a virtual machine
New-AzVM -ResourceGroupName $vmResourceGroup -Location $replRegion2 -VM $vmConfig

```

## Verify the customization

Create a Remote Desktop connection to the VM using the username and password you set when you created the VM. Inside the VM, open a cmd prompt and type:

```
dir c:\
```

You should see a directory named `buildActions` that was created during image customization.

## Clean up resources

If you want to now try re-customizing the image version to create a new version of the same image, **skip this step** and go on to [Use Azure Image Builder to create another image version](#).

This will delete the image that was created, along with all of the other resource files. Make sure you are finished with this deployment before deleting the resources.

Delete the resource group template first, otherwise the staging resource group (`IT_`) used by AIB will not be cleaned up.

Get ResourceID of the image template.

```
$resTemplateId = Get-AzResource -ResourceName $imageTemplateName -ResourceGroupName $imageResourceGroup -
 ResourceType Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2019-05-01-preview"
```

Delete image template.

```
Remove-AzResource -ResourceId $resTemplateId.ResourceId -Force
```

delete the resource group.

```
Remove-AzResourceGroup $imageResourceGroup -Force
```

## Next Steps

To learn how to update the image version you created, see [Use Azure Image Builder to create another image version](#).

# Preview: Create a new VM image version from an existing image version using Azure Image Builder

12/9/2019 • 3 minutes to read • [Edit Online](#)

This article shows you how to take an existing image version in a [Shared Image Gallery](#), update it, and publish it as a new image version to the gallery.

We will be using a sample .json template to configure the image. The .json file we are using is here: [helloImageTemplateforSIGfromWinSIG.json](#).

## IMPORTANT

Azure Image Builder is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Register the features

To use Azure Image Builder during the preview, you need to register the new feature.

```
az feature register --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview
```

Check the status of the feature registration.

```
az feature show --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview | grep state
```

Check your registration.

```
az provider show -n Microsoft.VirtualMachineImages | grep registrationState
az provider show -n Microsoft.Storage | grep registrationState
az provider show -n Microsoft.Compute | grep registrationState
```

If they do not say registered, run the following:

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Storage
az provider register -n Microsoft.Compute
```

## Set variables and permissions

If you used [Create an image and distribute to a Shared Image Gallery](#) to create your Shared Image Gallery, you've already created the variables we need. If not, please setup some variables to be used for this example.

For Preview, image builder will only support creating custom images in the same Resource Group as the source managed image. Update the resource group name in this example to be the same resource group as your source managed image.

```

Resource group name - we are using ibsigRG in this example
sigResourceGroup=myIBWinRG
Datacenter location - we are using West US 2 in this example
location=westus
Additional region to replicate the image to - we are using East US in this example
additionalRegion=eastus
name of the shared image gallery - in this example we are using myGallery
sigName=my22stsSIG
name of the image definition to be created - in this example we are using myImageDef
imageDefName=winSvrimages
image distribution metadata reference name
runOutputName=w2019SigRo
User name and password for the VM
username="user name for the VM"
vmPassword="password for the VM"

```

Create a variable for your subscription ID. You can get this using `az account show | grep id`.

```
subscriptionID=<Subscription ID>
```

Get the image version that you want to update.

```

sigDefImgVersionId=$(az sig image-version list \
-g $sigResourceGroup \
--gallery-name $sigName \
--gallery-image-definition $imageDefName \
--subscription $subscriptionID --query [].'id' -o json | grep 0. | tr -d '"' | tr -d '[:space:]')

```

If you already have your own Shared Image Gallery, and did not follow the previous example, you will need to assign permissions for Image Builder to access the Resource Group, so it can access the gallery.

```

az role assignment create \
--assignee cf32a0cc-373c-47c9-9156-0db11f6a6dfc \
--role Contributor \
--scope /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup

```

## Modify helloImage example

You can review the example we are about to use by opening the .json file here:

[helloImageTemplateforSIGfromSIG.json](#) along with the [Image Builder template reference](#).

Download the .json example and configure it with your variables.

```

curl
https://raw.githubusercontent.com/danielsollondon/azvmimagebuilder/master/quickstarts/8_Creating_a_Custom_Win_Shared_Image_Gallery_Image_from_SIG/helloImageTemplateforSIGfromWinSIG.json -o
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<rgName>/$sigResourceGroup/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<imageDefName>/$imageDefName/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<sharedImageGalName>/$sigName/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s%<sigDefImgVersionId>%$sigDefImgVersionId%g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<region1>/$location/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<region2>/$additionalRegion/g" helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<runOutputName>/$runOutputName/g" helloImageTemplateforSIGfromWinSIG.json

```

## Create the image

Submit the image configuration to the VM Image Builder Service.

```
az resource create \
--resource-group $sigResourceGroup \
--properties @helloImageTemplateforSIGfromWinSIG.json \
--is-full-object \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n imageTemplateforSIGfromWinSIG01
```

Start the image build.

```
az resource invoke-action \
--resource-group $sigResourceGroup \
--resource-type Microsoft.VirtualMachineImages/imageTemplates \
-n imageTemplateforSIGfromWinSIG01 \
--action Run
```

Wait until the image has been built and replication before moving on to the next step.

## Create the VM

```
az vm create \
--resource-group $sigResourceGroup \
--name aibImgWinVm002 \
--admin-username $username \
--admin-password $vmpassword \
--image
"/subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/Microsoft.Compute/galleries/$sigName/images/$imageDefName/versions/latest" \
--location $location
```

## Verify the customization

Create a Remote Desktop connection to the VM using the username and password you set when you created the VM. Inside the VM, open a cmd prompt and type:

```
dir c:\
```

You should now see two directories:

- `buildActions` that was created in the first image version.
- `buildActions2` that was created as part up updating the first image version to create the second image version.

## Next steps

To learn more about the components of the json file used in this article, see [Image builder template reference](#).

# Create an image and use a user-assigned managed identity to access files in Azure Storage

1/23/2020 • 4 minutes to read • [Edit Online](#)

Azure Image Builder supports using scripts, or copying files from multiple locations, such as GitHub and Azure storage etc. To use these, they must have been externally accessible to Azure Image Builder, but you could protect Azure Storage blobs using SAS Tokens.

This article shows how to create a customized image using the Azure VM Image Builder, where the service will use a [User-assigned Managed Identity](#) to access files in Azure storage for the image customization, without you having to make the files publicly accessible, or setting up SAS tokens.

In the example below, you will create two resource groups, one will be used for the custom image, and the other will host an Azure Storage Account, that contains a script file. This simulates a real life scenario, where you may have build artifacts, or image files in different storage accounts, outside of Image Builder. You will create a user-assigned identity, then grant that read permissions on the script file, but you will not set any public access to that file. You will then use the Shell customizer to download and run that script from the storage account.

## IMPORTANT

Azure Image Builder is currently in public preview. This preview version is provided without a service level agreement, and it's not recommended for production workloads. Certain features might not be supported or might have constrained capabilities. For more information, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

## Register the features

To use Azure Image Builder during the preview, you need to register the new feature.

```
az feature register --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview
```

Check the status of the feature registration.

```
az feature show --namespace Microsoft.VirtualMachineImages --name VirtualMachineTemplatePreview | grep state
```

Check your registration.

```
az provider show -n Microsoft.VirtualMachineImages | grep registrationState
```

```
az provider show -n Microsoft.Storage | grep registrationState
```

If they do not say registered, run the following:

```
az provider register -n Microsoft.VirtualMachineImages
```

```
az provider register -n Microsoft.Storage
```

## Create a resource group

We will be using some pieces of information repeatedly, so we will create some variables to store that information.

```
Image resource group name
imageResourceGroup=aibmdimsi
storage resource group
strResourceGroup=aibmdimsistor
Location
location=WestUS2
name of the image to be created
imageName=aibCustLinuxImgMsi01
image distribution metadata reference name
runOutputName=u1804ManImgMsiro
```

Create a variable for your subscription ID. You can get this using `az account show | grep id`.

```
subscriptionID=<Your subscription ID>
```

Create the resource groups for both the image and the script storage.

```
create resource group for image template
az group create -n $imageResourceGroup -l $location
create resource group for the script storage
az group create -n $strResourceGroup -l $location
```

Create the storage and copy the sample script into it from GitHub.

```
script storage account
scriptStorageAcc=aibstorscript$(date +'%s')

script container
scriptStorageAccContainer=scriptscont$(date +'%s')

script url
scriptUrl=https://$scriptStorageAcc.blob.core.windows.net/$scriptStorageAccContainer/customizeScript.sh

create storage account and blob in resource group
az storage account create -n $scriptStorageAcc -g $strResourceGroup -l $location --sku Standard_LRS

az storage container create -n $scriptStorageAccContainer --fail-on-exist --account-name $scriptStorageAcc

copy in an example script from the GitHub repo
az storage blob copy start \
 --destination-blob customizeScript.sh \
 --destination-container $scriptStorageAccContainer \
 --account-name $scriptStorageAcc \
 --source-uri
 https://raw.githubusercontent.com/danielsollondon/azvmimagebuilder/master/quickstarts/customizeScript.sh
```

Give Image Builder permission to create resources in the image resource group. The `--assignee` value is the app registration ID for the Image Builder service.

```
az role assignment create \
 --assignee cf32a0cc-373c-47c9-9156-0db11f6a6dfc \
 --role Contributor \
 --scope /subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
```

## Create user-assigned managed identity

Create the identity and assign permissions for the script storage account. For more information, see [User-Assigned Managed Identity](#).

```
Create the user assigned identity
identityName=aibBuiUserId$(date +'%s')
az identity create -g $imageResourceGroup -n $identityName
assign the identity permissions to the storage account, so it can read the script blob
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n $identityName | grep "clientId" | cut -c16- | tr -d '')
az role assignment create \
 --assignee $imgBuilderCliId \
 --role "Storage Blob Data Reader" \
 --scope
/subscriptions/$subscriptionID/resourceGroups/$strResourceGroup/providers/Microsoft.Storage/storageAccounts/$scriptStorageAcc/blobServices/default/containers/$scriptStorageAccContainer
create the user identity URI
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$imageResourceGroup/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName
```

## Modify the example

Download the example .json file and configure it with the variables you created.

```
curl
https://raw.githubusercontent.com/danielsollondon/azvmimagebuilder/master/quickstarts/7_Creating_Custom_Image_using_MSI_to_Access_Storage/helloImageTemplateMsi.json -o helloImageTemplateMsi.json
sed -i -e "s/<subscriptionID>/${subscriptionID}/g" helloImageTemplateMsi.json
sed -i -e "s/<rgName>/${imageResourceGroup}/g" helloImageTemplateMsi.json
sed -i -e "s/<region>/${location}/g" helloImageTemplateMsi.json
sed -i -e "s/<imageName>/${imageName}/g" helloImageTemplateMsi.json
sed -i -e "s%<scriptUrl>%$scriptUrl%g" helloImageTemplateMsi.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" helloImageTemplateMsi.json
sed -i -e "s%<runOutputName>%$runOutputName%g" helloImageTemplateMsi.json
```

## Create the image

Submit the image configuration to the Azure Image Builder service.

```
az resource create \
 --resource-group $imageResourceGroup \
 --properties @helloImageTemplateMsi.json \
 --is-full-object \
 --resource-type Microsoft.VirtualMachineImages/imageTemplates \
 -n helloImageTemplateMsi01
```

Start the image build.

```
az resource invoke-action \
 --resource-group $imageResourceGroup \
 --resource-type Microsoft.VirtualMachineImages/imageTemplates \
 -n helloImageTemplateMsi01 \
 --action Run
```

Wait for the build to complete. This can take about 15 minutes.

## Create a VM

Create a VM from the image.

```
az vm create \
--resource-group $imageResourceGroup \
--name aibImgVm00 \
--admin-username aibuser \
--image $imageName \
--location $location \
--generate-ssh-keys
```

After the VM has been created, start an SSH session with the VM.

```
ssh aibuser@<publicIp>
```

You should see the image was customized with a Message of the Day as soon as your SSH connection is established!

```

** This VM was built from the: **
** !! AZURE VM IMAGE BUILDER Custom Image !! **
** You have just been Customized :-) **

```

## Clean up

When you are finished, you can delete the resources if they are no longer needed.

```
az identity delete --ids $imgBuilderId
az resource delete \
 --resource-group $imageResourceGroup \
 --resource-type Microsoft.VirtualMachineImages/imageTemplates \
 -n helloImageTemplateMsi01
az group delete -n $imageResourceGroup
az group delete -n $strResourceGroup
```

## Next steps

If you have any trouble working with Azure Image Builder, see [Troubleshooting](#).

# Preview: Create an Azure Image Builder template

1/30/2020 • 15 minutes to read • [Edit Online](#)

Azure Image Builder uses a .json file to pass information into the Image Builder service. In this article we will go over the sections of the json file, so you can build your own. To see examples of full json files, see the [Azure Image Builder GitHub](#).

This is the basic template format:

```
{
 "type": "Microsoft.VirtualMachineImages/imageTemplates",
 "apiVersion": "2019-05-01-preview",
 "location": "<region>",
 "tags": {
 "<name>": "<value>",
 "<name>": "<value>"
 },
 "identity": {},
 "dependsOn": [],
 "properties": {
 "buildTimeoutInMinutes": <minutes>,
 "vmProfile":
 {
 "vmSize": "<vmSize>"
 },
 "build": {},
 "customize": {},
 "distribute": {}
 }
}
```

## Type and API version

The `type` is the resource type, which must be `"Microsoft.VirtualMachineImages/imageTemplates"`. The `apiVersion` will change over time as the API changes, but should be `"2019-05-01-preview"` for preview.

```
"type": "Microsoft.VirtualMachineImages/imageTemplates",
"apiVersion": "2019-05-01-preview",
```

## Location

The location is the region where the custom image will be created. For the Image Builder preview, the following regions are supported:

- East US
- East US 2
- West Central US
- West US
- West US 2

```
"location": "<region>",
```

## vmProfile

By default Image Builder will use a "Standard\_D1\_v2" build VM, you can override this, for example, if you want to customize an Image for a GPU VM, you need a GPU VM size. This is optional.

```
{
 "vmSize": "Standard_D1_v2"
},
```

## osDiskSizeGB

By default, Image Builder will not change the size of the image, it will use the size from the source image. You can adjust the size of the OS Disk (Win and Linux), note, do not go too small than the minimum required space required for the OS. This is optional, and a value of 0 means leave the same size as the source image. This is optional.

```
{
 "osDiskSizeGB": 100
},
```

## Tags

These are key/value pairs you can specify for the image that's generated.

## Depends on (optional)

This optional section can be used to ensure that dependencies are completed before proceeding.

```
"dependsOn": [],
```

For more information, see [Define resource dependencies](#).

## Identity

By default, Image Builder supports using scripts, or copying files from multiple locations, such as GitHub and Azure storage. To use these, they must be publicly accessible.

You can also use an Azure User-Assigned Managed Identity, defined by you, to allow Image Builder access Azure Storage, as long as the identity has been granted a minimum of 'Storage Blob Data Reader' on the Azure storage account. This means you do not need to make the storage blobs externally accessible, or setup SAS Tokens.

```
"identity": {
 "type": "UserAssigned",
 "userAssignedIdentities": {
 "<imgBuilderId>": {}
 }
},
```

For a complete example, see [Use an Azure User-Assigned Managed Identity to access files in Azure Storage](#).

Image Builder support for a User-Assigned Identity:

- Supports a single identity only
- Does not support custom domain names

To learn more, see [What is managed identities for Azure resources?](#). For more information on deploying this feature, see [Configure managed identities for Azure resources on an Azure VM using Azure CLI](#).

## Properties: source

The `source` section contains information about the source image that will be used by Image Builder.

The API requires a 'SourceType' that defines the source for the image build, currently there are three types:

- ISO - use this when the source is a RHEL ISO.
- PlatformImage - indicated the source image is a Marketplace image.
- ManagedImage - use this when starting from a regular managed image.
- SharedImageVersion - this is used when you are using an image version in a Shared Image Gallery as the source.

### ISO source

Azure Image Builder only supports using published Red Hat Enterprise Linux 7.x Binary DVD ISOs, for preview. Image Builder supports:

- RHEL 7.3
- RHEL 7.4
- RHEL 7.5

```
"source": {
 "type": "ISO",
 "sourceURI": "<sourceURI from the download center>",
 "sha256Checksum": "<checksum associated with ISO>"
}
```

To get the `sourceURI` and `sha256Checksum` values, go to <https://access.redhat.com/downloads> then select the product **Red Hat Enterprise Linux**, and a supported version.

In the list of **Installers and Images for Red Hat Enterprise Linux Server**, you need to copy the link for Red Hat Enterprise Linux 7.x Binary DVD, and the checksum.

#### NOTE

The access tokens of the links are refreshed at frequent intervals, so every time you want to submit a template, you must check if the RH link address has changed.

### PlatformImage source

Azure Image Builder supports Windows Server and client, and Linux Azure Marketplace images, see [here](#) for the full list.

```
"source": {
 "type": "PlatformImage",
 "publisher": "Canonical",
 "offer": "UbuntuServer",
 "sku": "18.04-LTS",
 "version": "18.04.201903060"
},
```

The properties here are the same that are used to create VM's, using AZ CLI, run the below to get the properties:

```
az vm image list -l westus -f UbuntuServer -p Canonical --output table --all
```

**NOTE**

Version cannot be 'latest', you must use the command above to get a version number.

### ManagedImage source

Sets the source image as an existing managed image of a generalized VHD or VM. The source managed image must be of a supported OS, and be in the same region as your Azure Image Builder template.

```
"source": {
 "type": "ManagedImage",
 "imageId": "/subscriptions/<subscriptionId>/resourceGroups/{destinationResourceGroupName}/providers/Microsoft.Compute/images/<imageName>"
}
```

The `imageId` should be the ResourceId of the managed image. Use `az image list` to list available images.

### SharedImageVersion source

Sets the source image an existing image version in a Shared Image Gallery. The image version must be of a supported OS, and the image must be replicated to the same region as your Azure Image Builder template.

```
"source": {
 "type": "SharedImageVersion",
 "imageVersionID": "/subscriptions/<subscriptionId>/resourceGroups/<resourceGroup>/p
roviders/Microsoft.Compute/galleries/<sharedImageGalleryName>/images/<imageDefinitionName>/versions/<imageVersion>"
}
```

The `imageVersionId` should be the ResourceId of the image version. Use `az sig image-version list` to list image versions.

## Properties: buildTimeoutInMinutes

By default, the Image Builder will run for 240 minutes. After that, it will timeout and stop, whether or not the image build is complete. If the timeout is hit, you will see an error similar to this:

```
[ERROR] Failed while waiting for packerizer: Timeout waiting for microservice to
[ERROR] complete: 'context deadline exceeded'
```

If you do not specify a `buildTimeoutInMinutes` value, or set it to 0, it will use the default value. You can increase or decrease the value, up to the maximum of 960mins (16hrs). For Windows, we do not recommend setting this below 60 minutes. If you find you are hitting the timeout, review the [logs](#), to see if the customization step is waiting on something like user input.

If you find you need more time for customizations to complete, set this to what you think you need, with a little overhead. But, do not set it too high because you might have to wait for it to timeout before seeing an error.

## Properties: customize

Image Builder supports multiple 'customizers'. Customizers are functions that are used to customize your image, such as running scripts, or rebooting servers.

When using `customize`:

- You can use multiple customizers, but they must have a unique `name`.
- Customizers execute in the order specified in the template.
- If one customizer fails, then the whole customization component will fail and report back an error.
- It is strongly advised you test the script thoroughly before using it in a template. Debugging the script on your own VM will be easier.
- Do not put sensitive data in the scripts.
- The script locations need to be publicly accessible, unless you are using [MSI](#).

```
"customize": [
 {
 "type": "Shell",
 "name": "<name>",
 "scriptUri": "<path to script>",
 "sha256Checksum": "<sha256 checksum>"
 },
 {
 "type": "Shell",
 "name": "<name>",
 "inline": [
 "<command to run inline>"
]
 }
,
```

The `customize` section is an array. Azure Image Builder will run through the customizers in sequential order. Any failure in any customizer will fail the build process.

### Shell customizer

The shell customizer supports running shell scripts, these must be publicly accessible for the IB to access them.

```

"customize": [
 {
 "type": "Shell",
 "name": "<name>",
 "scriptUri": "<link to script>",
 "sha256Checksum": "<sha256 checksum>"
 },
],
 "customize": [
 {
 "type": "Shell",
 "name": "<name>",
 "inline": "<commands to run>"
 },
],

```

OS Support: Linux

Customize properties:

- **type** – Shell
- **name** – name for tracking the customization
- **scriptUri** - URI to the location of the file
- **inline** - array of shell commands, separated by commas.
- **sha256Checksum** - Value of sha256 checksum of the file, you generate this locally, and then Image Builder will checksum and validate.
  - To generate the sha256Checksum, using a terminal on Mac/Linux run: `sha256sum <fileName>`

For commands to run with super user privileges, they must be prefixed with `sudo`.

#### NOTE

When running the shell customizer with RHEL ISO source, you need to ensure your first customization shell handles registering with a Red Hat entitlement server before any customization occurs. Once customization is complete, the script should unregister with the entitlement server.

### Windows restart customizer

The Restart customizer allows you to restart a Windows VM and wait for it come back online, this allows you to install software that requires a reboot.

```

"customize": [
 {
 "type": "WindowsRestart",
 "restartCommand": "shutdown /r /f /t 0 /c",
 "restartCheckCommand": "echo Azure-Image-Builder-Restarted-the-VM > c:\\buildArtifacts\\azureImageBuilderRestart.txt",
 "restartTimeout": "5m"
 }
],

```

OS Support: Windows

Customize properties:

- **Type**: WindowsRestart
- **restartCommand** - Command to execute the restart (optional). The default is `'shutdown /r /f /t 0 /c \"packer restart\"'`.
- **restartCheckCommand** – Command to check if restart succeeded (optional).
- **restartTimeout** - Restart timeout specified as a string of magnitude and unit. For example, `5m` (5 minutes) or `2h` (2 hours). The default is: '5m'

### Linux restart

There is no Linux Restart customizer, however, if you are installing drivers, or components that require a restart, you can install them and invoke a restart using the Shell customizer, there is a 20min SSH timeout to the build VM.

### PowerShell customizer

The shell customizer supports running PowerShell scripts and inline command, the scripts must be publicly accessible for the IB to access them.

```

"customize": [
 {
 "type": "PowerShell",
 "name": "<name>",
 "scriptUri": "<path to script>",
 "runElevated": "<true false>",
 "sha256Checksum": "<sha256 checksum>"
 },
 {
 "type": "PowerShell",
 "name": "<name>",
 "inline": "<PowerShell syntax to run>",
 "valid_exit_codes": "<exit code>",
 "runElevated": "<true or false>"
 }
],

```

OS support: Windows and Linux

Customize properties:

- **type** – PowerShell.
- **scriptUri** - URI to the location of the PowerShell script file.
- **inline** – Inline commands to be run, separated by commas.
- **valid\_exit\_codes** – Optional, valid codes that can be returned from the script/inline command, this will avoid reported failure of the script/inline command.
- **runElevated** – Optional, boolean, support for running commands and scripts with elevated permissions.
- **sha256Checksum** - Value of sha256 checksum of the file, you generate this locally, and then Image Builder will checksum and validate.
  - To generate the sha256Checksum, using a PowerShell on Windows [Get-Hash](#)

#### File customizer

The File customizer lets image builder download a file from a GitHub or Azure storage. If you have an image build pipeline that relies on build artifacts, you can then set the file customizer to download from the build share, and move the artifacts into the image.

```
"customize": [
 {
 "type": "File",
 "name": "<name>",
 "sourceUri": "<source location>",
 "destination": "<destination>",
 "sha256Checksum": "<sha256 checksum>"
 }
]
```

OS support: Linux and Windows

File customizer properties:

- **sourceUri** - an accessible storage endpoint, this can be GitHub or Azure storage. You can only download one file, not an entire directory. If you need to download a directory, use a compressed file, then uncompress it using the Shell or PowerShell customizers.
- **destination** – this is the full destination path and file name. Any referenced path and subdirectories must exist, use the Shell or PowerShell customizers to set these up beforehand. You can use the script customizers to create the path.

This is supported by Windows directories and Linux paths, but there are some differences:

- Linux OS's – the only path Image builder can write to is /tmp.
- Windows – No path restriction, but the path must exist.

If there is an error trying to download the file, or put it in a specified directory, the customize step will fail, and this will be in the customization.log.

#### NOTE

The file customizer is only suitable for small file downloads, < 20MB. For larger file downloads use a script or inline command, the use code to download files, such as, Linux [wget](#) or [curl](#), Windows, [Invoke-WebRequest](#).

Files in the File customizer can be downloaded from Azure Storage using [MSI](#).

#### Generalize

By default, Azure Image Builder will also run 'deprovision' code at the end of each image customization phase, to 'generalize' the image. Generalizing is a process where the image is set up so it can be reused to create multiple VMs. For Windows VMs, Azure Image Builder uses Sysprep. For Linux, Azure Image Builder runs 'waagent - deprovision'.

The commands Image Builder users to generalize may not be suitable for every situation, so Azure Image Builder will allow you to customize this command, if needed.

If you are migrating existing customization, and you are using different Sysprep/waagent commands, you can use the Image Builder generic commands, and if the VM creation fails, use your own Sysprep or waagent commands.

If Azure Image Builder creates a Windows custom image successfully, and you create a VM from it, then find that the VM creation fails or does not complete successfully, you will need to review the Windows Server Sysprep documentation or raise a support request with the Windows Server Sysprep Customer Services Support team, who can troubleshoot and advise on the correct Sysprep usage.

#### Default Sysprep command

```
echo '>>> Waiting for GA to start ...'
while ((Get-Service RdAgent).Status -ne 'Running') { Start-Sleep -s 5 }
while ((Get-Service WindowsAzureTelemetryService).Status -ne 'Running') { Start-Sleep -s 5 }
while ((Get-Service WindowsAzureGuestAgent).Status -ne 'Running') { Start-Sleep -s 5 }
echo '>>> Sysprepping VM ...'
if (Test-Path $Env:SystemRoot\windows\system32\Sysprep\unattend.xml){ rm $Env:SystemRoot\windows\system32\Sysprep\unattend.xml -Force } &
$Env:SystemRoot\System32\Sysprep\Sysprep.exe /oobe /generalize /quiet /quit
while($true) { $imageState = Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\State | Select ImageState; if($imageState.ImageState -ne 'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { Write-Output $imageState.ImageState; Start-Sleep -s 5 } else { break } }
```

#### Default Linux deprovision command

```
/usr/sbin/waagent -force -deprovision+user && export HISTSIZE=0 && sync
```

#### Overriding the Commands

To override the commands, use the PowerShell or Shell script provisioners to create the command files with the exact file name, and put them in the correct directories:

- Windows: c:\DeprovisioningScript.ps1
- Linux: /tmp/DeprovisioningScript.sh

Image Builder will read these commands, these are written out to the AIB logs, 'customization.log'. See [troubleshooting](#) on how to collect logs.

## Properties: distribute

Azure Image Builder supports three distribution targets:

- **managedImage** - managed image.
- **sharedImage** - Shared Image Gallery.
- **VHD** - VHD in a storage account.

You can distribute an image to both of the target types in the same configuration, please see [examples](#).

Because you can have more than one target to distribute to, Image Builder maintains a state for every distribution target that can be accessed by querying the `runOutputName`. The `runOutputName` is an object you can query post distribution for information about that distribution. For example, you can query the location of the VHD, or regions where the image version was replicated to, or SIG Image version created. This is a property of every distribution target. The `runOutputName` must be unique to each distribution target. Here is an example, this is querying a Shared Image Gallery distribution:

```
subscriptionID=<subscriptionID>
imageResourceGroup=<resourceGroup of image template>
runOutputName=<runOutputName>

az resource show \
 --ids
"/subscriptions/$subscriptionID/resourcegroups/$imageResourceGroup/providers/Microsoft.VirtualMachineImages/imageTemplates/ImageTemplateLinuxRHEL77/runOutputs/$runOutputName" \
 --api-version=2019-05-01-preview
```

Output:

```
{
 "id": "/subscriptions/xxxxxx/resourcegroups/rheltest/providers/Microsoft.VirtualMachineImages/imageTemplates/ImageTemplateLinuxRHEL77/runOutputs/rhel77",
 "identity": null,
 "kind": null,
 "location": null,
 "managedBy": null,
 "name": "rhel77",
 "plan": null,
 "properties": {
 "artifactId": "/subscriptions/xxxxxx/resourceGroups/aibDevOpsImg/providers/Microsoft.Compute/galleries/devOpsSIG/images/rhel/versions/0.24105.52755",
 "provisioningState": "Succeeded"
 },
 "resourceGroup": "rheltest",
 "sku": null,
 "tags": null,
 "type": "Microsoft.VirtualMachineImages/imageTemplates/runOutputs"
}
```

### Distribute: managedImage

The image output will be a managed image resource.

```
"distribute": [
 {
 "type": "managedImage",
 "imageId": "<resource ID>",
 "location": "<region>",
 "runOutputName": "<name>",
 "artifactTags": {
 "<name>": "<value>",
 "<name>": "<value>"
 }
 }
]
```

Distribute properties:

- **type** - managedImage
- **imageId** - Resource ID of the destination image, expected format  
`/subscriptions/<subscriptionId>/resourceGroups/<destinationResourceGroupName>/providers/Microsoft.Compute/images/<imageName>`
- **location** - location of the managed image.
- **runOutputName** - unique name for identifying the distribution.
- **artifactTags** - Optional user specified key value pair tags.

#### NOTE

The destination resource group must exist. If you want the image distributed to a different region, it will increase the deployment time .

### Distribute: sharedImage

The Azure Shared Image Gallery is a new Image Management service that allows managing of image region replication, versioning and sharing custom images. Azure Image Builder supports distributing with this service, so you can distribute images to regions supported by Shared Image Galleries.

A Shared Image Gallery is made up of:

- Gallery - Container for multiple shared images. A gallery is deployed in one region.
- Image definitions - a conceptual grouping for images.
- Image versions - this is an image type used for deploying a VM or scale set. Image versions can be replicated to other regions where VMs need to be deployed.

Before you can distribute to the Image Gallery, you must create a gallery and an image definition, see [Shared images](#).

```
{
 "type": "sharedImage",
 "galleryImageId": "<resource ID>",
 "runOutputName": "<name>",
 "artifactTags": {
 "<name>": "<value>",
 "<name>": "<value>"
 },
 "replicationRegions": [
 "<region where the gallery is deployed>",
 "<region>"
]
}
```

Distribute properties for shared image galleries:

- **type** - sharedImage
- **galleryImageId** – ID of the shared image gallery. The format is:  
`/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/galleries/<sharedImageGalleryName>/images/<imageGalleryName>`
- **runOutputName** – unique name for identifying the distribution.
- **artifactTags** - Optional user specified key value pair tags.
- **replicationRegions** - Array of regions for replication. One of the regions must be the region where the Gallery is deployed.

#### NOTE

You can use Azure Image Builder in a different region to the gallery, but the Azure Image Builder service will need to transfer the image between the datacenters and this will take longer. Image Builder will automatically version the image, based on a monotonic integer, you cannot specify it currently.

### Distribute: VHD

You can output to a VHD. You can then copy the VHD, and use it to publish to Azure MarketPlace, or use with Azure Stack.

```
{
 "type": "VHD",
 "runOutputName": "<VHD name>",
 "tags": {
 "<name>": "<value>",
 "<name>": "<value>"
 }
}
```

OS Support: Windows and Linux

Distribute VHD parameters:

- **type** - VHD.
- **runOutputName** – unique name for identifying the distribution.
- **tags** - Optional user specified key value pair tags.

Azure Image Builder does not allow the user to specify a storage account location, but you can query the status of the `runOutputs` to get the location.

```
az resource show \
 --ids
"/subscriptions/$subscriptionId/resourcegroups/<imageResourceGroup>/providers/Microsoft.VirtualMachineImages/imageTemplates/<imageTemplateName>/runOutputs/<runOutputName>" | grep artifactUri
```

#### NOTE

Once the VHD has been created, copy it to a different location, as soon as possible. The VHD is stored in a storage account in the temporary resource group created when the image template is submitted to the Azure Image Builder service. If you delete the image template, then you will lose the VHD.

## Next steps

There are sample json files for different scenarios in the [Azure Image Builder GitHub](#).

# Find Windows VM images in the Azure Marketplace with Azure PowerShell

11/13/2019 • 6 minutes to read • [Edit Online](#)

This article describes how to use Azure PowerShell to find VM images in the Azure Marketplace. You can then specify a Marketplace image when you create a VM programmatically with PowerShell, Resource Manager templates, or other tools.

You can also browse available images and offers using the [Azure Marketplace storefront](#), the [Azure portal](#), or the [Azure CLI](#).

## Terminology

A Marketplace image in Azure has the following attributes:

- **Publisher:** The organization that created the image. Examples: Canonical, MicrosoftWindowsServer
- **Offer:** The name of a group of related images created by a publisher. Examples: UbuntuServer, WindowsServer
- **SKU:** An instance of an offer, such as a major release of a distribution. Examples: 18.04-LTS, 2019-Datacenter
- **Version:** The version number of an image SKU.

To identify a Marketplace image when you deploy a VM programmatically, supply these values individually as parameters. Some tools accept an image *URN*, which combines these values, separated by the colon (:) character: *Publisher:Offer:SKU:Version*. In a URN, you can replace the version number with "latest", which selects the latest version of the image.

If the image publisher provides additional license and purchase terms, then you must accept those terms and enable programmatic deployment. You'll also need to supply *purchase plan* parameters when deploying a VM programmatically. See [Deploy an image with Marketplace terms](#).

## Table of commonly used Windows images

This table shows a subset of available Skus for the indicated Publishers and Offers.

| PUBLISHER              | OFFER         | SKU                             |
|------------------------|---------------|---------------------------------|
| MicrosoftWindowsServer | WindowsServer | 2019-Datacenter                 |
| MicrosoftWindowsServer | WindowsServer | 2019-Datacenter-Core            |
| MicrosoftWindowsServer | WindowsServer | 2019-Datacenter-with-Containers |
| MicrosoftWindowsServer | WindowsServer | 2016-Datacenter                 |
| MicrosoftWindowsServer | WindowsServer | 2016-Datacenter-Server-Core     |
| MicrosoftWindowsServer | WindowsServer | 2016-Datacenter-with-Containers |
| MicrosoftWindowsServer | WindowsServer | 2012-R2-Datacenter              |

| PUBLISHER              | OFFER                     | SKU             |
|------------------------|---------------------------|-----------------|
| MicrosoftWindowsServer | WindowsServer             | 2012-Datacenter |
| MicrosoftDynamicsNAV   | DynamicsNAV               | 2017            |
| MicrosoftSharePoint    | MicrosoftSharePointServer | 2019            |
| MicrosoftSQLServer     | SQL2019-WS2016            | Enterprise      |
| MicrosoftRServer       | RServer-WS2016            | Enterprise      |

## Navigate the images

One way to find an image in a location is to run the [Get-AzVMImagePublisher](#), [Get-AzVMImageOffer](#), and [Get-AzVMImageSku](#) cmdlets in order:

1. List the image publishers.
2. For a given publisher, list their offers.
3. For a given offer, list their SKUs.

Then, for a selected SKU, run [Get-AzVMImage](#) to list the versions to deploy.

1. List the publishers:

```
$locName=<Azure location, such as West US>
Get-AzVMImagePublisher -Location $locName | Select PublisherName
```

2. Fill in your chosen publisher name and list the offers:

```
$pubName=<publisher>
Get-AzVMImageOffer -Location $locName -PublisherName $pubName | Select Offer
```

3. Fill in your chosen offer name and list the SKUs:

```
$offerName=<offer>
Get-AzVMImageSku -Location $locName -PublisherName $pubName -Offer $offerName | Select Skus
```

4. Fill in your chosen SKU name and get the image version:

```
$skuName=<SKU>
Get-AzVMImage -Location $locName -PublisherName $pubName -Offer $offerName -Sku $skuName | Select Version
```

From the output of the `Get-AzVMImage` command, you can select a version image to deploy a new virtual machine.

The following example shows the full sequence of commands and their outputs:

```
$locName="West US"
Get-AzVMImagePublisher -Location $locName | Select PublisherName
```

Partial output:

```
PublisherName

...
abiquo
accedian
acellion
accessdata-group
accops
Acronis
Acronis.Backup
actian-corp
actian_matrix
actifio
activeeon
adgs
advantech
advantech-webaccess
advantys
...
```

For the *MicrosoftWindowsServer* publisher:

```
$pubName="MicrosoftWindowsServer"
Get-AzVMImageOffer -Location $locName -PublisherName $pubName | Select Offer
```

Output:

```
Offer

Windows-HUB
WindowsServer
WindowsServerSemiAnnual
```

For the *WindowsServer* offer:

```
$offerName="WindowsServer"
Get-AzVMImageSku -Location $locName -PublisherName $pubName -Offer $offerName | Select Skus
```

Partial output:

```
Skus

2008-R2-SP1
2008-R2-SP1-smalldisk
2012-Datacenter
2012-Datacenter-smalldisk
2012-R2-Datacenter
2012-R2-Datacenter-smalldisk
2016-Datacenter
2016-Datacenter-Server-Core
2016-Datacenter-Server-Core-smalldisk
2016-Datacenter-smalldisk
2016-Datacenter-with-Containers
2016-Datacenter-with-RDSH
2019-Datacenter
2019-Datacenter-Core
2019-Datacenter-Core-smalldisk
2019-Datacenter-Core-with-Containers
...
```

Then, for the *2019-Datacenter* SKU:

```
$skuName="2019-Datacenter"
Get-AzVMImage -Location $locName -PublisherName $pubName -Offer $offerName -Sku $skuName | Select Version
```

Now you can combine the selected publisher, offer, SKU, and version into a URN (values separated by `:`). Pass this URN with the `--image` parameter when you create a VM with the `New-AzVM` cmdlet. You can optionally replace the version number in the URN with "latest" to get the latest version of the image.

If you deploy a VM with a Resource Manager template, then you'll set the image parameters individually in the `imageReference` properties. See the [template reference](#).

## Deploy an image with Marketplace terms

Some VM images in the Azure Marketplace have additional license and purchase terms that you must accept before you can deploy them programmatically.

To deploy a VM from such an image, you'll need to both accept the image's terms and enable programmatic deployment. You'll only need to do this once per subscription. Afterward, each time you deploy a VM programmatically from the image you'll also need to specify *purchase plan* parameters.

The following sections show how to:

- Find out whether a Marketplace image has additional license terms
- Accept the terms programmatically
- Provide purchase plan parameters when you deploy a VM programmatically

### View plan properties

To view an image's purchase plan information, run the `Get-AzVMImage` cmdlet. If the `PurchasePlan` property in the output is not `null`, the image has terms you need to accept before programmatic deployment.

For example, the *Windows Server 2016 Datacenter* image doesn't have additional terms, so the `PurchasePlan` information is `null`:

```
$version = "2016.127.20170406"
Get-AzVMImage -Location $locName -PublisherName $pubName -Offer $offerName -Skus $skuName -Version $version
```

Output:

```
Id : /Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/MicrosoftWindowsServer/ArtifactTypes/VMImage/Offers/WindowsServer/Skus/2016-Datacenter/Versions/2019.0.20190115
Location : westus
PublisherName : MicrosoftWindowsServer
Offer : WindowsServer
Skus : 2019-Datacenter
Version : 2019.0.20190115
FilterExpression :
Name : 2019.0.20190115
OSDiskImage : {
 "operatingSystem": "Windows"
 }
PurchasePlan : null
DataDiskImages : []
```

The example below shows a similar command for the *Data Science Virtual Machine - Windows 2016* image,

which has the following `PurchasePlan` properties: `name`, `product`, and `publisher`. Some images also have a `promotion code` property. To deploy this image, see the following sections to accept the terms and to enable programmatic deployment.

```
Get-AzVMImage -Location "westus" -PublisherName "microsoft-ads" -Offer "windows-data-science-vm" -Skus "windows2016" -Version "0.2.02"
```

Output:

```
Id : /Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/microsoft-ads/ArtifactTypes/VMImage/Offers/windows-data-science-vm/Skus/windows2016/Versions/19.01.14
Location : westus
PublisherName : microsoft-ads
Offer : windows-data-science-vm
Skus : windows2016
Version : 19.01.14
FilterExpression :
Name : 19.01.14
OSDiskImage : {
 "operatingSystem": "Windows"
}
PurchasePlan : {
 "publisher": "microsoft-ads",
 "name": "windows2016",
 "product": "windows-data-science-vm"
}
DataDiskImages : []
```

## Accept the terms

To view the license terms, use the [Get-AzMarketplaceterms](#) cmdlet and pass in the purchase plan parameters. The output provides a link to the terms for the Marketplace image and shows whether you previously accepted the terms. Be sure to use all lowercase letters in the parameter values.

```
Get-AzMarketplaceterms -Publisher "microsoft-ads" -Product "windows-data-science-vm" -Name "windows2016"
```

Output:

```
Publisher : microsoft-ads
Product : windows-data-science-vm
Plan : windows2016
LicenseTextLink :
https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_MICROSOFT%253a2DADS%253a24WINDOWS%253a2DDATA%253a2DSCIENCE%253a2DVM%253a24WINDOWS2016%253a240C5SKMQOXSED6BBSNTF4XRC54XLOHP7QMPV54DQU7JCBZNYFP35IDPOWTUKXUC7ZAG7W6ZMDD6NHWNKUIVSYBUTZ245F44SU5AD7Q.txt
PrivacyPolicyLink : https://www.microsoft.com/EN-US/privacystatement/OnlineServices/Default.aspx
Signature :
2UWH6PHSAIM4U22HXPXW25AL2NHUJ7Y7GRV27EBL6SUIDURGMYG6IID03P47FFIBBDHZHSQTR7PNK6VIIRYJRQ3WXSE6BTNUNENXA
Accepted : False
Signdate : 1/25/2019 7:43:00 PM
```

Use the [Set-AzMarketplaceterms](#) cmdlet to accept or reject the terms. You only need to accept terms once per subscription for the image. Be sure to use all lowercase letters in the parameter values.

```
$agreementTerms=Get-AzMarketplaceTerms -Publisher "microsoft-ads" -Product "windows-data-science-vm" -Name "windows2016"

Set-AzMarketplaceTerms -Publisher "microsoft-ads" -Product "windows-data-science-vm" -Name "windows2016" -Terms $agreementTerms -Accept
```

Output:

```
Publisher : microsoft-ads
Product : windows-data-science-vm
Plan : windows2016
LicenseTextLink :
https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_MICROSOFT%253a2DADS%253a24WINDOWS%253a2DDATA%253a2DSCIENCE%253a2DV

M%253a24WINDOWS2016%253a240C5SKMQ0XED66BBSNTF4XRCS4XLOHP7QMPV54DQU7JCBZYFP35IDPOWTUKXUC7ZAG7W6ZMDD6NHWNKUIVS
YBZUTZ245F44SU5AD7Q.txt
PrivacyPolicyLink : https://www.microsoft.com/EN-US/privacystatement/OnlineServices/Default.aspx
Signature :
XXXXXXXX3MNJ5SROEG2BYDA2YGECU33GXTD3UFPLPC4BAVKAUL3PDYL3KBKBLG4ZCDJZVNSA7KJWTGMDSYDD6KRLV3LV274DLBXXXXXX
Accepted : True
Signdate : 2/23/2018 7:49:31 PM
```

## Deploy using purchase plan parameters

After accepting the terms for an image, you can deploy a VM in that subscription. As shown in the following snippet, use the [Set-AzVMPlan](#) cmdlet to set the Marketplace plan information for the VM object. For a complete script to create network settings for the VM and complete the deployment, see the [PowerShell script examples](#).

```
...

$vmConfig = New-AzVMConfig -VMName "myVM" -VMSize Standard_D1

Set the Marketplace plan information

$publisherName = "microsoft-ads"

$productName = "windows-data-science-vm"

$planName = "windows2016"

$vmConfig = Set-AzVMPlan -VM $vmConfig -Publisher $publisherName -Product $productName -Name $planName

$cred=Get-Credential

$vmConfig = Set-AzVMOperatingSystem -Windows -VM $vmConfig -ComputerName "myVM" -Credential $cred

Set the Marketplace image

$offerName = "windows-data-science-vm"

$skuName = "Windows2016"

$version = "19.01.14"

$vmConfig = Set-AzVMSourceImage -VM $vmConfig -PublisherName $publisherName -Offer $offerName -Skus $skuName -Version $version

...
```

You'll then pass the VM configuration along with network configuration objects to the [New-AzVM](#) cmdlet.

## Next steps

To create a virtual machine quickly with the `New-AzVM` cmdlet by using basic image information, see [Create a Windows virtual machine with PowerShell](#).

See a PowerShell script example to [create a fully configured virtual machine](#).

# Prepare a Windows VHD or VHDX to upload to Azure

1/30/2020 • 18 minutes to read • [Edit Online](#)

Before you upload a Windows virtual machine (VM) from on-premises to Azure, you must prepare the virtual hard disk (VHD or VHDX). Azure supports both generation 1 and generation 2 VMs that are in VHD file format and that have a fixed-size disk. The maximum size allowed for the VHD is 1,023 GB.

In a generation 1 VM, you can convert a VHDX file system to VHD. You can also convert a dynamically expanding disk to a fixed-size disk. But you can't change a VM's generation. For more information, see [Should I create a generation 1 or 2 VM in Hyper-V?](#) and [Azure support for generation 2 VMs \(preview\)](#).

For information about the support policy for Azure VMs, see [Microsoft server software support for Azure VMs](#).

## NOTE

The instructions in this article apply to:

1. The 64-bit version of Windows Server 2008 R2 and later Windows Server operating systems. For information about running a 32-bit operating system in Azure, see [Support for 32-bit operating systems in Azure VMs](#).
2. If any Disaster Recovery tool will be used to migrate the workload, like Azure Site Recovery or Azure Migrate, this process is still required to be done and followed on the Guest OS to prepare the image prior the migration.

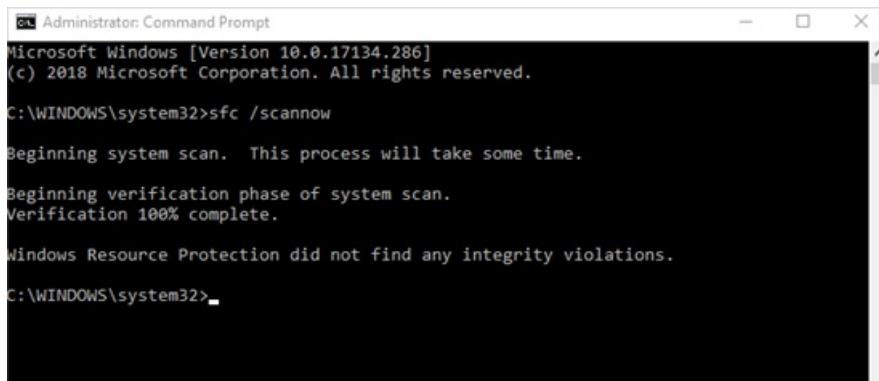
## System File Checker (SFC) command

**Run Windows System File Checker utility (run sfc /scannow) on OS prior to generalization step of creating customer OS image**

The System File Checker (SFC) command is used to verify and replace Windows system files.

To run the SFC command:

1. Open an elevated CMD prompt as Administrator.
2. Type `sfc /scannow` and select **Enter**.



The screenshot shows an Administrator Command Prompt window. The title bar says "Administrator: Command Prompt". The window displays the following text:  
Microsoft Windows [Version 10.0.17134.286]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\WINDOWS\system32>sfc /scannow  
Beginning system scan. This process will take some time.  
Beginning verification phase of system scan.  
Verification 100% complete.  
Windows Resource Protection did not find any integrity violations.  
C:\WINDOWS\system32>

After the SFC scan is completed, try to install Windows Updates and restart the computer.

## Convert the virtual disk to a fixed size and to VHD

If you need to convert your virtual disk to the required format for Azure, use one of the methods in this section:

1. Back up the VM before you run the virtual disk conversion process.
2. Make sure that the Windows VHD works correctly on the local server. Resolve any errors within the VM itself before you try to convert or upload it to Azure.
3. Regarding the size of the VHD:
  - a. All VHDs on Azure must have a virtual size aligned to 1MB. When converting from a raw disk to VHD you must ensure that the raw disk size is a multiple of 1 MB before conversion. Fractions of a megabyte will cause errors when creating images from the uploaded VHD.
  - b. The maximum size allowed for the OS VHD is 2TB.

After you convert the disk, create a VM that uses the disk. Start and sign in to the VM to finish preparing it for uploading.

### Use Hyper-V Manager to convert the disk

1. Open Hyper-V Manager and select your local computer on the left. In the menu above the computer list, select **Action > Edit Disk**.
2. On the **Locate Virtual Hard Disk** page, select your virtual disk.
3. On the **Choose Action** page, select **Convert > Next**.
4. If you need to convert from VHDX, select **VHD > Next**.
5. If you need to convert from a dynamically expanding disk, select **Fixed size > Next**.
6. Locate and select a path to save the new VHD file to.
7. Select **Finish**.

#### NOTE

Use an elevated PowerShell session to run the commands in this article.

### Use PowerShell to convert the disk

You can convert a virtual disk by using the [Convert-VHD](#) command in Windows PowerShell. Select **Run as administrator** when you start PowerShell.

The following example command converts the disk from VHDX to VHD. The command also converts the disk from a dynamically expanding disk to a fixed-size disk.

```
Convert-VHD -Path c:\test\MY-VM.vhdx -DestinationPath c:\test\MY-NEW-VM.vhd -VHDTtype Fixed
```

In this command, replace the value for `-Path` with the path to the virtual hard disk that you want to convert. Replace the value for `-DestinationPath` with the new path and name of the converted disk.

### Convert from VMware VMDK disk format

If you have a Windows VM image in the [VMDK file format](#), use the [Microsoft Virtual Machine Converter](#) to convert it to VHD format. For more information, see [How to convert a VMware VMDK to Hyper-V VHD](#).

## Set Windows configurations for Azure

#### NOTE

Azure platform mounts an ISO file to the DVD-ROM when a Windows VM is created from a generalized image. For this reason, the DVD-ROM must be enabled in the OS in the generalized image. If it is disabled, the Windows VM will be stuck at OOBE.

On the VM that you plan to upload to Azure, run the following commands from an [elevated command prompt window](#):

1. Remove any static persistent route on the routing table:

- To view the route table, run `route print` at the command prompt.
- Check the `Persistence Routes` sections. If there's a persistent route, use the `route delete` command to remove it.

2. Remove the WinHTTP proxy:

```
netsh winhttp reset proxy
```

If the VM needs to work with a specific proxy, add a proxy exception to the Azure IP address ([168.63.129.16](#)) so the VM can connect to Azure:

```
$proxyAddress=<your proxy server>
$proxyBypassList=<your list of bypasses>;168.63.129.16

netsh winhttp set proxy $proxyAddress $proxyBypassList
```

3. Set the disk SAN policy to `Onlineall`:

```
diskpart
```

In the open command prompt window, type the following commands:

```
san policy=onlineall
exit
```

4. Set Coordinated Universal Time (UTC) time for Windows. Also set the startup type of the Windows time service (`w32time`) to `Automatic`:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\TimeZoneInformation' -Name
"RealTimeIsUniversal" -Value 1 -Type DWord -Force

Set-Service -Name w32time -StartupType Automatic
```

5. Set the power profile to high performance:

```
powercfg /setactive SCHEME_MIN
```

6. Make sure the environmental variables `TEMP` and `TMP` are set to their default values:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name
"TEMP" -Value "%SystemRoot%\TEMP" -Type ExpandString -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name
"TMP" -Value "%SystemRoot%\TEMP" -Type ExpandString -Force
```

## Check the Windows services

Make sure that each of the following Windows services is set to the Windows default values. These services are

the minimum that must be set up to ensure VM connectivity. To reset the startup settings, run the following commands:

```
Get-Service -Name bfe | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'
Get-Service -Name dhcp | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'
Get-Service -Name dnscache | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType
'Automatic'
Get-Service -Name IKEEXT | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType
'Automatic'
Get-Service -Name iphlpsvc | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType
'Automatic'
Get-Service -Name netlogon | Where-Object { $_.StartType -ne 'Manual' } | Set-Service -StartupType 'Manual'
Get-Service -Name netman | Where-Object { $_.StartType -ne 'Manual' } | Set-Service -StartupType 'Manual'
Get-Service -Name nsi | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType 'Automatic'
Get-Service -Name TermService | Where-Object { $_.StartType -ne 'Manual' } | Set-Service -StartupType
'Manual'
Get-Service -Name MpsSvc | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType
'Automatic'
Get-Service -Name RemoteRegistry | Where-Object { $_.StartType -ne 'Automatic' } | Set-Service -StartupType
'Automatic'
```

## Update remote-desktop registry settings

Make sure the following settings are configured correctly for remote access:

### NOTE

You might receive an error message when you run

```
Set-ItemProperty -Path 'HKLM:\SOFTWARE\ Policies\Microsoft\Windows NT\Terminal Services' -Name <object name>
-Value <value>
```

. You can safely ignore this message. It means only that the domain isn't pushing that configuration through a Group Policy Object.

1. Remote Desktop Protocol (RDP) is enabled:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server' -Name
"fDenyTSConnections" -Value 0 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SOFTWARE\ Policies\Microsoft\Windows NT\Terminal Services' -Name
"fDenyTSConnections" -Value 0 -Type DWord -Force
```

2. The RDP port is set up correctly. The default port is 3389:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "PortNumber" -Value 3389 -Type DWord -Force
```

When you deploy a VM, the default rules are created against port 3389. If you want to change the port number, do that after the VM is deployed in Azure.

3. The listener is listening in every network interface:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "LanAdapter" -Value 0 -Type DWord -Force
```

4. Configure the network-level authentication (NLA) mode for the RDP connections:

```

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -
Name "UserAuthentication" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -
Name "SecurityLayer" -Value 1 -Type DWord -Force

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -
Name "fAllowSecProtocolNegotiation" -Value 1 -Type DWord -Force

```

5. Set the keep-alive value:

```

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
"KeepAliveEnable" -Value 1 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
"KeepAliveInterval" -Value 1 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "KeepAliveTimeout" -Value 1 -Type DWord -Force

```

6. Reconnect:

```

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
"fDisableAutoReconnect" -Value 0 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "fInheritReconnectSame" -Value 1 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "fReconnectSame" -Value 0 -Type DWord -Force

```

7. Limit the number of concurrent connections:

```

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -
Name "MaxInstanceCount" -Value 4294967295 -Type DWord -Force

```

8. Remove any self-signed certificates tied to the RDP listener:

```

if ((Get-Item -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-
Tcp').Property -contains "SSLCertificateSHA1Hash")
{
 Remove-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-
Tcp' -Name "SSLCertificateSHA1Hash" -Force
}

```

This code ensures that you can connect at the beginning when you deploy the VM. If you need to review this later, you can do so after the VM is deployed in Azure.

9. If the VM will be part of a domain, check the following policies to make sure the former settings aren't reverted.

| GOAL           | POLICY                                                                                                                                               | VALUE                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| RDP is enabled | Computer Configuration\Policies\Windows Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Allow users to connect remotely by using Remote Desktop |

| GOAL                                  | POLICY                                                                                                                                                       | VALUE                                                      |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| NLA group policy                      | Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Security                                                    | Require user authentication for remote access by using NLA |
| Keep-alive settings                   | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Configure keep-alive connection interval                   |
| Reconnect settings                    | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Reconnect automatically                                    |
| Limited number of connection settings | Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections | Limit number of connections                                |

## Configure Windows Firewall rules

- Turn on Windows Firewall on the three profiles (domain, standard, and public):

```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
```

- Run the following command in PowerShell to allow WinRM through the three firewall profiles (domain, private, and public), and enable the PowerShell remote service:

```
Enable-PSRemoting -Force

Set-NetFirewallRule -DisplayName "Windows Remote Management (HTTP-In)" -Enabled True
```

- Enable the following firewall rules to allow the RDP traffic:

```
Set-NetFirewallRule -DisplayGroup "Remote Desktop" -Enabled True
```

- Enable the rule for file and printer sharing so the VM can respond to a ping command inside the virtual network:

```
Set-NetFirewallRule -DisplayName "File and Printer Sharing (Echo Request - ICMPv4-In)" -Enabled True
```

- If the VM will be part of a domain, check the following Azure AD policies to make sure the former settings aren't reverted.

| GOAL                                 | POLICY                                                                                                                                                  | VALUE                                   |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| Enable the Windows Firewall profiles | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall   | Protect all network connections         |
| Enable RDP                           | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall   | Allow inbound Remote Desktop exceptions |
|                                      | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall | Allow inbound Remote Desktop exceptions |
| Enable ICMP-V4                       | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall   | Allow ICMP exceptions                   |
|                                      | Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall | Allow ICMP exceptions                   |

## Verify the VM

Make sure the VM is healthy, secure, and RDP accessible:

1. To make sure the disk is healthy and consistent, check the disk at the next VM restart:

```
Chkdsk /f
```

Make sure the report shows a clean and healthy disk.

2. Set the Boot Configuration Data (BCD) settings.

### NOTE

Use an elevated PowerShell window to run these commands.

```

bcdedit /set "{bootmgr}" integrityservices enable
bcdedit /set "{default}" device partition=C:
bcdedit /set "{default}" integrityservices enable
bcdedit /set "{default}" recoveryenabled Off
bcdedit /set "{default}" osdevice partition=C:
bcdedit /set "{default}" bootstatuspolicy IgnoreAllFailures

#Enable Serial Console Feature
bcdedit /set "{bootmgr}" displaybootmenu yes
bcdedit /set "{bootmgr}" timeout 5
bcdedit /set "{bootmgr}" bootevents yes
bcdedit /ems "{current}" ON
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200

```

3. The dump log can be helpful in troubleshooting Windows crash issues. Enable the dump log collection:

```

Set up the guest OS to collect a kernel dump on an OS crash event
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name CrashDumpEnabled -Type DWord -Force -Value 2
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name DumpFile -Type ExpandString -Force -Value "%SystemRoot%\MEMORY.DMP"
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name NMICrashDump -Type DWord -Force -Value 1

Set up the guest OS to collect user mode dumps on a service crash event
$key = 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps'
if ((Test-Path -Path $key) -eq $false) {(New-Item -Path 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error Reporting' -Name LocalDumps)}
New-ItemProperty -Path $key -Name DumpFolder -Type ExpandString -Force -Value "c:\CrashDumps"
New-ItemProperty -Path $key -Name CrashCount -Type DWord -Force -Value 10
New-ItemProperty -Path $key -Name DumpType -Type DWord -Force -Value 2
Set-Service -Name WerSvc -StartupType Manual

```

4. Verify that the Windows Management Instrumentation (WMI) repository is consistent:

```
winmgmt /verifyrepository
```

If the repository is corrupted, see [WMI: Repository corruption or not.](#)

5. Make sure no other application is using port 3389. This port is used for the RDP service in Azure. To see which ports are used on the VM, run `netstat -anob`:

```
netstat -anob
```

6. To upload a Windows VHD that's a domain controller:

- Follow [these extra steps](#) to prepare the disk.
- Make sure you know the Directory Services Restore Mode (DSRM) password in case you have to start the VM in DSRM at some point. For more information, see [Set a DSRM password](#).

7. Make sure you know the built-in administrator account and password. You might want to reset the current local administrator password and make sure you can use this account to sign in to Windows through the RDP connection. This access permission is controlled by the "Allow log on through Remote Desktop Services" Group Policy Object. View this object in the Local Group Policy Editor here:

Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment

8. Check the following Azure AD policies to make sure you're not blocking your RDP access through RDP or

from the network:

- Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny access to this computer from the network
- Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny log on through Remote Desktop Services

9. Check the following Azure AD policy to make sure you're not removing any of the required access accounts:

- Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Access this computer from the network

The policy should list the following groups:

- Administrators
- Backup Operators
- Everyone
- Users

10. Restart the VM to make sure that Windows is still healthy and can be reached through the RDP connection. At this point, you might want to create a VM in your local Hyper-V to make sure the VM starts completely. Then test to make sure you can reach the VM through RDP.

11. Remove any extra Transport Driver Interface (TDI) filters. For example, remove software that analyzes TCP packets or extra firewalls. If you need to review this later, you can do so after the VM is deployed in Azure.

12. Uninstall any other third-party software or driver that's related to physical components or any other virtualization technology.

### Install Windows updates

Ideally, you should keep the machine updated at the *patch level*. If this isn't possible, make sure the following updates are installed. To get the latest updates, see the Windows update history pages: [Windows 10 and Windows Server 2019](#), [Windows 8.1 and Windows Server 2012 R2](#) and [Windows 7 SP1 and Windows Server 2008 R2 SP1](#).

| COMPONENT | BINARY       | WINDOWS 7 SP1, WINDOWS SERVER 2008 R2 SP1 | WINDOWS 8, WINDOWS SERVER 2012              | WINDOWS 8.1, WINDOWS SERVER 2012 R2 | WINDOWS 10 V1607, WINDOWS SERVER 2016 V1607 | WINDOWS 10 V1703 | WINDOWS 10 V1709, WINDOWS SERVER 2016 V1709 | WINDOWS 10 V1803, WINDOWS SERVER 2016 V1803 |
|-----------|--------------|-------------------------------------------|---------------------------------------------|-------------------------------------|---------------------------------------------|------------------|---------------------------------------------|---------------------------------------------|
| Storage   | disk.sys     | 6.1.7601.23403 - KB3125574                | 6.2.9200.17638 / 6.2.9200.21757 - KB3137061 | 6.3.9600.18203 - KB3137061          | -                                           | -                | -                                           | -                                           |
|           | storport.sys | 6.1.7601.23403 - KB3125574                | 6.2.9200.17188 / 6.2.9200.21306 - KB3018489 | 6.3.9600.18573 - KB4022726          | 10.0.1439.3.1358 - KB4022715                | 10.0.1506.3.332  | -                                           | -                                           |

| <b>COMPONENT</b> | <b>BINARY</b> | <b>WINDOWS 7 SP1, WINDOWS SERVER 2008 R2 SP1</b> | <b>WINDOWS 8, WINDOWS SERVER 2012</b>       | <b>WINDOWS 8.1, WINDOWS SERVER 2012 R2</b> | <b>WINDOWS 10 V1607, WINDOWS SERVER 2016 V1607</b> | <b>WINDOWS 10 V1703</b> | <b>WINDOWS 10 V1709, WINDOWS SERVER 2016 V1709</b> | <b>WINDOWS 10 V1803, WINDOWS SERVER 2016 V1803</b> |
|------------------|---------------|--------------------------------------------------|---------------------------------------------|--------------------------------------------|----------------------------------------------------|-------------------------|----------------------------------------------------|----------------------------------------------------|
|                  | ntfs.sys      | 6.1.7601.23403 - KB3125574                       | 6.2.9200.17623 / 6.2.9200.21743 - KB3121255 | 6.3.9600.18654 - KB4022726                 | 10.0.14393.1198 - KB4022715                        | 10.0.15063.447          | -                                                  | -                                                  |
|                  | lologmsg.dll  | 6.1.7601.23403 - KB3125574                       | 6.2.9200.16384 - KB2995387                  | -                                          | -                                                  | -                       | -                                                  | -                                                  |
|                  | Classpnp.sys  | 6.1.7601.23403 - KB3125574                       | 6.2.9200.17061 / 6.2.9200.21180 - KB2995387 | 6.3.9600.18334 - KB3172614                 | 10.0.14393.953 - KB4022715                         | -                       | -                                                  | -                                                  |
|                  | Volsnap.sys   | 6.1.7601.23403 - KB3125574                       | 6.2.9200.17047 / 6.2.9200.21165 - KB2975331 | 6.3.9600.18265 - KB3145384                 | -                                                  | 10.0.15063.0            | -                                                  | -                                                  |
|                  | partmgr.sys   | 6.1.7601.23403 - KB3125574                       | 6.2.9200.16681 - KB2877114                  | 6.3.9600.17401 - KB3000850                 | 10.0.14393.953 - KB4022715                         | 10.0.15063.0            | -                                                  | -                                                  |
|                  | volmgr.sys    |                                                  |                                             |                                            |                                                    | 10.0.15063.0            | -                                                  | -                                                  |
|                  | Volmgrx.sys   | 6.1.7601.23403 - KB3125574                       | -                                           | -                                          | -                                                  | 10.0.15063.0            | -                                                  | -                                                  |
|                  | Mscsi.sys     | 6.1.7601.23403 - KB3125574                       | 6.2.9200.21006 - KB2955163                  | 6.3.9600.18624 - KB4022726                 | 10.0.14393.1066 - KB4022715                        | 10.0.15063.447          | -                                                  | -                                                  |
|                  | Msdsm.sys     | 6.1.7601.23403 - KB3125574                       | 6.2.9200.21474 - KB3046101                  | 6.3.9600.18592 - KB4022726                 | -                                                  | -                       | -                                                  | -                                                  |
|                  | Mpio.sys      | 6.1.7601.23403 - KB3125574                       | 6.2.9200.21190 - KB3046101                  | 6.3.9600.18616 - KB4022726                 | 10.0.14393.1198 - KB4022715                        | -                       | -                                                  | -                                                  |

| COMPONENT | BINARY          | WINDOWS 7 SP1, WINDOWS SERVER 2008 R2 SP1 | WINDOWS 8, WINDOWS SERVER 2012 | WINDOWS 8.1, WINDOWS SERVER 2012 R2 | WINDOWS 10 V1607, WINDOWS SERVER 2016 | WINDOWS 10 V1703           | WINDOWS 10 V1709, WINDOWS SERVER 2016 | WINDOWS 10 V1803, WINDOWS SERVER 2016 |
|-----------|-----------------|-------------------------------------------|--------------------------------|-------------------------------------|---------------------------------------|----------------------------|---------------------------------------|---------------------------------------|
|           | vmstorfl.sys    | 6.3.9600.18907 - KB4072650                | 6.3.9600.18080 - KB3063109     | 6.3.9600.18907 - KB4072650          | 10.0.14393.2007 - KB4345418           | 10.0.15063.850 - KB4345419 | 10.0.16299.371 - KB4345420            | -                                     |
|           | Fveapi.dll      | 6.1.7601.23311 - KB3125574                | 6.2.9200.20930 - KB2930244     | 6.3.9600.18294 - KB3172614          | 10.0.14393.576 - KB4022715            | -                          | -                                     | -                                     |
|           | Fveapibased.dll | 6.1.7601.23403 - KB3125574                | 6.2.9200.20930 - KB2930244     | 6.3.9600.17415 - KB3172614          | 10.0.14393.206 - KB4022715            | -                          | -                                     | -                                     |
| Network   | netvsc.sys      | -                                         | -                              | -                                   | 10.0.14393.1198 - KB4022715           | 10.0.15063.250 - KB4020001 | -                                     | -                                     |
|           | mrxsmb10.sys    | 6.1.7601.23816 - KB4022722                | 6.2.9200.22108 - KB4022724     | 6.3.9600.18603 - KB4022726          | 10.0.14393.479 - KB4022715            | 10.0.15063.483             | -                                     | -                                     |
|           | mrxsmb20.sys    | 6.1.7601.23816 - KB4022722                | 6.2.9200.21548 - KB4022724     | 6.3.9600.18586 - KB4022726          | 10.0.14393.953 - KB4022715            | 10.0.15063.483             | -                                     | -                                     |
|           | mrxsmb.sys      | 6.1.7601.23816 - KB4022722                | 6.2.9200.22074 - KB4022724     | 6.3.9600.18586 - KB4022726          | 10.0.14393.953 - KB4022715            | 10.0.15063.0               | -                                     | -                                     |
|           | tcpip.sys       | 6.1.7601.23761 - KB4022722                | 6.2.9200.22070 - KB4022724     | 6.3.9600.18478 - KB4022726          | 10.0.14393.1358 - KB4022715           | 10.0.15063.447             | -                                     | -                                     |
|           | http.sys        | 6.1.7601.23403 - KB3125574                | 6.2.9200.17285 - KB3042553     | 6.3.9600.18574 - KB4022726          | 10.0.14393.251 - KB4022715            | 10.0.15063.483             | -                                     | -                                     |
|           | vmswitch.sys    | 6.1.7601.23727 - KB4022719                | 6.2.9200.22117 - KB4022724     | 6.3.9600.18654 - KB4022726          | 10.0.14393.1358 - KB4022715           | 10.0.15063.138             | -                                     | -                                     |

| COMPONENT               | BINARY        | WINDOWS 7 SP1, WINDOWS SERVER 2008 R2 SP1 | WINDOWS 8, WINDOWS SERVER 2012 | WINDOWS 8.1, WINDOWS SERVER 2012 R2 | WINDOWS 10 V1607, WINDOWS SERVER 2016 | WINDOWS 10 V1703 | WINDOWS 10 V1709, WINDOWS SERVER 2016 | WINDOWS 10 V1803, WINDOWS SERVER 2016 |
|-------------------------|---------------|-------------------------------------------|--------------------------------|-------------------------------------|---------------------------------------|------------------|---------------------------------------|---------------------------------------|
| Core                    | ntoskrnl.exe  | 6.1.7601.23807 - KB4022719                | 6.2.9200.22170 - KB4022718     | 6.3.9600.18696 - KB4022726          | 10.0.14393.1358 - KB4022715           | 10.0.15063.483   | -                                     | -                                     |
| Remote Desktop Services | rdpcorets.dll | 6.2.9200.21506 - KB4022719                | 6.2.9200.22104 - KB4022724     | 6.3.9600.18619 - KB4022726          | 10.0.14393.1198 - KB4022715           | 10.0.15063.0     | -                                     | -                                     |
|                         | termsrv.dll   | 6.1.7601.23403 - KB3125574                | 6.2.9200.17048 - KB2973501     | 6.3.9600.17415 - KB3000850          | 10.0.14393.0 - KB4022715              | 10.0.15063.0     | -                                     | -                                     |
|                         | termdd.sys    | 6.1.7601.23403 - KB3125574                | -                              | -                                   | -                                     | -                | -                                     | -                                     |
|                         | win32k.sys    | 6.1.7601.23807 - KB4022719                | 6.2.9200.22168 - KB4022718     | 6.3.9600.18698 - KB4022726          | 10.0.14393.594 - KB4022715            | -                | -                                     | -                                     |
|                         | rdpdd.dll     | 6.1.7601.23403 - KB3125574                | -                              | -                                   | -                                     | -                | -                                     | -                                     |
|                         | rdpwd.sys     | 6.1.7601.23403 - KB3125574                | -                              | -                                   | -                                     | -                | -                                     | -                                     |
| Security                | MS17-010      | KB4012212                                 | KB4012213                      | KB4012213                           | KB4012606                             | KB4012606        | -                                     | -                                     |
|                         |               |                                           | KB4012216                      |                                     | KB4013198                             | KB4013198        | -                                     | -                                     |
|                         |               | KB4012215                                 | KB4012214                      | KB4012216                           | KB4013429                             | KB4013429        | -                                     | -                                     |
|                         |               |                                           | KB4012217                      |                                     | KB4013429                             | KB4013429        | -                                     | -                                     |
|                         | CVE-2018-0886 | KB4103718                                 | KB4103730                      | KB4103725                           | KB4103723                             | KB4103731        | KB4103727                             | KB4103721                             |

| COMPONENT | BINARY | WINDOWS 7 SP1, WINDOWS SERVER 2008 R2 SP1 | WINDOWS 8, WINDOWS SERVER 2012 | WINDOWS 8.1, WINDOWS SERVER 2012 R2 | WINDOWS 10 V1607, WINDOWS SERVER 2016 V1607 | WINDOWS 10 V1703 | WINDOWS 10 V1709 | WINDOWS 10 V1803, WINDOWS SERVER 2016 V1803 |
|-----------|--------|-------------------------------------------|--------------------------------|-------------------------------------|---------------------------------------------|------------------|------------------|---------------------------------------------|
|           |        | KB41037<br>12                             | KB41037<br>26                  | KB41037<br>15                       |                                             |                  |                  |                                             |

#### NOTE

To avoid an accidental reboot during VM provisioning, we recommend ensuring that all Windows Update installations are finished and that no updates are pending. One way to do this is to install all possible Windows updates and reboot once before you run the Sysprep command.

### Determine when to use Sysprep

System Preparation Tool (Sysprep) is a process you can run to reset a Windows installation. Sysprep provides an "out of the box" experience by removing all personal data and resetting several components.

You typically run Sysprep to create a template from which you can deploy several other VMs that have a specific configuration. The template is called a *generalized image*.

If you want to create only one VM from one disk, you don't have to use Sysprep. Instead, you can create the VM from a *specialized image*. For information about how to create a VM from a specialized disk, see:

- [Create a VM from a specialized disk](#)
- [Create a VM from a specialized VHD disk](#)

If you want to create a generalized image, you need to run Sysprep. For more information, see [How to use Sysprep: An introduction](#).

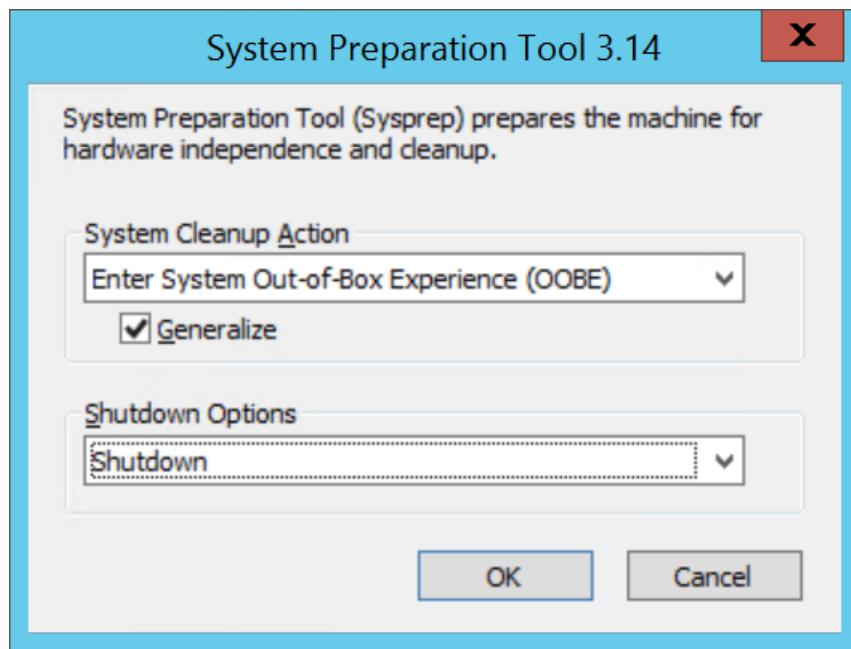
Not every role or application that's installed on a Windows-based computer supports generalized images. So before you run this procedure, make sure Sysprep supports the role of the computer. For more information, see [Sysprep support for server roles](#).

### Generalize a VHD

#### NOTE

After you run `sysprep.exe` in the following steps, turn off the VM. Don't turn it back on until you create an image from it in Azure.

1. Sign in to the Windows VM.
2. Run **Command Prompt** as an administrator.
3. Change the directory to `%windir%\system32\sysprep`. Then run `sysprep.exe`.
4. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that the **Generalize** check box is selected.



5. In **Shutdown Options**, select **Shutdown**.
6. Select **OK**.
7. When Sysprep finishes, shut down the VM. Don't use **Restart** to shut down the VM.

Now the VHD is ready to be uploaded. For more information about how to create a VM from a generalized disk, see [Upload a generalized VHD and use it to create a new VM in Azure](#).

**NOTE**

A custom `unattend.xml` file is not supported. Although we do support the `additionalUnattendContent` property, that provides only limited support for adding `microsoft-windows-shell-setup` options into the `unattend.xml` file that the Azure provisioning agent uses. You can use, for example, `additionalUnattendContent` to add `FirstLogonCommands` and `LogonCommands`. For more information, see [additionalUnattendContent FirstLogonCommands example](#).

## Complete the recommended configurations

The following settings don't affect VHD uploading. However, we strongly recommend that you configured them.

- Install the [Azure Virtual Machine Agent](#). Then you can enable VM extensions. The VM extensions implement most of the critical functionality that you might want to use with your VMs. You'll need the extensions, for example, to reset passwords or configure RDP. For more information, see [Azure Virtual Machine Agent overview](#).
- After you create the VM in Azure, we recommend that you put the page file on the *temporal drive volume* to improve performance. You can set up the file placement as follows:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management' -
Name "PagingFiles" -Value "D:\pagefile.sys" -Type MultiString -Force
```

If a data disk is attached to the VM, the temporal drive volume's letter is typically *D*. This designation could be different, depending on your settings and the number of available drives.

- We recommend disabling script blockers that might be provided by anti-virus software. They might interfere and block the Windows Provisioning Agent scripts executed when you deploy a new VM from your image.

## Next steps

- [Upload a Windows VM image to Azure for Resource Manager deployments](#)
- [Troubleshoot Azure Windows VM activation problems](#)

# Create a managed image of a generalized VM in Azure

12/4/2019 • 5 minutes to read • [Edit Online](#)

A managed image resource can be created from a generalized virtual machine (VM) that is stored as either a managed disk or an unmanaged disk in a storage account. The image can then be used to create multiple VMs. For information on how managed images are billed, see [Managed Disks pricing](#).

## Generalize the Windows VM using Sysprep

Sysprep removes all your personal account and security information, and then prepares the machine to be used as an image. For information about Sysprep, see [Sysprep overview](#).

Make sure the server roles running on the machine are supported by Sysprep. For more information, see [Sysprep support for server roles](#) and [Unsupported scenarios](#).

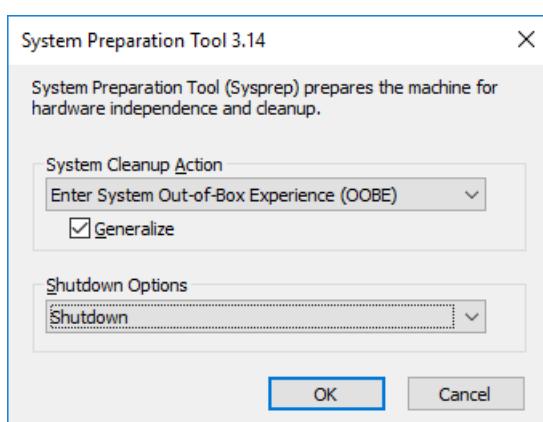
### IMPORTANT

After you have run Sysprep on a VM, that VM is considered *generalized* and cannot be restarted. The process of generalizing a VM is not reversible. If you need to keep the original VM functioning, you should create a [copy of the VM](#) and generalize its copy.

If you plan to run Sysprep before uploading your virtual hard disk (VHD) to Azure for the first time, make sure you have [prepared your VM](#).

To generalize your Windows VM, follow these steps:

1. Sign in to your Windows VM.
2. Open a Command Prompt window as an administrator. Change the directory to %windir%\system32\sysprep, and then run `sysprep.exe`.
3. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)** and select the **Generalize** check box.
4. For **Shutdown Options**, select **Shutdown**.
5. Select **OK**.



6. When Sysprep completes, it shuts down the VM. Do not restart the VM.

#### TIP

**Optional** Use [DISM](#) to optimize your image and reduce your VM's first boot time.

To optimize your image, mount your VHD by double-clicking on it in Windows explorer, and then run DISM with the `/optimize-image` parameter.

```
DISM /image:D:\ /optimize-image /boot
```

Where D: is the mounted VHD's path.

Running `DISM /optimize-image` should be the last modification you make to your VHD. If you make any changes to your VHD prior to deployment, you'll have to run `DISM /optimize-image` again.

## Create a managed image in the portal

1. Go to the [Azure portal](#) to manage the VM image. Search for and select **Virtual machines**.

2. Select your VM from the list.

3. In the **Virtual machine** page for the VM, on the upper menu, select **Capture**.

The **Create image** page appears.

4. For **Name**, either accept the pre-populated name or enter a name that you would like to use for the image.

5. For **Resource group**, either select **Create new** and enter a name, or select a resource group to use from the drop-down list.

6. If you want to delete the source VM after the image has been created, select **Automatically delete this virtual machine after creating the image**.

7. If you want the ability to use the image in any [availability zone](#), select **On** for **Zone resiliency**.

8. Select **Create** to create the image.

After the image is created, you can find it as an **Image** resource in the list of resources in the resource group.

## Create an image of a VM using Powershell

Creating an image directly from the VM ensures that the image includes all of the disks associated with the VM, including the OS disk and any data disks. This example shows how to create a managed image from a VM that uses managed disks.

Before you begin, make sure that you have the latest version of the Azure PowerShell module. To find the version, run `Get-Module -ListAvailable Az` in PowerShell. If you need to upgrade, see [Install Azure PowerShell on Windows with PowerShellGet](#). If you are running PowerShell locally, run `Connect-AzAccount` to create a connection with Azure.

#### NOTE

If you would like to store your image in zone-redundant storage, you need to create it in a region that supports [availability zones](#) and include the `-ZoneResilient` parameter in the image configuration (`New-AzImageConfig` command).

To create a VM image, follow these steps:

1. Create some variables.

```
$vmName = "myVM"
$rgName = "myResourceGroup"
$location = "EastUS"
$imageName = "myImage"
```

2. Make sure the VM has been deallocated.

```
Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force
```

3. Set the status of the virtual machine to **Generalized**.

```
Set-AzVm -ResourceGroupName $rgName -Name $vmName -Generalized
```

4. Get the virtual machine.

```
$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
```

5. Create the image configuration.

```
$image = New-AzImageConfig -Location $location -SourceVirtualMachineId $vm.Id
```

6. Create the image.

```
New-AzImage -Image $image -ImageName $imageName -ResourceGroupName $rgName
```

## Create an image from a managed disk using PowerShell

If you want to create an image of only the OS disk, specify the managed disk ID as the OS disk:

1. Create some variables.

```
$vmName = "myVM"
$rgName = "myResourceGroup"
$location = "EastUS"
$imageName = "myImage"
```

2. Get the VM.

```
$vm = Get-AzVm -Name $vmName -ResourceGroupName $rgName
```

3. Get the ID of the managed disk.

```
$diskID = $vm.StorageProfile.OsDisk.ManagedDisk.Id
```

4. Create the image configuration.

```
$imageConfig = New-AzImageConfig -Location $location
$imageConfig = Set-AzImageOsDisk -Image $imageConfig -OsState Generalized -OsType Windows -
ManagedDiskId $diskID
```

## 5. Create the image.

```
New-AzImage -ImageName $imageName -ResourceGroupName $rgName -Image $imageConfig
```

# Create an image from a snapshot using Powershell

You can create a managed image from a snapshot of a generalized VM by following these steps:

### 1. Create some variables.

```
$rgName = "myResourceGroup"
$location = "EastUS"
$snapshotName = "mySnapshot"
$imageName = "myImage"
```

### 2. Get the snapshot.

```
$snapshot = Get-AzSnapshot -ResourceGroupName $rgName -SnapshotName $snapshotName
```

### 3. Create the image configuration.

```
$imageConfig = New-AzImageConfig -Location $location
$imageConfig = Set-AzImageOsDisk -Image $imageConfig -OsState Generalized -OsType Windows -SnapshotId
$snapshot.Id
```

### 4. Create the image.

```
New-AzImage -ImageName $imageName -ResourceGroupName $rgName -Image $imageConfig
```

# Create an image from a VM that uses a storage account

To create a managed image from a VM that doesn't use managed disks, you need the URI of the OS VHD in the storage account, in the following format:

<https://mystorageaccount.blob.core.windows.net/vhdcontainer/vhdfilename.vhd>. In this example, the VHD is in *mystorageaccount*, in a container named *vhdcontainer*, and the VHD filename is *vhdfilename.vhd*.

### 1. Create some variables.

```
$vmName = "myVM"
$rgName = "myResourceGroup"
$location = "EastUS"
$imageName = "myImage"
$osVhdUri = "https://mystorageaccount.blob.core.windows.net/vhdcontainer/vhdfilename.vhd"
```

### 2. Stop/deallocate the VM.

```
Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force
```

### 3. Mark the VM as generalized.

```
Set-AzVm -ResourceGroupName $rgName -Name $vmName -Generalized
```

4. Create the image by using your generalized OS VHD.

```
$imageConfig = New-AzImageConfig -Location $location
$imageConfig = Set-AzImageOsDisk -Image $imageConfig -OsType Windows -OsState Generalized -BlobUri
$osVhdUri
$image = New-AzImage -ImageName $imageName -ResourceGroupName $rgName -Image $imageConfig
```

## Next steps

- [Create a VM from a managed image.](#)

# Create a VM from a managed image

12/4/2019 • 2 minutes to read • [Edit Online](#)

You can create multiple virtual machines (VMs) from an Azure managed VM image using the Azure portal or PowerShell. A managed VM image contains the information necessary to create a VM, including the OS and data disks. The virtual hard disks (VHDs) that make up the image, including both the OS disks and any data disks, are stored as managed disks.

Before creating a new VM, you'll need to [create a managed VM image](#) to use as the source image and grant read access on the image to any user who should have access to the image.

## Use the portal

1. Go to the [Azure portal](#) to find a managed image. Search for and select **Images**.
2. Select the image you want to use from the list. The image **Overview** page opens.
3. Select **Create VM** from the menu.
4. Enter the virtual machine information. The user name and password entered here will be used to log in to the virtual machine. When complete, select **OK**. You can create the new VM in an existing resource group, or choose **Create new** to create a new resource group to store the VM.
5. Select a size for the VM. To see more sizes, select **View all** or change the **Supported disk type** filter.
6. Under **Settings**, make changes as necessary and select **OK**.
7. On the summary page, you should see your image name listed as a **Private image**. Select **Ok** to start the virtual machine deployment.

## Use PowerShell

You can use PowerShell to create a VM from an image by using the simplified parameter set for the [New-AzVm](#) cmdlet. The image needs to be in the same resource group where you'll create the VM.

The simplified parameter set for [New-AzVm](#) only requires that you provide a name, resource group, and image name to create a VM from an image. New-AzVm will use the value of the **-Name** parameter as the name of all of the resources that it creates automatically. In this example, we provide more detailed names for each of the resources but let the cmdlet create them automatically. You can also create resources beforehand, such as the virtual network, and pass the resource name into the cmdlet. New-AzVm will use the existing resources if it can find them by their name.

The following example creates a VM named *myVMFromImage*, in the *myResourceGroup* resource group, from the image named *myImage*.

```
New-AzVm `
 -ResourceGroupName "myResourceGroup" `
 -Name "myVMfromImage" `
 -ImageName "myImage" `
 -Location "East US" `
 -VirtualNetworkName "myImageVnet" `
 -SubnetName "myImageSubnet" `
 -SecurityGroupName "myImageNSG" `
 -PublicIpAddressName "myImagePIP" `
 -OpenPorts 3389
```

## Next steps

[Create and manage Windows VMs with the Azure PowerShell module](#)

# How to use Packer to create Windows virtual machine images in Azure

11/13/2019 • 6 minutes to read • [Edit Online](#)

Each virtual machine (VM) in Azure is created from an image that defines the Windows distribution and OS version. Images can include pre-installed applications and configurations. The Azure Marketplace provides many first and third-party images for most common OS' and application environments, or you can create your own custom images tailored to your needs. This article details how to use the open-source tool [Packer](#) to define and build custom images in Azure.

This article was last tested on 2/21/2019 using the [Az PowerShell module](#) version 1.3.0 and [Packer](#) version 1.3.4.

## NOTE

Azure now has a service, Azure Image Builder (preview), for defining and creating your own custom images. Azure Image Builder is built on Packer, so you can even use your existing Packer shell provisioner scripts with it. To get started with Azure Image Builder, see [Create a Windows VM with Azure Image Builder](#).

## Create Azure resource group

During the build process, Packer creates temporary Azure resources as it builds the source VM. To capture that source VM for use as an image, you must define a resource group. The output from the Packer build process is stored in this resource group.

Create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup* in the *eastus* location:

```
$rgName = "myResourceGroup"
$location = "East US"
New-AzResourceGroup -Name $rgName -Location $location
```

## Create Azure credentials

Packer authenticates with Azure using a service principal. An Azure service principal is a security identity that you can use with apps, services, and automation tools like Packer. You control and define the permissions as to what operations the service principal can perform in Azure.

Create a service principal with [New-AzADServicePrincipal](#) and assign permissions for the service principal to create and manage resources with [New-AzRoleAssignment](#). The value for `-DisplayName` needs to be unique; replace with your own value as needed.

```
$sp = New-AzADServicePrincipal -DisplayName "PackerServicePrincipal"
$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($sp.Secret)
$plainPassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
New-AzRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName $sp.ApplicationId
```

Then output the password and application ID.

```
$plainPassword
$sp.ApplicationId
```

To authenticate to Azure, you also need to obtain your Azure tenant and subscription IDs with [Get-AzSubscription](#):

```
Get-AzSubscription
```

## Define Packer template

To build images, you create a template as a JSON file. In the template, you define builders and provisioners that carry out the actual build process. Packer has a [builder for Azure](#) that allows you to define Azure resources, such as the service principal credentials created in the preceding step.

Create a file named *windows.json* and paste the following content. Enter your own values for the following:

| PARAMETER                                | WHERE TO OBTAIN                                                    |
|------------------------------------------|--------------------------------------------------------------------|
| <i>client_id</i>                         | View service principal ID with <code>\$sp.applicationId</code>     |
| <i>client_secret</i>                     | View the auto-generated password with <code>\$plainPassword</code> |
| <i>tenant_id</i>                         | Output from <code>\$sub.TenantId</code> command                    |
| <i>subscription_id</i>                   | Output from <code>\$sub.SubscriptionId</code> command              |
| <i>managed_image_resource_group_name</i> | Name of resource group you created in the first step               |
| <i>managed_image_name</i>                | Name for the managed disk image that is created                    |

```
{
 "builders": [
 {
 "type": "azure-arm",
 "client_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
 "client_secret": "ppppppp-pppp-pppp-pppp-pppppppppp",
 "tenant_id": "zzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzz",
 "subscription_id": "yyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyy",
 "managed_image_resource_group_name": "myResourceGroup",
 "managed_image_name": "myPackerImage",
 "os_type": "Windows",
 "image_publisher": "MicrosoftWindowsServer",
 "image_offer": "WindowsServer",
 "image_sku": "2016-Datacenter",
 "communicator": "winrm",
 "winrm_use_ssl": true,
 "winrm_insecure": true,
 "winrm_timeout": "5m",
 "winrm_username": "packer",
 "azure_tags": {
 "dept": "Engineering",
 "task": "Image deployment"
 },
 "location": "East US",
 "vm_size": "Standard_DS2_v2"
 }
],
 "provisioners": [
 {
 "type": "powershell",
 "inline": [
 "Add-WindowsFeature Web-Server",
 "& $env:SystemRoot\\System32\\Sysprep\\Sysprep.exe /oobe /generalize /quiet /quit",
 "while($true) { $imageState = Get-ItemProperty HKLM:\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Setup\\State | Select ImageState; if($imageState.ImageState -ne 'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { Write-Output $imageState.ImageState; Start-Sleep -s 10 } else { break } }"
]
 }
]
}
```

This template builds a Windows Server 2016 VM, installs IIS, then generalizes the VM with Sysprep. The IIS install shows how you can use the PowerShell provisioner to run additional commands. The final Packer image then includes the required software install and configuration.

## Build Packer image

If you don't already have Packer installed on your local machine, [follow the Packer installation instructions](#).

Build the image by opening a cmd prompt and specifying your Packer template file as follows:

```
./packer build windows.json
```

An example of the output from the preceding commands is as follows:

```
azure-arm output will be in this color.

==> azure-arm: Running builder ...
 azure-arm: Creating Azure Resource Manager (ARM) client ...
--> azure-arm: Creating resource group
```

```

--> azure-arm: Creating resource group ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> Location : 'East US'
==> azure-arm: -> Tags :
==> azure-arm: ->> task : Image deployment
==> azure-arm: ->> dept : Engineering
==> azure-arm: Validating deployment template ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> DeploymentName : 'pkrdppq0mthtbtt'
==> azure-arm: Deploying deployment template ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> DeploymentName : 'pkrdppq0mthtbtt'
==> azure-arm: Getting the certificate's URL ...
==> azure-arm: -> Key Vault Name : 'pkrvvpq0mthtbtt'
==> azure-arm: -> Key Vault Secret Name : 'packerKeyVaultSecret'
==> azure-arm: -> Certificate URL :
'https://pkrvvpq0mthtbtt.vault.azure.net/secrets/packerKeyVaultSecret/8c7bd823e4fa44e1abb747636128adbb'
==> azure-arm: Setting the certificate's URL ...
==> azure-arm: Validating deployment template ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> DeploymentName : 'pkrdppq0mthtbtt'
==> azure-arm: Deploying deployment template ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> DeploymentName : 'pkrdppq0mthtbtt'
==> azure-arm: Getting the VM's IP address ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> PublicIPAddressName : 'packerPublicIP'
==> azure-arm: -> NicName : 'packerNic'
==> azure-arm: -> Network Connection : 'PublicEndpoint'
==> azure-arm: -> IP Address : '40.76.55.35'
==> azure-arm: Waiting for WinRM to become available...
==> azure-arm: Connected to WinRM!
==> azure-arm: Provisioning with Powershell...
==> azure-arm: Provisioning with shell script: /var/folders/h1/ymh5bdx15wgdn5hvgj1wc0zh0000gn/T/packer-
powershell-provisioner902510110
 azure-arm: #< CLIXML
 azure-arm:
 azure-arm: Success Restart Needed Exit Code Feature Result
 azure-arm: ----- ----- ----- -----
 azure-arm: True No Success {Common HTTP Features, Default Document, D...
 azure-arm: <Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04"><Obj
S="progress" RefId="0"><TN RefId="0"><T>System.Management.Automation.PSCustomObject</T><T>System.Object</T>
</TN><MS><I64 N="SourceId">1</I64><PR N="Record"><AV>Preparing modules for first use.</AV><AI>0</AI><Nil />
<PI>-1</PI><PC>-1</PC><T>Completed</T><SR>-1</SR><SD> </SD></PR></MS></Obj></Objs>
==> azure-arm: Querying the machine's properties ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> ComputeName : 'pkrvmpq0mthtbtt'
==> azure-arm: -> Managed OS Disk : '/subscriptions/guid/resourceGroups/packer-Resource-Group-
pq0mthtbtt/providers/Microsoft.Compute/disks/osdisk'
==> azure-arm: Powering off machine ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> ComputeName : 'pkrvmpq0mthtbtt'
==> azure-arm: Capturing image ...
==> azure-arm: -> Compute ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: -> Compute Name : 'pkrvmpq0mthtbtt'
==> azure-arm: -> Compute Location : 'East US'
==> azure-arm: -> Image ResourceGroupName : 'myResourceGroup'
==> azure-arm: -> Image Name : 'myPackerImage'
==> azure-arm: -> Image Location : 'eastus'
==> azure-arm: Deleting resource group ...
==> azure-arm: -> ResourceGroupName : 'packer-Resource-Group-pq0mthtbtt'
==> azure-arm: Deleting the temporary OS disk ...
==> azure-arm: -> OS Disk : skipping, managed disk was used...
Build 'azure-arm' finished.

==> Builds finished. The artifacts of successful builds are:
--> azure-arm: Azure.ResourceManagement.VMImage:

ManagedImageResourceGroupName: myResourceGroup

```

```
ManagedImageName: myPackerImage
ManagedImageLocation: eastus
```

It takes a few minutes for Packer to build the VM, run the provisioners, and clean up the deployment.

## Create a VM from the Packer image

You can now create a VM from your Image with [New-AzVM](#). The supporting network resources are created if they do not already exist. When prompted, enter an administrative username and password to be created on the VM.

The following example creates a VM named *myVM* from *myPackerImage*:

```
New-AzVm `
-ResourceGroupName $rgName `
-Name "myVM" `
-Location $location `
-VirtualNetworkName "myVnet" `
-SubnetName "mySubnet" `
-SecurityGroupName "myNetworkSecurityGroup" `
-PublicIpAddressName "myPublicIpAddress" `
-OpenPorts 80 `
-Image "myPackerImage"
```

If you wish to create VMs in a different resource group or region than your Packer image, specify the image ID rather than image name. You can obtain the image ID with [Get-AzImage](#).

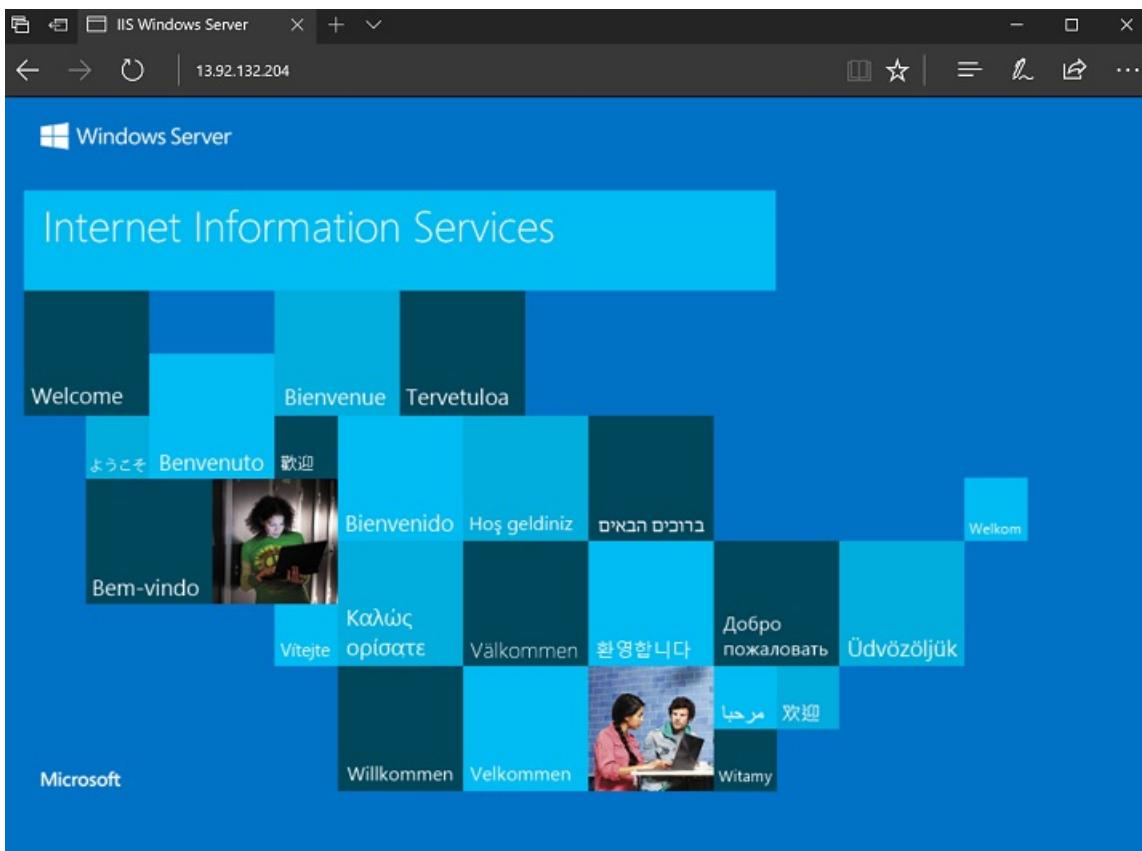
It takes a few minutes to create the VM from your Packer image.

## Test VM and webserver

Obtain the public IP address of your VM with [Get-AzPublicIPAddress](#). The following example obtains the IP address for *myPublicIP* created earlier:

```
Get-AzPublicIPAddress `
-ResourceGroupName $rgName `
-Name "myPublicIPAddress" | select "IpAddress"
```

To see your VM, that includes the IIS install from the Packer provisioner, in action, enter the public IP address in to a web browser.



## Next steps

You can also use existing Packer provisioner scripts with [Azure Image Builder](#).

# Use Windows client in Azure for dev/test scenarios

11/13/2019 • 2 minutes to read • [Edit Online](#)

You can use Windows 7, Windows 8, or Windows 10 Enterprise (x64) in Azure for dev/test scenarios provided you have an appropriate Visual Studio (formerly MSDN) subscription. This article outlines the eligibility requirements for running Windows 7, Windows 8.1, Windows 10 Enterprise in Azure and use of the following Azure Gallery images.

|                                                                                                                         |           |         |
|-------------------------------------------------------------------------------------------------------------------------|-----------|---------|
|  Windows 10 Enterprise N (x64)         | Microsoft | Compute |
|  Windows 8.1 Enterprise N (x64)        | Microsoft | Compute |
|  Windows 7 Enterprise N with SP1 (x64) | Microsoft | Compute |

## NOTE

For Windows 10 Pro and Windows 10 Pro N image in Azure Gallery, please refer to [How to deploy Windows 10 on Azure with Multitenant Hosting Rights](#)

| NAME                                                                                                               | PUBLISHER | CATEGORY |
|--------------------------------------------------------------------------------------------------------------------|-----------|----------|
|  Windows 10 Pro, Version 1709    | Microsoft | Compute  |
|  Windows 10 Pro N, Version 1709 | Microsoft | Compute  |

## Subscription eligibility

Active Visual Studio subscribers (people who have acquired a Visual Studio subscription license) can use Windows client for development and testing purposes. Windows client can be used on your own hardware and Azure virtual machines running in any type of Azure subscription. Windows client may not be deployed to or used on Azure for normal production use, or used by people who are not active Visual Studio subscribers.

For your convenience, certain Windows 10 images are available from the Azure Gallery within [eligible dev/test offers](#). Visual Studio subscribers within any type of offer can also [adequately prepare and create](#) a 64-bit Windows 7, Windows 8, or Windows 10 image and then [upload to Azure](#). The use remains limited to dev/test by active Visual Studio subscribers.

## Eligible offers

The following table details the offer IDs that are eligible to deploy Windows 10 through the Azure Gallery. The Windows 10 images are only visible to the following offers. Visual Studio subscribers who need to run Windows client in a different offer type require you to [adequately prepare and create](#) a 64-bit Windows 7, Windows 8, or Windows 10 image and [then upload to Azure](#).

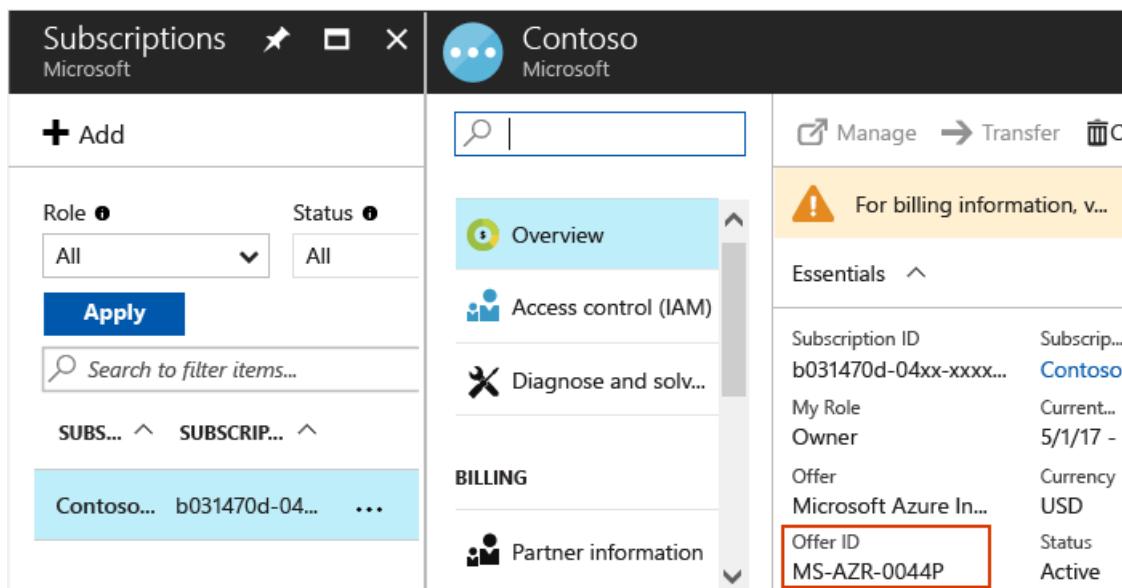
| OFFER NAME                                                 | OFFER NUMBER | AVAILABLE CLIENT IMAGES |
|------------------------------------------------------------|--------------|-------------------------|
| <a href="#">Pay-As-You-Go Dev/Test</a>                     | 0023P        | Windows 10              |
| <a href="#">Visual Studio Enterprise (MPN) subscribers</a> | 0029P        | Windows 10              |

| OFFER NAME                                      | OFFER NUMBER | AVAILABLE CLIENT IMAGES |
|-------------------------------------------------|--------------|-------------------------|
| Visual Studio Professional subscribers          | 0059P        | Windows 10              |
| Visual Studio Test Professional subscribers     | 0060P        | Windows 10              |
| Visual Studio Premium with MSDN (benefit)       | 0061P        | Windows 10              |
| Visual Studio Enterprise subscribers            | 0063P        | Windows 10              |
| Visual Studio Enterprise (BizSpark) subscribers | 0064P        | Windows 10              |
| Enterprise Dev/Test                             | 0148P        | Windows 10              |

## Check your Azure subscription

If you do not know your offer ID, you can obtain it through the Azure portal in one of these two ways:

- On the *Subscriptions* window:



The screenshot shows the Azure Subscriptions window for the Contoso Microsoft account. The left pane displays a list of subscriptions, with the first one, 'Contoso...', selected. The right pane shows the subscription details under the 'BILLING' tab. The 'Offer ID' field, which contains 'MS-AZR-0044P', is highlighted with a red box.

| Subscription                     | Offer ID     |
|----------------------------------|--------------|
| Contoso... b031470d-04xx-xxxx... | MS-AZR-0044P |

- Or, click **Billing** and then click your subscription ID. The offer ID appears in the *Billing* window.

You can also view the offer ID from the '[Subscriptions](#)' tab of the Azure Account portal:

**ACCOUNT ADMINISTRATOR**

**SUBSCRIPTION ID**

**ORDER ID**

**OFFER**

[Visual Studio Enterprise](#)

**OFFER ID**

MS-AZR-0063P

**CURRENCY**

USD

**STATUS**

Active

## Next steps

You can now deploy your VMs using [PowerShell](#), [Resource Manager templates](#), or [Visual Studio](#).

# Download a Windows VHD from Azure

1/14/2020 • 2 minutes to read • [Edit Online](#)

In this article, you learn how to download a Windows virtual hard disk (VHD) file from Azure using the Azure portal.

## Optional: Generalize the VM

If you want to use the VHD as an [image](#) to create other VMs, you should use [Sysprep](#) to generalize the operating system.

To use the VHD as an image to create other VMs, generalize the VM.

1. If you haven't already done so, sign in to the [Azure portal](#).
2. [Connect to the VM](#).
3. On the VM, open the Command Prompt window as an administrator.
4. Change the directory to `%windir%\system32\sysprep` and run `sysprep.exe`.
5. In the System Preparation Tool dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that **Generalize** is selected.
6. In Shutdown Options, select **Shutdown**, and then click **OK**.

## Stop the VM

A VHD can't be downloaded from Azure if it's attached to a running VM. You need to stop the VM to download a VHD.

1. On the Hub menu in the Azure portal, click **Virtual Machines**.
2. Select the VM from the list.
3. On the blade for the VM, click **Stop**.

## Generate download URL

To download the VHD file, you need to generate a [shared access signature \(SAS\)](#) URL. When the URL is generated, an expiration time is assigned to the URL.

1. On the page for the VM, click **Disk**s in the left menu.
2. Select the operating system disk for the VM.
3. On the page for the disk, select **Disk Export** from the left menu.
4. The default expiration time of the URL is 3600 seconds. Increase this to **36000** for Windows OS disks.
5. Click **Generate URL**.

### NOTE

The expiration time is increased from the default to provide enough time to download the large VHD file for a Windows Server operating system. You can expect a VHD file that contains the Windows Server operating system to take several hours to download depending on your connection. If you are downloading a VHD for a data disk, the default time is sufficient.

## Download VHD

1. Under the URL that was generated, click Download the VHD file.
2. You may need to click **Save** in your browser to start the download. The default name for the VHD file is *abcd*.

## Next steps

- Learn how to [upload a VHD file to Azure](#).
- [Create managed disks from unmanaged disks in a storage account](#).
- [Manage Azure disks with PowerShell](#).

2 minutes to read

# Manage the availability of Windows virtual machines in Azure

2/28/2020 • 9 minutes to read • [Edit Online](#)

Learn ways to set up and manage multiple virtual machines to ensure high availability for your Windows application in Azure. You can also [manage the availability of Linux virtual machines](#).

## Understand VM Reboots - maintenance vs. downtime

There are three scenarios that can lead to virtual machine in Azure being impacted: unplanned hardware maintenance, unexpected downtime, and planned maintenance.

- **Unplanned Hardware Maintenance Event** occurs when the Azure platform predicts that the hardware or any platform component associated to a physical machine, is about to fail. When the platform predicts a failure, it will issue an unplanned hardware maintenance event to reduce the impact to the virtual machines hosted on that hardware. Azure uses [Live Migration](#) technology to migrate the Virtual Machines from the failing hardware to a healthy physical machine. Live Migration is a VM preserving operation that only pauses the Virtual Machine for a short time. Memory, open files, and network connections are maintained, but performance might be reduced before and/or after the event. In cases where Live Migration cannot be used, the VM will experience Unexpected Downtime, as described below.
- **An Unexpected Downtime** is when the hardware or the physical infrastructure for the virtual machine fails unexpectedly. This can include local network failures, local disk failures, or other rack level failures. When detected, the Azure platform automatically migrates (heals) your virtual machine to a healthy physical machine in the same datacenter. During the healing procedure, virtual machines experience downtime (reboot) and in some cases loss of the temporary drive. The attached OS and data disks are always preserved.

Virtual machines can also experience downtime in the unlikely event of an outage or disaster that affects an entire datacenter, or even an entire region. For these scenarios, Azure provides protection options including [availability zones](#) and [paired regions](#).

- **Planned Maintenance events** are periodic updates made by Microsoft to the underlying Azure platform to improve overall reliability, performance, and security of the platform infrastructure that your virtual machines run on. Most of these updates are performed without any impact upon your Virtual Machines or Cloud Services (see [VM Preserving Maintenance](#)). While the Azure platform attempts to use VM Preserving Maintenance in all possible occasions, there are rare instances when these updates require a reboot of your virtual machine to apply the required updates to the underlying infrastructure. In this case, you can perform Azure Planned Maintenance with Maintenance-Redeploy operation by initiating the maintenance for their VMs in the suitable time window. For more information, see [Planned Maintenance for Virtual Machines](#).

To reduce the impact of downtime due to one or more of these events, we recommend the following high availability best practices for your virtual machines:

- [Configure multiple virtual machines in an availability set for redundancy](#)
- [Use managed disks for VMs in an availability set](#)
- [Use scheduled events to proactively response to VM impacting events](#)
- [Configure each application tier into separate availability sets](#)
- [Combine a Load Balancer with availability sets](#)

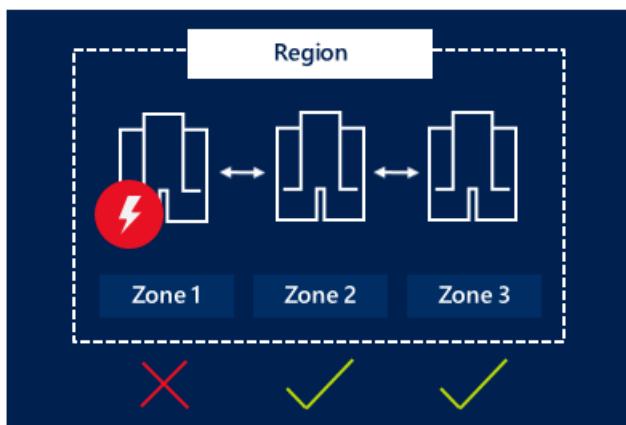
- Use availability zones to protect from datacenter level failures

## Use availability zones to protect from datacenter level failures

**Availability zones** expand the level of control you have to maintain the availability of the applications and data on your VMs. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there are a minimum of three separate zones in all enabled regions. The physical separation of Availability Zones within a region protects applications and data from datacenter failures. Zone-redundant services replicate your applications and data across Availability Zones to protect from single-points-of-failure.

An Availability Zone in an Azure region is a combination of a **fault domain** and an **update domain**. For example, if you create three or more VMs across three zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains. The Azure platform recognizes this distribution across update domains to make sure that VMs in different zones are not updated at the same time.

With Availability Zones, Azure offers industry best 99.99% VM uptime SLA. By architecting your solutions to use replicated VMs in zones, you can protect your applications and data from the loss of a datacenter. If one zone is compromised, then replicated apps and data are instantly available in another zone.



Learn more about deploying a [Windows](#) or [Linux](#) VM in an Availability Zone.

## Configure multiple virtual machines in an availability set for redundancy

Availability sets are another datacenter configuration to provide VM redundancy and availability. This configuration within a datacenter ensures that during either a planned or unplanned maintenance event, at least one virtual machine is available and meets the 99.95% Azure SLA. For more information, see the [SLA for Virtual Machines](#).

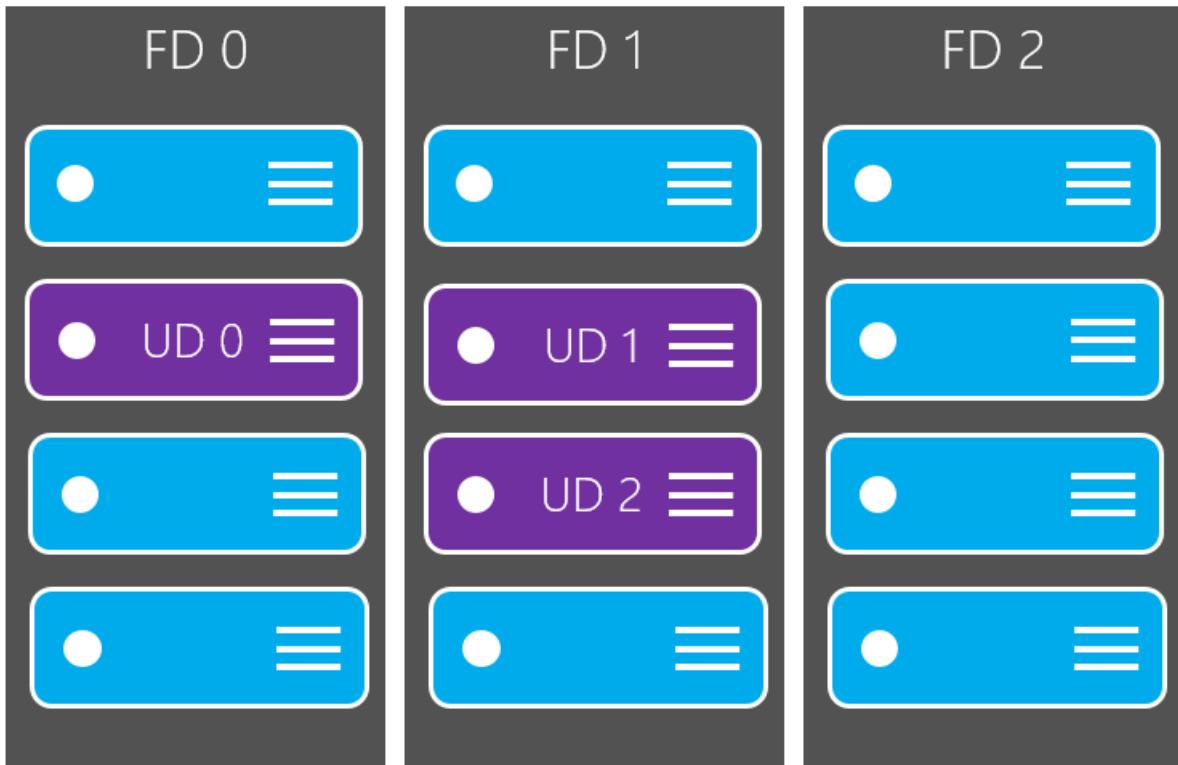
### IMPORTANT

A single instance virtual machine in an availability set by itself should use Premium SSD or Ultra Disk for all operating system disks and data disks in order to qualify for the SLA for Virtual Machine connectivity of at least 99.9%.

Each virtual machine in your availability set is assigned an **update domain** and a **fault domain** by the underlying Azure platform. For a given availability set, five non-user-configurable update domains are assigned by default (Resource Manager deployments can then be increased to provide up to 20 update domains) to indicate groups of virtual machines and underlying physical hardware that can be rebooted at the same time. When more than five virtual machines are configured within a single availability set, the sixth virtual machine is placed into the same update domain as the first virtual machine, the seventh in the same update domain as the second virtual machine, and so on. The order of update domains being rebooted may not proceed sequentially.

during planned maintenance, but only one update domain is rebooted at a time. A rebooted update domain is given 30 minutes to recover before maintenance is initiated on a different update domain.

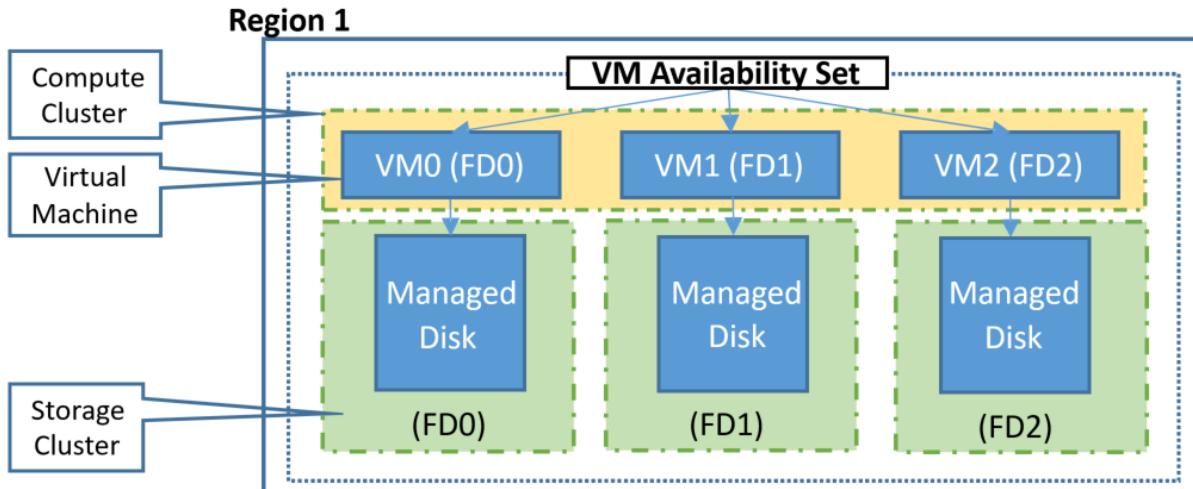
Fault domains define the group of virtual machines that share a common power source and network switch. By default, the virtual machines configured within your availability set are separated across up to three fault domains for Resource Manager deployments (two fault domains for Classic). While placing your virtual machines into an availability set does not protect your application from operating system or application-specific failures, it does limit the impact of potential physical hardware failures, network outages, or power interruptions.



## Use managed disks for VMs in an availability set

If you are currently using VMs with unmanaged disks, we highly recommend you [convert VMs in Availability Set to use Managed Disks](#).

[Managed disks](#) provide better reliability for Availability Sets by ensuring that the disks of VMs in an Availability Set are sufficiently isolated from each other to avoid single points of failure. It does this by automatically placing the disks in different storage fault domains (storage clusters) and aligning them with the VM fault domain. If a storage fault domain fails due to hardware or software failure, only the VM instance with disks on the storage fault domain fails.



#### IMPORTANT

The number of fault domains for managed availability sets varies by region - either two or three per region. You can see the fault domain for each region by running the following scripts.

```
Get-AzComputeResourceSku | where{$_ . ResourceType -eq 'availabilitySets' -and $_ . Name -eq 'Aligned'}
```

```
az vm list-skus --resource-type availabilitySets --query '[?name==`Aligned`].{Location:locationInfo[0].location, MaximumFaultDomainCount:capabilities[0].value}' -o Table
```

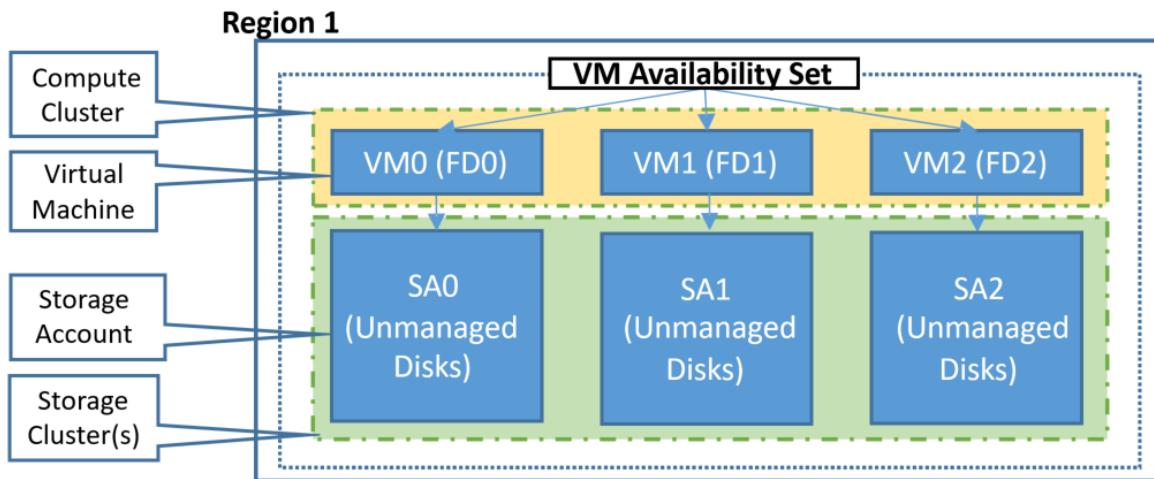
#### NOTE

Under certain circumstances, 2 VMs in the same AvailabilitySet could share the same FaultDomain. This can be confirmed by going into your availability set and checking the **Fault Domain** column. This can be caused from the following sequence while deploying the VMs:

- Deploy the 1st VM
- Stop/Deallocate the 1st VM
- Deploy the 2nd VM Under these circumstances, the OS Disk of the 2nd VM might be created on the same Fault Domain as the 1st VM, and so the 2nd VM will also land on the same FaultDomain. To avoid this issue, it's recommended to not stop/deallocate the VMs between deployments.

If you plan to use VMs with unmanaged disks, follow below best practices for Storage accounts where virtual hard disks (VHDs) of VMs are stored as [page blobs](#).

1. **Keep all disks (OS and data) associated with a VM in the same storage account**
2. **Review the limits on the number of unmanaged disks in an Azure Storage account** before adding more VHDs to a storage account
3. **Use a separate storage account for each VM in an Availability Set.** Do not share Storage accounts with multiple VMs in the same Availability Set. It is acceptable for VMs across different Availability Sets to share storage accounts if above best practices are followed



## Use scheduled events to proactively respond to VM impacting events

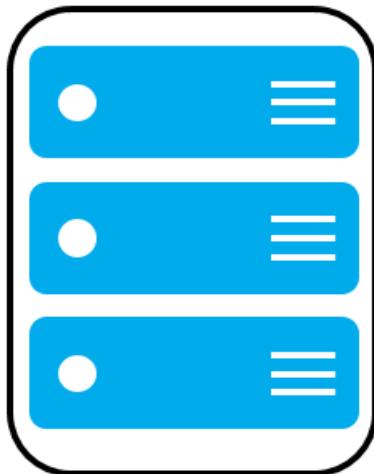
When you subscribe to [scheduled events](#), your VM is notified about upcoming maintenance events that can impact your VM. When scheduled events are enabled, your virtual machine is given a minimum amount of time before the maintenance activity is performed. For example, Host OS updates that might impact your VM are queued up as events that specify the impact, as well as a time at which the maintenance will be performed if no action is taken. Schedule events are also queued up when Azure detects imminent hardware failure that might impact your VM, which allows you to decide when the healing should be performed. Customers can use the event to perform tasks prior to the maintenance, such as saving state, failing over to the secondary, and so on. After you complete your logic for gracefully handling the maintenance event, you can approve the outstanding scheduled event to allow the platform to proceed with maintenance.

## Configure each application tier into separate availability zones or availability sets

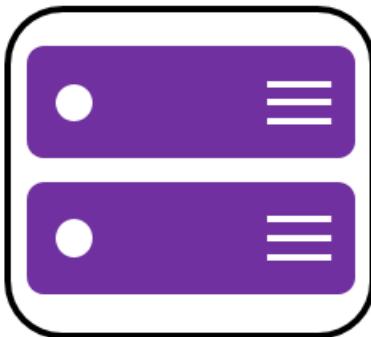
If your virtual machines are all nearly identical and serve the same purpose for your application, we recommend that you configure an availability zone or availability set for each tier of your application. If you place two different tiers in the same availability zone or set, all virtual machines in the same application tier can be rebooted at once. By configuring at least two virtual machines in an availability zone or set for each tier, you guarantee that at least one virtual machine in each tier is available.

For example, you could put all the virtual machines in the front end of your application running IIS, Apache, and Nginx in a single availability zone or set. Make sure that only front-end virtual machines are placed in the same availability zone or set. Similarly, make sure that only data-tier virtual machines are placed in their own availability zone or set, like your replicated SQL Server virtual machines, or your MySQL virtual machines.

## Web Tier Availability Set



## Data Tier Availability Set



## Combine a load balancer with availability zones or sets

Combine the [Azure Load Balancer](#) with an availability zone or set to get the most application resiliency. The Azure Load Balancer distributes traffic between multiple virtual machines. For our Standard tier virtual machines, the Azure Load Balancer is included. Not all virtual machine tiers include the Azure Load Balancer. For more information about load balancing your virtual machines, see [Load Balancing virtual machines](#).

If the load balancer is not configured to balance traffic across multiple virtual machines, then any planned maintenance event affects the only traffic-serving virtual machine, causing an outage to your application tier. Placing multiple virtual machines of the same tier under the same load balancer and availability set enables traffic to be continuously served by at least one instance.

For a tutorial on how to load balance across availability zones, see [Load balance VMs across all availability zones by using the Azure CLI](#).

## Next steps

To learn more about load balancing your virtual machines, see [Load Balancing virtual machines](#).

View Reference Architectures for running N-tier applications on SQL Server in IaaS

- [Windows N-tier application on Azure with SQL Server](#)
- [Run an N-tier application in multiple Azure regions for high availability](#)

# Create a proximity placement group using the portal

10/30/2019 • 2 minutes to read • [Edit Online](#)

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a [proximity placement group](#).

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

## Create the proximity placement group

1. Type **proximity placement group** in the search.
2. Under **Services** in the search results, select **Proximity placement groups**.
3. In the **Proximity placement groups** page, select **Add**.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected.
5. In **Resource group** either select **Create new** to create a new group or select an existing resource group from the drop-down.
6. In **Region** select the location where you want the proximity placement group to be created.
7. In **Proximity placement group name** type a name and then select **Review + create**.
8. After validation passes, select **Create** to create the proximity placement group.

Home > New > Proximity Placement Group > Create Proximity Placement Group

## Create Proximity Placement Group

Basics Tags Review + create

Fill out the required fields and then review the information on the Review + create tab. Once you're satisfied, click Create to deploy.

**Project details**

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

Subscription \* ⓘ Pay-As-You-Go

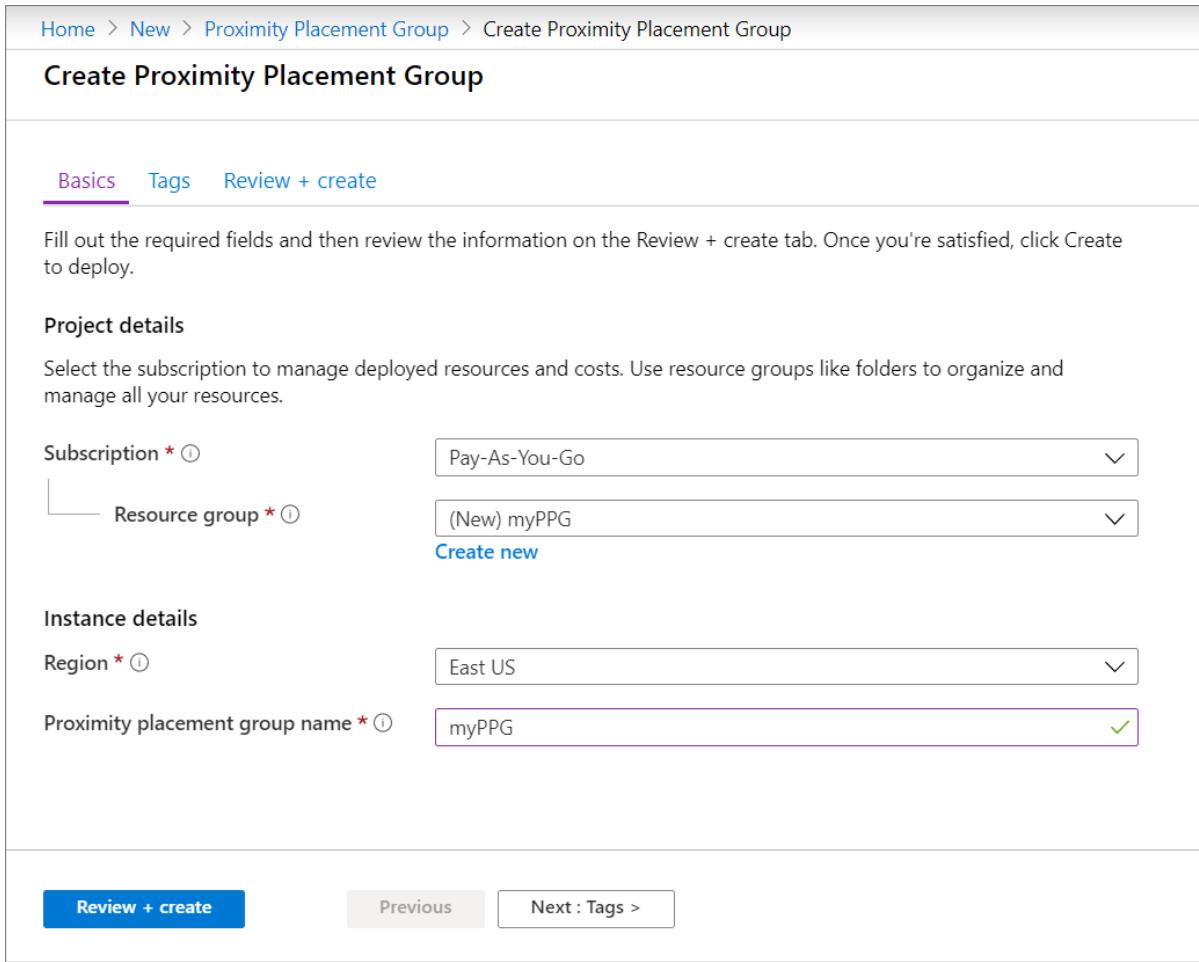
Resource group \* ⓘ (New) myPPG Create new

**Instance details**

Region \* ⓘ East US

Proximity placement group name \* ⓘ myPPG ✓

**Review + create** Previous Next : Tags >



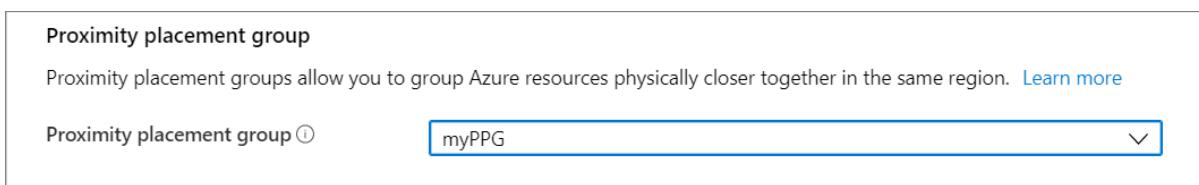
## Create a VM

1. While creating a VM in the portal, go to the **Advanced** tab.
2. In the **Proximity placement group** selection, select the correct placement group.

Proximity placement group

Proximity placement groups allow you to group Azure resources physically closer together in the same region. [Learn more](#)

Proximity placement group ⓘ myPPG



3. When you are done making all of the other required selections, select **Review + create**.
4. After it passes validation, select **Create** to deploy the VM in the placement group.

## Next steps

You can also use the [Azure PowerShell](#) to create proximity placement groups.

# Deploy VMs to proximity placement groups using PowerShell

2/11/2020 • 3 minutes to read • [Edit Online](#)

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a [proximity placement group](#).

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

## Create a proximity placement group

Create a proximity placement group using the [New-AzProximityPlacementGroup](#) cmdlet.

```
$resourceGroup = "myPPGResourceGroup"
$location = "East US"
$ppgName = "myPPG"
New-AzResourceGroup -Name $resourceGroup -Location $location
$ppg = New-AzProximityPlacementGroup `
 -Location $location `
 -Name $ppgName `
 -ResourceGroupName $resourceGroup `
 -ProximityPlacementGroupType Standard
```

## List proximity placement groups

You can list all of the proximity placement groups using the [Get-AzProximityPlacementGroup](#) cmdlet.

```
Get-AzProximityPlacementGroup
```

## Create a VM

Create a VM in the proximity placement group using `-ProximityPlacementGroup $ppg.Id` to refer to the proximity placement group ID when you use [New-AzVM](#) to create the VM.

```
$vmName = "myVM"

New-AzVm `
 -ResourceGroupName $resourceGroup `
 -Name $vmName `
 -Location $location `
 -OpenPorts 3389 `
 -ProximityPlacementGroup $ppg.Id
```

You can see the VM in the placement group using [Get-AzProximityPlacementGroup](#).

```
Get-AzProximityPlacementGroup -ResourceId $ppg.Id |
 Format-Table -Property VirtualMachines -Wrap
```

## Move an existing VM into a proximity placement group

You can also add an existing VM to a proximity placement group. You need to stop\deallocate the VM first, then update the VM and restart.

```
$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPGResourceGroup -Name myPPG
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
Update-AzVM -VM $vm -ResourceGroupName $vm.ResourceGroupName -ProximityPlacementGroupId $ppg.Id
Restart-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
```

## Move an existing VM out of a proximity placement group

To remove a VM from a proximity placement group, you need to stop\deallocate the VM first, then update the VM and restart.

```
$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPGResourceGroup -Name myPPG
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
$vm.ProximityPlacementGroupId = ""
Update-AzVM -VM $vm -ResourceGroupName $vm.ResourceGroupName
Restart-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
```

# Availability Sets

You can also create an availability set in your proximity placement group. Use the same `-ProximityPlacementGroup` parameter with the [New-AzAvailabilitySet](#) cmdlet to create an availability set and all of the VMs created in the availability set will also be created in the same proximity placement group.

To add or remove an existing availability set to a proximity placement group, you first need to stop all of the VMs in the availability set.

## Move an existing availability set into a proximity placement group

```
$resourceGroup = "myResourceGroup"
$avSetName = "myAvailabilitySet"
$avSet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup -Name $avSetName
$vmIDs = $avSet.VirtualMachinesReferences
foreach ($vmId in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
}

$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPG -Name myPPG
Update-AzAvailabilitySet -AvailabilitySet $avSet -ProximityPlacementGroupId $ppg.Id
foreach ($vmId in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
}
```

## Move an existing availability set out of a proximity placement group

```

$resourceGroup = "myResourceGroup"
$avSetName = "myAvailabilitySet"
$avSet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup -Name $avSetName
$vmIDs = $avSet.VirtualMachineReferences
foreach ($vmId in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
}

$avSet.ProximityPlacementGroup = ""
Update-AzAvailabilitySet -AvailabilitySet $avSet
foreach ($vmId in $vmIDs){
 $string = $vmID.Id.Split("/")
 $vmName = $string[8]
 Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
}

```

## Scale sets

You can also create a scale set in your proximity placement group. Use the same `-ProximityPlacementGroup` parameter with [New-AzVmss](#) to create a scale set and all of the instances will be created in the same proximity placement group.

To add or remove an existing scale set to a proximity placement group, you first need to stop the scale set.

### Move an existing scale set into a proximity placement group

```

$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPG -Name myPPG
$vmss = Get-AzVmss -ResourceGroupName myVMSSResourceGroup -VMScaleSetName myScaleSet
Stop-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName $vmss.ResourceGroupName
Update-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName $vmss.ResourceGroupName -ProximityPlacementGroupId
$ppg.Id
Restart-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName $vmss.ResourceGroupName

```

### Move an existing scale set out of a proximity placement group

```

$vmss = Get-AzVmss -ResourceGroupName myVMSSResourceGroup -VMScaleSetName myScaleSet
Stop-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName $vmss.ResourceGroupName
$vmss.ProximityPlacementGroup = ""
Update-AzVmss -VirtualMachineScaleSet $vmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName
Restart-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName $vmss.ResourceGroupName

```

## Next steps

You can also use the [Azure CLI](#) to create proximity placement groups.

# Create a Windows virtual machine in an availability zone with PowerShell

12/26/2019 • 4 minutes to read • [Edit Online](#)

This article details using Azure PowerShell to create an Azure virtual machine running Windows Server 2016 in an Azure availability zone. An [availability zone](#) is a physically separate zone in an Azure region. Use availability zones to protect your apps and data from an unlikely failure or loss of an entire datacenter.

To use an availability zone, create your virtual machine in a [supported Azure region](#).

## Sign in to Azure

Sign in to your Azure subscription with the `Connect-AzAccount` command and follow the on-screen directions.

```
Connect-AzAccount
```

## Check VM SKU availability

The availability of VM sizes, or SKUs, may vary by region and zone. To help you plan for the use of Availability Zones, you can list the available VM SKUs by Azure region and zone. This ability makes sure that you choose an appropriate VM size, and obtain the desired resiliency across zones. For more information on the different VM types and sizes, see [VM Sizes overview](#).

You can view the available VM SKUs with the `Get-AzComputeResourceSku` command. The following example lists available VM SKUs in the *eastus2* region:

```
Get-AzComputeResourceSku | where {$_.Locations.Contains("eastus2")};
```

The output is similar to the following condensed example, which shows the Availability Zones in which each VM size is available:

| ResourceType    | Name            | Location | Zones     | [...] |
|-----------------|-----------------|----------|-----------|-------|
| virtualMachines | Standard_DS1_v2 | eastus2  | {1, 2, 3} |       |
| virtualMachines | Standard_DS2_v2 | eastus2  | {1, 2, 3} |       |
| [...]           |                 |          |           |       |
| virtualMachines | Standard_F1s    | eastus2  | {1, 2, 3} |       |
| virtualMachines | Standard_F2s    | eastus2  | {1, 2, 3} |       |
| [...]           |                 |          |           |       |
| virtualMachines | Standard_D2s_v3 | eastus2  | {1, 2, 3} |       |
| virtualMachines | Standard_D4s_v3 | eastus2  | {1, 2, 3} |       |
| [...]           |                 |          |           |       |
| virtualMachines | Standard_E2_v3  | eastus2  | {1, 2, 3} |       |
| virtualMachines | Standard_E4_v3  | eastus2  | {1, 2, 3} |       |

## Create resource group

Create an Azure resource group with [New-AzResourceGroup](#). A resource group is a logical container into which Azure resources are deployed and managed. In this example, a resource group named *myResourceGroup* is created in the *eastus2* region.

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS2
```

## Create networking resources

### Create a virtual network, subnet, and a public IP address

These resources are used to provide network connectivity to the virtual machine and connect it to the internet.

Create the IP address in an availability zone, 2 in this example. In a later step, you create the VM in the same zone used to create the IP address.

```
Create a subnet configuration
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24

Create a virtual network
$vnet = New-AzVirtualNetwork -ResourceGroupName myResourceGroup -Location eastus2 `
 -Name myVNet -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

Create a public IP address in an availability zone and specify a DNS name
$pip = New-AzPublicIpAddress -ResourceGroupName myResourceGroup -Location eastus2 -Zone 2 `
 -AllocationMethod Static -IdleTimeoutInMinutes 4 -Name "mypublicdns$(Get-Random)" -Sku Standard
```

### Create a network security group and a network security group rule

The network security group secures the virtual machine using inbound and outbound rules. In this case, an inbound rule is created for port 3389, which allows incoming remote desktop connections. We also want to create an inbound rule for port 80, which allows incoming web traffic.

```
Create an inbound network security group rule for port 3389
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp `
 -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `
 -DestinationPortRange 3389 -Access Allow

Create an inbound network security group rule for port 80
$nsgRuleWeb = New-AzNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleWWW -Protocol Tcp `
 -Direction Inbound -Priority 1001 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix * `
 -DestinationPortRange 80 -Access Allow

Create a network security group
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName myResourceGroup -Location eastus2 `
 -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP,$nsgRuleWeb
```

### Create a network card for the virtual machine

Create a network card with [New-AzNetworkInterface](#) for the virtual machine. The network card connects the virtual machine to a subnet, network security group, and public IP address.

```
Create a virtual network card and associate with public IP address and NSG
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName myResourceGroup -Location eastus2 `
 -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id
```

## Create virtual machine

Create a virtual machine configuration. This configuration includes the settings that are used when deploying the virtual machine such as a virtual machine image, size, and authentication configuration. The *Standard\_DS1\_v2* size in this example is supported in availability zones. This configuration also specifies the availability zone you set when creating the IP address. When running this step, you are prompted for credentials. The values that you enter are configured as the user name and password for the virtual machine.

```
Define a credential object
$cred = Get-Credential

Create a virtual machine configuration
$vmbConfig = New-AzVMConfig -VMName myVM -VMSize Standard_DS1_v2 -Zone 2 | `
 Set-AzVMOperatingSystem -Windows -ComputerName myVM -Credential $cred | `
 Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer `
 -Skus 2016-Datacenter -Version latest | Add-AzVMNetworkInterface -Id $nic.Id
```

Create the virtual machine with [New-AzVM](#).

```
New-AzVM -ResourceGroupName myResourceGroup -Location eastus2 -VM $vmConfig
```

## Confirm zone for managed disk

You created the VM's IP address resource in the same availability zone as the VM. The managed disk resource for the VM is created in the same availability zone. You can verify this with [Get-AzDisk](#):

```
Get-AzDisk -ResourceGroupName myResourceGroup
```

The output shows that the managed disk is in the same availability zone as the VM:

```
ResourceGroupName : myResourceGroup
AccountType : PremiumLRS
OwnerId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroup/providers/Microsoft.
 Compute/virtualMachines/myVM
ManagedBy : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroup/providers/Microsoft.
 Compute/virtualMachines/myVM
Sku : Microsoft.Azure.Management.Compute.Models.DiskSku
Zones : {2}
TimeCreated : 9/7/2017 6:57:26 PM
OsType : Windows
CreationData : Microsoft.Azure.Management.Compute.Models.CreationData
DiskSizeGB : 127
EncryptionSettings :
ProvisioningState : Succeeded
Id : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroup/providers/Microsoft.
 Compute/disks/myVM_OsDisk_1_bd921920bb0a4650becfc2d830000000
Name : myVM_OsDisk_1_bd921920bb0a4650becfc2d830000000
Type : Microsoft.Compute/disks
Location : eastus2
Tags : {}
```

## Next steps

In this article, you learned how to create a VM in an availability zone. Learn more about [availability](#) for Azure VMs.

# Create a Windows virtual machine in an availability zone with the Azure portal

11/13/2019 • 2 minutes to read • [Edit Online](#)

This article steps through using the Azure portal to create a virtual machine in an Azure availability zone. An [availability zone](#) is a physically separate zone in an Azure region. Use availability zones to protect your apps and data from an unlikely failure or loss of an entire datacenter.

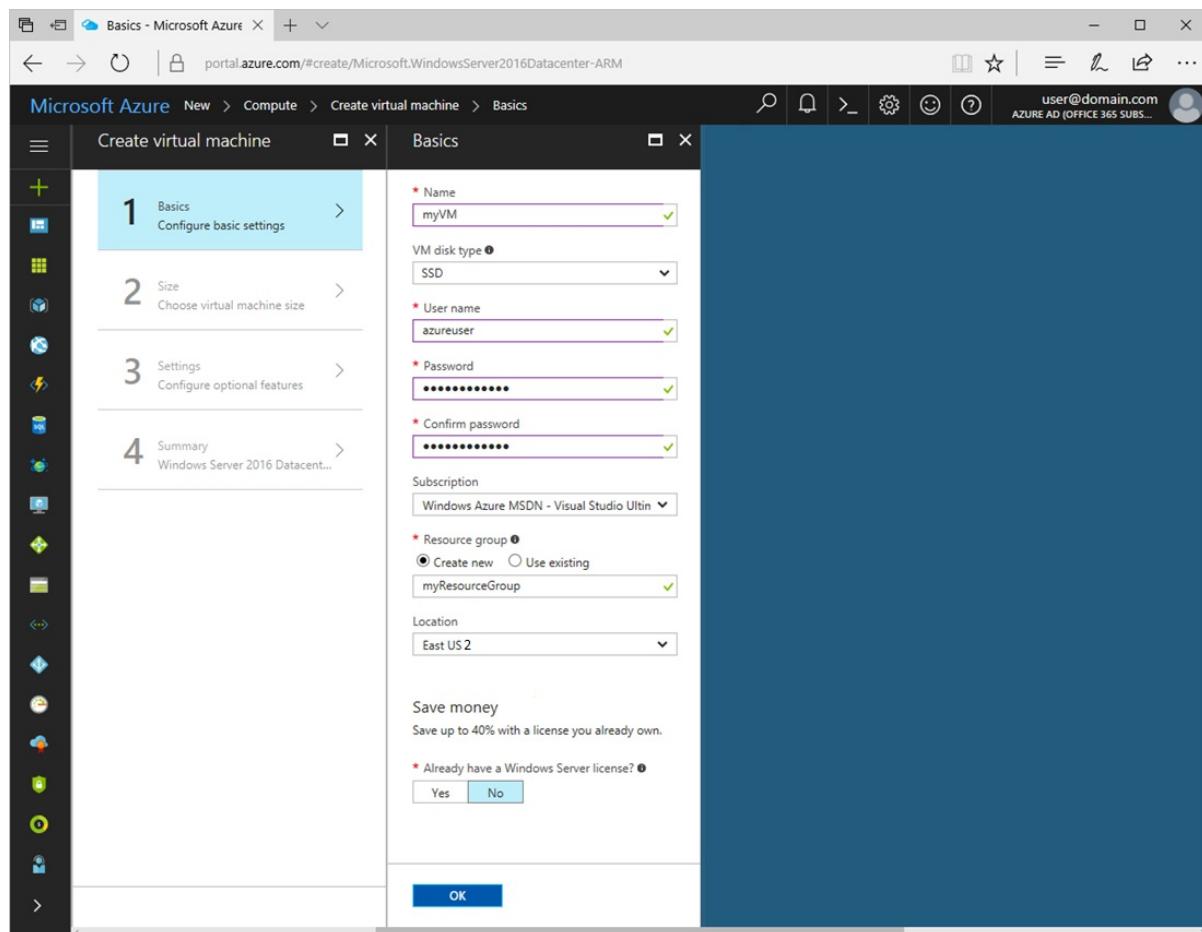
To use an availability zone, create your virtual machine in a [supported Azure region](#).

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create virtual machine

1. Click **Create a resource** in the upper left-hand corner of the Azure portal.
2. Select **Compute**, and then select **Windows Server 2016 Datacenter**.
3. Enter the virtual machine information. The user name and password entered here is used to sign in to the virtual machine. The password must be at least 12 characters long and meet the [defined complexity requirements](#). Choose a Location such as East US 2 that supports availability zones. When complete, click **OK**.

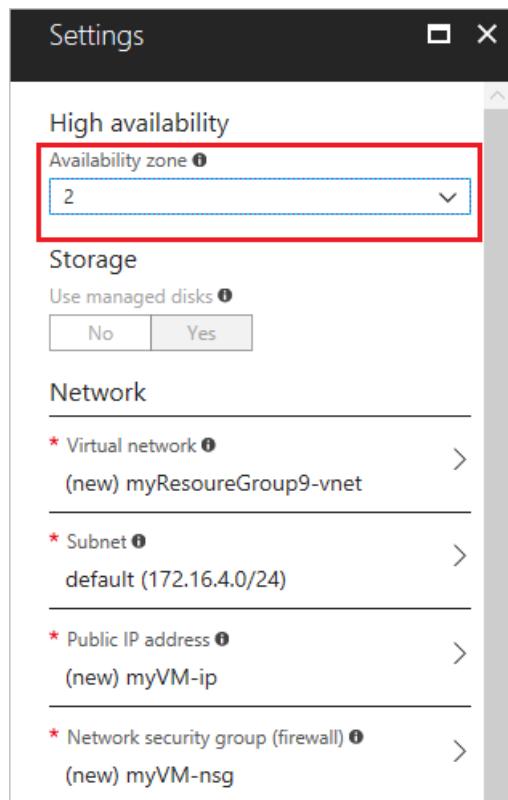


4. Choose a size for the VM. Select a recommended size, or filter based on features. Confirm the size is

available in the zone you want to use.

| Choose a size  |        |          |       |              |                    |                     |                    |               |                 |          |            |
|----------------|--------|----------|-------|--------------|--------------------|---------------------|--------------------|---------------|-----------------|----------|------------|
| Search         |        |          |       | Compute type |                    | Supported disk type |                    | Minimum vCPUs |                 |          |            |
| RECOMMENDATION | SKU    | TYPE     | VCPUS | GB RAM       | DATA DISK CAPACITY | MAX IOPS            | LOCAL SSD CAPACITY | RDMA SUPPORT  | PREMIUM SUPPORT | GRAPHICS | ZONES      |
| DS             | DS1_v2 | Standard | 1     | 3.5          | 4                  | 3200                | 7 GB               | ✓             |                 | 1,2,3    | \$91.51    |
|                | DS2_v2 | Standard | 2     | 7            | 8                  | 6400                | 14 GB              | ✓             |                 | 1,2,3    | \$183.02   |
|                | DS3_v2 | Standard | 4     | 14           | 16                 | 12800               | 28 GB              | ✓             |                 | 1,2,3    | \$365.30   |
|                | DS4_v2 | Standard | 8     | 28           | 32                 | 25600               | 56 GB              | ✓             |                 | 1,2,3    | \$731.35   |
|                | DS5_v2 | Standard | 16    | 56           | 64                 | 51200               | 112 GB             | ✓             |                 | 1,2,3    | \$1,392.77 |
|                | DS2_v2 | Promo    | 2     | 7            | 8                  | 8000                | 14 GB              | ✓             |                 | 1,2,3    | \$156.98   |
|                | DS3_v2 | Promo    | 4     | 14           | 16                 | 16000               | 28 GB              | ✓             |                 | 1,2,3    | \$313.97   |
|                | DS4_v2 | Promo    | 8     | 28           | 32                 | 32000               | 56 GB              | ✓             |                 | 1,2,3    | \$628.68   |
|                | DS5_v2 | Promo    | 16    | 56           | 64                 | 64000               | 112 GB             | ✓             |                 | 1,2,3    | \$1,257.36 |
|                | DS1    | Standard | 1     | 3.5          | 4                  | 3200                | 7 GB               | ✓             |                 | 2,3      | \$96.72    |
|                | DS2    | Standard | 2     | 7            | 8                  | 6400                | 14 GB              | ✓             |                 | 2,3      | \$193.44   |
|                | DS3    | Standard | 4     | 14           | 16                 | 12800               | 28 GB              | ✓             |                 | 2,3      | \$386.88   |
|                | DS4    | Standard | 8     | 28           | 32                 | 25600               | 56 GB              | ✓             |                 | 2,3      | \$773.76   |

5. Under **Settings > High availability**, select one of the numbered zones from the **Availability zone** dropdown, keep the remaining defaults, and click **OK**.



6. On the summary page, click **Create** to start the virtual machine deployment.  
7. The VM will be pinned to the Azure portal dashboard. Once the deployment has completed, the VM summary automatically opens.

## Confirm zone for managed disk and IP address

When the VM is deployed in an availability zone, a managed disk for the VM is created in the same availability zone. By default, a public IP address is also created in that zone.

You can confirm the zone settings for these resources in the portal.

1. Click **Resource groups** and then the name of the resource group for the VM, such as *myResourceGroup*.
2. Click the name of the Disk resource. The **Overview** page includes details about the location and availability zone of the resource.

The screenshot shows the Azure portal interface for managing a disk resource. The left sidebar lists navigation options: Home, myResourceGroup9, myVM\_OsDisk\_1\_9b339ea95c8a485183a6afc1324c8e7e, Disk, Overview (which is selected and highlighted with a red box), Activity log, Access control (IAM), Tags, Settings (Locks, Automation script), Support + Troubleshooting (New support request). The main content area displays the disk's properties:

- NAME:** myVM\_OsDisk\_1\_9b339ea95c8a485183a6afc1324c8e7e
- DISK STATE:** Attached
- \* Account type:** Premium (SSD)
- \* Size (GiB):** 128
- ESTIMATED PERFORMANCE:**
  - IOPS limit: 500
  - Throughput limit (MB/s): 100
- OWNER VM:** myVM
- OPERATING SYSTEM:** Windows
- SOURCE IMAGE:** MicrosoftWindowsServer / WindowsServer / 2016-Datacenter / 2016.127.20171217
- TIME CREATED:** 3/13/2018 2:26:24 PM
- RESOURCE GROUP:** MYRESOUREGROUP9
- LOCATION:** East US 2
- AVAILABILITY ZONE:** 2 (highlighted with a red box)

3. Click the name of the Public IP address resource. The **Overview** page includes details about the location and availability zone of the resource.

The screenshot shows the Azure portal interface for managing a Public IP address resource. The left sidebar lists navigation options: Home, myResourceGroup9, myVM-ip, Public IP address, Overview (selected and highlighted with a red box), Activity log, Access control (IAM), Tags, Settings. The main content area displays the resource's properties:

- Resource group (change):** myResourceGroup9
- Location:** East US 2 (Zone 2) (highlighted with a red box)
- Subscription name (change):** Internal
- Subscription ID:** XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX
- SKU:** Basic
- IP address:** 40.70.67.133
- DNS name:** -
- Associated to:** myvm712
- Virtual machine:** myVM

## Next steps

In this article, you learned how to create a VM in an availability zone. Learn more about [availability](#) for Azure VMs.

2 minutes to read

# Publish an ASP.NET Web App to an Azure VM from Visual Studio

10/4/2019 • 2 minutes to read • [Edit Online](#)

This document describes how to publish an ASP.NET web application to an Azure virtual machine (VM) using the **Microsoft Azure Virtual Machines** publishing feature in Visual Studio 2019.

## Prerequisites

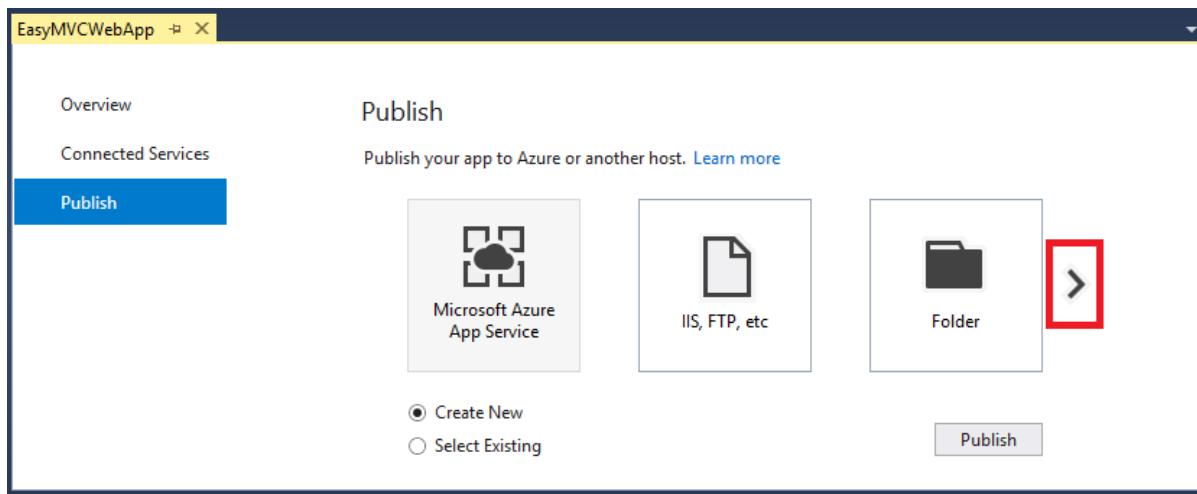
In order to use Visual Studio to publish an ASP.NET project to an Azure VM, the VM must be correctly set up.

- Machine must be configured to run an ASP.NET web application and have WebDeploy installed.
- The VM must have a DNS name configured. For more information, see [Create a fully qualified domain name in the Azure portal for a Windows VM](#).

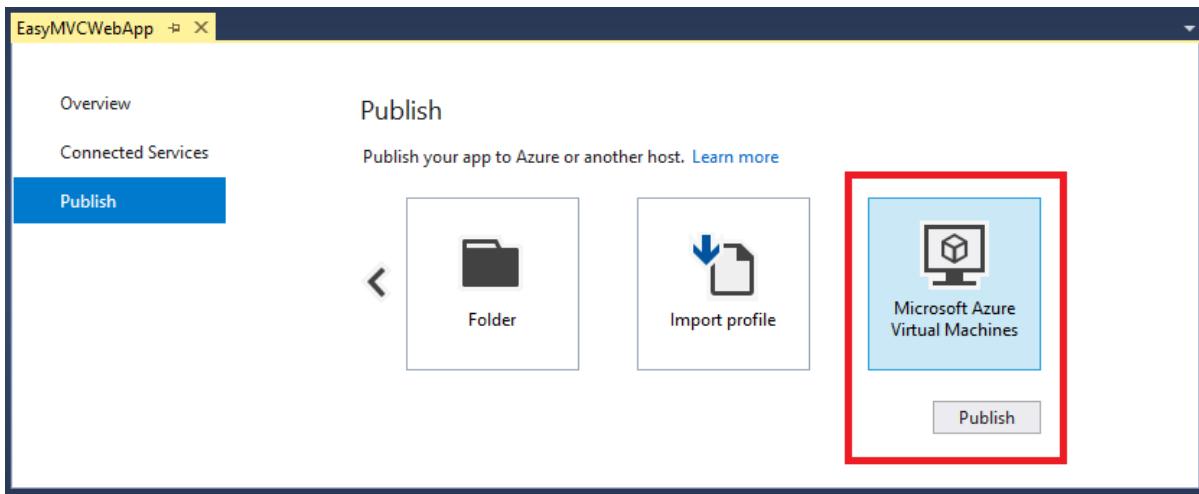
## Publish your ASP.NET web app to the Azure VM using Visual Studio

The following section describes how to publish an existing ASP.NET web application to an Azure virtual machine.

1. Open your web app solution in Visual Studio 2019.
2. Right-click the project in Solution Explorer and choose **Publish...**
3. Use the arrow on the right of the page to scroll through the publishing options until you find **Microsoft Azure Virtual Machines**.

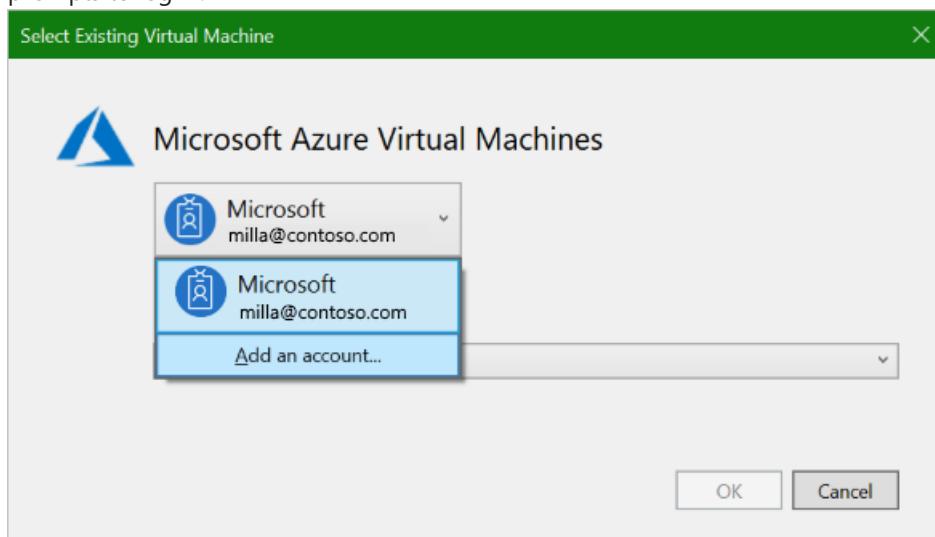


4. Select the **Microsoft Azure Virtual Machines** icon and select **Publish**.



5. Choose the appropriate account (with Azure subscription connected to your virtual machine).

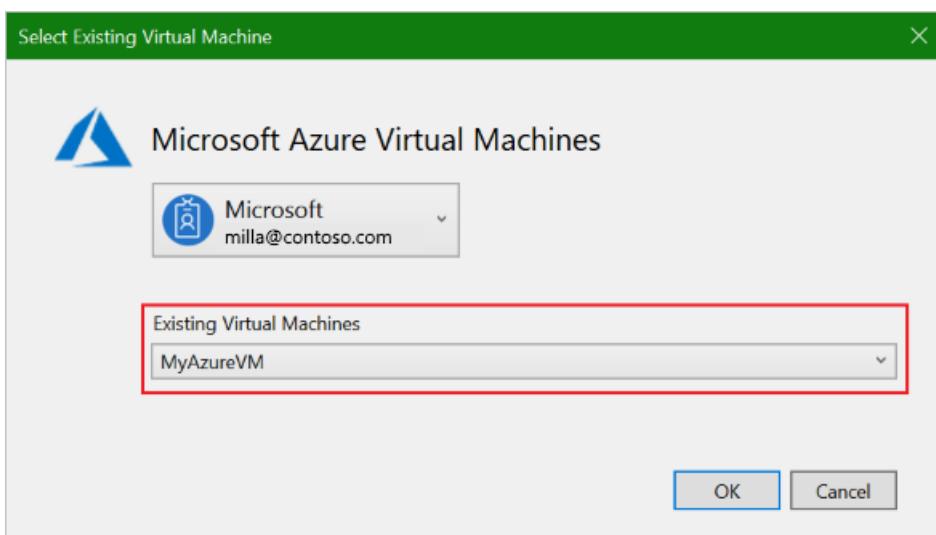
- If you're signed in to Visual Studio, the account list is populated with all your authenticated accounts.
- If you are not signed in, or if the account you need is not listed, choose "Add an account..." and follow the prompts to log in.



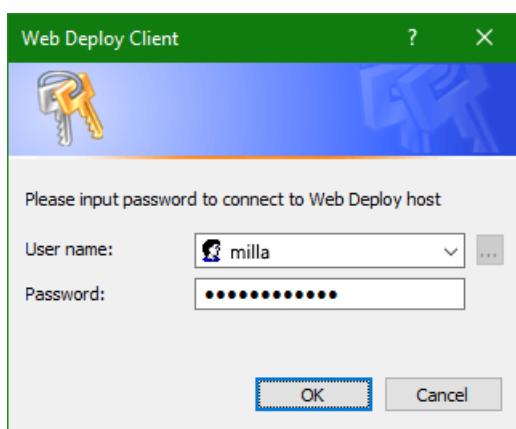
6. Select the appropriate VM from the list of Existing Virtual Machines.

**NOTE**

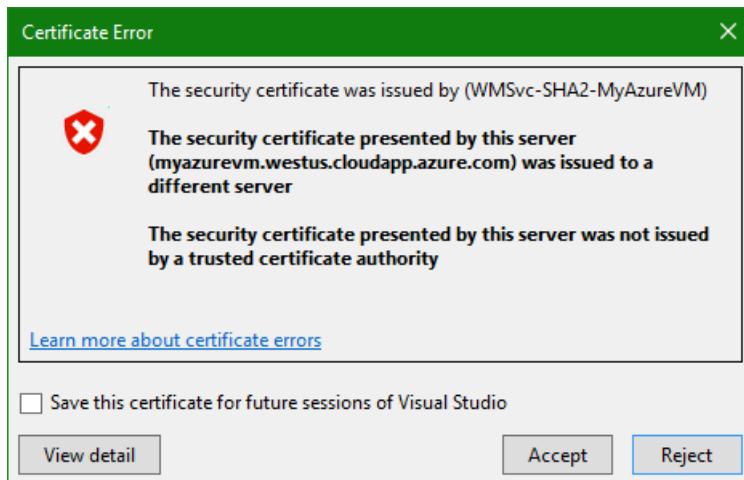
Populating this list can take some time.



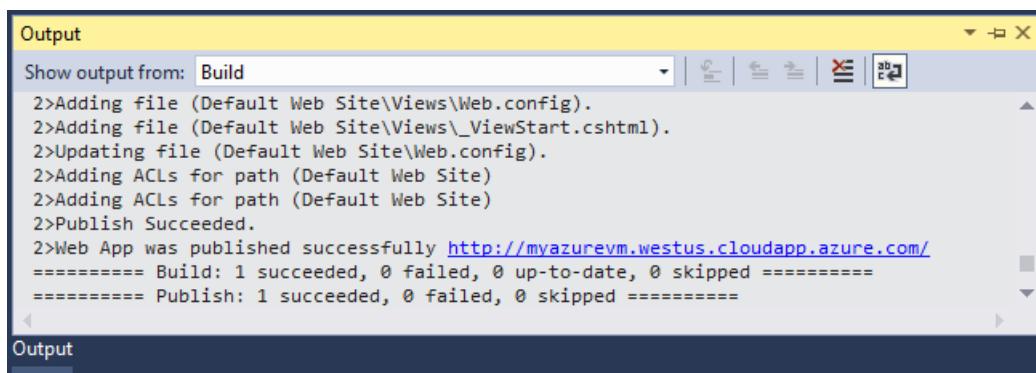
7. Click OK to begin publishing.
8. When prompted for credentials, supply the username and password of a user account on the target VM that is configured with publishing rights. These credentials are typically the admin username and password used when creating the VM.



9. Accept the security certificate.



10. Watch the Output window to check the progress of the publish operation.



11. If publishing is successful, a browser launches to open the URL of the newly published site.

**Success!**

You have now successfully published your web app to an Azure virtual machine.

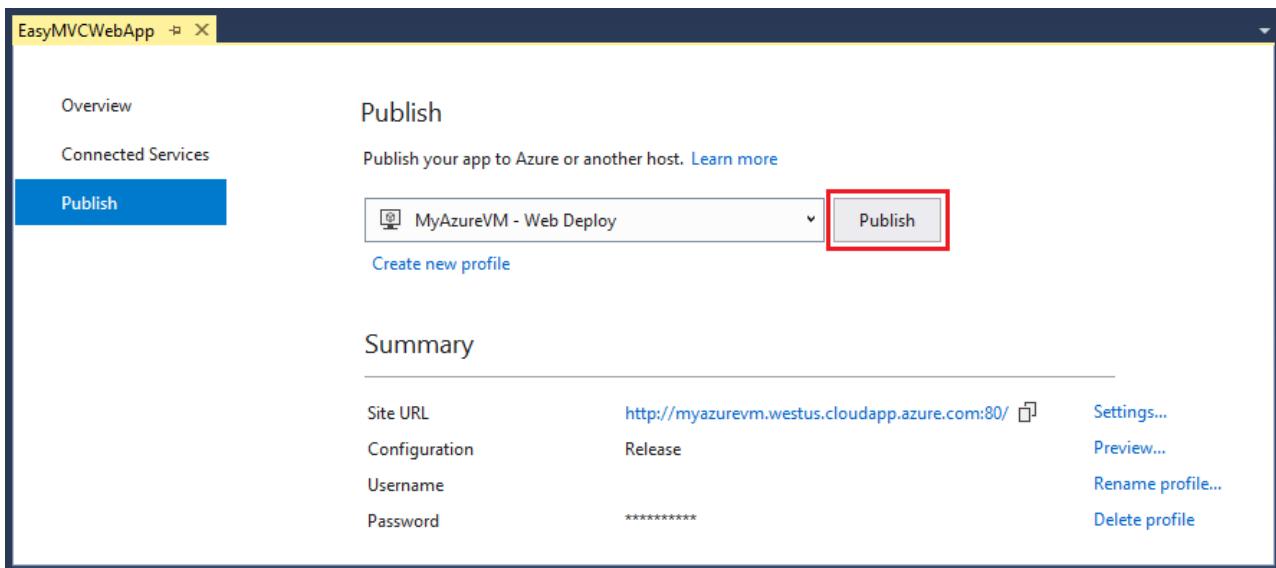
## Publish Page Options

After completing the publish wizard, the Publish page is opened in the document well with the new publishing profile selected.

## Re-publish

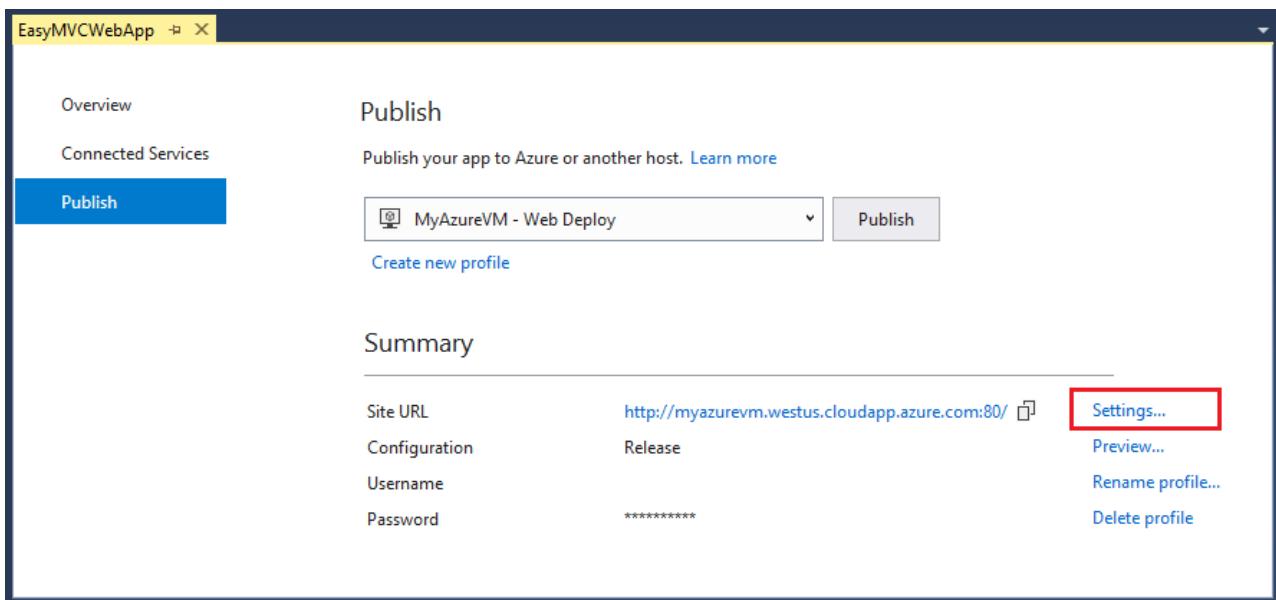
To publish updates to your web application, select the **Publish** button on the Publish page.

- If prompted, enter username and password.
- Publishing begins immediately.

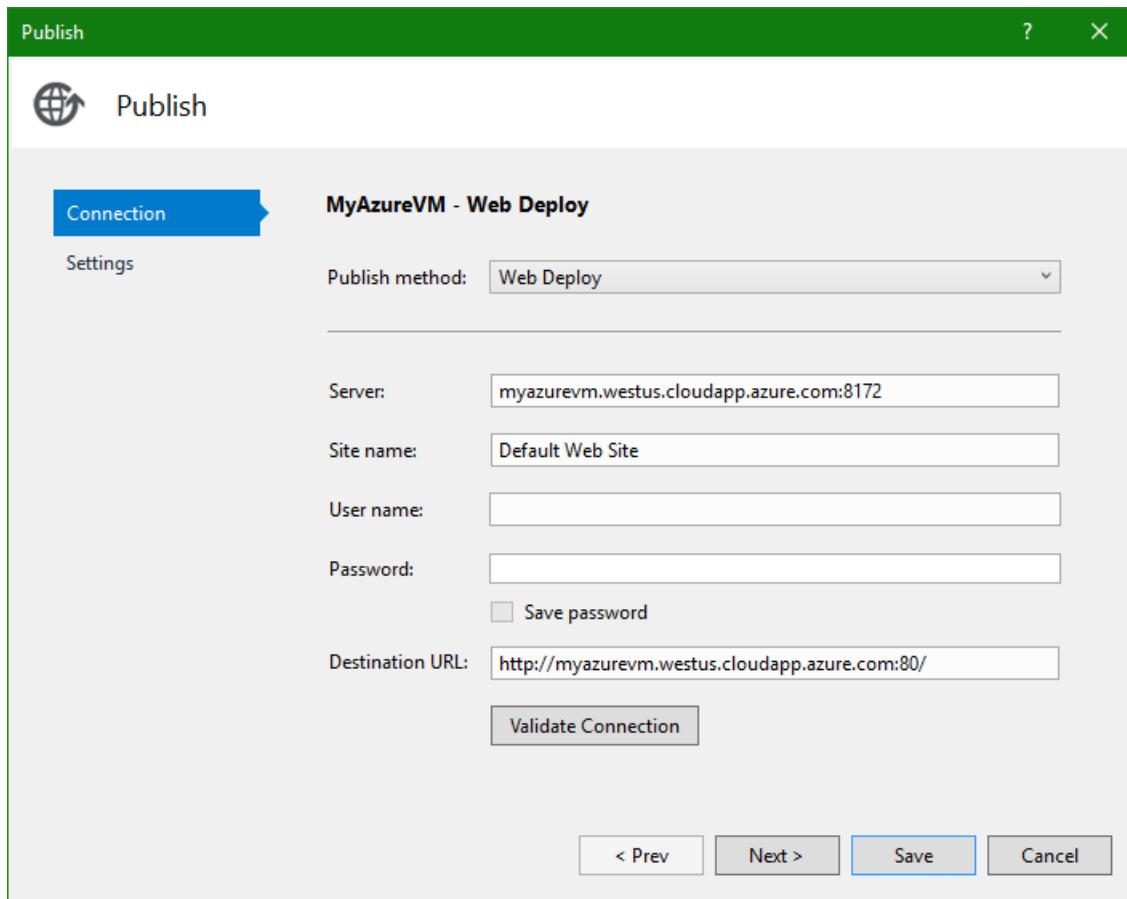


## Modify publish profile settings

To view and modify the publish profile settings, select **Settings....**



Your settings should look something like this:

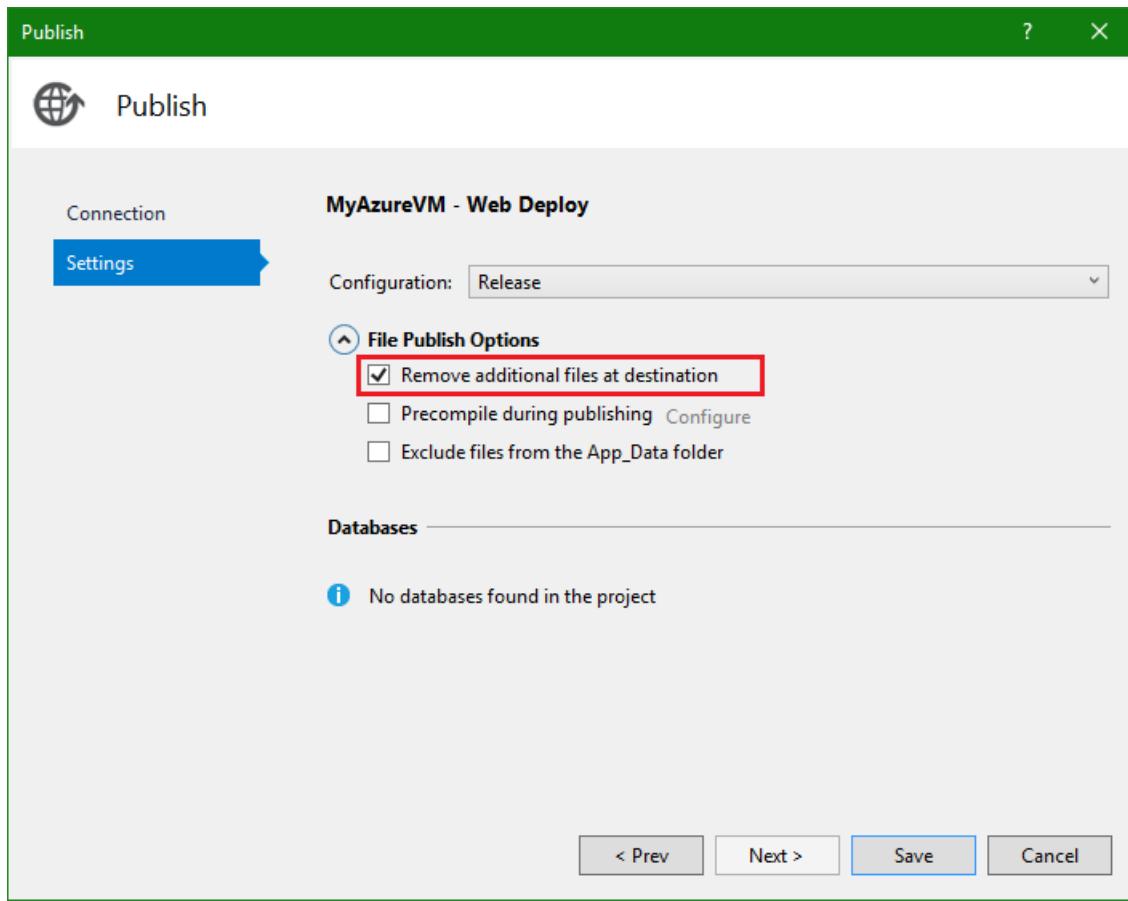


#### Save User name and Password

- Avoid providing authentication information every time you publish. To do so, populate the **User name** and **Password** fields, and select the **Save password** box.
- Use the **Validate Connection** button to confirm that you have entered the right information.

#### Deploy to clean web server

- If you want to ensure that the web server has a clean copy of the web application after each upload and that no other files are left from a previous deployment, you can check the **Remove additional files at destination** checkbox in the **Settings** tab.
- Warning: Publishing with this setting deletes all files that exist on the web server (wwwroot directory). Be sure you know the state of the machine before publishing with this option enabled.



## Next steps

### Set up CI/CD for automated deployment to Azure VM

To set up a continuous delivery pipeline with Azure Pipelines, see [Deploy to a Windows Virtual Machine](#).

# What is SQL Server on Azure Virtual Machines? (Windows)

12/13/2019 • 5 minutes to read • [Edit Online](#)

SQL Server on Azure virtual machines enables you to use full versions of SQL Server in the Cloud without having to manage any on-premises hardware. SQL Server VMs also simplify licensing costs when you pay as you go.

Azure virtual machines run in many different [geographic regions](#) around the world. They also offer a variety of [machine sizes](#). The virtual machine image gallery allows you to create a SQL Server VM with the right version, edition, and operating system. This makes virtual machines a good option for a many different SQL Server workloads.

## Automated updates

SQL Server Azure VMs can use [Automated Patching](#) to schedule a maintenance window for installing important windows and SQL Server updates automatically.

## Automated backups

SQL Server Azure VMs can take advantage of [Automated Backup](#), which regularly creates backups of your database to blob storage. You can also manually use this technique. For more information, see [Use Azure Storage for SQL Server Backup and Restore](#).

## High availability

If you require high availability, consider configuring SQL Server Availability Groups. This involves multiple SQL Server Azure VMs in a virtual network. You can configure your high availability solution manually, or you can use templates in the Azure portal for automatic configuration. For an overview of all high availability options, see [High Availability and Disaster Recovery for SQL Server in Azure Virtual Machines](#).

## Performance

Azure virtual machines offer different machine sizes to meet various workload demands. SQL VMs also provide automated storage configuration, which is optimized for your performance requirements. For more information about configuring storage for SQL VMs, see [Storage configuration for SQL Server VMs](#). To fine-tune performance, see the [Performance best practices for SQL Server in Azure Virtual Machines](#).

## Get started with SQL VMs

To get started, choose a SQL Server virtual machine image with your required version, edition, and operating system. The following sections provide direct links to the Azure portal for the SQL Server virtual machine gallery images.

### TIP

For more information about how to understand pricing for SQL images, see [Pricing guidance for SQL Server Azure VMs](#).

## Pay as you go

The following table provides a matrix of pay-as-you-go SQL Server images.

| VERSION                       | OPERATING SYSTEM       | EDITION                                       |
|-------------------------------|------------------------|-----------------------------------------------|
| <b>SQL Server 2019</b>        | Windows Server 2019    | Enterprise, Standard, Web, Developer          |
| <b>SQL Server 2017</b>        | Windows Server 2016    | Enterprise, Standard, Web, Express, Developer |
| <b>SQL Server 2016 SP2</b>    | Windows Server 2016    | Enterprise, Standard, Web, Express, Developer |
| <b>SQL Server 2014 SP2</b>    | Windows Server 2012 R2 | Enterprise, Standard, Web, Express            |
| <b>SQL Server 2012 SP4</b>    | Windows Server 2012 R2 | Enterprise, Standard, Web, Express            |
| <b>SQL Server 2008 R2 SP3</b> | Windows Server 2008 R2 | Enterprise, Standard, Web, Express            |

To see the available Linux SQL Server virtual machine images, see [Overview of SQL Server on Azure Virtual Machines \(Linux\)](#).

#### NOTE

It is now possible to change the licensing model of a pay-per-usage SQL Server VM to use your own license. For more information, see [How to change the licensing model for a SQL VM](#).

#### Bring your own license

You can also bring your own license (BYOL). In this scenario, you only pay for the VM without any additional charges for SQL Server licensing. Bringing your own license can save you money over time for continuous production workloads. For requirements to use this option, see [Pricing guidance for SQL Server Azure VMs](#).

To bring your own license, you can either convert an existing pay-per-usage SQL VM, or you can deploy an image with the prefixed **{BYOL}**. For more information about switching your licensing model between pay-per-usage and BYOL, see [How to change the licensing model for a SQL VM](#).

| VERSION                    | OPERATING SYSTEM       | EDITION                        |
|----------------------------|------------------------|--------------------------------|
| <b>SQL Server 2019</b>     | Windows Server 2019    | Enterprise BYOL, Standard BYOL |
| <b>SQL Server 2017</b>     | Windows Server 2016    | Enterprise BYOL, Standard BYOL |
| <b>SQL Server 2016 SP2</b> | Windows Server 2016    | Enterprise BYOL, Standard BYOL |
| <b>SQL Server 2014 SP2</b> | Windows Server 2012 R2 | Enterprise BYOL, Standard BYOL |
| <b>SQL Server 2012 SP4</b> | Windows Server 2012 R2 | Enterprise BYOL, Standard BYOL |

It is possible to deploy an older image of SQL Server that is not available in the Azure portal using PowerShell. To view all available images using Powershell, use the following command:

```
Get-AzVMImageOffer -Location $Location -Publisher 'MicrosoftSQLServer'
```

For more information about deploying SQL Server VMs using PowerShell, view [How to provision SQL Server virtual machines with Azure PowerShell](#).

## Connect to the VM

After creating your SQL Server VM, connect to it from applications or tools, such as SQL Server Management Studio (SSMS). For instructions, see [Connect to a SQL Server Virtual Machine on Azure](#).

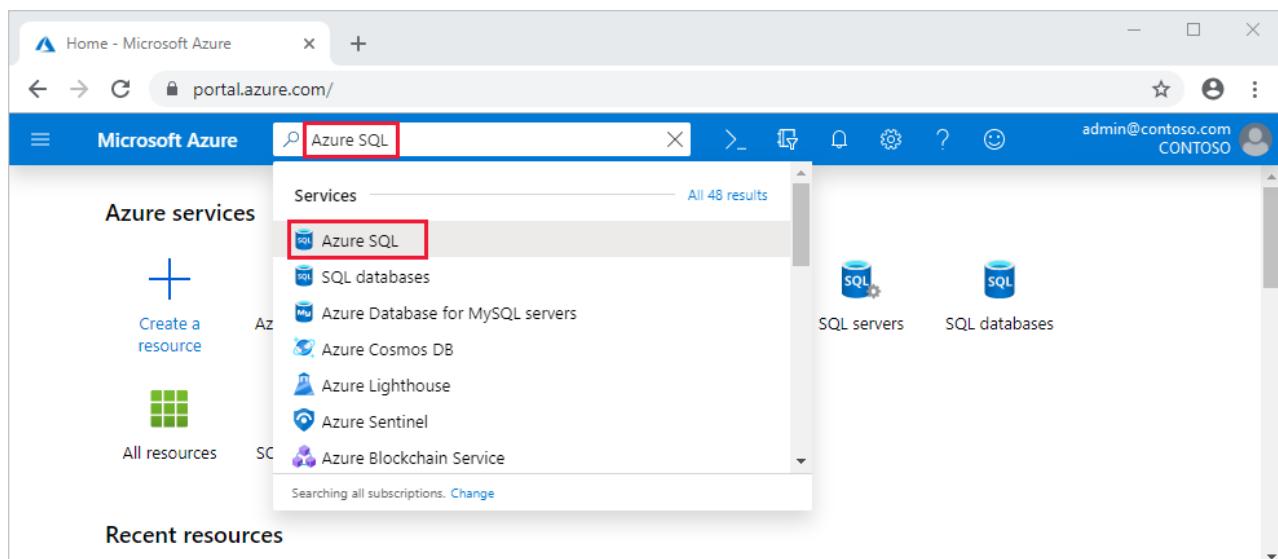
## Migrate your data

If you have an existing database, you'll want to move that to the newly provisioned SQL VM. For a list of migration options and guidance, see [Migrating a Database to SQL Server on an Azure VM](#).

# Create and manage Azure SQL resources with the Azure portal

The Azure portal provides a single page where you can manage [all of your Azure SQL resources](#) including your SQL virtual machines.

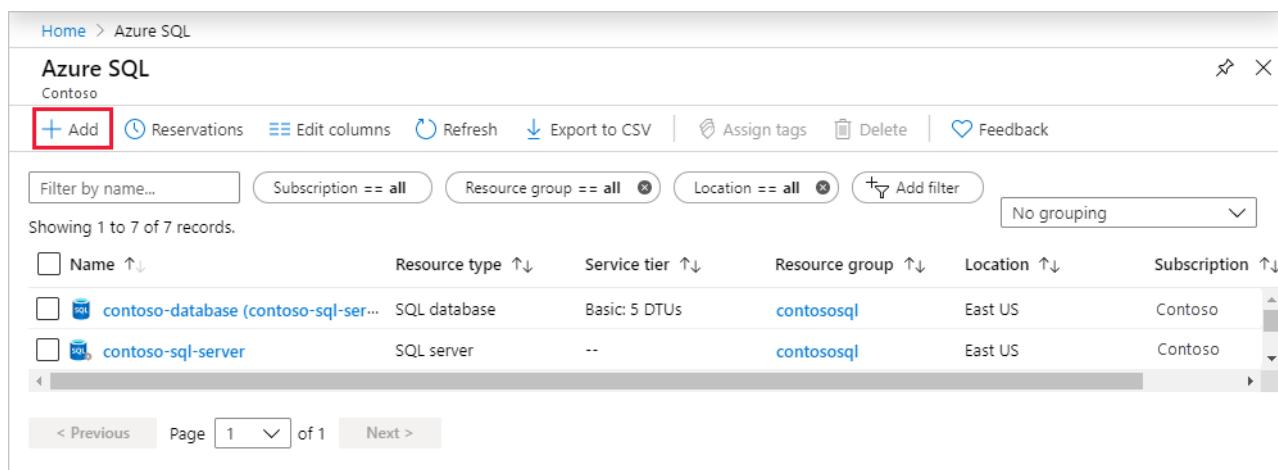
To access the **Azure SQL resources** page, select **Azure SQL** in the Azure portal menu, or search for and select **Azure SQL** from any page.



### NOTE

**Azure SQL** provides a quick and easy way to access all of your SQL databases, elastic pools, database servers, SQL managed instances, and SQL virtual machines. Azure SQL is not a service or resource.

To manage existing resources, select the desired item in the list. To create new Azure SQL resources, select **+ Add**.



After selecting **+ Add**, view additional information about the different options by selecting **Show details** on any tile.

Home > Azure SQL resources > Select SQL deployment option

## Select SQL deployment option

Microsoft

**How do you plan to use the service?**

**Databases**  
Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type  
Single database

Create Show details

**Managed instances**  
Best for most migrations to the cloud. Lift-and-shift ready.

Resource type  
Single instance

Create Show details

**SQL virtual machines**  
Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image  
Free SQL Server License: SQL Server 2017 Developer

Create Show details

**SQL virtual machine**  
SQL virtual machines offer full administrative control over the SQL Server instance and underlying OS for migration to Azure.

**Featured capabilities:**

- ✓ SQL Server and OS access
- ✓ Expansive SQL Server and OS version support
- ✓ Automated manageability features for SQL Server

For details, see:

- [Create a single database](#)
- [Create an elastic pool](#)
- [Create a managed instance](#)
- [Create a SQL virtual machine](#)

## SQL VM image refresh policy

Azure only maintains one virtual machine image for each supported operating system, version, and edition combination. This means that over time images are refreshed, and older images are removed. For more information, see the **Images** section of the [SQL Server VMs FAQ](#).

## Customer experience improvement program (CEIP)

The Customer Experience Improvement Program (CEIP) is enabled by default. This periodically sends reports to Microsoft to help improve SQL Server. There is no management task required with CEIP unless you want to disable it after provisioning. You can customize or disable the CEIP by connecting to the VM with remote desktop. Then run the **SQL Server Error and Usage Reporting** utility. Follow the instructions to disable reporting. For more information about data collection, see the [SQL Server Privacy Statement](#).

## Related products and services

### Windows Virtual Machines

- [Virtual Machines overview](#)

### Storage

- [Introduction to Microsoft Azure Storage](#)

### Networking

- [Virtual Network overview](#)
- [IP addresses in Azure](#)
- [Create a Fully Qualified Domain Name in the Azure portal](#)

### SQL

- [SQL Server documentation](#)

- [Azure SQL Database comparison](#)

## Next steps

Get started with SQL Server on Azure virtual machines:

- [Create a SQL Server VM in the Azure portal](#)

Get answers to commonly asked questions about SQL VMs:

- [SQL Server on Azure Virtual Machines FAQ](#)

View Reference Architectures for running N-tier applications on SQL Server in IaaS

- [Windows N-tier application on Azure with SQL Server](#)
- [Run an N-tier application in multiple Azure regions for high availability](#)

# Install and configure MongoDB on a Windows VM in Azure

11/13/2019 • 5 minutes to read • [Edit Online](#)

MongoDB is a popular open-source, high-performance NoSQL database. This article guides you through installing and configuring MongoDB on a Windows Server 2016 virtual machine (VM) in Azure. You can also [install MongoDB on a Linux VM in Azure](#).

## Prerequisites

Before you install and configure MongoDB, you need to create a VM and, ideally, add a data disk to it. See the following articles to create a VM and add a data disk:

- Create a Windows Server VM using [the Azure portal](#) or [Azure PowerShell](#).
- Attach a data disk to a Windows Server VM using [the Azure portal](#) or [Azure PowerShell](#).

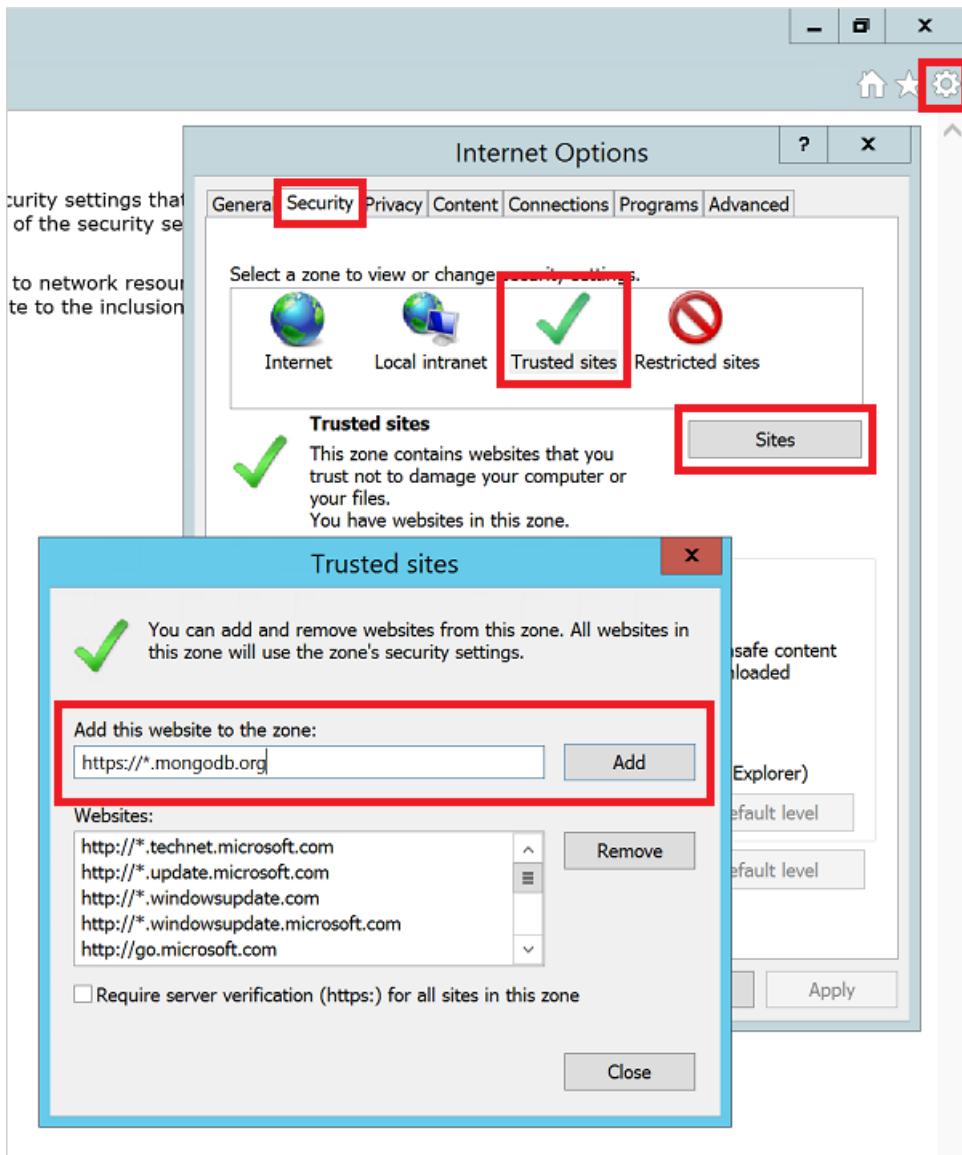
To begin installing and configuring MongoDB, [log on to your Windows Server VM](#) by using Remote Desktop.

## Install MongoDB

### IMPORTANT

MongoDB security features, such as authentication and IP address binding, are not enabled by default. Security features should be enabled before deploying MongoDB to a production environment. For more information, see [MongoDB Security and Authentication](#).

1. After you've connected to your VM using Remote Desktop, open Internet Explorer from the taskbar.
2. Select **Use recommended security, privacy, and compatibility settings** when Internet Explorer first opens, and click **OK**.
3. Internet Explorer enhanced security configuration is enabled by default. Add the MongoDB website to the list of allowed sites:
  - Select the **Tools** icon in the upper-right corner.
  - In **Internet Options**, select the **Security** tab, and then select the **Trusted Sites** icon.
  - Click the **Sites** button. Add *https://\*.mongodb.com* to the list of trusted sites, and then close the dialog box.



4. Browse to the [MongoDB - Downloads](https://www.mongodb.com/downloads) page (<https://www.mongodb.com/downloads>).
5. If needed, select the **Community Server** edition and then select the latest current stable release for *Windows Server 2008 R2 64-bit and later*. To download the installer, click **DOWNLOAD (msi)**.

A screenshot of the MongoDB Downloads page. The 'Community Server' edition is selected and highlighted with a red box. Below it, the 'Current Stable Release (3.2.10)' is shown, also with a red box. A note below the release says '09/30/2016, Release Notes | Changelog' and 'Download Source: tgz | zip'. Under the 'Version:' section, there is a dropdown menu set to 'Windows Server 2008 R2 64-bit and later, with SSL support x64', which is also highlighted with a red box. In the 'Installation Package:' section, a large green button labeled 'DOWNLOAD (msi)' is highlighted with a red box. At the bottom left, there is a link 'Binary: Installation Instructions | All Version Binaries'.

Run the installer after the download is complete.

6. Read and accept the license agreement. When you're prompted, select **Complete** install.

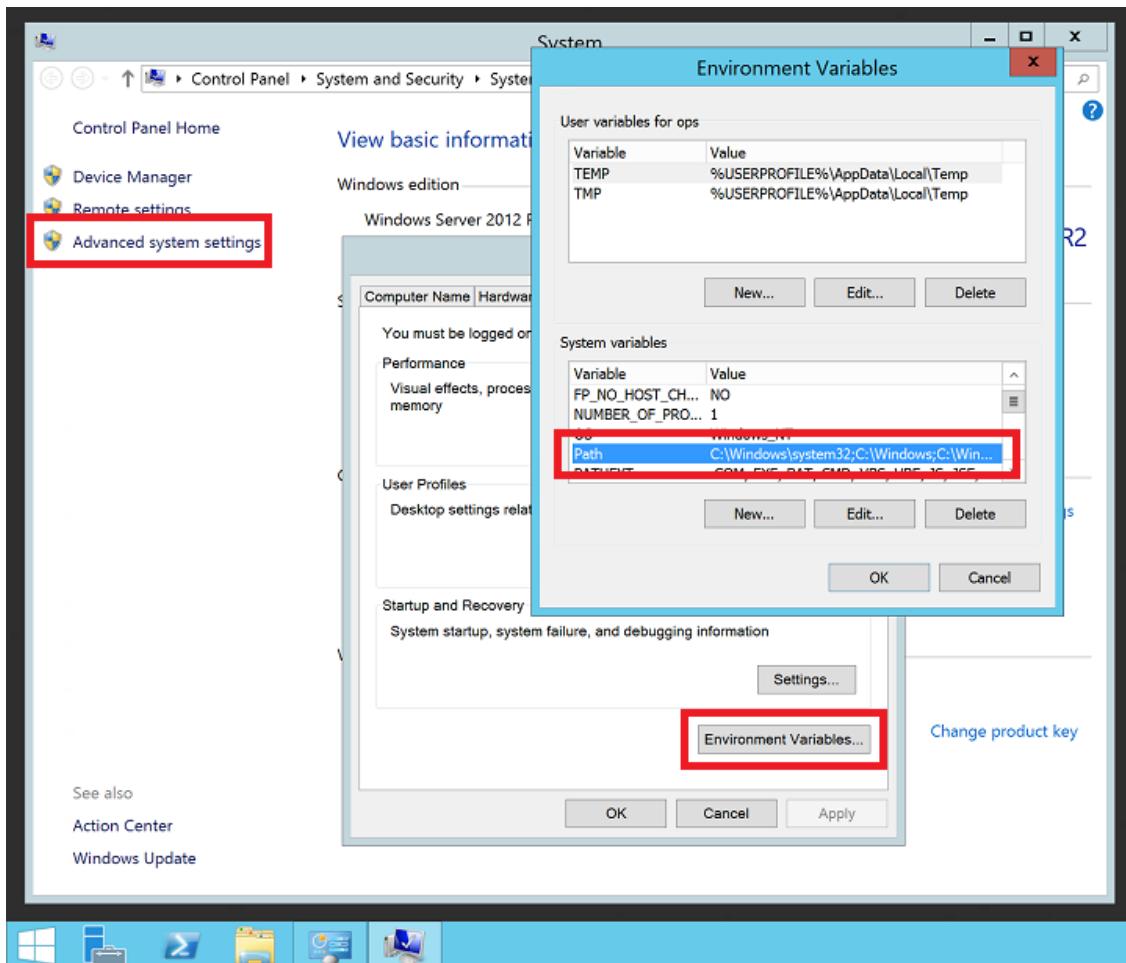
7. If desired, you can choose to also install Compass, a graphical interface for MongoDB.

8. On the final screen, click **Install**.

# Configure the VM and MongoDB

1. The path variables are not updated by the MongoDB installer. Without the MongoDB `bin` location in your path variable, you need to specify the full path each time you use a MongoDB executable. To add the location to your path variable:

- Right-click the **Start** menu, and select **System**.
  - Click **Advanced system settings**, and then click **Environment Variables**.
  - Under **System variables**, select **Path**, and then click **Edit**.



Add the path to your MongoDB `bin` folder. MongoDB is typically installed in `C:\Program Files\MongoDB`. Verify the installation path on your VM. The following example adds the default MongoDB install location to the `PATH` variable:

`;C:\Program Files\MongoDB\Server\3.6\bin`

## NOTE

Be sure to add the leading semicolon ( ; ) to indicate that you are adding a location to your PATH variable.

2. Create MongoDB data and log directories on your data disk. From the **Start** menu, select **Command Prompt**. The following examples create the directories on drive F:

```
mkdir F:\MongoData
mkdir F:\MongoLogs
```

3. Start a MongoDB instance with the following command, adjusting the path to your data and log directories accordingly:

```
mongod --dbpath F:\MongoData\ --logpath F:\MongoLogs\mongolog.log
```

It may take several minutes for MongoDB to allocate the journal files and start listening for connections. All log messages are directed to the *F:\MongoLogs\mongolog.log* file as `mongod.exe` server starts and allocates journal files.

**NOTE**

The command prompt stays focused on this task while your MongoDB instance is running. Leave the command prompt window open to continue running MongoDB. Or, install MongoDB as service, as detailed in the next step.

4. For a more robust MongoDB experience, install the `mongod.exe` as a service. Creating a service means you don't need to leave a command prompt running each time you want to use MongoDB. Create the service as follows, adjusting the path to your data and log directories accordingly:

```
mongod --dbpath F:\MongoData\ --logpath F:\MongoLogs\mongolog.log --logappend --install
```

The preceding command creates a service named MongoDB, with a description of "Mongo DB". The following parameters are also specified:

- The `--dbpath` option specifies the location of the data directory.
- The `--logpath` option must be used to specify a log file, because the running service does not have a command window to display output.
- The `--logappend` option specifies that a restart of the service causes output to append to the existing log file.

To start the MongoDB service, run the following command:

```
net start MongoDB
```

For more information about creating the MongoDB service, see [Configure a Windows Service for MongoDB](#).

## Test the MongoDB instance

With MongoDB running as a single instance or installed as a service, you can now start creating and using your databases. To start the MongoDB administrative shell, open another command prompt window from the **Start** menu, and enter the following command:

```
mongo
```

You can list the databases with the `db` command. Insert some data as follows:

```
db.foo.insert({ a : 1 })
```

Search for data as follows:

```
db.foo.find()
```

The output is similar to the following example:

```
{ "_id" : "ObjectId("57f6a86cee873a6232d74842")", "a" : 1 }
```

Exit the `mongo` console as follows:

```
exit
```

## Configure firewall and Network Security Group rules

Now that MongoDB is installed and running, open a port in Windows Firewall so you can remotely connect to MongoDB. To create a new inbound rule to allow TCP port 27017, open an administrative PowerShell prompt and enter the following command:

```
New-NetFirewallRule `
-DisplayName "Allow MongoDB" `
-Direction Inbound `
-Protocol TCP `
-LocalPort 27017 `
-Action Allow
```

You can also create the rule by using the **Windows Firewall with Advanced Security** graphical management tool. Create a new inbound rule to allow TCP port 27017.

If needed, create a Network Security Group rule to allow access to MongoDB from outside of the existing Azure virtual network subnet. You can create the Network Security Group rules by using the [Azure portal](#) or [Azure PowerShell](#). As with the Windows Firewall rules, allow TCP port 27017 to the virtual network interface of your MongoDB VM.

### NOTE

TCP port 27017 is the default port used by MongoDB. You can change this port by using the `--port` parameter when starting `mongod.exe` manually or from a service. If you change the port, make sure to update the Windows Firewall and Network Security Group rules in the preceding steps.

## Next steps

In this tutorial, you learned how to install and configure MongoDB on your Windows VM. You can now access MongoDB on your Windows VM, by following the advanced topics in the [MongoDB documentation](#).

# Use Azure to host and run SAP workload scenarios

2/27/2020 • 12 minutes to read • [Edit Online](#)

When you use Microsoft Azure, you can reliably run your mission-critical SAP workloads and scenarios on a scalable, compliant, and enterprise-proven platform. You get the scalability, flexibility, and cost savings of Azure. With the expanded partnership between Microsoft and SAP, you can run SAP applications across development and test and production scenarios in Azure and be fully supported. From SAP NetWeaver to SAP S/4HANA, SAP BI on Linux to Windows, and SAP HANA to SQL, we've got you covered.

Besides hosting SAP NetWeaver scenarios with the different DBMS on Azure, you can host other SAP workload scenarios, like SAP BI on Azure.

The uniqueness of Azure for SAP HANA is an offer that sets Azure apart. To enable hosting more memory and CPU resource-demanding SAP scenarios that involve SAP HANA, Azure offers the use of customer-dedicated bare-metal hardware. Use this solution to run SAP HANA deployments that require up to 24 TB (120-TB scale-out) of memory for S/4HANA or other SAP HANA workload.

Hosting SAP workload scenarios in Azure also can create requirements of identity integration and single sign-on. This situation can occur when you use Azure Active Directory (Azure AD) to connect different SAP components and SAP software-as-a-service (SaaS) or platform-as-a-service (PaaS) offers. A list of such integration and single sign-on scenarios with Azure AD and SAP entities is described and documented in the section "AAD SAP identity integration and single sign-on."

## Changes to the SAP workload section

Changes to documents in the SAP on Azure workload section are listed at the end of this article. The entries in the change log are kept for around 180 days.

## You want to know

If you have specific questions, we are going to point you to specific documents or flows in this section of the start page. You want to know:

- What Azure VMs and HANA Large Instance units are supported for which SAP software releases and which operating system versions. Read the document [What SAP software is supported for Azure deployment](#) for answers and the process to find the information
- What SAP deployment scenarios are supported with Azure VMs and HANA Large Instances. Information about the supported scenarios can be found in the documents:
  - [SAP workload on Azure virtual machine supported scenarios](#)
  - [Supported scenarios for HANA Large Instance](#)

## SAP HANA on Azure (Large Instances)

A series of documents leads you through SAP HANA on Azure (Large Instances), or for short, HANA Large Instances. For information on HANA Large Instances start with the document [Overview and architecture of SAP HANA on Azure \(Large Instances\)](#) and go through the related documentation in the HANA Large Instance section

## SAP HANA on Azure virtual machines

This section of the documentation covers different aspects of SAP HANA. As a prerequisite, you should be familiar with the principal services of Azure that provide elementary services of Azure IaaS. So, you need knowledge of

Azure compute, storage, and networking. Many of these subjects are handled in the SAP NetWeaver-related [Azure planning guide](#).

For information on HANA on Azure, see the following articles and their subarticles:

- [Quickstart: Manual installation of single-instance SAP HANA on Azure VMs](#)
- [Deploy SAP S/4HANA or BW/4HANA on Azure](#)
- [SAP HANA infrastructure configurations and operations on Azure](#)
- [SAP HANA high availability for Azure virtual machines](#)
- [SAP HANA availability within one Azure region](#)
- [SAP HANA availability across Azure regions](#)
- [High availability of SAP HANA on Azure virtual machines](#)
- [Backup guide for SAP HANA on Azure virtual machines](#)
- [SAP HANA Azure Backup on file level](#)
- [SAP HANA backup based on storage snapshots](#)

## SAP NetWeaver deployed on Azure virtual machines

This section lists planning and deployment documentation for SAP NetWeaver and Business One on Azure. The documentation focuses on the basics and the use of non-HANA databases with an SAP workload on Azure. The documents and articles for high availability are also the foundation for HANA high availability in Azure, such as:

- [SAP Business One on Azure virtual machines](#)
- [Deploy SAP IDES EHP7 SP3 for SAP ERP 6.0 on Azure](#)
- [Run SAP NetWeaver on Microsoft Azure SUSE Linux VMs](#)
- [Azure Virtual Machines planning and implementation for SAP NetWeaver](#)
- [Azure Virtual Machines deployment for SAP NetWeaver](#)
- [Protect a multitier SAP NetWeaver application deployment by using Site Recovery](#)
- [SAP LaMa connector for Azure](#)

For information on non-HANA databases under an SAP workload on Azure, see:

- [Considerations for Azure Virtual Machines DBMS deployment for SAP workload](#)
- [SQL Server Azure Virtual Machines DBMS deployment for SAP NetWeaver](#)
- [Oracle Azure Virtual Machines DBMS deployment for SAP workload](#)
- [IBM DB2 Azure Virtual Machines DBMS deployment for SAP workload](#)
- [SAP ASE Azure Virtual Machines DBMS deployment for SAP workload](#)
- [SAP MaxDB, Live Cache, and Content Server deployment on Azure VMs](#)

For information on SAP HANA databases on Azure, see the section "SAP HANA on Azure virtual machines."

For information on high availability of an SAP workload on Azure, see:

- [Azure Virtual Machines high availability for SAP NetWeaver](#)

This document points to various other architecture and scenario documents. In later scenario documents, links to detailed technical documents that explain the deployment and configuration of the different high-availability methods are provided. The different documents that show how to establish and configure high availability for an SAP NetWeaver workload cover Linux and Windows operating systems.

For information on integration between Azure Active Directory (Azure AD) and SAP services and single sign-on, see:

- [Tutorial: Azure Active Directory integration with SAP Cloud for Customer](#)

- [Tutorial: Azure Active Directory integration with SAP Cloud Platform Identity Authentication](#)
- [Tutorial: Azure Active Directory integration with SAP Cloud Platform](#)
- [Tutorial: Azure Active Directory integration with SAP NetWeaver](#)
- [Tutorial: Azure Active Directory integration with SAP Business ByDesign](#)
- [Tutorial: Azure Active Directory integration with SAP HANA](#)
- [Your S/4HANA environment: Fiori Launchpad SAML single sign-on with Azure AD](#)

For information on integration of Azure services into SAP components, see:

- [Use SAP HANA in Power BI Desktop](#)
- [DirectQuery and SAP HANA](#)
- [Use the SAP BW Connector in Power BI Desktop](#)
- [Azure Data Factory offers SAP HANA and Business Warehouse data integration](#)

## Change Log

- 02/26/2020: Change in [SAP HANA Azure virtual machine storage configurations](#) to clarify file system choice for HANA on Azure
- 02/25/2020: Change in [High availability architecture and scenarios for SAP](#) to include the link to the HA for SAP NetWeaver on Azure VMs on RHEL multi-SID guide
- 02/26/2020: Change in [High availability for SAP NW on Azure VMs on SLES for SAP applications](#), [High availability for SAP NW on Azure VMs on SLES with ANF for SAP applications](#), [Azure VMs high availability for SAP NetWeaver on RHEL](#) and [Azure VMs high availability for SAP NetWeaver on RHEL with Azure NetApp Files](#) to remove the statement that multi-SID ASCS/ERS cluster is not supported
- 02/26/2020: Release of [High availability for SAP NetWeaver on Azure VMs on RHEL multi-SID guide](#) to add a link to the SUSE multi-SID cluster guide
- 02/25/2020: Change in [High availability architecture and scenarios for SAP](#) to add links to newer HA articles
- 02/25/2020: Change in [High availability of IBM Db2 LUW on Azure VMs on SUSE Linux Enterprise Server with Pacemaker](#) to point to document that describes access to public endpoint with Standard Azure Load balancer
- 02/21/2020: Complete revision of the article [SAP ASE Azure Virtual Machines DBMS deployment for SAP workload](#)
- 02/21/2020: Change in [SAP HANA Azure virtual machine storage configuration](#) to represent new recommendation in stripe size for /hana/data and adding setting of I/O scheduler
- 02/21/2020: Changes in HANA Large Instance documents to represent newly certified SKUs of S224 and S224m
- 02/21/2020: Change in [Azure VMs high availability for SAP NetWeaver on RHEL](#) and [Azure VMs high availability for SAP NetWeaver on RHEL with Azure NetApp Files](#) to adjust the cluster constraints for enqueue server replication 2 architecture (ENSA2)
- 02/20/2020: Change in [High availability for SAP NetWeaver on Azure VMs on SLES multi-SID guide](#) to add a link to the SUSE multi-SID cluster guide
- 02/13/2020: Changes to [Azure Virtual Machines planning and implementation for SAP NetWeaver](#) to implement links to new documents
- 02/13/2020: Added new document [SAP workload on Azure virtual machine supported scenario](#)
- 02/13/2020: Added new document [What SAP software is supported for Azure deployment](#)
- 02/13/2020: Change in [High availability of IBM Db2 LUW on Azure VMs on Red Hat Enterprise Linux Server](#) to point to document that describes access to public endpoint with Standard Azure Load balancer
- 02/13/2020: Add the new VM types to [SAP certifications and configurations running on Microsoft Azure](#)
- 02/13/2020: Add new SAP support notes [SAP workloads on Azure: planning and deployment checklist](#)
- 02/13/2020: Change in [Azure VMs high availability for SAP NetWeaver on RHEL](#) and [Azure VMs high availability for SAP NetWeaver on RHEL with Azure NetApp Files](#)

availability for SAP NetWeaver on RHEL with Azure NetApp Files to align the cluster resources timeouts to the Red Hat timeout recommendations

- 02/11/2020: Release of [SAP HANA on Azure Large Instance migration to Azure Virtual Machines](#)
- 02/07/2020: Change in [Public endpoint connectivity for VMs using Azure Standard ILB in SAP HA scenarios](#) to update sample NSG screenshot
- 02/03/2020: Change in [High availability for SAP NW on Azure VMs on SLES for SAP applications](#) and [High availability for SAP NW on Azure VMs on SLES with ANF for SAP applications](#) to remove the warning about using dash in the host names of cluster nodes on SLES
- 01/28/2020: Change in [High availability of SAP HANA on Azure VMs on RHEL](#) to align the SAP HANA cluster resources timeouts to the Red Hat timeout recommendations
- 01/17/2020: Change in [Azure proximity placement groups for optimal network latency with SAP applications](#) to change the section of moving existing VMs into a proximity placement group
- 01/17/2020: Change in [SAP workload configurations with Azure Availability Zones](#) to point to procedure that automates measurements of latency between Availability Zones
- 01/16/2020: Change in [How to install and configure SAP HANA \(Large Instances\) on Azure](#) to adapt OS releases to HANA IaaS hardware directory
- 01/16/2020: Changes in [High availability for SAP NetWeaver on Azure VMs on SLES multi-SID guide](#) to add instructions for SAP systems, using enqueue server 2 architecture (ENSA2)
- 01/10/2020: Changes in [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on SLES](#) and in [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on RHEL](#) to add instructions on how to make `nfs4_disable_idmapping` changes permanent.
- 01/10/2020: Changes in [High availability for SAP NetWeaver on Azure VMs on SLES with Azure NetApp Files for SAP applications](#) and in [Azure Virtual Machines high availability for SAP NetWeaver on RHEL with Azure NetApp Files for SAP applications](#) to add instructions how to mount Azure NetApp Files NFSv4 volumes.
- 12/23/2019: Release of [High availability for SAP NetWeaver on Azure VMs on SLES multi-SID guide](#)
- 12/18/2019: Release of [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on RHEL](#)
- 11/21/2019: Changes in [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on SUSE Linux Enterprise Server](#) to simplify the configuration for NFS ID mapping and change the recommended primary network interface to simplify routing.
- 11/15/2019: Minor changes in [High availability for SAP NetWeaver on SUSE Linux Enterprise Server with Azure NetApp Files for SAP applications](#) and [High availability for SAP NetWeaver on Red Hat Enterprise Linux with Azure NetApp Files for SAP applications](#) to clarify capacity pool size restrictions and remove statement that only NFSv3 version is supported.
- 11/12/2019: Release of [High availability for SAP NetWeaver on Windows with Azure NetApp Files \(SMB\)](#)
- 11/08/2019: Changes in [High availability of SAP HANA on Azure VMs on SUSE Linux Enterprise Server](#), [Set up SAP HANA System Replication on Azure virtual machines \(VMs\)](#), [Azure Virtual Machines high availability for SAP NetWeaver on SUSE Linux Enterprise Server for SAP applications](#), [Azure Virtual Machines high availability for SAP NetWeaver on SUSE Linux Enterprise Server with Azure NetApp Files](#), [Azure Virtual Machines high availability for SAP NetWeaver on Red Hat Enterprise Linux](#), [Azure Virtual Machines high availability for SAP NetWeaver on Red Hat Enterprise Linux with Azure NetApp Files](#), [High availability for NFS on Azure VMs on SUSE Linux Enterprise Server](#), [GlusterFS on Azure VMs on Red Hat Enterprise Linux for SAP NetWeaver](#) to recommend Azure standard load balancer
- 11/08/2019: Changes in [SAP workload planning and deployment checklist](#) to clarify encryption recommendation
- 11/04/2019: Changes in [Setting up Pacemaker on SUSE Linux Enterprise Server in Azure](#) to create the cluster directly with unicast configuration
- 10/29/2019: Release of [Public endpoint connectivity for Virtual Machines using Azure Standard Load Balancer in SAP high-availability scenarios](#)

- 10/25/2019: Changes in [SAP HANA Azure virtual machine storage configurations](#) and [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on SUSE Linux Enterprise Server](#) to clarify NFS protocol for /hana/shared volume
- 10/22/2019: Change in [High availability for SAP NetWeaver on Azure VMs on SUSE Linux Enterprise Server for SAP applications](#), [High availability for SAP NetWeaver on Azure VMs on SUSE Linux Enterprise Server with Azure NetApp Files for SAP applications](#), [High availability for NFS on Azure VMs on SUSE Linux Enterprise Server](#), [Setting up Pacemaker on SUSE Linux Enterprise Server in Azure](#), [High availability of IBM Db2 LUW on Azure VMs on SUSE Linux Enterprise Server with Pacemaker](#), and [High availability of SAP HANA on Azure VMs on SUSE Linux Enterprise Server](#) for Azure Load-Balancer Detection Hardening
- Changes ANF section and header section in [SAP HANA Azure virtual machine storage configurations](#)
- 10/21/2019: Release of [SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on SLES](#)
- 10/16/2019: Fix broken links in [Backup and restore](#)
- 10/16/2019: Change the minimum recommended OS from SLES 12 SP3 to SLES 12 SP4 in [High availability of IBM Db2 LUW on Azure VMs on SUSE Linux Enterprise Server with Pacemaker](#)
- 10/11/2019: Changes to Ultra disk storage configurations and introduction of ANF in [SAP HANA Azure virtual machine storage configurations](#)
- 10/01/2019: Change in [graphics of Azure proximity placement groups for optimal network latency with SAP applications](#) to get more clarity
- 10/01/2019: Change in [SAP HANA infrastructure configurations and operations on Azure](#) to correct statements around highly available NFS share for /hana/shared.
- 09/28/2019: Change in [Setting up Pacemaker on Red Hat Enterprise Linux in Azure](#) to clarify SBD as a fencing mechanism is not supported on RHEL clusters
- 09/17/2019: Change in NetWeaver Planning and Deployment Guide to unify terms around VM Extension for SAP
- 08/22/2019: Changes in [Setting up Pacemaker on SUSE Linux Enterprise Server in Azure](#) to update the URLs for custom role creation
- 08/16/2019: Changes in [Setting up Pacemaker on Red Hat Enterprise Linux in Azure](#) to remind customers to update the actions in the custom role, if updating to the new version of the Azure fence agent
- 08/15/2019: Changes in [SAP HANA Azure virtual machine storage configurations](#) to reflect General Availability of Ultra disk (formerly Ultra SSD)
- 08/01/2019: Changes to [Setting up Pacemaker on SUSE Linux Enterprise Server in Azure](#) to integrate changes specifically for SLES 15

# Create MATLAB Distributed Computing Server clusters on Azure VMs

11/13/2019 • 3 minutes to read • [Edit Online](#)

Use Microsoft Azure virtual machines to create one or more MATLAB Distributed Computing Server clusters to run your compute-intensive parallel MATLAB workloads. Install your MATLAB Distributed Computing Server software on a VM to use as a base image and use an Azure quickstart template or Azure PowerShell script (available on [GitHub](#)) to deploy and manage the cluster. After deployment, connect to the cluster to run your workloads.

## About MATLAB and MATLAB Distributed Computing Server

The [MATLAB](#) platform is optimized for solving engineering and scientific problems. MATLAB users with large-scale simulations and data processing tasks can use MathWorks parallel computing products to speed up their compute-intensive workloads by taking advantage of compute clusters and grid services. [Parallel Computing Toolbox](#) lets MATLAB users parallelize applications and take advantage of multi-core processors, GPUs, and compute clusters. [MATLAB Distributed Computing Server](#) enables MATLAB users to utilize many computers in a compute cluster.

By using Azure virtual machines, you can create MATLAB Distributed Computing Server clusters that have all the same mechanisms available to submit parallel work as on-premises clusters, such as interactive jobs, batch jobs, independent tasks, and communicating tasks. Using Azure in conjunction with the MATLAB platform has many benefits compared to provisioning and using traditional on-premises hardware: a range of virtual machine sizes, creation of clusters on-demand so you pay only for the compute resources you use, and the ability to test models at scale.

## Prerequisites

- **Client computer** - You'll need a Windows-based client computer to communicate with Azure and the MATLAB Distributed Computing Server cluster after deployment.
- **Azure PowerShell** - See [How to install and configure Azure PowerShell](#) to install it on your client computer.
- **Azure subscription** - If you don't have a subscription, you can create a [free account](#) in just a couple of minutes. For larger clusters, consider a pay-as-you-go subscription or other purchase options.
- **vCPUs quota** - You might need to increase the vCPU quota to deploy a large cluster or more than one MATLAB Distributed Computing Server cluster. To increase a quota, [open an online customer support request](#) at no charge.
- **MATLAB, Parallel Computing Toolbox, and MATLAB Distributed Computing Server licenses** - The scripts assume that the [MathWorks Hosted License Manager](#) is used for all licenses.
- **MATLAB Distributed Computing Server software** - Will be installed on a VM that will be used as the base VM image for the cluster VMs.

## High level steps

To use Azure virtual machines for your MATLAB Distributed Computing Server clusters, the following high-level steps are required. Detailed instructions are in the documentation accompanying the quickstart template and scripts on [GitHub](#).

1. **Create a base VM image**

- Download and install MATLAB Distributed Computing Server software onto this VM.

**NOTE**

This process can take a couple of hours, but you only have to do it once for each version of MATLAB you use.

## 2. Create one or more clusters

- Use the supplied PowerShell script or use the quickstart template to create a cluster from the base VM image.
- Manage the clusters using the supplied PowerShell script which allows you to list, pause, resume, and delete clusters.

## Cluster configurations

Currently, the cluster creation script and template enable you to create a single MATLAB Distributed Computing Server topology. If you want, create one or more additional clusters, with each cluster having a different number of worker VMs, using different VM sizes, and so on.

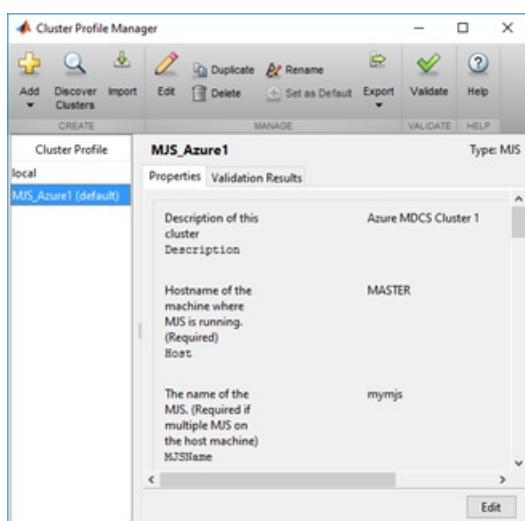
### MATLAB client and cluster in Azure

The MATLAB client node, MATLAB Job Scheduler node, and MATLAB Distributed Computing Server "worker" nodes are all configured as Azure VMs in a virtual network, as shown in the following figure.

- To use the cluster, connect by Remote Desktop to the client node. The client node runs the MATLAB client.
- The client node has a file share that can be accessed by all workers.
- MathWorks Hosted License Manager is used for the license checks for all MATLAB software.
- By default, one MATLAB Distributed Computing Server worker per vCPU is created on the worker VMs, but you can specify any number.

## Use an Azure-based Cluster

As with other types of MATLAB Distributed Computing Server clusters, you need to use the Cluster Profile Manager in the MATLAB client (on the client VM) to create a MATLAB Job Scheduler cluster profile.



## Next steps

- For detailed instructions to deploy and manage MATLAB Distributed Computing Server clusters in Azure, see the [GitHub](#) repository containing the templates and scripts.
- Go to the [MathWorks site](#) for detailed documentation for MATLAB and MATLAB Distributed Computing

Server.

# Visual Studio images on Azure

1/10/2020 • 4 minutes to read • [Edit Online](#)

Using Visual Studio in a preconfigured Azure virtual machine (VM) is a quick, easy way to go from nothing to an up-and-running development environment. System images with different Visual Studio configurations are available in the [Azure Marketplace](#).

New to Azure? [Create a free Azure account](#).

## NOTE

Not all subscriptions are eligible to deploy Windows 10 images. For more information see [Use Windows client in Azure for dev/test scenarios](#)

## What configurations and versions are available?

Images for the most recent major versions, Visual Studio 2019, Visual Studio 2017 and Visual Studio 2015, can be found in the Azure Marketplace. For each released major version, you see the originally "released to web" (RTW) version and the latest updated versions. Each of these versions offers the Visual Studio Enterprise and the Visual Studio Community editions. These images are updated at least every month to include the latest Visual Studio and Windows updates. While the names of the images remain the same, each image's description includes the installed product version and the image's "as of" date.

| RELEASE VERSION                                           | EDITIONS              | PRODUCT VERSION       |
|-----------------------------------------------------------|-----------------------|-----------------------|
| <a href="#">Visual Studio 2019: Latest (Version 16.4)</a> | Enterprise, Community | Version 16.4.0        |
| <a href="#">Visual Studio 2019: RTW</a>                   | Enterprise, Community | Version 16.0.9        |
| <a href="#">Visual Studio 2017: Latest (Version 15.9)</a> | Enterprise, Community | Version 15.9.17       |
| <a href="#">Visual Studio 2017: RTW</a>                   | Enterprise, Community | Version 15.0.27       |
| <a href="#">Visual Studio 2015: Latest (Update 3)</a>     | Enterprise, Community | Version 14.0.25431.01 |

## NOTE

In accordance with Microsoft servicing policy, the originally released (RTW) version of Visual Studio 2015 has expired for servicing. Visual Studio 2015 Update 3 is the only remaining version offered for the Visual Studio 2015 product line.

For more information, see the [Visual Studio Servicing Policy](#).

## What features are installed?

Each image contains the recommended feature set for that Visual Studio edition. Generally, the installation includes:

- All available workloads, including each workload's recommended optional components

- .NET 4.6.2 and .NET 4.7 SDKs, Targeting Packs, and Developer Tools
- Visual F#
- GitHub Extension for Visual Studio
- LINQ to SQL Tools

The command line used to install Visual Studio when building the images is as follows:

```
vs_enterprise.exe --allWorkloads --includeRecommended --passive ^
 add Microsoft.Net.Component.4.7.SDK ^
 add Microsoft.Net.Component.4.7.TargetingPack ^
 add Microsoft.Net.Component.4.6.2.SDK ^
 add Microsoft.Net.Component.4.6.2.TargetingPack ^
 add Microsoft.Net.ComponentGroup.4.7.DeveloperTools ^
 add Microsoft.VisualStudio.Component.FSharp ^
 add Component.GitHub.VisualStudio ^
 add Microsoft.VisualStudio.Component.LinqToSql
```

If the images don't include a Visual Studio feature that you require, provide feedback through the feedback tool in the upper-right corner of the page.

## What size VM should I choose?

Azure offers a full range of virtual machine sizes. Because Visual Studio is a powerful, multi-threaded application, you want a VM size that includes at least two processors and 7 GB of memory. We recommend the following VM sizes for the Visual Studio images:

- Standard\_D2\_v3
- Standard\_D2s\_v3
- Standard\_D4\_v3
- Standard\_D4s\_v3
- Standard\_D2\_v2
- Standard\_D2S\_v2
- Standard\_D3\_v2

For more information on the latest machine sizes, see [Sizes for Windows virtual machines in Azure](#).

With Azure, you can rebalance your initial choice by resizing the VM. You can either provision a new VM with a more appropriate size, or resize your existing VM to different underlying hardware. For more information, see [Resize a Windows VM](#).

## After the VM is running, what's next?

Visual Studio follows the "bring your own license" model in Azure. As with an installation on proprietary hardware, one of the first steps is licensing your Visual Studio installation. To unlock Visual Studio, either:

- Sign in with a Microsoft account that's associated with a Visual Studio subscription
- Unlock Visual Studio with the product key that came with your initial purchase

For more information, see [Sign in to Visual Studio](#) and [How to unlock Visual Studio](#).

## How do I save the development VM for future or team use?

The spectrum of development environments is huge, and there's real cost associated with building out the more complex environments. Regardless of your environment's configuration, you can save, or capture, your configured VM as a "base image" for future use or for other members of your team. Then, when booting a new VM, you

provision it from the base image rather than the Azure Marketplace image.

A quick summary: Use the System Preparation tool (Sysprep) and shut down the running VM, and then capture (*Figure 1*) the VM as an image through the UI in the Azure portal. Azure saves the `.vhdx` file that contains the image in the storage account of your choosing. The new image then shows up as an Image resource in your subscription's list of resources.



(*Figure 1*) Capture an image through the Azure portal UI.

For more information, see [Create a managed image of a generalized VM in Azure](#).

#### IMPORTANT

Don't forget to use Sysprep to prepare the VM. If you miss that step, Azure can't provision a VM from the image.

#### NOTE

You still incur some cost for storage of the images, but that incremental cost can be insignificant compared to the overhead costs to rebuild the VM from scratch for each team member who needs one. For instance, it costs a few dollars to create and store a 127-GB image for a month that's reusable by your entire team. However, these costs are insignificant compared to hours each employee invests to build out and validate a properly configured dev box for their individual use.

Additionally, your development tasks or technologies might need more scale, like varieties of development configurations and multiple machine configurations. You can use Azure DevTest Labs to create *recipes* that automate construction of your "golden image." You can also use DevTest Labs to manage policies for your team's running VMs. [Using Azure DevTest Labs for developers](#) is the best source for more information on DevTest Labs.

## Next steps

Now that you know about the preconfigured Visual Studio images, the next step is to create a new VM:

- [Create a VM through the Azure portal](#)
- [Windows Virtual Machines overview](#)

2 minutes to read

# Attach a data disk to a Windows VM with PowerShell

11/13/2019 • 2 minutes to read • [Edit Online](#)

This article shows you how to attach both new and existing disks to a Windows virtual machine by using PowerShell.

First, review these tips:

- The size of the virtual machine controls how many data disks you can attach. For more information, see [Sizes for virtual machines](#).
- To use premium SSDs, you'll need a [premium storage-enabled VM type](#), like the DS-series or GS-series virtual machine.

This article uses PowerShell within the [Azure Cloud Shell](#), which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

## Add an empty data disk to a virtual machine

This example shows how to add an empty data disk to an existing virtual machine.

### Using managed disks

```
$rgName = 'myResourceGroup'
$vmName = 'myVM'
$location = 'East US'
$storageType = 'Premium_LRS'
$dataDiskName = $vmName + '_datadisk1'

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -CreateOption Empty -DiskSizeGB 128
$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -ResourceGroupName $rgName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
$vm = Add-AzVMDisk -VM $vm -Name $dataDiskName -CreateOption Attach -ManagedDiskId $dataDisk1.Id -Lun 1

Update-AzVM -VM $vm -ResourceGroupName $rgName
```

### Using managed disks in an Availability Zone

To create a disk in an Availability Zone, use [New-AzDiskConfig](#) with the `-Zone` parameter. The following example creates a disk in zone 1.

```
$rgName = 'myResourceGroup'
$vmName = 'myVM'
$location = 'East US 2'
$storageType = 'Premium_LRS'
$dataDiskName = $vmName + '_datadisk1'

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -CreateOption Empty -DiskSizeGB 128
-Zone 1
$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -ResourceGroupName $rgName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
$vm = Add-AzVMDisk -VM $vm -Name $dataDiskName -CreateOption Attach -ManagedDiskId $dataDisk1.Id -Lun 1

Update-AzVM -VM $vm -ResourceGroupName $rgName
```

## Initialize the disk

After you add an empty disk, you'll need to initialize it. To initialize the disk, you can sign in to a VM and use disk management. If you enabled [WinRM](#) and a certificate on the VM when you created it, you can use remote PowerShell to initialize the disk. You can also use a custom script extension:

```
$location = "location-name"
$scriptName = "script-name"
$fileName = "script-file-name"
Set-AzVMCustomScriptExtension -ResourceGroupName $rgName -Location $locName -VMName $vmName -Name
$scriptName -TypeHandlerVersion "1.4" -StorageAccountName "mystore1" -StorageAccountKey "primary-key" -
FileName $fileName -ContainerName "scripts"
```

The script file can contain code to initialize the disks, for example:

```
$disks = Get-Disk | Where partitionstyle -eq 'raw' | sort number

$letters = 70..89 | ForEach-Object { [char]$_. }
$count = 0
$labels = "data1","data2"

foreach ($disk in $disks) {
 $driveLetter = $letters[$count].ToString()
 $disk |
 Initialize-Disk -PartitionStyle MBR -PassThru |
 New-Partition -UseMaximumSize -DriveLetter $driveLetter |
 Format-Volume -FileSystem NTFS -NewFileSystemLabel $labels[$count] -Confirm:$false -Force
$count++
}
```

## Attach an existing data disk to a VM

You can attach an existing managed disk to a VM as a data disk.

```
$rgName = "myResourceGroup"
$vmName = "myVM"
$location = "East US"
$dataDiskName = "myDisk"
$disk = Get-AzDisk -ResourceGroupName $rgName -DiskName $dataDiskName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName

$vm = Add-AzVMDataDisk -CreateOption Attach -Lun 0 -VM $vm -ManagedDiskId $disk.Id

Update-AzVM -VM $vm -ResourceGroupName $rgName
```

## Next steps

You can also deploy managed disks using templates. For more information, see [Using Managed Disks in Azure Resource Manager Templates](#) or the [quickstart template](#) for deploying multiple data disks.

# Attach a managed data disk to a Windows VM by using the Azure portal

2/28/2020 • 2 minutes to read • [Edit Online](#)

This article shows you how to attach a new managed data disk to a Windows virtual machine (VM) by using the Azure portal. The size of the VM determines how many data disks you can attach. For more information, see [Sizes for virtual machines](#).

## Add a data disk

1. Go to the [Azure portal](#) to add a data disk. Search for and select **Virtual machines**.
2. Select a virtual machine from the list.
3. On the **Virtual machine** page, select **Disk**s.
4. On the **Disk**s page, select **Add data disk**.
5. In the drop-down for the new disk, select **Create disk**.
6. In the **Create managed disk** page, type in a name for the disk and adjust the other settings as necessary. When you're done, select **Create**.
7. In the **Disk**s page, select **Save** to save the new disk configuration for the VM.
8. After Azure creates the disk and attaches it to the virtual machine, the new disk is listed in the virtual machine's disk settings under **Data disks**.

## Initialize a new data disk

1. Connect to the VM.
2. Select the Windows **Start** menu inside the running VM and enter **diskmgmt.msc** in the search box. The **Disk Management** console opens.
3. Disk Management recognizes that you have a new, uninitialized disk and the **Initialize Disk** window appears.
4. Verify the new disk is selected and then select **OK** to initialize it.
5. The new disk appears as **unallocated**. Right-click anywhere on the disk and select **New simple volume**. The **New Simple Volume Wizard** window opens.
6. Proceed through the wizard, keeping all of the defaults, and when you're done select **Finish**.
7. Close **Disk Management**.
8. A pop-up window appears notifying you that you need to format the new disk before you can use it. Select **Format disk**.
9. In the **Format new disk** window, check the settings, and then select **Start**.
10. A warning appears notifying you that formatting the disks erases all of the data. Select **OK**.
11. When the formatting is complete, select **OK**.

## Next steps

- You can also [attach a data disk by using PowerShell](#).
- If your application needs to use the *D:* drive to store data, you can [change the drive letter of the Windows temporary disk](#).

# How to detach a data disk from a Windows virtual machine

1/9/2020 • 2 minutes to read • [Edit Online](#)

When you no longer need a data disk that's attached to a virtual machine, you can easily detach it. This removes the disk from the virtual machine, but doesn't remove it from storage.

## WARNING

If you detach a disk it is not automatically deleted. If you have subscribed to Premium storage, you will continue to incur storage charges for the disk. For more information, see [Pricing and Billing when using Premium Storage](#).

If you want to use the existing data on the disk again, you can reattach it to the same virtual machine, or another one.

## Detach a data disk using PowerShell

You can *hot* remove a data disk using PowerShell, but make sure nothing is actively using the disk before detaching it from the VM.

In this example, we remove the disk named **myDisk** from the VM **myVM** in the **myResourceGroup** resource group. First you remove the disk using the [Remove-AzVMDataDisk](#) cmdlet. Then, you update the state of the virtual machine, using the [Update-AzVM](#) cmdlet, to complete the process of removing the data disk.

```
$VirtualMachine = Get-AzVM `
 -ResourceGroupName "myResourceGroup" `
 -Name "myVM" `
Remove-AzVMDataDisk `
 -VM $VirtualMachine `
 -Name "myDisk" `
Update-AzVM `
 -ResourceGroupName "myResourceGroup" `
 -VM $VirtualMachine
```

The disk stays in storage but is no longer attached to a virtual machine.

## Detach a data disk using the portal

You can *hot* remove a data disk, but make sure nothing is actively using the disk before detaching it from the VM.

1. In the left menu, select **Virtual Machines**.
2. Select the virtual machine that has the data disk you want to detach.
3. Under **Settings**, select **Disks**.
4. At the top of the **Disks** pane, select **Edit**.
5. In the **Disks** pane, to the far right of the data disk that you would like to detach, select **Detach**.
6. Select **Save** on the top of the page to save your changes.

The disk stays in storage but is no longer attached to a virtual machine.

## Next steps

If you want to reuse the data disk, you can just [attach it to another VM](#)

# Using Managed Disks in Azure Resource Manager Templates

12/10/2019 • 5 minutes to read • [Edit Online](#)

This document walks through the differences between managed and unmanaged disks when using Azure Resource Manager templates to provision virtual machines. The examples help you to update existing templates that are using unmanaged Disks to managed disks. For reference, we are using the [101-vm-simple-windows](#) template as a guide. You can see the template using both [managed Disks](#) and a prior version using [unmanaged disks](#) if you'd like to directly compare them.

## Unmanaged Disks template formatting

To begin, let's take a look at how unmanaged disks are deployed. When creating unmanaged disks, you need a storage account to hold the VHD files. You can create a new storage account or use one that already exists. This article shows you how to create a new storage account. Create a storage account resource in the resources block as shown below.

```
{
 "type": "Microsoft.Storage/storageAccounts",
 "apiVersion": "2018-07-01",
 "name": "[variables('storageAccountName')]",
 "location": "[resourceGroup().location]",
 "sku": {
 "name": "Standard_LRS"
 },
 "kind": "Storage",
 "properties": {}
}
```

Within the virtual machine object, add a dependency on the storage account to ensure that it's created before the virtual machine. Within the `storageProfile` section, specify the full URI of the VHD location, which references the storage account and is needed for the OS disk and any data disks.

```
{
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2018-10-01",
 "name": "[variables('vmName')]",
 "location": "[resourceGroup().location]",
 "dependsOn": [
 "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
 "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
],
 "properties": {
 "hardwareProfile": {...},
 "osProfile": {...},
 "storageProfile": {
 "imageReference": {
 "publisher": "MicrosoftWindowsServer",
 "offer": "WindowsServer",
 "sku": "[parameters('windowsOSVersion')]",
 "version": "latest"
 },
 "osDisk": {
 "name": "osdisk",
 "vhd": {
 "uri": "[concat(reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob, 'vhds/osdisk.vhd')]"
 },
 "caching": "ReadWrite",
 "createOption": "FromImage"
 },
 "dataDisks": [
 {
 "name": "datadisk1",
 "diskSizeGB": 1023,
 "lun": 0,
 "vhd": {
 "uri": "[concat(reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob, 'vhds/datadisk1.vhd')]"
 },
 "createOption": "Empty"
 }
]
 },
 "networkProfile": {...},
 "diagnosticsProfile": {...}
 }
}
```

## Managed disks template formatting

With Azure Managed Disks, the disk becomes a top-level resource and no longer requires a storage account to be created by the user. Managed disks were first exposed in the [2016-04-30-preview](#) API version, they are available in all subsequent API versions and are now the default disk type. The following sections walk through the default settings and detail how to further customize your disks.

### NOTE

It is recommended to use an API version later than [2016-04-30-preview](#) as there were breaking changes between [2016-04-30-preview](#) and [2017-03-30](#).

### Default managed disk settings

To create a VM with managed disks, you no longer need to create the storage account resource. Referencing the template example below, there are some differences from the previous unmanged disk examples to note:

- The `apiVersion` is a version that supports managed disks.
- `osDisk` and `dataDisks` no longer refer to a specific URI for the VHD.
- When deploying without specifying additional properties, the disk will use a storage type based on the size of the VM. For example, if you are using a VM size that supports premium storage (sizes with "s" in their name such as Standard\_D2s\_v3) then premium disks will be configured by default. You can change this by using the `sku` setting of the disk to specify a storage type.
- If no name for the disk is specified, it takes the format of `<VMName>_OsDisk_1_<randomstring>` for the OS disk and `<VMName>_disk<#>_<randomstring>` for each data disk.
  - If a VM is being created from a custom image then the default settings for storage account type and disk name are retrieved from the disk properties defined in the custom image resource. These can be overridden by specifying values for these in the template.
- By default, Azure disk encryption is disabled.
- By default, disk caching is Read/Write for the OS disk and None for data disks.
- In the example below there is still a storage account dependency, though this is only for storage of diagnostics and is not needed for disk storage.

```
{
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2018-10-01",
 "name": "[variables('vmName')]",
 "location": "[resourceGroup().location]",
 "dependsOn": [
 "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
 "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
],
 "properties": {
 "hardwareProfile": {...},
 "osProfile": {...},
 "storageProfile": {
 "imageReference": {
 "publisher": "MicrosoftWindowsServer",
 "offer": "WindowsServer",
 "sku": "[parameters('windowsOSVersion')]",
 "version": "latest"
 },
 "osDisk": {
 "createOption": "FromImage"
 },
 "dataDisks": [
 {
 "diskSizeGB": 1023,
 "lun": 0,
 "createOption": "Empty"
 }
]
 },
 "networkProfile": {...},
 "diagnosticsProfile": {...}
 }
}
```

## Using a top-level managed disk resource

As an alternative to specifying the disk configuration in the virtual machine object, you can create a top-level disk resource and attach it as part of the virtual machine creation. For example, you can create a disk resource as follows to use as a data disk.

```
{
 "type": "Microsoft.Compute/disks",
 "apiVersion": "2018-06-01",
 "name": "[concat(variables('vmName'),'-datadisk1')]",
 "location": "[resourceGroup().location]",
 "sku": {
 "name": "Standard_LRS"
 },
 "properties": {
 "creationData": {
 "createOption": "Empty"
 },
 "diskSizeGB": 1023
 }
}
```

Within the VM object, reference the disk object to be attached. Specifying the resource ID of the managed disk created in the `managedDisk` property allows the attachment of the disk as the VM is created. The `apiVersion` for the VM resource is set to `2017-03-30`. A dependency on the disk resource is added to ensure it's successfully created before VM creation.

```
{
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2018-10-01",
 "name": "[variables('vmName')]",
 "location": "[resourceGroup().location]",
 "dependsOn": [
 "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
 "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]",
 "[resourceId('Microsoft.Compute/disks/', concat(variables('vmName'),'-datadisk1'))]"
],
 "properties": {
 "hardwareProfile": {...},
 "osProfile": {...},
 "storageProfile": {
 "imageReference": {
 "publisher": "MicrosoftWindowsServer",
 "offer": "WindowsServer",
 "sku": "[parameters('windowsOSVersion')]",
 "version": "latest"
 },
 "osDisk": {
 "createOption": "FromImage"
 },
 "dataDisks": [
 {
 "lun": 0,
 "name": "[concat(variables('vmName'),'-datadisk1')]",
 "createOption": "attach",
 "managedDisk": {
 "id": "[resourceId('Microsoft.Compute/disks/', concat(variables('vmName'),'-datadisk1'))]"
 }
 }
]
 },
 "networkProfile": {...},
 "diagnosticsProfile": {...}
 }
}
```

## Create managed availability sets with VMs using managed disks

To create managed availability sets with VMs using managed disks, add the `sku` object to the availability set

resource and set the `name` property to `Aligned`. This property ensures that the disks for each VM are sufficiently isolated from each other to avoid single points of failure. Also note that the `apiVersion` for the availability set resource is set to `2018-10-01`.

```
{
 "type": "Microsoft.Compute/availabilitySets",
 "apiVersion": "2018-10-01",
 "location": "[resourceGroup().location]",
 "name": "[variables('avSetName')]",
 "properties": {
 "PlatformUpdateDomainCount": 3,
 "PlatformFaultDomainCount": 2
 },
 "sku": {
 "name": "Aligned"
 }
}
```

## Standard SSD disks

Below are the parameters needed in the Resource Manager template to create Standard SSD Disks:

- `apiVersion` for Microsoft.Compute must be set as `2018-04-01` (or later)
- Specify `managedDisk.storageAccountType` as `StandardSSD_LRS`

The following example shows the `properties.storageProfile.osDisk` section for a VM that uses Standard SSD Disks:

```
"osDisk": {
 "osType": "Windows",
 "name": "myOsDisk",
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "managedDisk": {
 "storageAccountType": "StandardSSD_LRS"
 }
}
```

For a complete template example of how to create a Standard SSD disk with a template, see [Create a VM from a Windows Image with Standard SSD Data Disks](#).

## Additional scenarios and customizations

To find full information on the REST API specifications, please review the [create a managed disk REST API documentation](#). You will find additional scenarios, as well as default and acceptable values that can be submitted to the API through template deployments.

## Next steps

- For full templates that use managed disks visit the following Azure Quickstart Repo links.
  - [Windows VM with managed disk](#)
  - [Linux VM with managed disk](#)
- Visit the [Azure Managed Disks Overview](#) document to learn more about managed disks.
- Review the template reference documentation for virtual machine resources by visiting the [Microsoft.Compute/virtualMachines template reference](#) document.
- Review the template reference documentation for disk resources by visiting the [Microsoft.Compute/disks template reference](#) document.
- For information on how to use managed disks in Azure virtual machine scale sets, visit the [Use data disks with scale sets](#) document.



# Enable shared disk

2/19/2020 • 3 minutes to read • [Edit Online](#)

This article covers how to enable the shared disks (preview) feature for Azure managed disks. Azure shared disks (preview) is a new feature for Azure managed disks that enables you to attach a managed disk to multiple virtual machines (VMs) simultaneously. Attaching a managed disk to multiple VMs allows you to either deploy new or migrate existing clustered applications to Azure.

If you are looking for conceptual information on managed disks that have shared disks enabled, refer to [Azure shared disks](#).

## Limitations

While in preview, managed disks that have shared disks enabled are subject to the following limitations:

- Currently only available with premium SSDs.
- Currently only supported in the West Central US region.
- All virtual machines sharing a disk must be deployed in the same [proximity placement groups](#).
- Can only be enabled on data disks, not OS disks.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see [Failover clustering hardware requirements and storage options](#).
- ReadOnly host caching is not available for premium SSDs with `maxShares>1`.
- Availability sets and virtual machine scale sets can only be used with `FaultDomainCount` set to 1.
- Azure Backup and Azure Site Recovery support is not yet available.

If you're interested in trying shared disks then [sign up for our preview](#).

## Disk sizes

For now, only premium SSDs can enable shared disks. The disk sizes that support this feature are P15 and greater. Different disk sizes may have a different `maxShares` limit, which you cannot exceed when setting the `maxShares` value.

For each disk, you can define a `maxShares` value that represents the maximum number of nodes that can simultaneously share the disk. For example, if you plan to set up a 2-node failover cluster, you would set `maxShares=2`. The maximum value is an upper bound. Nodes can join or leave the cluster (mount or unmount the disk) as long as the number of nodes is lower than the specified `maxShares` value.

### NOTE

The `maxShares` value can only be set or edited when the disk is detached from all nodes.

The following table illustrates the allowed maximum values for `maxShares` by disk size:

| DISK SIZES    | MAXSHARES LIMIT |
|---------------|-----------------|
| P15, P20      | 2               |
| P30, P40, P50 | 5               |

| DISK SIZES    | MAXSHARES LIMIT |
|---------------|-----------------|
| P60, P70, P80 | 10              |

The IOPS and bandwidth limits for a disk are not affected by the `maxShares` value. For example, the max IOPS of a P15 disk are 1100 whether `maxShares = 1` or `maxShares > 1`.

## Deploy an Azure shared disk

To deploy a managed disk with the shared disk feature enabled, use the new property `maxShares` and define a value `>1`. This makes the disk shareable across multiple VMs.

### IMPORTANT

The value of `maxShares` can only be set or changed when a disk is unmounted from all VMs. See the [Disk sizes](#) for the allowed values for `maxShares`.

Before using the following template, replace `[parameters('dataDiskName')]`, `[resourceGroup().location]`, `[parameters('dataDiskSizeGB')]`, and `[parameters('maxShares')]` with your own values.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "dataDiskName": {
 "type": "string",
 "defaultValue": "mySharedDisk"
 },
 "dataDiskSizeGB": {
 "type": "int",
 "defaultValue": 1024
 },
 "maxShares": {
 "type": "int",
 "defaultValue": 2
 }
 },
 "resources": [
 {
 "type": "Microsoft.Compute/disks",
 "name": "[parameters('dataDiskName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2019-07-01",
 "sku": {
 "name": "Premium_LRS"
 },
 "properties": {
 "creationData": {
 "createOption": "Empty"
 },
 "diskSizeGB": "[parameters('dataDiskSizeGB')]",
 "maxShares": "[parameters('maxShares')]"
 }
 }
]
}
```

## Using Azure shared disks with your VMs

Once you've deployed a shared disk with `maxShares>1`, you can mount the disk to one or more of your VMs.

## IMPORTANT

All VMs sharing a disk must be deployed in the same [proximity placement group](#).

```
$resourceGroup = "myResourceGroup"
.setLocation = "WestCentralUS"
$ppgName = "myPPG"
$ppg = New-AzProximityPlacementGroup `
 -Location $location `
 -Name $ppgName `
 -ResourceGroupName $resourceGroup `
 -ProximityPlacementGroupType Standard

$vm = New-AzVm -ResourceGroupName $resourceGroup -Name "myVM" -Location $location -VirtualNetworkName "myVnet"
-SubnetName "mySubnet" -SecurityGroupName "myNetworkSecurityGroup" -PublicIpAddressName "myPublicIpAddress" -
ProximityPlacementGroup $ppg.Id

$dataDisk = Get-AzDisk -ResourceGroupName $resourceGroup -DiskName "mySharedDisk"

$vm = Add-AzVMDataDisk -VM $vm -Name "mySharedDisk" -CreateOption Attach -ManagedDiskId $dataDisk.Id -Lun 0

update-AzVm -VM $vm -ResourceGroupName $resourceGroup
```

## Supported SCSI PR commands

Once you've mounted the shared disk to your VMs in your cluster, you can establish quorum and read/write to the disk using SCSI PR. The following PR commands are available when using Azure shared disks:

To interact with the disk, start with the persistent-reservation-action list:

```
PR_REGISTER_KEY
PR_REGISTER_AND_IGNORE
PR_GET_CONFIGURATION
PR_RESERVE
PR_PREEMPT_RESERVATION
PR_CLEAR_RESERVATION
PR_RELEASE_RESERVATION
```

When using PR\_RESERVE, PR\_PREEMPT\_RESERVATION, or PR\_RELEASE\_RESERVATION, provide one of the following persistent-reservation-type:

```
PR_NONE
PR_WRITE_EXCLUSIVE
PR_EXCLUSIVE_ACCESS
PR_WRITE_EXCLUSIVE_REGISTRANTS_ONLY
PR_EXCLUSIVE_ACCESS_REGISTRANTS_ONLY
PR_WRITE_EXCLUSIVE_ALL_REGISTRANTS
PR_EXCLUSIVE_ACCESS_ALL_REGISTRANTS
```

You also need to provide a persistent-reservation-key when using PR\_RESERVE, PR\_REGISTER\_AND\_IGNORE, PR\_REGISTER\_KEY, PR\_PREEMPT\_RESERVATION, PR\_CLEAR\_RESERVATION, or PR\_RELEASE-RESERVATION.

## Next steps

If you're interested in trying shared disks, [sign up for our preview](#).

# Upload a vhd to Azure using Azure PowerShell

12/20/2019 • 5 minutes to read • [Edit Online](#)

This article explains how to upload a vhd from your local machine to an Azure managed disk. Previously, you had to follow a more involved process that included staging your data in a storage account, and managing that storage account. Now, you no longer need to manage a storage account, or stage data in it to upload a vhd. Instead, you create an empty managed disk, and upload a vhd directly to it. This simplifies uploading on-premises VMs to Azure and enables you to upload a vhd up to 32 TiB directly into a large managed disk.

If you are providing a backup solution for IaaS VMs in Azure, we recommend you use direct upload to restore customer backups to managed disks. If you are uploading a VHD from a machine external to Azure, speeds will depend on your local bandwidth. If you are using an Azure VM, then your bandwidth will be the same as standard HDDs.

Currently, direct upload is supported for standard HDD, standard SSD, and premium SSD managed disks. It is not yet supported for ultra SSDs.

## Prerequisites

- Download the latest [version of AzCopy v10](#).
- [Install Azure PowerShell module](#).
- If you intend to upload a VHD from on-premises: A VHD that [has been prepared for Azure](#), stored locally.
- Or, a managed disk in Azure, if you intend to perform a copy action.

## Create an empty managed disk

To upload your vhd to Azure, you'll need to create an empty managed disk that is configured for this upload process. Before you create one, there's some additional information you should know about these disks.

This kind of managed disk has two unique states:

- ReadToUpload, which means the disk is ready to receive an upload but, no [secure access signature](#) (SAS) has been generated.
- ActiveUpload, which means that the disk is ready to receive an upload and the SAS has been generated.

While in either of these states, the managed disk will be billed at [standard HDD pricing](#), regardless of the actual type of disk. For example, a P10 will be billed as an S10. This will be true until `revoke-access` is called on the managed disk, which is required in order to attach the disk to a VM.

Before you create an empty standard HDD for uploading, you'll need the file size in bytes of the vhd you want to upload. The example code will get that for you but, to do it yourself you can use:

`$vhdSizeBytes = (Get-Item "<fullFilePathHere>").length`. This value is used when specifying the -

**UploadSizeInBytes** parameter.

Now, on your local shell, create an empty standard HDD for uploading by specifying the **Upload** setting in the - **CreateOption** parameter as well as the **-UploadSizeInBytes** parameter in the [New-AzDiskConfig](#) cmdlet. Then call [New-AzDisk](#) to create the disk:

```
$vhdSizeBytes = (Get-Item "<fullFilePathHere>").length

$diskconfig = New-AzDiskConfig -SkuName 'Standard_LRS' -OsType 'Windows' -UploadSizeInBytes $vhdSizeBytes -
Location 'West US' -CreateOption 'Upload'

New-AzDisk -ResourceGroupName 'myResourceGroup' -DiskName 'myDiskName' -Disk $diskconfig
```

If you would like to upload either a premium SSD or a standard SSD, replace **Standard\_LRS** with either **Premium\_LRS** or **StandardSSD\_LRS**. Ultra SSD is not yet supported.

You have now created an empty managed disk that is configured for the upload process. To upload a vhd to the disk, you'll need a writeable SAS, so that you can reference it as the destination for your upload.

To generate a writable SAS of your empty managed disk, use the following command:

```
$diskSas = Grant-AzDiskAccess -ResourceGroupName 'myResourceGroup' -DiskName 'myDiskName' -DurationInSecond
86400 -Access 'Write'

$disk = Get-AzDisk -ResourceGroupName 'myResourceGroup' -DiskName 'myDiskName'
```

## Upload vhd

Now that you have a SAS for your empty managed disk, you can use it to set your managed disk as the destination for your upload command.

Use AzCopy v10 to upload your local VHD file to a managed disk by specifying the SAS URI you generated.

This upload has the same throughput as the equivalent [standard HDD](#). For example, if you have a size that equates to S4, you will have a throughput of up to 60 MiB/s. But, if you have a size that equates to S70, you will have a throughput of up to 500 MiB/s.

```
AzCopy.exe copy "c:\somewhere\mydisk.vhd"$diskSas.AccessSAS --blob-type PageBlob
```

If your SAS expires during the upload, and you haven't called `revoke-access` yet, you can get a new SAS to continue the upload using `grant-access`, again.

After the upload is complete, and you no longer need to write any more data to the disk, revoke the SAS. Revoking the SAS will change the state of the managed disk and allow you to attach the disk to a VM.

```
Revoke-AzDiskAccess -ResourceGroupName 'myResourceGroup' -DiskName 'myDiskName'
```

## Copy a managed disk

Direct upload also simplifies the process of copying a managed disk. You can either copy within the same region or cross-region (to another region).

The follow script will do this for you, the process is similar to the steps described earlier, with some differences since you're working with an existing disk.

## IMPORTANT

You need to add an offset of 512 when you're providing the disk size in bytes of a managed disk from Azure. This is because Azure omits the footer when returning the disk size. The copy will fail if you do not do this. The following script already does this for you.

Replace the `<sourceResourceGroupHere>`, `<sourceDiskNameHere>`, `<targetDiskNameHere>`, `<targetResourceGroupHere>`, `<yourOSTypeHere>` and `<yourTargetLocationHere>` (an example of a location value would be uswest2) with your values, then run the following script in order to copy a managed disk.

```
$sourceRG = <sourceResourceGroupHere>
$sourceDiskName = <sourceDiskNameHere>
$targetDiskName = <targetDiskNameHere>
$targetRG = <targetResourceGroupHere>
$targetLocate = <yourTargetLocationHere>
#Expected value for OS is either "Windows" or "Linux"
$targetOS = <yourOSTypeHere>

$sourceDisk = Get-AzDisk -ResourceGroupName $sourceRG -DiskName $sourceDiskName

Adding the sizeInBytes with the 512 offset, and the -Upload flag
$targetDiskconfig = New-AzDiskConfig -SkuName 'Standard_LRS' -osType $targetOS -UploadSizeInBytes
$($sourceDisk.DiskSizeBytes+512) -Location $targetLocate -CreateOption 'Upload'

$targetDisk = New-AzDisk -ResourceGroupName $targetRG -DiskName $targetDiskName -Disk $targetDiskconfig

$sourceDiskSas = Grant-AzDiskAccess -ResourceGroupName $sourceRG -DiskName $sourceDiskName -DurationInSecond
86400 -Access 'Read'

$targetDiskSas = Grant-AzDiskAccess -ResourceGroupName $targetRG -DiskName $targetDiskName -DurationInSecond
86400 -Access 'Write'

azcopy copy $sourceDiskSas.AccessSAS $targetDiskSas.AccessSAS --blob-type PageBlob

Revoke-AzDiskAccess -ResourceGroupName $sourceRG -DiskName $sourceDiskName

Revoke-AzDiskAccess -ResourceGroupName $targetRG -DiskName $targetDiskName
```

## Next steps

Now that you've successfully uploaded a vhd to a managed disk, you can attach your disk to a VM and begin using it.

To learn how to attach a data disk to a VM, see our article on the subject: [Attach a data disk to a Windows VM with PowerShell](#). To use the disk as the OS disk, see [Create a Windows VM from a specialized disk](#).

# How to expand the OS drive of a virtual machine

11/13/2019 • 4 minutes to read • [Edit Online](#)

When you create a new virtual machine (VM) in a Resource Group by deploying an image from [Azure Marketplace](#), the default OS drive is often 127 GB (some images have smaller OS disk sizes by default). Even though it's possible to add data disks to the VM (how many depending upon the SKU you've chosen) and moreover it's recommended to install applications and CPU intensive workloads on these addendum disks, oftentimes customers need to expand the OS drive to support certain scenarios such as following:

- Support legacy applications that install components on OS drive.
- Migrate a physical PC or virtual machine from on-premises with a larger OS drive.

## IMPORTANT

Resizing the OS Disk of an Azure Virtual Machine requires the virtual machine to be deallocated.

After expanding the disks, you need to [expand the volume within the OS](#) to take advantage of the larger disk.

## Resize a managed disk

Open your Powershell ISE or Powershell window in administrative mode and follow the steps below:

1. Sign in to your Microsoft Azure account in resource management mode and select your subscription as follows:

```
Connect-AzAccount
Select-AzSubscription -SubscriptionName 'my-subscription-name'
```

2. Set your resource group name and VM name as follows:

```
$rgName = 'my-resource-group-name'
$vmName = 'my-vm-name'
```

3. Obtain a reference to your VM as follows:

```
$vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
```

4. Stop the VM before resizing the disk as follows:

```
Stop-AzVM -ResourceGroupName $rgName -Name $vmName
```

5. Obtain a reference to the managed OS disk. Set the size of the managed OS disk to the desired value and update the Disk as follows:

```
$disk= Get-AzDisk -ResourceGroupName $rgName -DiskName $vm.StorageProfile.OsDisk.Name
$disk.DiskSizeGB = 1023
Update-AzDisk -ResourceGroupName $rgName -Disk $disk -DiskName $disk.Name
```

**WARNING**

The new size should be greater than the existing disk size. The maximum allowed is 2048 GB for OS disks. (It is possible to expand the VHD blob beyond that size, but the OS will only be able to work with the first 2048 GB of space.)

6. Updating the VM may take a few seconds. Once the command finishes executing, restart the VM as follows:

```
Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

And that's it! Now RDP into the VM, open Computer Management (or Disk Management) and expand the drive using the newly allocated space.

## Resize an unmanaged disk

Open your Powershell ISE or Powershell window in administrative mode and follow the steps below:

1. Sign in to your Microsoft Azure account in resource management mode and select your subscription as follows:

```
Connect-AzAccount
Select-AzSubscription -SubscriptionName 'my-subscription-name'
```

2. Set your resource group name and VM name as follows:

```
$rgName = 'my-resource-group-name'
$vmName = 'my-vm-name'
```

3. Obtain a reference to your VM as follows:

```
$vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
```

4. Stop the VM before resizing the disk as follows:

```
Stop-AzVM -ResourceGroupName $rgName -Name $vmName
```

5. Set the size of the unmanaged OS disk to the desired value and update the VM as follows:

```
$vm.StorageProfile.OSDisk.DiskSizeGB = 1023
Update-AzVM -ResourceGroupName $rgName -VM $vm
```

**WARNING**

The new size should be greater than the existing disk size. The maximum allowed is 2048 GB for OS disks. (It is possible to expand the VHD blob beyond that size, but the OS will only be able to work with the first 2048 GB of space.)

6. Updating the VM may take a few seconds. Once the command finishes executing, restart the VM as follows:

```
Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

## Scripts for OS disk

Below is the complete script for your reference for both managed and unmanaged disks:

### Managed disks

```
Connect-AzAccount
Select-AzSubscription -SubscriptionName 'my-subscription-name'
$rgName = 'my-resource-group-name'
$vmName = 'my-vm-name'
$vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
Stop-AzVM -ResourceGroupName $rgName -Name $vmName
$disk= Get-AzDisk -ResourceGroupName $rgName -DiskName $vm.StorageProfile.OsDisk.Name
$disk.DiskSizeGB = 1023
Update-AzDisk -ResourceGroupName $rgName -Disk $disk -DiskName $disk.Name
Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

### Unmanaged disks

```
Connect-AzAccount
Select-AzSubscription -SubscriptionName 'my-subscription-name'
$rgName = 'my-resource-group-name'
$vmName = 'my-vm-name'
$vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
Stop-AzVM -ResourceGroupName $rgName -Name $vmName
$vm.StorageProfile.OSDisk.DiskSizeGB = 1023
Update-AzVM -ResourceGroupName $rgName -VM $vm
Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

## Resizing data disks

This article is focused primarily on expanding the OS disk of the VM, but the script can also be used for expanding the data disks attached to the VM. For example, to expand the first data disk attached to the VM, replace the `OSDisk` object of `StorageProfile` with `DataDisks` array and use a numeric index to obtain a reference to first attached data disk, as shown below:

### Managed disk

```
$disk= Get-AzDisk -ResourceGroupName $rgName -DiskName $vm.StorageProfile.DataDisks[0].Name
$disk.DiskSizeGB = 1023
```

### Unmanaged disk

```
$vm.StorageProfile.DataDisks[0].DiskSizeGB = 1023
```

Similarly you may reference other data disks attached to the VM, either by using an index as shown above or the `Name` property of the disk:

### Managed disk

```
(Get-AzDisk -ResourceGroupName $rgName -DiskName $($vm.StorageProfile.DataDisks | Where {$_.Name -eq 'my-second-data-disk'}).Name).DiskSizeGB = 1023
```

## Unmanaged disk

```
($vm.StorageProfile.DataDisks | Where ($_.Name -eq 'my-second-data-disk')).DiskSizeGB = 1023
```

## Expand the volume within the OS

Once you have expanded the disk for the VM, you need to go into the OS and expand the volume to encompass the new space. There are several methods for expanding a partition. This section covers connecting the VM using an RDP connection to expand the partition using **DiskPart**.

1. Open an RDP connection to your VM.
2. Open a command prompt and type **diskpart**.
3. At the **DISKPART** prompt, type `list volume`. Make note of the volume you want to extend.
4. At the **DISKPART** prompt, type `select volume <volumenumber>`. This selects the volume *volumenumber* that you want to extend into contiguous, empty space on the same disk.
5. At the **DISKPART** prompt, type `extend [size=<size>]`. This extends the selected volume by *size* in megabytes (MB).

## Next steps

You can also attach disks using the [Azure portal](#).

# Use Azure Storage Explorer to manage Azure managed disks

11/12/2019 • 2 minutes to read • [Edit Online](#)

Storage Explorer 1.10.0 enables users to upload, download, and copy managed disks, as well as create snapshots. Because of these additional capabilities, you can use Storage Explorer to migrate data from on-premises to Azure, and migrate data across Azure regions.

## Prerequisites

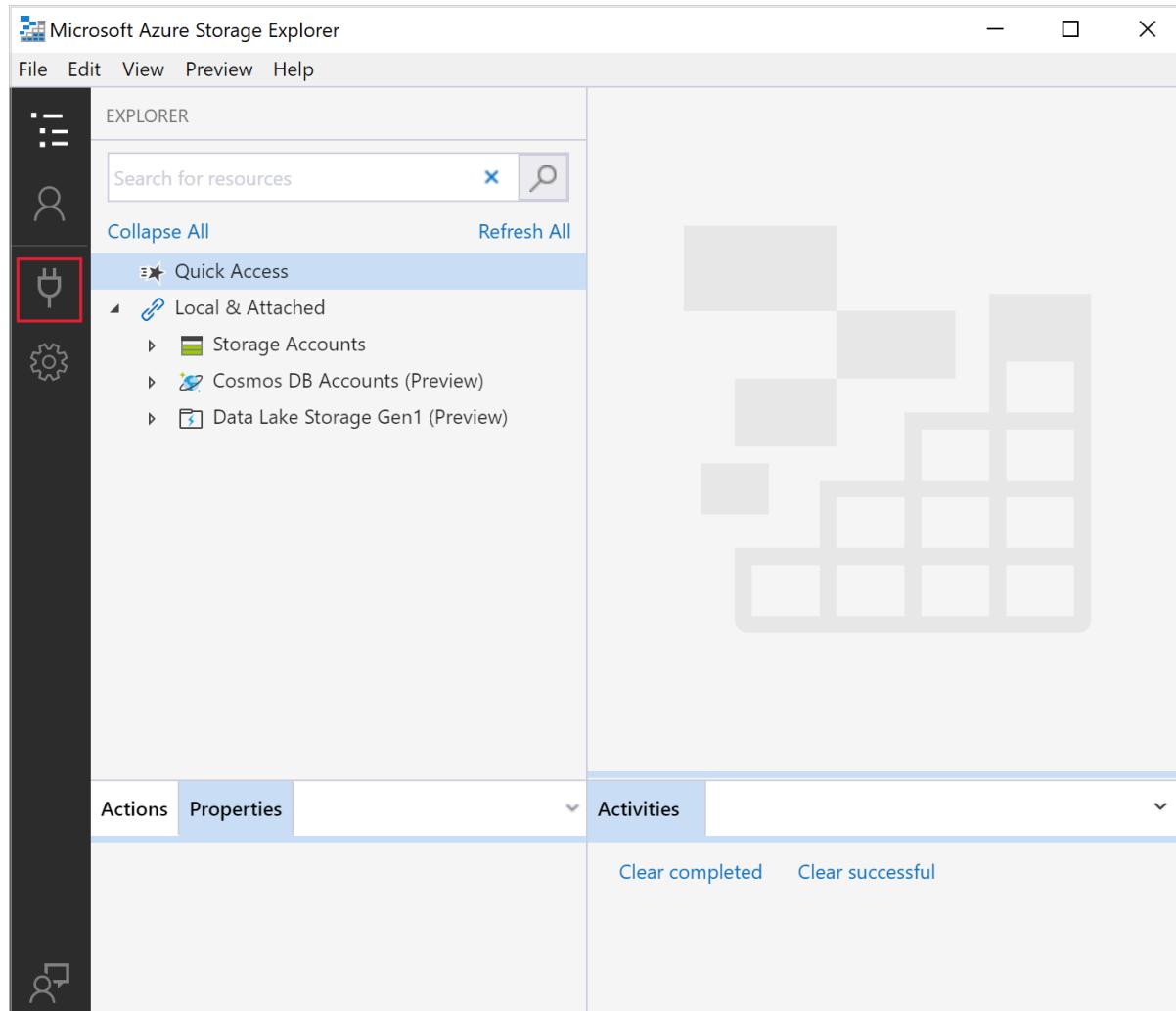
To complete this article, you'll need the following:

- An Azure subscription
- One or more Azure managed disks
- The latest version of [Azure Storage Explorer](#)

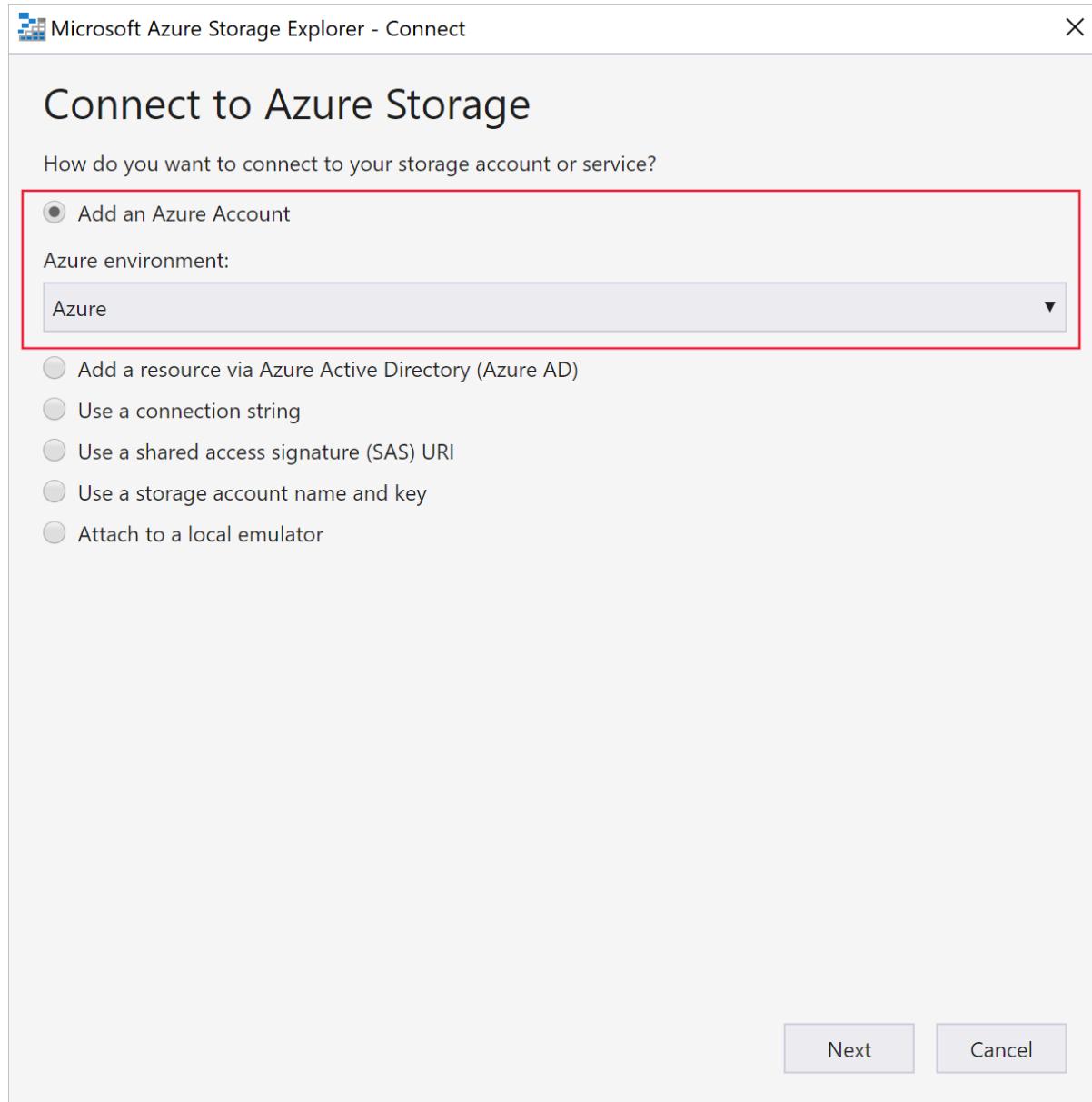
## Connect to an Azure subscription

If your Storage Explorer isn't connected to Azure, you will not be able to use it to manage resources. This section goes over connecting it to your Azure account so that you can manage resources using Storage Explorer.

1. Launch Azure Storage Explorer and click the **plug-in** icon on the left.



2. Select **Add an Azure Account**, and then click **Next**.



3. In the **Azure Sign in** dialog box, enter your Azure credentials.



# Sign in

Email, phone, or Skype

No account? [Create one!](#)

Can't access your account?

[Sign-in options](#)

Next

4. Select your subscription from the list and then click **Apply**.

Microsoft Azure Storage Explorer

File Edit View Preview Help

ACCOUNT MANAGEMENT

Show resources from these subscriptions:

|                                                                                         |        |
|-----------------------------------------------------------------------------------------|--------|
| Microsoft<br>contoso@microsoft.com                                                      | Remove |
| <input type="checkbox"/> All subscriptions                                              |        |
| <input type="checkbox"/> contoso subscription 1<br>00000000-0000-0000-000000000000      |        |
| <input type="checkbox"/> contoso subscription 2<br>11111111-1111-1111-1111-111111111111 |        |

Add an account...

Activities

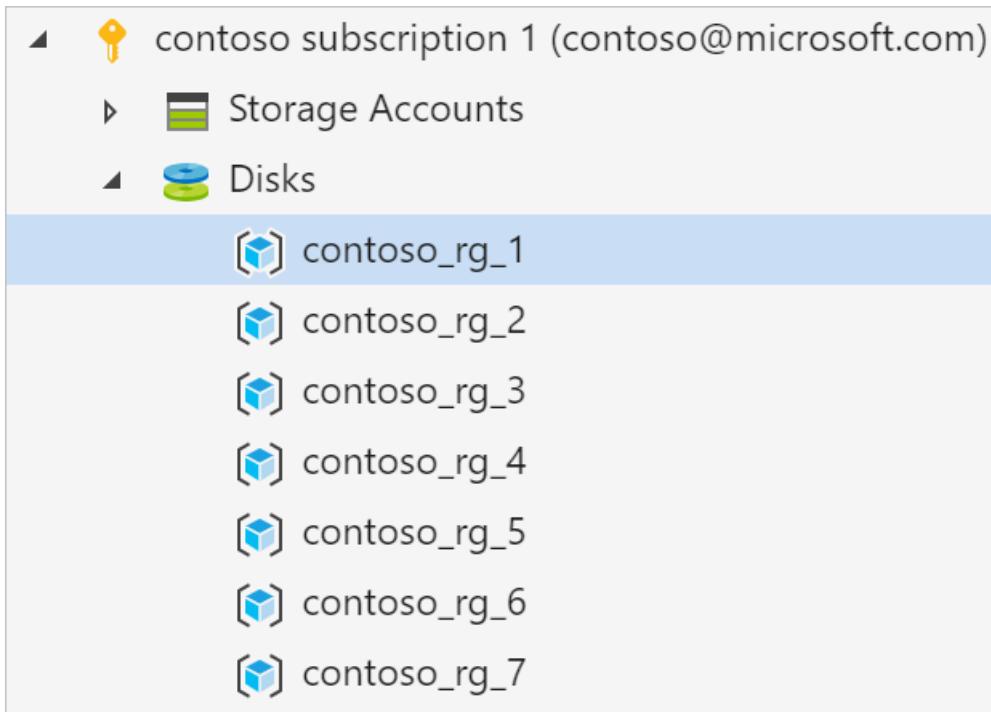
Clear completed Clear successful

Apply Cancel



## Upload a managed disk from an on-prem VHD

1. On the left pane, expand **Disks** and select the resource group that you want to upload your disk to.



2. Select **Upload**.

The screenshot shows the "contoso\_rg\_1" resource group details page. The top navigation bar includes "contoso\_rg\_1", a refresh icon, and a close button. Below the bar is a toolbar with "Upload" (highlighted with a red box), "Download", "Copy", "Paste", "Delete", "Create Snapshot", and "Refresh" buttons. A table below lists disks with columns: Disk Name, SKU, Size, Disk State, Owner VM, and Location. The message "No disks found" is displayed.

3. In **Upload VHD** specify your source VHD, the name of the disk, the OS type, the region you want to upload the disk to, as well as the account type. In some regions Availability zones are supported, for those regions you can select a zone of your choice.
4. Select **Create** to begin uploading your disk.

 Microsoft Azure Storage Explorer - Upload VHD X

## Upload VHD

Select a VHD from which to create a disk and then specify the remaining parameters.

Source VHD:

 ...

Disk name:

OS type:

 ▼

Region:

 ▼

Availability zone:

 ▼

Account type:

 ▼

Create Cancel

5. The status of the upload will now display in **Activities**.

Activities

[Clear completed](#) [Clear successful](#)

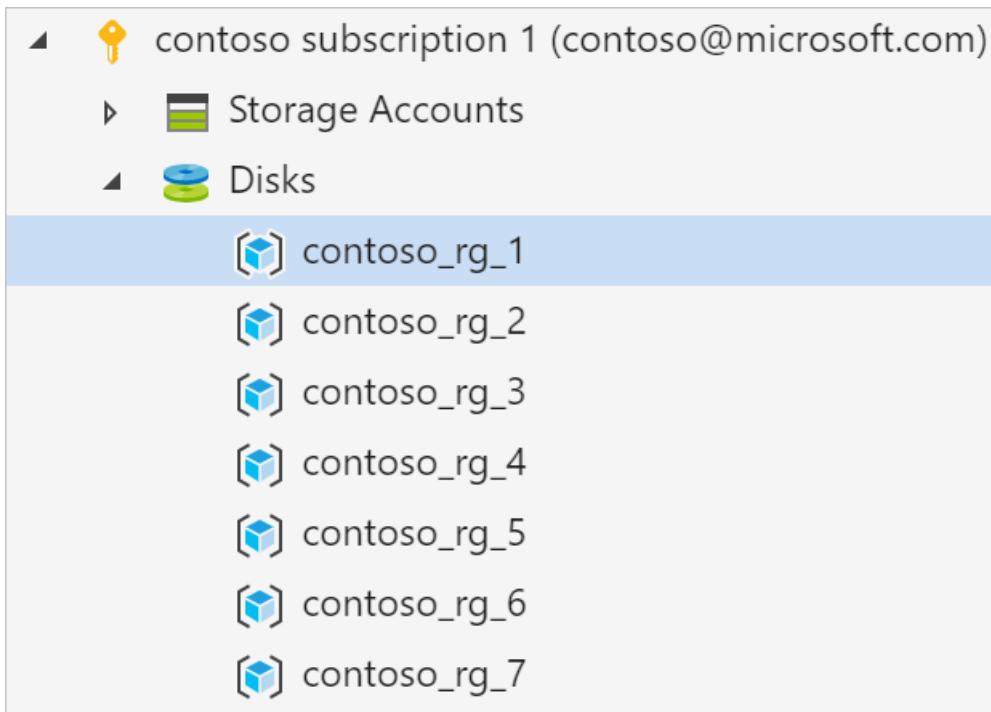
 Uploading 'C:\Users\Administrator\Downloads\mydisk\_onprem.vhd' to disk 'mydisk\_azure' in resource group 'contoso\_rg\_1'

6. If the upload has finished and you don't see the disk in the right pane, select **Refresh**.

## Download a managed disk

The following steps explain how to download a managed disk to an on-prem VHD. A disk's state must be **Unattached** in order to be downloaded, you cannot download an **Attached** disk.

1. On the left pane, if it isn't already expanded, expand **Disks** and select the resource group that you want to download your disk from.



2. On the right pane, select the disk you want to download.
3. Select **Download** and then choose where you would like to save the disk.

The screenshot shows the Azure portal's 'Disks' blade for the 'contoso\_rg\_1' resource group. The top navigation bar includes 'Upload', 'Download' (which is highlighted with a red box), 'Copy', 'Paste', 'Delete', 'Create Snapshot', and 'Refresh' buttons. Below the navigation bar is a table with columns: Disk Name, SKU, Size, Disk State, Owner VM, and Location. A single row is selected, showing 'mydisk\_azure' as the disk name, 'Premium' as the SKU, '32 GB' as the size, 'Unattached' as the disk state, 'centralus' as the location, and no owner VM listed. At the bottom of the table, it says 'Showing 1 to 1 of 1 discovered disks'.

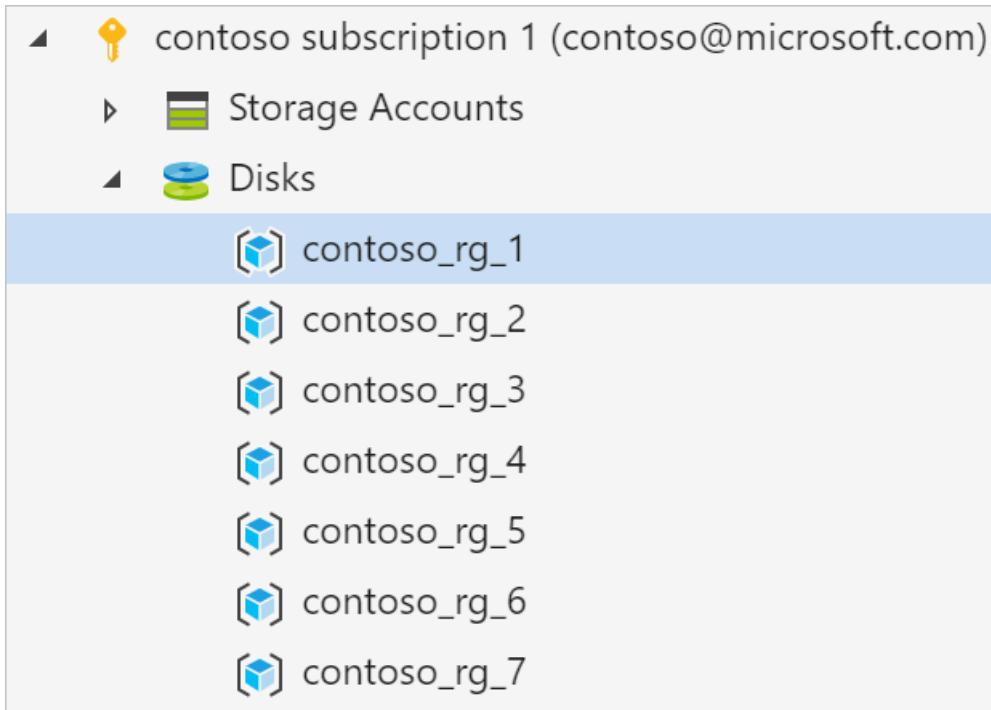
4. Select **Save** and your disk will begin downloading. The status of the download will display in **Activities**.

The screenshot shows the 'Activities' blade. It has tabs for 'Activities' (which is selected and highlighted in blue) and 'Jobs'. Below the tabs are 'Clear completed' and 'Clear successful' buttons. A single activity is listed: 'Downloading disk 'mydisk\_azure' in resource group 'contoso\_rg\_1' to 'C:\Users\Administrator\Downloads\mydisk\_download\_from\_azure.vhd''. The status of the activity is shown as a blue circular icon followed by the task description.

## Copy a managed disk

With Storage Explorer, you can copy a managed disk within or across regions. To copy a disk:

1. From the **Disks** dropdown on the left, select the resource group that contains the disk you want to copy.

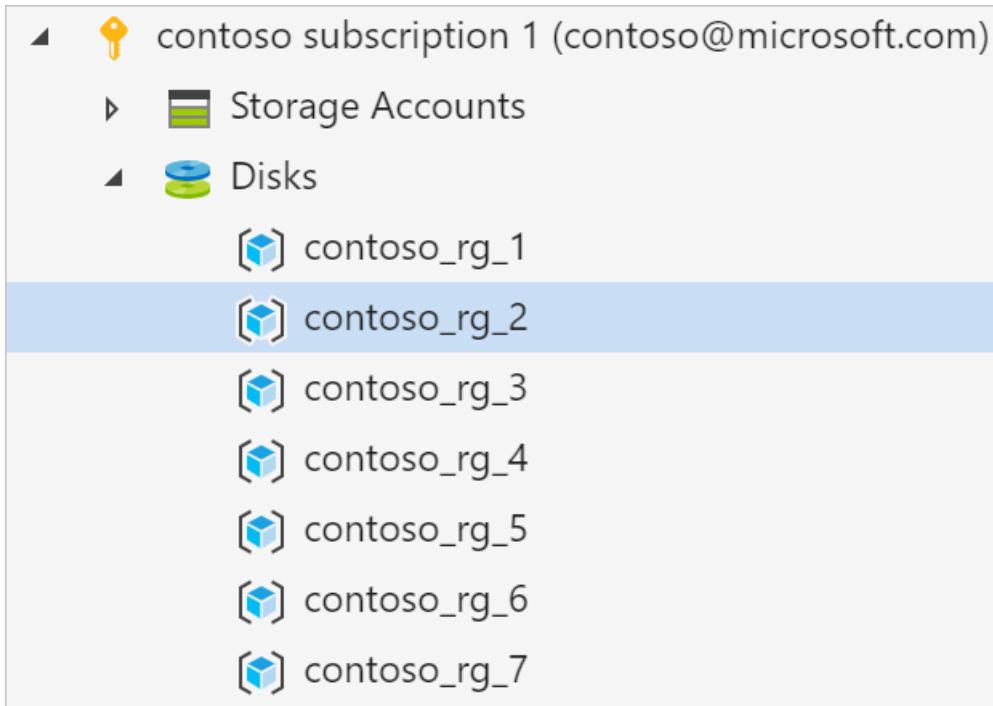


2. On the right pane, select the disk you'd like to copy and select **Copy**.

| Disk Name    | SKU     | Size  | Disk State | Owner VM | Location  |
|--------------|---------|-------|------------|----------|-----------|
| mydisk_azure | Premium | 32 GB | Unattached |          | centralus |

Showing 1 to 1 of 1 discovered disks

3. On the left pane, select the resource group you'd like to paste the disk in.



4. Select **Paste** on the right pane.

A screenshot of the 'Disks' blade for the resource group 'contoso\_rg\_2'. The title bar shows 'contoso\_rg\_2'. The toolbar includes buttons for Upload, Download, Copy, Paste (which is highlighted with a red box), Delete, Create Snapshot, and Refresh. Below the toolbar is a table header with columns: Disk Name, SKU, Size, Disk State, Owner VM, and Location. A message 'No disks found' is displayed below the table. At the bottom, it says 'Showing 0 to 0 of 0 entries'.

5. In the **Paste Disk** dialog, fill in the values. You can also specify an Availability zone in supported regions.



## Paste Disk

Choose the name for the disk and and then specify the remaining parameters.

Disk name:

Region:



Availability zone:



Account type:



6. Select **Paste** and your disk will begin copying, the status is displayed in **Activities**.

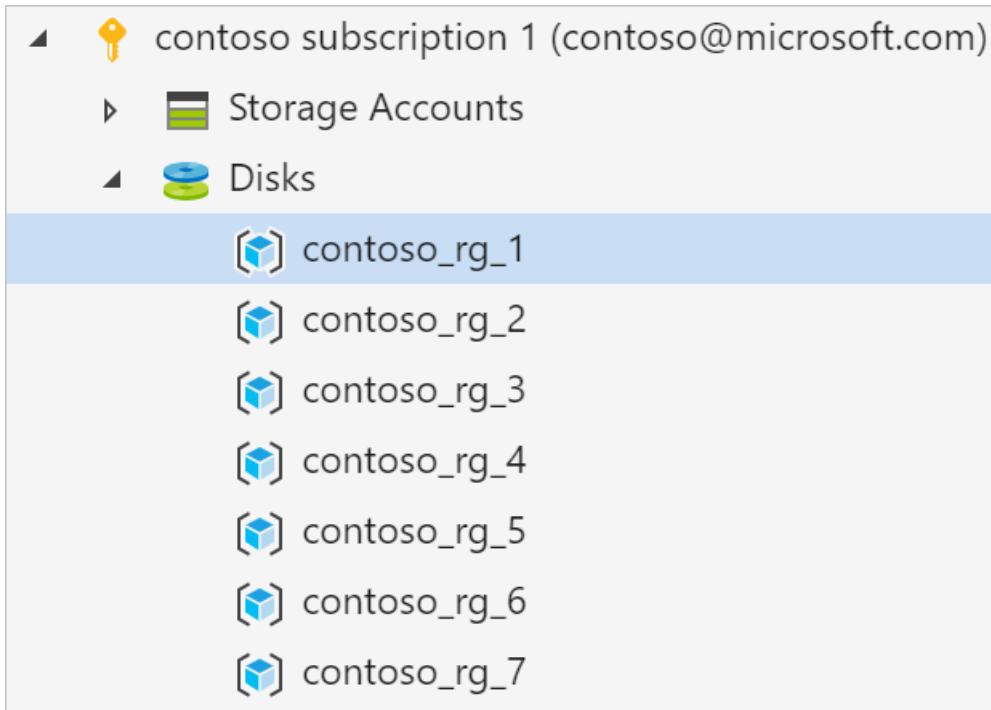
Activities

Clear completed   Clear successful

🕒 Copying disk 'mydisk\_azure' in resource group 'contoso\_rg\_1' to disk 'mydisk\_azure\_paste' in resource group 'contoso\_rg\_2'

## Create a snapshot

1. From the **Disks** dropdown on the left, select the resource group that contains the disk you want to snapshot.



2. On the right, select the disk you'd like to snapshot and select **Create Snapshot**.

This screenshot shows the 'contoso\_rg\_1' disk details page. At the top, there is a toolbar with icons for Upload, Download, Copy, Paste, Delete, and Create Snapshot (which is highlighted with a red box). Below the toolbar is a table with columns: Disk Name, SKU, Size, Disk State, Owner VM, and Location. One row is visible, showing 'mydisk\_azure' with a Premium SKU, 32 GB size, Unattached state, no owner VM, and centralus location. At the bottom of the page, it says 'Showing 1 to 1 of 1 discovered disks'.

| Disk Name    | SKU     | Size  | Disk State | Owner VM | Location  |
|--------------|---------|-------|------------|----------|-----------|
| mydisk_azure | Premium | 32 GB | Unattached |          | centralus |

3. In **Create Snapshot**, specify the name of the snapshot as well as the resource group you want to create it in. Then select **Create**.



## Create Snapshot

Specify the name of the snapshot and what resource group to create it in.

Snapshot name:

Resource group:

- Once the snapshot has been created, you can select **Open in Portal** in **Activities** to view the snapshot in the Azure portal.

Activities

Clear completed   Clear successful

Successfully created snapshot 'mydisk\_azure\_snapshot\_20190901' from disk 'mydisk\_azure' in resource group 'contoso\_rg\_1'

[Open in Portal](#)

## Next steps

Learn how to [Create a VM from a VHD by using the Azure portal](#).

Learn how to [Attach a managed data disk to a Windows VM by using the Azure portal](#).

# Create a snapshot

12/23/2019 • 2 minutes to read • [Edit Online](#)

A snapshot is a full, read-only copy of a virtual hard drive (VHD). You can take a snapshot of an OS or data disk VHD to use as a backup, or to troubleshoot virtual machine (VM) issues.

If you are going to use the snapshot to create a new VM, we recommend that you cleanly shut down the VM before taking a snapshot, to clear out any processes that are in progress.

## Use the Azure portal

To create a snapshot, complete the following steps:

1. On the [Azure portal](#), select **Create a resource**.
2. Search for and select **Snapshot**.
3. In the **Snapshot** window, select **Create**. The **Create snapshot** window appears.
4. Enter a **Name** for the snapshot.
5. Select an existing [Resource group](#) or enter the name of a new one.
6. Select an Azure datacenter **Location**.
7. For **Source disk**, select the managed disk to snapshot.
8. Select the **Account type** to use to store the snapshot. Select **Standard\_HDD**, unless you need the snapshot to be stored on a high-performing disk.
9. Select **Create**.

## Use PowerShell

The following steps show how to copy the VHD disk and create the snapshot configuration. You can then take a snapshot of the disk by using the [New-AzSnapshot](#) cmdlet.

1. Set some parameters:

```
$resourceGroupName = 'myResourceGroup'
$location = 'eastus'
$vmName = 'myVM'
$snapshotName = 'mySnapshot'
```

2. Get the VM:

```
$vm = get-azvm `br/>-ResourceGroupName $resourceGroupName
-Name $vmName
```

3. Create the snapshot configuration. For this example, the snapshot is of the OS disk:

```
$snapshot = New-AzSnapshotConfig
-SourceUri $vm.StorageProfile.OsDisk.ManagedDisk.Id
-Location $location
-CreateOption copy
```

**NOTE**

If you would like to store your snapshot in zone-resilient storage, create it in a region that supports [availability zones](#) and include the `-SkuName Standard_ZRS` parameter.

4. Take the snapshot:

```
New-AzSnapshot
-Snapshot $snapshot
-SnapshotName $snapshotName
-ResourceGroupName $resourceGroupName
```

## Next steps

Create a virtual machine from a snapshot by creating a managed disk from a snapshot and then attaching the new managed disk as the OS disk. For more information, see the sample in [Create a VM from a snapshot with PowerShell](#).

# Reduce costs with Azure Disks Reservation

1/30/2020 • 6 minutes to read • [Edit Online](#)

Save on your Azure Disk Storage usage with reserved capacity. Azure Disk Storage reservations combined with Azure Reserved Virtual Machine Instances let you lower your total virtual machine (VM) costs. The reservation discount is applied automatically to the matching disks in the selected reservation scope. Because of this automatic application, you don't need to assign a reservation to a managed disk to get the discounts.

Discounts are applied hourly depending on the disk usage. Unused reserved capacity doesn't carry over. Azure Disk Storage reservation discounts don't apply to unmanaged disks, ultra disks, or page blob consumption.

## Determine your storage needs

Before you purchase a reservation, determine your storage needs. Currently, Azure Disk Storage reservations are available only for select Azure premium SSD SKUs. The SKU of a premium SSD determines the disk's size and performance.

When determining your storage needs, don't think of disks based on just capacity. For example, you can't have a reservation for a P40 disk and use that to pay for two smaller P30 disks. When purchasing a reservation, you're only purchasing a reservation for the total number of disks per SKU.

A disk reservation is made per disk SKU. As a result, the reservation consumption is based on the unit of the disk SKUs instead of the provided size.

For example, assume you reserve one P40 disk that has 2 TiB of provisioned storage capacity. Also assume you allocate only two P30 disks. The P40 reservation in that case doesn't account for P30 consumption, and you pay the pay-as-you-go rate on the P30 disks.

| PRE<br>MIU<br>M<br>SSD<br>SIZE<br>S | P1*        | P2*        | P3*        | P4         | P6         | P10         | P15         | P20         | P30         | P40         | P50         | P60         | P70         | P80         |
|-------------------------------------|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Disk size in GiB                    | 4          | 8          | 16         | 32         | 64         | 128         | 256         | 512         | 1,024       | 2,048       | 4,096       | 8,192       | 16,384      | 32,767      |
| IOP S per disk                      | 120        | 120        | 120        | 120        | 240        | 500         | 1,100       | 2,300       | 5,000       | 7,500       | 7,500       | 16,000      | 18,000      | 20,000      |
| Throughput per disk                 | 25 MiB/sec | 25 MiB/sec | 25 MiB/sec | 25 MiB/sec | 50 MiB/sec | 100 MiB/sec | 125 MiB/sec | 150 MiB/sec | 200 MiB/sec | 250 MiB/sec | 250 MiB/sec | 500 MiB/sec | 750 MiB/sec | 900 MiB/sec |

| PRE<br>MIU<br>M<br>SSD<br>SIZE<br>S | P1*          | P2*          | P3*          | P4           | P6           | P10          | P15          | P20                 | P30                 | P40                 | P50                 | P60                 | P70                 | P80                 |
|-------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Max burst IOP S per disk **         | 3,500        | 3,500        | 3,500        | 3,500        | 3,500        | 3,500        | 3,500        | 3,500               |                     |                     |                     |                     |                     |                     |
| Max burst throughput per disk **    | 170 MiB /sec        |                     |                     |                     |                     |                     |                     |
| Max burst duration**                | 30 min              |                     |                     |                     |                     |                     |                     |
| Eligible for reservation            | No           | Yes, up to one year |

\*Denotes a disk size that is currently in preview, for regional availability information see [New disk sizes: Managed and unmanaged](#).

\*\*Denotes a feature that is currently in preview, see [Disk bursting](#) for more information.

## Purchase considerations

We recommend the following practices when considering disk reservation purchase:

- Analyze your usage information to help determine which reservations you should purchase. Make sure you track the usage in disk SKUs instead of provisioned or used disk capacity.
- Examine your disk reservation along with your VM reservation. We highly recommend making reservations for both VM usage and disk usage for maximum savings. You can start with determining the right VM reservation and then evaluate the disk reservation. Generally, you'll have a standard configuration for each of your workloads. For example, a SQL Server server might have two P40 data disks and one P30 operating system disk.

This kind of pattern can help you determine the reserved amount you might purchase. This approach can simplify the evaluation process and ensure that you have an aligned plan for both your VM and disks. The plan contains considerations like subscriptions or regions.

# Purchase restrictions

Reservation discounts are currently unavailable for the following:

- Unmanaged disks or page blobs.
- Standard SSDs or standard hard-disk drives (HDDs).
- Premium SSD SKUs smaller than P30: P1, P2, P3, P4, P6, P10, P15, and P20 SSD SKUs.
- Disks in Azure Government, Azure Germany, or Azure China regions.

In rare circumstances, Azure limits the purchase of new reservations to a subset of disk SKUs because of low capacity in a region.

## Buy a disk reservation

You can purchase Azure Disk Storage reservations through the [Azure portal](#). You can pay for the reservation either up front or with monthly payments. For more information about purchasing with monthly payments, see [Purchase reservations with monthly payments](#).

Follow these steps to purchase reserved capacity:

1. Go to the [Purchase reservations](#) pane in the Azure portal.
2. Select **Azure Managed Disks** to purchase a reservation.

The screenshot shows the 'Purchase reservations' page in the Azure portal. It lists several services with their respective icons and descriptions, each featuring a 'Buy' button. The 'Azure Managed Disks' section is highlighted with a red border.

| Service                       | Description                                                                                                                             | Buy Button |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|------------|
| Virtual machine               | Save on virtual machine costs by buying reserved instances for 1 or 3 years                                                             | Buy        |
| SQL Database                  | Save on SQL Database compute costs by buying reserved vCores for 1 or 3 years                                                           | Buy        |
| Azure SQL Data Warehouse      | Save up to 65% on SQL Data Warehouse costs by buying reserved capacity for 1 or 3 years                                                 | Buy        |
| Azure Cosmos DB               | Save up to 65% on Cosmos DB by buying reserved throughput capacity for 1 or 3 years                                                     | Buy        |
| Azure Blob Storage            | Save on Azure Storage costs for Block Blobs and Azure Data Lake Storage by buying Azure Blob Storage Reserved Capacity for 1 or 3 years | Buy        |
| Azure Database for MySQL      | Save on Azure Database for MySQL compute costs by buying reserved vCores for 1 year                                                     | Buy        |
| Azure Database for MariaDB    | Save on Azure Database for MariaDB compute costs by buying reserved vCores for 1 year                                                   | Buy        |
| Azure Database for PostgreSQL | Save on Azure Database for PostgreSQL single server compute costs by buying reserved vCores for 1 year                                  | Buy        |
| Azure Managed Disks           | Save on Premium SSD Managed Disks by buying reserved disks for 1 year                                                                   | Buy        |
| Azure Databricks              | Save on your Azure Databricks costs by pre-purchasing DBUs for 1 or 3 years                                                             | Buy        |

3. Specify the required values described in the following table:

| ELEMENT | DESCRIPTION |
|---------|-------------|
|---------|-------------|

| ELEMENT                  | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scope</b>             | <p>How many subscriptions can use the billing benefit associated with the reservation. This value also specifies how the reservation is applied to specific subscriptions.</p> <p>If you select <b>Shared</b>, the reservation discount is applied to Azure Storage capacity in every subscription within your billing context. The billing context is based on how you signed up for Azure. For enterprise customers, the shared scope is the enrollment and includes all subscriptions within the enrollment. For pay-as-you-go customers, the shared scope includes all individual subscriptions with pay-as-you-go rates created by the account administrator.</p> <p>If you select <b>Single subscription</b>, the reservation discount is applied to Azure Storage capacity in the selected subscription.</p> <p>If you select <b>Single resource group</b>, the reservation discount is applied to Azure Storage capacity in the selected subscription and in that subscription's selected resource group.</p> <p>You can change the reservation scope after you purchase the reservation.</p> |
| <b>Subscription</b>      | <p>The subscription you use to pay for the Azure Storage reservation. The payment method on the selected subscription is used in charging the costs. The subscription must be one of the following types:</p> <ul style="list-style-type: none"> <li>• Enterprise Agreement (offer numbers MS-AZR-0017P and MS-AZR-0148P). For an Enterprise subscription, the charges are deducted from the enrollment's monetary commitment balance or charged as overage.</li> <li>• Individual subscription with pay-as-you-go rates (offer numbers MS-AZR-0003P and MS-AZR-0023P). For an individual subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Disks</b>             | The SKU you want to create.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Region</b>            | The region where the reservation is in effect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Billing frequency</b> | How often the account is billed for the reservation. Options include <b>Monthly</b> and <b>Upfront</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

## Select the product you want to purchase

X

Save on your Premium SSD Managed Disks usage by purchasing reserved capacity. Discounts are applied hourly on the disk usage, any unused reserved capacity does not carry over. Reservation discount does not apply to Premium SSD Unmanaged Disks or Page Blob consumption. [Learn More](#)

Scope \*  Subscription \*

Filter by name... Region : **West Europe** Disk : **Select a value** Billing frequency : **Select a value**

| ↑↓ Name                   | ↑↓ Disk | ↑↓ Region   | ↑↓ Term  | ↑↓ Billing frequency |
|---------------------------|---------|-------------|----------|----------------------|
| Premium SSD Managed Disks | P30     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P30     | West Europe | One Year | Monthly              |
| Premium SSD Managed Disks | P40     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P40     | West Europe | One Year | Monthly              |
| Premium SSD Managed Disks | P50     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P50     | West Europe | One Year | Monthly              |
| Premium SSD Managed Disks | P60     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P60     | West Europe | One Year | Monthly              |
| Premium SSD Managed Disks | P70     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P70     | West Europe | One Year | Monthly              |
| Premium SSD Managed Disks | P80     | West Europe | One Year | Upfront              |
| Premium SSD Managed Disks | P80     | West Europe | One Year | Monthly              |

- After you specify the values for your reservation, the Azure portal displays the cost. The portal also shows the discount percentage over pay-as-you-go billing. Select **Next** to continue to the **Purchase reservations** pane.
- On the **Purchase reservations** pane, you can name your reservation and select the total quantity of reservations you want to make. The number of reservations maps to the number of disks. For example, if you want to reserve a hundred disks, enter the **Quantity** value **100**.
- Review the total cost of the reservation.

Home > Reservations > Purchase reservations

Purchase reservations

Products

Azure Managed Disks

| ↑↓ Reservation name      | ↑↓ Product                                               | ↑↓ Sco... ↑↓ | Unit price↑↓ | Quantity↑↓                     | Subtotal (% Discount)↑↓ | Billing frequency↑↓ |
|--------------------------|----------------------------------------------------------|--------------|--------------|--------------------------------|-------------------------|---------------------|
| Disk_RI_01-14-2020_10-31 | Premium SSD Managed Disks   P30   West Europe   One Year | Shared       | <price>      | <input type="text" value="1"/> | <subtotal>              | Upfront             |

Total reservation cost <total-cost>

After you purchase a reservation, it's automatically applied to any existing Disk Storage resources that match the reservation terms. If you haven't created any Disk Storage resources yet, the reservation applies whenever you create a resource that matches the reservation terms. In either case, the reservation term begins immediately after a successful purchase.

## Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations within certain limitations. For more information, see [Self-service exchanges and refunds for Azure Reservations](#).

## Expiration of a reservation

When a reservation expires, any Azure Disk Storage capacity that you use under that reservation is billed at the pay-as-you-go rate. Reservations don't renew automatically.

You'll receive an email notification 30 days before the expiration of the reservation and again on the expiration date. To continue taking advantage of the cost savings that a reservation provides, renew it no later than the expiration date.

## Need help? Contact us

If you have questions or need help, [create a support request](#).

## Next steps

- [What are Azure Reservations?](#)
- [Understand how your reservation discount is applied to Azure Disk Storage](#)

# Creating an incremental snapshot (preview) for managed disks

11/13/2019 • 5 minutes to read • [Edit Online](#)

Incremental snapshots (preview) are point in time backups for managed disks that, when taken, consist only of all the changes since the last snapshot. When you attempt to download or otherwise use an incremental snapshot, the full VHD is used. This new capability for managed disk snapshots can potentially allow them to be more cost effective, since you are no longer required to store the entire disk with each individual snapshot, unless you choose to. Just like regular snapshots, incremental snapshots can be used to create a full managed disk or, to make a regular snapshot.

There are a few differences between an incremental snapshot and a regular snapshot. Incremental snapshots will always use standard HDDs storage, irrespective of the storage type of the disk, whereas regular snapshots can use premium SSDs. If you are using regular snapshots on Premium Storage to scale up VM deployments, we recommend you use custom images on standard storage in the [Shared Image Gallery](#). It will help you to achieve a more massive scale with lower cost. Additionally, incremental snapshots potentially offer better reliability with [zone-redundant storage](#) (ZRS). If ZRS is available in the selected region, an incremental snapshot will use ZRS automatically. If ZRS is not available in the region, then the snapshot will default to [locally-redundant storage](#) (LRS). You can override this behavior and select one manually but, we do not recommend that.

Incremental snapshots also offer a differential capability, which is uniquely available to managed disks. They enable you to get the changes between two incremental snapshots of the same managed disks, down to the block level. You can use this capability to reduce your data footprint when copying snapshots across regions.

## Supported regions

Only the following regions are currently supported:

- Available as a GA offering in the West Central US, Canada East, Canada Central regions.
- Available as a public preview in the East US, East US 2, Central US, North Europe, South East Asia regions.

## Restrictions

- Incremental snapshots currently cannot be created after you've changed the size of a disk (during preview only).
- Incremental snapshots currently cannot be moved between subscriptions.
- You can currently only generate SAS URIs of up to five snapshots of a particular snapshot family at any given time.
- You cannot create an incremental snapshot for a particular disk outside of that disk's subscription.
- Up to seven incremental snapshots per disk can be created every five minutes.
- A total of 200 incremental snapshots can be created for a single disk.

## PowerShell

You can use Azure PowerShell to create an incremental snapshot. You will need the latest version of Azure PowerShell, the following command will either install it or update your existing installation to latest:

```
Install-Module -Name Az -AllowClobber -Scope CurrentUser
```

Once that is installed, login to your PowerShell session with `az login`.

To create an incremental snapshot with Azure PowerShell, set the configuration with [New-AzSnapshotConfig](#) with the `-Incremental` parameter and then pass that as a variable to [New-AzSnapshot](#) through the `-Snapshot` parameter.

Replace `<yourDiskNameHere>`, `<yourResourceGroupNameHere>`, and `<yourDesiredSnapshotNameHere>` with your values, then you can use the following script to create an incremental snapshot:

```
Get the disk that you need to backup by creating an incremental snapshot
$yourDisk = Get-AzDisk -DiskName <yourDiskNameHere> -ResourceGroupName <yourResourceGroupNameHere>

Create an incremental snapshot by setting the SourceUri property with the value of the Id property of the disk
$snapshotConfig=New-AzSnapshotConfig -SourceUri $yourDisk.Id -Location $yourDisk.Location -CreateOption Copy -Incremental
New-AzSnapshot -ResourceGroupName <yourResourceGroupNameHere> -SnapshotName <yourDesiredSnapshotNameHere> -Snapshot $snapshotConfig
```

You can identify incremental snapshots from the same disk with the `SourceResourceId` and the `SourceUniqueId` properties of snapshots. `SourceResourceId` is the Azure Resource Manager resource ID of the parent disk.

`SourceUniqueId` is the value inherited from the `UniqueId` property of the disk. If you were to delete a disk and then create a new disk with the same name, the value of the `UniqueId` property changes.

You can use `SourceResourceId` and `SourceUniqueId` to create a list of all snapshots associated with a particular disk. Replace `<yourResourceGroupNameHere>` with your value and then you can use the following example to list your existing incremental snapshots:

```
$snapshots = Get-AzSnapshot -ResourceGroupName <yourResourceGroupNameHere>

$incrementalSnapshots = New-Object System.Collections.ArrayList
foreach ($snapshot in $snapshots)
{
 if($snapshot.Incremental -and $snapshot.CreationData.SourceResourceId -eq $yourDisk.Id -and $snapshot.CreationData.SourceUniqueId -eq $yourDisk.UniqueId){
 $incrementalSnapshots.Add($snapshot)
 }
}

$incrementalSnapshots
```

## CLI

You can create an incremental snapshot with the Azure CLI, you will need the latest version of Azure CLI.

On Windows, the following command will either install or update your existing installation to the latest version:

```
Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi; Start-Process msieexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'
```

On Linux, the CLI installation will vary depending on operating system version. See [Install the Azure CLI](#) for your particular Linux version.

To create an incremental snapshot, use `az snapshot create` with the `--incremental` parameter.

The following example creates an incremental snapshot, replace `<yourDesiredSnapshotNameHere>`, `<yourResourceGroupNameHere>`, `<exampleDiskName>`, and `<exampleLocation>` with your own values, then run the

example:

```
sourceResourceId=$(az disk show -g <yourResourceGroupNameHere> -n <exampleDiskName> --query '[id]' -o tsv)

az snapshot create -g <yourResourceGroupNameHere> \
-n <yourDesiredSnapshotNameHere> \
-l <exampleLocation> \
--source "$sourceResourceId" \
--incremental
```

You can identify incremental snapshots from the same disk with the `SourceResourceId` and the `SourceUniqueId` properties of snapshots. `SourceResourceId` is the Azure Resource Manager resource ID of the parent disk.

`SourceUniqueId` is the value inherited from the `UniqueId` property of the disk. If you were to delete a disk and then create a new disk with the same name, the value of the `UniqueId` property changes.

You can use `SourceResourceId` and `SourceUniqueId` to create a list of all snapshots associated with a particular disk. The following example will list all incremental snapshots associated with a particular disk but, it requires some setup.

This example uses jq for querying the data. To run the example, you must [install jq](#).

Replace `<yourResourceGroupNameHere>` and `<exampleDiskName>` with your values, then you can use the following example to list your existing incremental snapshots, as long as you've also installed jq:

```
sourceUniqueId=$(az disk show -g <yourResourceGroupNameHere> -n <exampleDiskName> --query '[uniqueId]' -o tsv)

sourceResourceId=$(az disk show -g <yourResourceGroupNameHere> -n <exampleDiskName> --query '[id]' -o tsv)

az snapshot list -g <yourResourceGroupNameHere> -o json \
| jq -cr --arg SUID "$sourceUniqueId" --arg SRID "$sourceResourceId" '.[] | select(.incremental==true and \
.creationData.sourceUniqueId==$SUID and .creationData.sourceResourceId==$SRID)'
```

## Resource Manager template

You can also use Azure Resource Manager templates to create an incremental snapshot. You'll need to make sure the `apiVersion` is set to **2019-03-01** and that the `incremental` property is also set to true. The following snippet is an example of how to create an incremental snapshot with Resource Manager templates:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
 "diskName": {
 "type": "string",
 "defaultValue": "contosodisk1"
 },
 "diskResourceId": {
 "defaultValue": "<your_managed_disk_resource_ID>",
 "type": "String"
 }
 },
 "resources": [
 {
 "type": "Microsoft.Compute/snapshots",
 "name": "[concat(parameters('diskName'), '_snapshot1')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2019-03-01",
 "properties": {
 "creationData": {
 "createOption": "Copy",
 "sourceResourceId": "[parameters('diskResourceId')]"
 },
 "incremental": true
 }
 }
]
}
```

## Next steps

If you'd like to see sample code demonstrating the differential capability of incremental snapshots, using .NET, see [Copy Azure Managed Disks backups to another region with differential capability of incremental snapshots](#).

# Back up Azure unmanaged VM disks with incremental snapshots

11/13/2019 • 7 minutes to read • [Edit Online](#)

## Overview

Azure Storage provides the capability to take snapshots of blobs. Snapshots capture the blob state at that point in time. In this article, we describe a scenario in which you can maintain backups of virtual machine disks using snapshots. You can use this methodology when you choose not to use Azure Backup and Recovery Service, and wish to create a custom backup strategy for your virtual machine disks.

Azure virtual machine disks are stored as page blobs in Azure Storage. Since we are describing a backup strategy for virtual machine disks in this article, we refer to snapshots in the context of page blobs. To learn more about snapshots, refer to [Creating a Snapshot of a Blob](#).

## What is a snapshot?

A blob snapshot is a read-only version of a blob that is captured at a point in time. Once a snapshot has been created, it can be read, copied, or deleted, but not modified. Snapshots provide a way to back up a blob as it appears at a moment in time. Until REST version 2015-04-05, you had the ability to copy full snapshots. With the REST version 2015-07-08 and above, you can also copy incremental snapshots.

## Full snapshot copy

Snapshots can be copied to another storage account as a blob to keep backups of the base blob. You can also copy a snapshot over its base blob, which is like restoring the blob to an earlier version. When a snapshot is copied from one storage account to another, it occupies the same space as the base page blob. Therefore, copying whole snapshots from one storage account to another is slow and consumes much space in the target storage account.

### NOTE

If you copy the base blob to another destination, the snapshots of the blob are not copied along with it. Similarly, if you overwrite a base blob with a copy, snapshots associated with the base blob are not affected and stay intact under the base blob name.

### Back up disks using snapshots

As a backup strategy for your virtual machine disks, you can take periodic snapshots of the disk or page blob, and copy them to another storage account using tools like [Copy Blob](#) operation or [AzCopy](#). You can copy a snapshot to a destination page blob with a different name. The resulting destination page blob is a writeable page blob and not a snapshot. Later in this article, we describe steps to take backups of virtual machine disks using snapshots.

### Restore disks using snapshots

When it is time to restore your disk to a stable version that was previously captured in one of the backup snapshots, you can copy a snapshot over the base page blob. After the snapshot is promoted to the base page blob, the snapshot remains, but its source is overwritten with a copy that can be both read and written. Later in this article we describe steps to restore a previous version of your disk from its snapshot.

### Implementing full snapshot copy

You can implement a full snapshot copy by doing the following,

- First, take a snapshot of the base blob using the [Snapshot Blob](#) operation.
- Then, copy the snapshot to a target storage account using [Copy Blob](#).
- Repeat this process to maintain backup copies of your base blob.

## Incremental snapshot copy

The new feature in the [GetPageRanges](#) API provides a much better way to back up the snapshots of your page blobs or disks. The API returns the list of changes between the base blob and the snapshots, which reduces the amount of storage space used on the backup account. The API supports page blobs on Premium Storage as well as Standard Storage. Using this API, you can build faster and more efficient backup solutions for Azure VMs. This API will be available with the REST version 2015-07-08 and higher.

Incremental Snapshot Copy allows you to copy from one storage account to another the difference between,

- Base blob and its Snapshot OR
- Any two snapshots of the base blob

Provided the following conditions are met,

- The blob was created on Jan-1-2016 or later.
- The blob was not overwritten with [PutPage](#) or [Copy Blob](#) between two snapshots.

**Note:** This feature is available for Premium and Standard Azure Page Blobs.

When you have a custom backup strategy using snapshots, copying the snapshots from one storage account to another can be slow and can consume much storage space. Instead of copying the entire snapshot to a backup storage account, you can write the difference between consecutive snapshots to a backup page blob. This way, the time to copy and the space to store backups is substantially reduced.

### Implementing Incremental Snapshot Copy

You can implement incremental snapshot copy by doing the following,

- Take a snapshot of the base blob using [Snapshot Blob](#).
- Copy the snapshot to the target backup storage account in same or any other Azure region using [Copy Blob](#).  
This is the backup page blob. Take a snapshot of the backup page blob and store it in the backup account.
- Take another snapshot of the base blob using [Snapshot Blob](#).
- Get the difference between the first and second snapshots of the base blob using [GetPageRanges](#). Use the new parameter **prevsnapshot**, to specify the DateTime value of the snapshot you want to get the difference with.  
When this parameter is present, the REST response includes only the pages that were changed between target snapshot and previous snapshot including clear pages.
- Use [PutPage](#) to apply these changes to the backup page blob.
- Finally, take a snapshot of the backup page blob and store it in the backup storage account.

In the next section, we will describe in more detail how you can maintain backups of disks using Incremental Snapshot Copy

## Scenario

In this section, we describe a scenario that involves a custom backup strategy for virtual machine disks using snapshots.

Consider a DS-series Azure VM with a premium storage P30 disk attached. The P30 disk called *mypremiumdisk* is stored in a premium storage account called *mypremiumaccount*. A standard storage account called *mybackupstdaccount* is used for storing the backup of *mypremiumdisk*. We would like to keep a snapshot of *mypremiumdisk* every 12 hours.

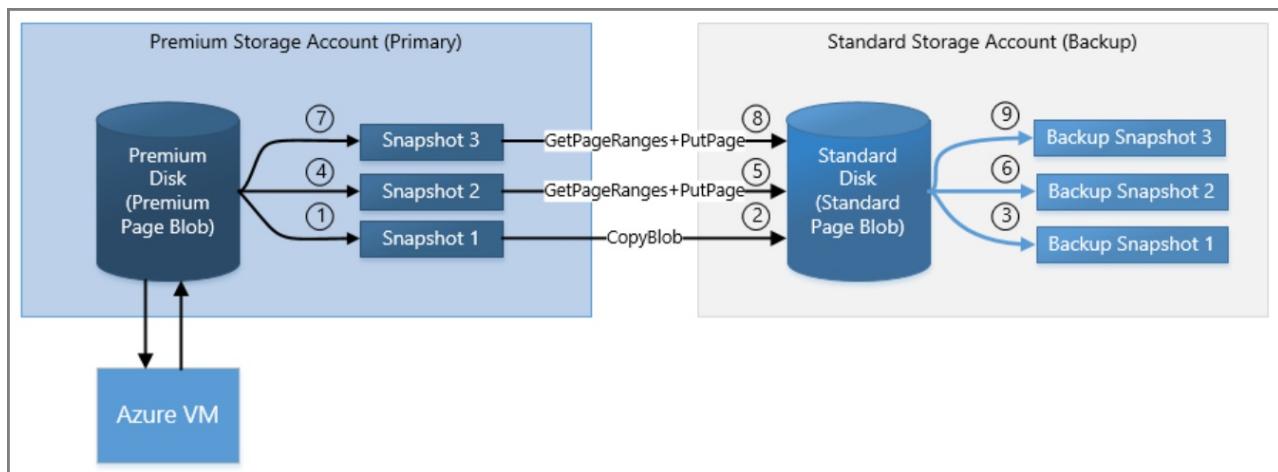
To learn about creating a storage account, see [Create a storage account](#).

To learn about backing up Azure VMs, refer to [Plan Azure VM backups](#).

## Steps to maintain backups of a disk using incremental snapshots

The following steps describe how to take snapshots of *mypremiumdisk* and maintain the backups in *mybackupsdaccount*. The backup is a standard page blob called *mybackupsdpageblob*. The backup page blob always reflects the same state as the last snapshot of *mypremiumdisk*.

1. Create the backup page blob for your premium storage disk, by taking a snapshot of *mypremiumdisk* called *mypremiumdisk\_ss1*.
2. Copy this snapshot to *mybackupsdaccount* as a page blob called *mybackupsdpageblob*.
3. Take a snapshot of *mybackupsdpageblob* called *mybackupsdpageblob\_ss1*, using [Snapshot Blob](#) and store it in *mybackupsdaccount*.
4. During the backup window, create another snapshot of *mypremiumdisk*, say *mypremiumdisk\_ss2*, and store it in *mypremiumaccount*.
5. Get the incremental changes between the two snapshots, *mypremiumdisk\_ss2* and *mypremiumdisk\_ss1*, using [GetPageRanges](#) on *mypremiumdisk\_ss2* with the **prevsnapshot** parameter set to the timestamp of *mypremiumdisk\_ss1*. Write these incremental changes to the backup page blob *mybackupsdpageblob* in *mybackupsdaccount*. If there are deleted ranges in the incremental changes, they must be cleared from the backup page blob. Use [PutPage](#) to write incremental changes to the backup page blob.
6. Take a snapshot of the backup page blob *mybackupsdpageblob*, called *mybackupsdpageblob\_ss2*. Delete the previous snapshot *mypremiumdisk\_ss1* from premium storage account.
7. Repeat steps 4–6 every backup window. In this way, you can maintain backups of *mypremiumdisk* in a standard storage account.



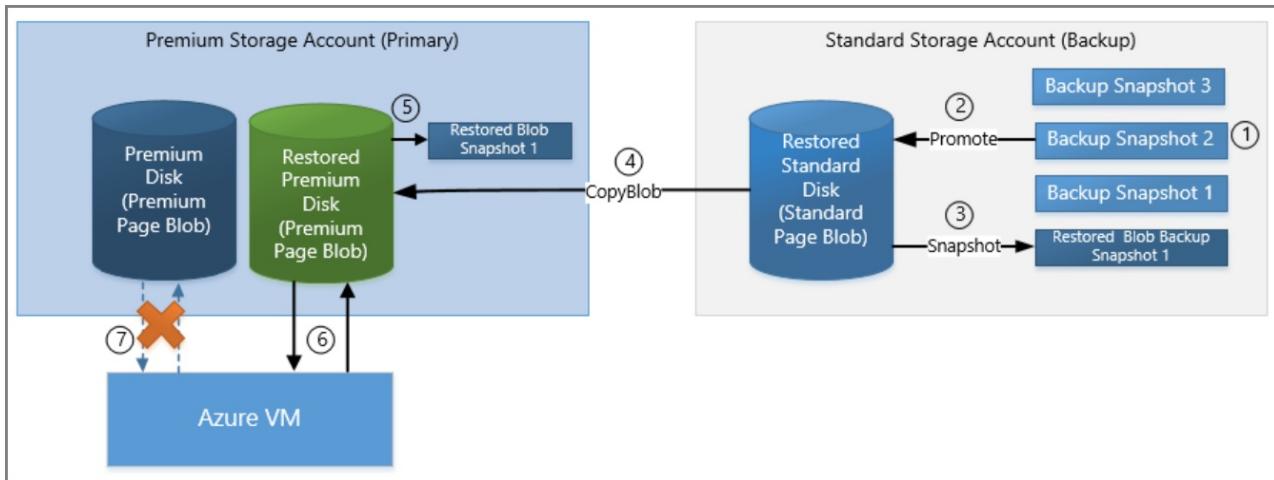
## Steps to restore a disk from snapshots

The following steps, describe how to restore the premium disk, *mypremiumdisk* to an earlier snapshot from the backup storage account *mybackupsdaccount*.

1. Identify the point in time that you wish to restore the premium disk to. Let's say that it is snapshot *mybackupsdpageblob\_ss2*, which is stored in the backup storage account *mybackupsdaccount*.
2. In *mybackupsdaccount*, promote the snapshot *mybackupsdpageblob\_ss2* as the new backup base page blob *mybackupsdpageblobrestored*.
3. Take a snapshot of this restored backup page blob, called *mybackupsdpageblobrestored\_ss1*.
4. Copy the restored page blob *mybackupsdpageblobrestored* from *mybackupsdaccount* to *mypremiumaccount* as the new premium disk *mypremiumdiskrestored*.
5. Take a snapshot of *mypremiumdiskrestored*, called *mypremiumdiskrestored\_ss1* for making future incremental

backups.

6. Point the DS series VM to the restored disk *mypremiumdiskrestored* and detach the old *mypremiumdisk* from the VM.
7. Begin the Backup process described in previous section for the restored disk *mypremiumdiskrestored*, using the *mybackupstdpageblobrestored* as the backup page blob.



## Next Steps

Use the following links to learn more about creating snapshots of a blob and planning your VM backup infrastructure.

- [Creating a Snapshot of a Blob](#)
- [Plan your VM Backup Infrastructure](#)

# Update the storage type of a managed disk

1/23/2020 • 3 minutes to read • [Edit Online](#)

There are four disk types of Azure managed disks: Azure ultra SSDs (preview), premium SSD, standard SSD, and standard HDD. You can switch between the three GA disk types (premium SSD, standard SSD, and standard HDD) based on your performance needs. You are not yet able to switch from or to an ultra SSD, you must deploy a new one.

This functionality is not supported for unmanaged disks. But you can easily [convert an unmanaged disk to a managed disk](#) to be able to switch between disk types.

## Prerequisites

- Because conversion requires a restart of the virtual machine (VM), you should schedule the migration of your disk storage during a pre-existing maintenance window.
- If your disk is unmanaged, first [convert it to a managed disk](#) so you can switch between storage options.

## Switch all managed disks of a VM between Premium and Standard

This example shows how to convert all of a VM's disks from Standard to Premium storage or from Premium to Standard storage. To use Premium managed disks, your VM must use a [VM size](#) that supports Premium storage. This example also switches to a size that supports premium storage:

```

Name of the resource group that contains the VM
$rgName = 'yourResourceGroup'

Name of the your virtual machine
$vmName = 'yourVM'

Choose between Standard_LRS and Premium_LRS based on your scenario
$storageType = 'Premium_LRS'

Premium capable size
Required only if converting storage from Standard to Premium
$size = 'Standard_DS2_v2'

Stop and deallocate the VM before changing the size
Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force

$vm = Get-AzVM -Name $vmName -resourceGroupName $rgName

Change the VM size to a size that supports Premium storage
Skip this step if converting storage from Premium to Standard
$vm.HardwareProfile.VmSize = $size
Update-AzVM -VM $vm -ResourceGroupName $rgName

Get all disks in the resource group of the VM
$vmDisks = Get-AzDisk -ResourceGroupName $rgName

For disks that belong to the selected VM, convert to Premium storage
foreach ($disk in $vmDisks)
{
 if ($disk.ManagedBy -eq $vm.Id)
 {
 $disk.Sku = [Microsoft.Azure.Management.Compute.Models.DiskSku]::new($storageType)
 $disk | Update-AzDisk
 }
}

Start-AzVM -ResourceGroupName $rgName -Name $vmName

```

## Switch individual managed disks between Standard and Premium

For your dev/test workload, you might want a mix of Standard and Premium disks to reduce your costs. You can choose to upgrade only those disks that need better performance. This example shows how to convert a single VM disk from Standard to Premium storage or from Premium to Standard storage. To use Premium managed disks, your VM must use a [VM size](#) that supports Premium storage. This example also shows how to switch to a size that supports Premium storage:

```

$diskName = 'yourDiskName'
resource group that contains the managed disk
$rgName = 'yourResourceGroupName'
Choose between Standard_LRS and Premium_LRS based on your scenario
$storageType = 'Premium_LRS'
Premium capable size
$size = 'Standard_DS2_v2'

$disk = Get-AzDisk -DiskName $diskName -ResourceGroupName $rgName

Get parent VM resource
$vmResource = Get-AzResource -ResourceId $disk.ManagedBy

Stop and deallocate the VM before changing the storage type
Stop-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name $vmResource.Name -Force

$vm = Get-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name $vmResource.Name

Change the VM size to a size that supports Premium storage
Skip this step if converting storage from Premium to Standard
$vm.HardwareProfile.VmSize = $size
Update-AzVM -VM $vm -ResourceGroupName $rgName

Update the storage type
$disk.Sku = [Microsoft.Azure.Management.Compute.Models.DiskSku]::new($storageType)
$disk | Update-AzDisk

Start-AzVM -ResourceGroupName $vm.ResourceGroupName -Name $vm.Name

```

## Convert managed disks from Standard to Premium in the Azure portal

Follow these steps:

1. Sign in to the [Azure portal](#).
2. Select the VM from the list of **Virtual machines** in the portal.
3. If the VM isn't stopped, select **Stop** at the top of VM **Overview** pane, and wait for the VM to stop.
4. In the pane for the VM, select **Disks** from the menu.
5. Select the disk that you want to convert.
6. Select **Configuration** from the menu.
7. Change the **Account type** from **Standard HDD** to **Premium SSD**.
8. Click **Save**, and close the disk pane.

The disk type conversion is instantaneous. You can start your VM after the conversion.

## Switch managed disks between Standard HDD and Standard SSD

This example shows how to convert a single VM disk from Standard HDD to Standard SSD or from Standard SSD to Standard HDD:

```
$diskName = 'yourDiskName'
resource group that contains the managed disk
$rgName = 'yourResourceGroupName'
Choose between Standard_LRS and StandardSSD_LRS based on your scenario
$storageType = 'StandardSSD_LRS'

$disk = Get-AzDisk -DiskName $diskName -ResourceGroupName $rgName

Get parent VM resource
$vmResource = Get-AzResource -ResourceId $disk.ManagedBy

Stop and deallocate the VM before changing the storage type
Stop-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name $vmResource.Name -Force

$vm = Get-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name $vmResource.Name

Update the storage type
$disk.Sku = [Microsoft.Azure.Management.Compute.Models.DiskSku]::new($storageType)
$disk | Update-AzDisk

Start-AzVM -ResourceGroupName $vm.ResourceGroupName -Name $vm.Name
```

## Next steps

Make a read-only copy of a VM by using a [snapshot](#).

# Migrate to Premium Storage by using Azure Site Recovery

12/4/2019 • 11 minutes to read • [Edit Online](#)

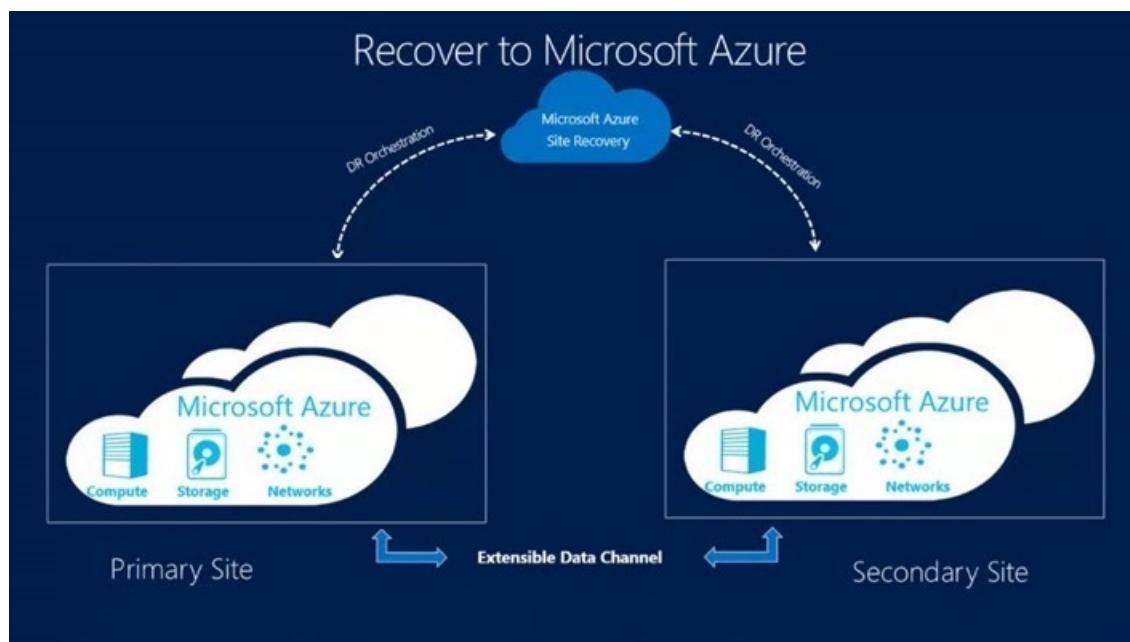
Azure premium SSDs deliver high-performance, low-latency disk support for virtual machines (VMs) that are running I/O-intensive workloads. This guide helps you migrate your VM disks from a standard storage account to a premium storage account by using [Azure Site Recovery](#).

Site Recovery is an Azure service that contributes to your strategy for business continuity and disaster recovery by orchestrating the replication of on-premises physical servers and VMs to the cloud (Azure) or to a secondary datacenter. When outages occur in your primary location, you fail over to the secondary location to keep applications and workloads available. You fail back to your primary location when it returns to normal operation.

Site Recovery provides test failovers to support disaster recovery drills without affecting production environments. You can run failovers with minimal data loss (depending on replication frequency) for unexpected disasters. In the scenario of migrating to Premium Storage, you can use the [failover in Site Recovery](#) to migrate target disks to a premium storage account.

We recommend migrating to Premium Storage by using Site Recovery because this option provides minimal downtime. This option also avoids the manual execution of copying disks and creating new VMs. Site Recovery will systematically copy your disks and create new VMs during failover.

Site Recovery supports a number of types of failover with minimal or no downtime. To plan your downtime and estimate data loss, see the [types of failover in Site Recovery](#). If you [prepare to connect to Azure VMs after failover](#), you should be able to connect to the Azure VM by using RDP after failover.



## Azure Site Recovery components

These Site Recovery components are relevant to this migration scenario:

- **Configuration server** is an Azure VM that coordinates communication and manages data replication and recovery processes. On this VM, you run a single setup file to install the configuration server and an additional component, called a process server, as a replication gateway. Read about [configuration server](#)

[prerequisites](#). You set up the configuration server only once, and you can use it for all migrations to the same region.

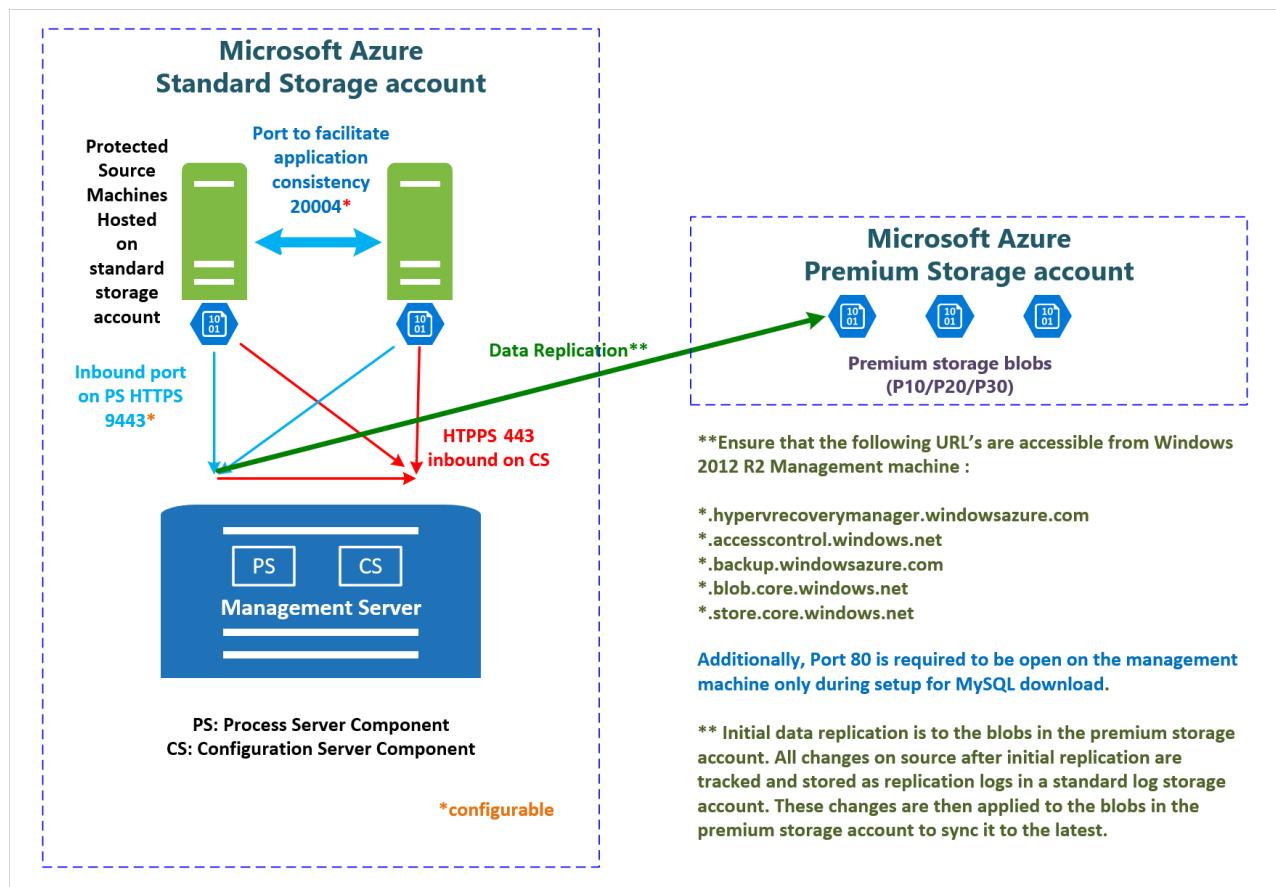
- **Process server** is a replication gateway that:

1. Receives replication data from source VMs.
2. Optimizes the data with caching, compression, and encryption.
3. Sends the data to a storage account.

It also handles push installation of the mobility service to source VMs and performs automatic discovery of source VMs. The default process server is installed on the configuration server. You can deploy additional standalone process servers to scale your deployment. Read about [best practices for process server deployment](#) and [deploying additional process servers](#). You set up the process server only once, and you can use it for all migrations to the same region.

- **Mobility service** is a component that is deployed on every standard VM that you want to replicate. It captures data writes on the standard VM and forwards them to the process server. Read about [replicated machine prerequisites](#).

This graphic shows how these components interact:



#### NOTE

Site Recovery does not support the migration of Storage Spaces disks.

For additional components for other scenarios, see [Scenario architecture](#).

## Azure essentials

These are the Azure requirements for this migration scenario:

- An Azure subscription.

- An Azure premium storage account to store replicated data.
- An Azure virtual network to which VMs will connect when they're created at failover. The Azure virtual network must be in the same region as the one in which Site Recovery runs.
- An Azure standard storage account to store replication logs. This can be the same storage account for the VM disks that are being migrated.

## Prerequisites

- Understand the relevant migration scenario components in the preceding section.
- Plan your downtime by learning about [failover in Site Recovery](#).

## Setup and migration steps

You can use Site Recovery to migrate Azure IaaS VMs between regions or within same region. The following instructions are tailored for this migration scenario from the article [Replicate VMware VMs or physical servers to Azure](#). Please follow the links for detailed steps in addition to the instructions in this article.

### Step 1: Create a Recovery Services vault

1. Open the [Azure portal](#).
2. Select **Create a resource > Management > Backup and Site Recovery (OMS)**. Alternatively, you can select **Browse > Recovery Services Vault > Add**.

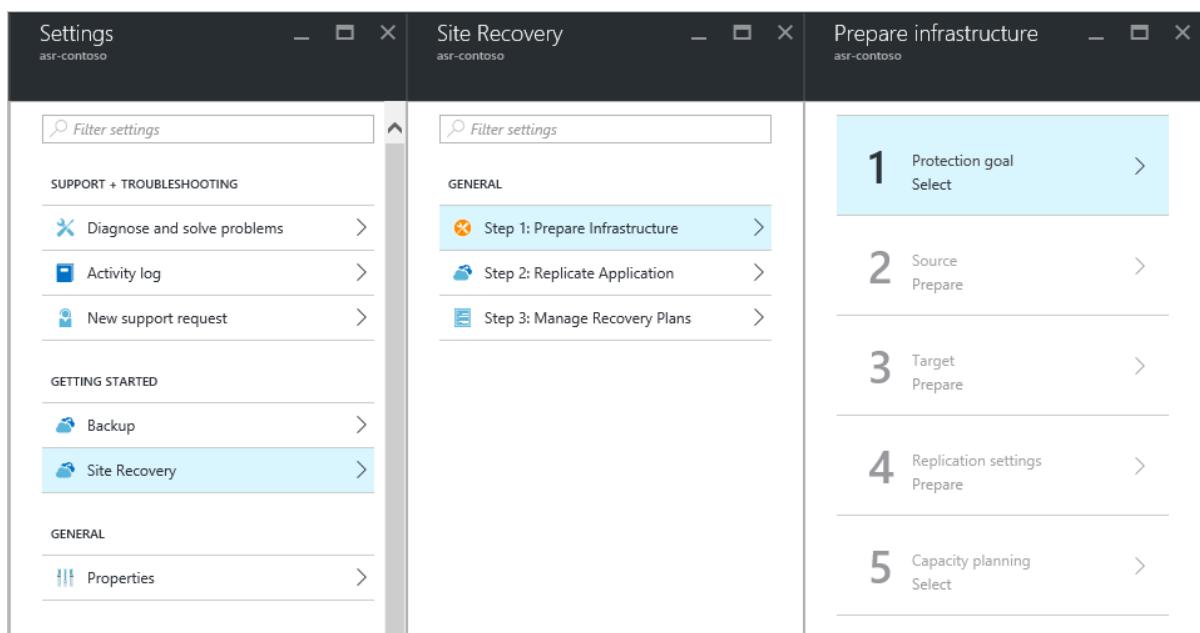
#### NOTE

Backup and Site Recovery was formerly part of the [OMS suite](#).

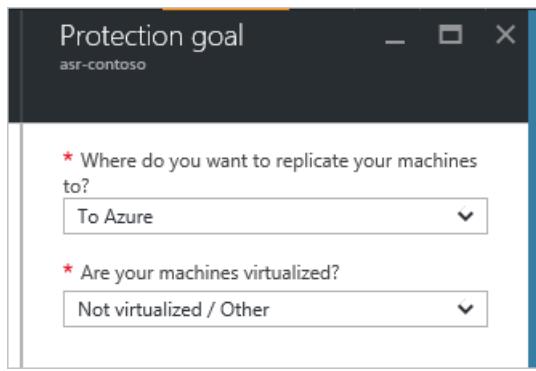
3. Specify a region that VMs will be replicated to. For the purpose of migration in the same region, select the region where your source VMs and source storage accounts are.

### Step 2: Choose your protection goals

1. On the VM where you want to install the configuration server, open the [Azure portal](#).
2. Go to **Recovery Services vaults > Settings > Site Recovery > Step 1: Prepare Infrastructure > Protection goal**.



3. Under **Protection goal**, in the first drop-down list, select **To Azure**. In the second drop-down list, select **Not virtualized / Other**, and then select **OK**.

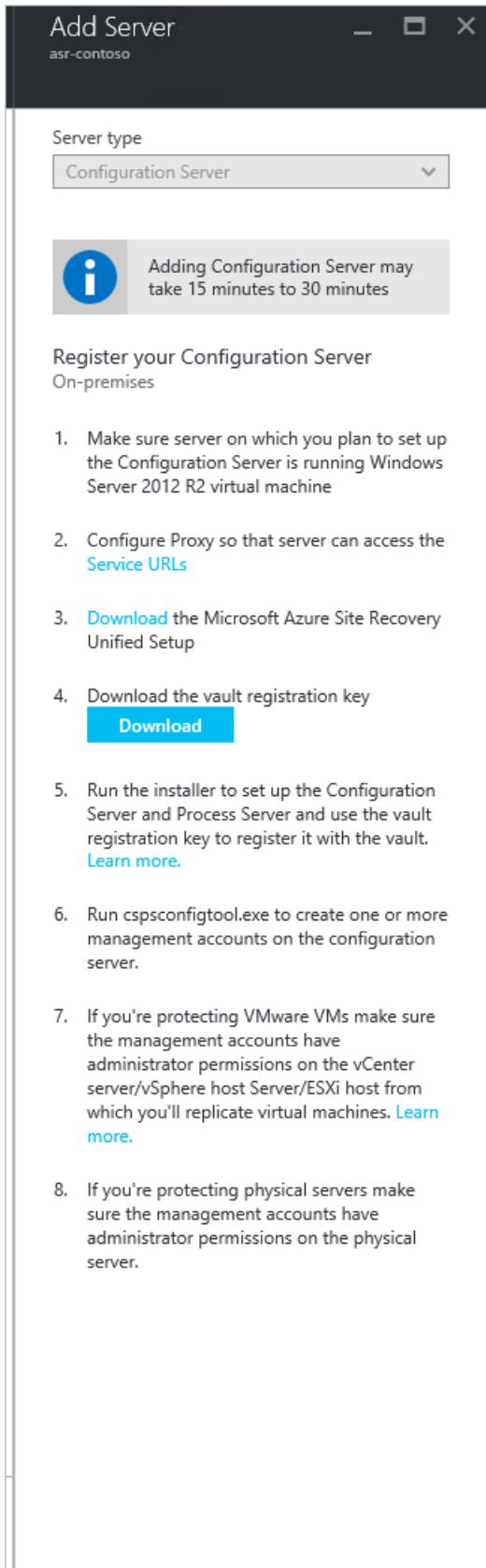


### Step 3: Set up the source environment (configuration server)

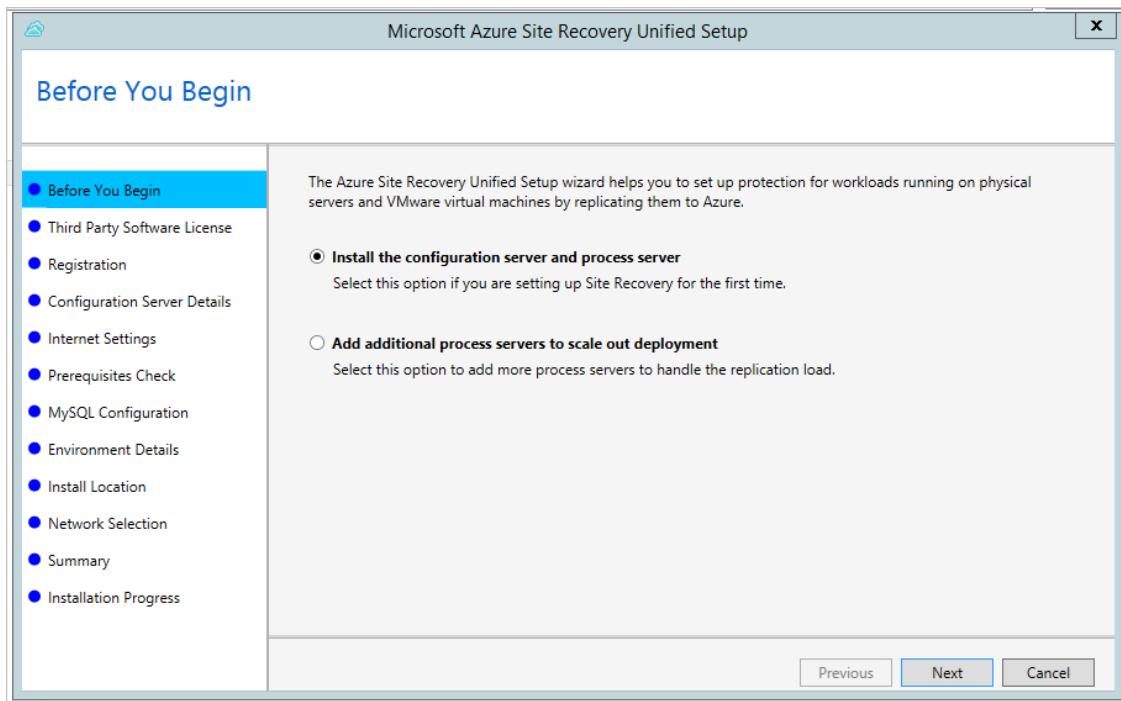
1. Download **Azure Site Recovery Unified Setup** and the vault registration key by going to the **Prepare infrastructure > Prepare source > Add Server** panes.

You will need the vault registration key to run the unified setup. The key is valid for five days after you generate it.

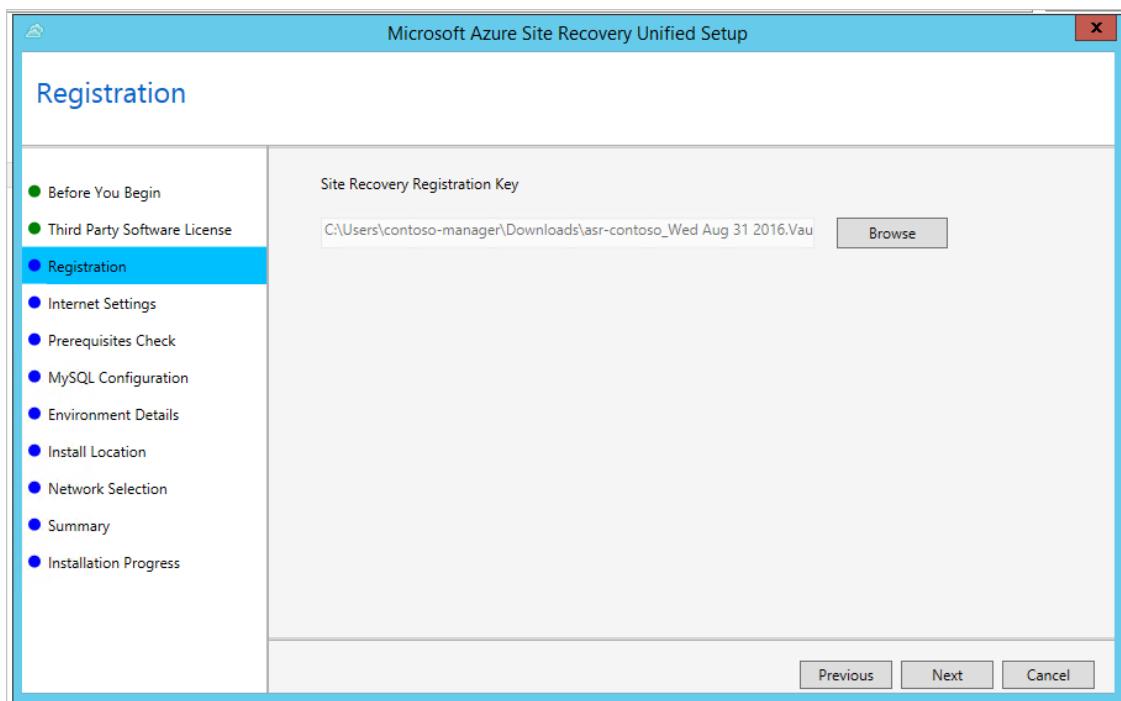
2. In the **Add Server** pane, add a configuration server.



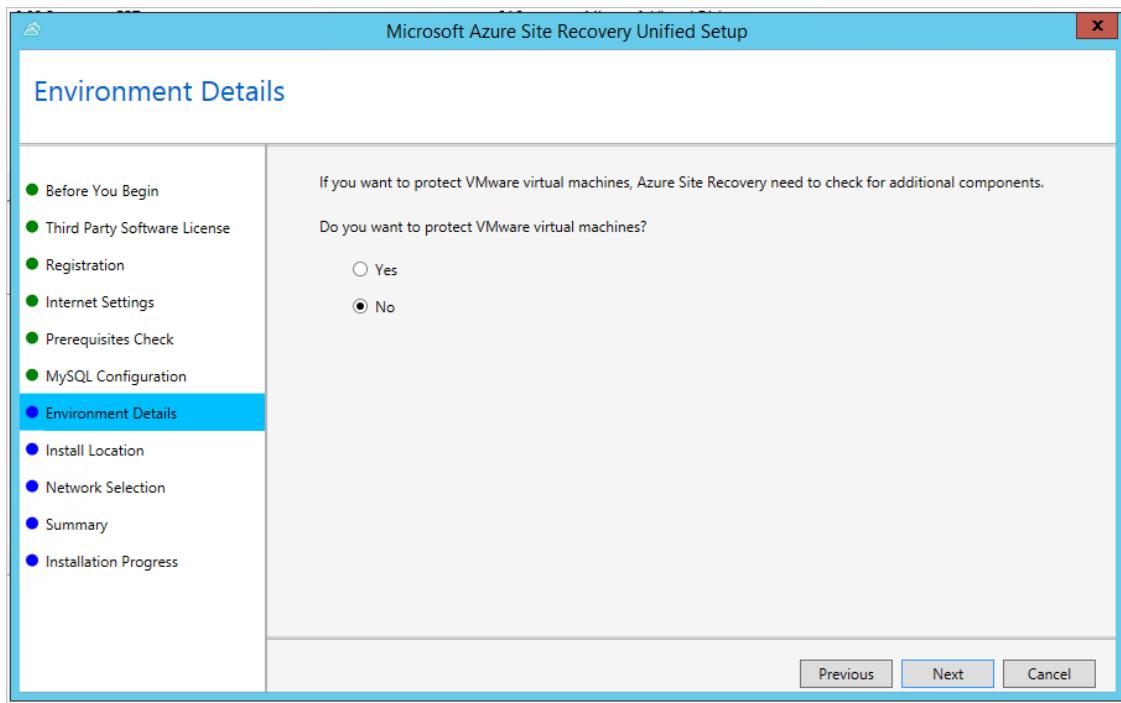
3. On the VM that you're using as the configuration server, run Unified Setup to install the configuration server and the process server. You can [walk through the screenshots](#) to complete the installation. You can refer to the following screenshots for steps specified for this migration scenario.
    - a. In **Before You Begin**, select **Install the configuration server and process server**.



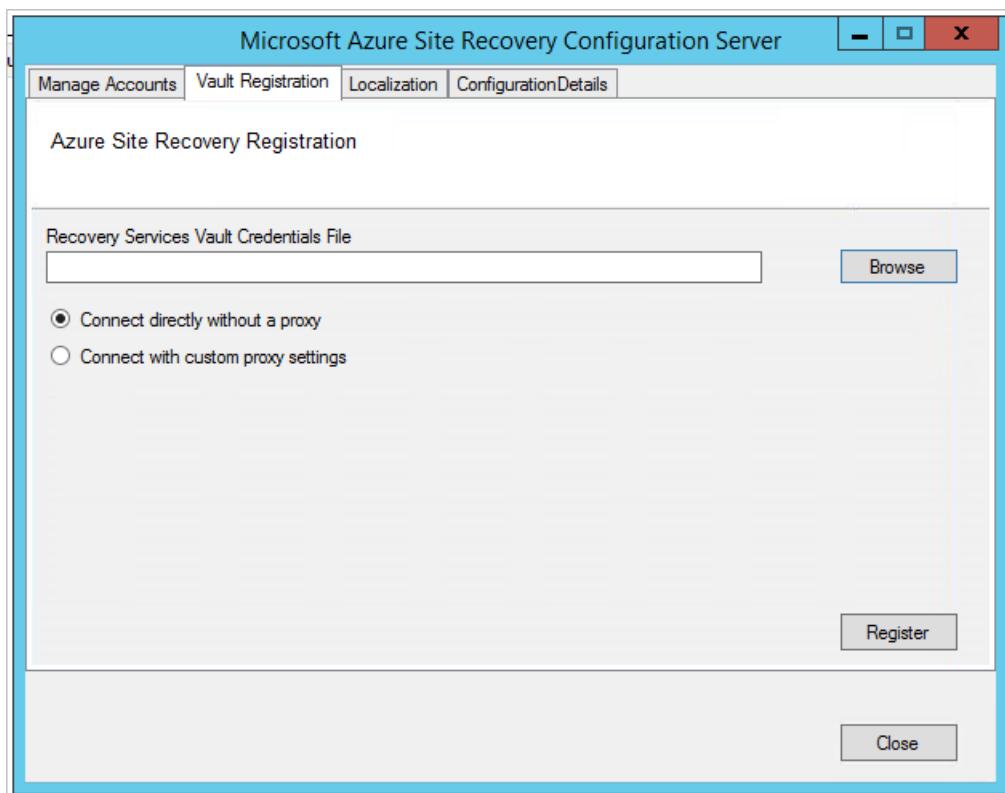
- b. In **Registration**, browse and select the registration key that you downloaded from the vault.



- c. In **Environment Details**, select whether you're going to replicate VMware VMs. For this migration scenario, choose **No**.



4. After the installation is complete, do the following in the **Microsoft Azure Site Recovery Configuration Server** window:
  - a. Use the **Manage Accounts** tab to create the account that Site Recovery can use for automatic discovery. (In the scenario about protecting physical machines, setting up the account isn't relevant, but you need at least one account to enable one of the following steps. In this case, you can name the account and password as any.)
  - b. Use the **Vault Registration** tab to upload the vault credential file.



#### Step 4: Set up the target environment

Select **Prepare infrastructure > Target**, and specify the deployment model that you want to use for VMs after failover. You can choose **Classic** or **Resource Manager**, depending on your scenario.

The screenshot shows two windows side-by-side. The left window is titled 'Prepare infrastructure' and lists five steps: 1. Protection goal (VMware VMs/physical servers t...), 2. Source (CONTOSO-CONFIG), 3. Target (Prepare, highlighted in blue), 4. Replication settings (Prepare), and 5. Capacity planning (Select). The right window is titled 'Target' and shows the configuration for step 3. It includes sections for 'Step 1 : Select Azure subscription' (Subscription: Visual Studio Enterprise, Deployment model: Classic selected), 'Step 2 : Ensure that at least one compatible Azure storage account exist' (Storage account(s): Found 5 compatible Azure storage accounts out of 6 available in the subscription), and 'Step 3 : Ensure that at least one compatible Azure virtual network exist' (Network(s): Found 2 compatible Azure virtual networks out of 2 available in the subscription).

Site Recovery checks that you have one or more compatible Azure storage accounts and networks.

#### NOTE

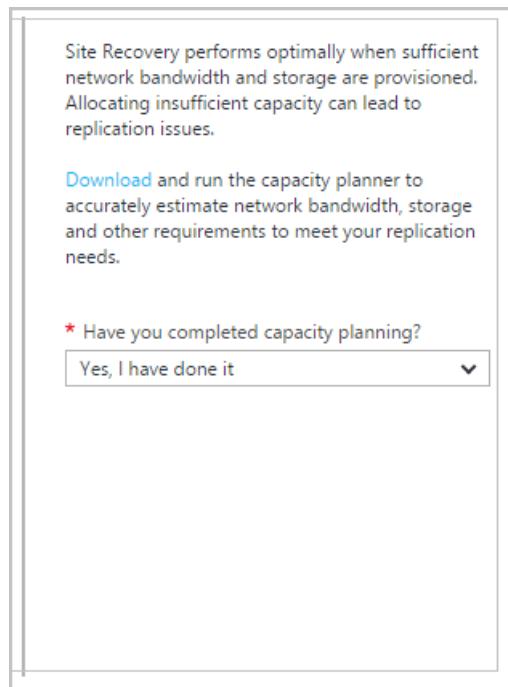
If you're using a premium storage account for replicated data, you need to set up an additional standard storage account to store replication logs.

#### Step 5: Set up replication settings

To verify that your configuration server is successfully associated with the replication policy that you create, follow [Set up replication settings](#).

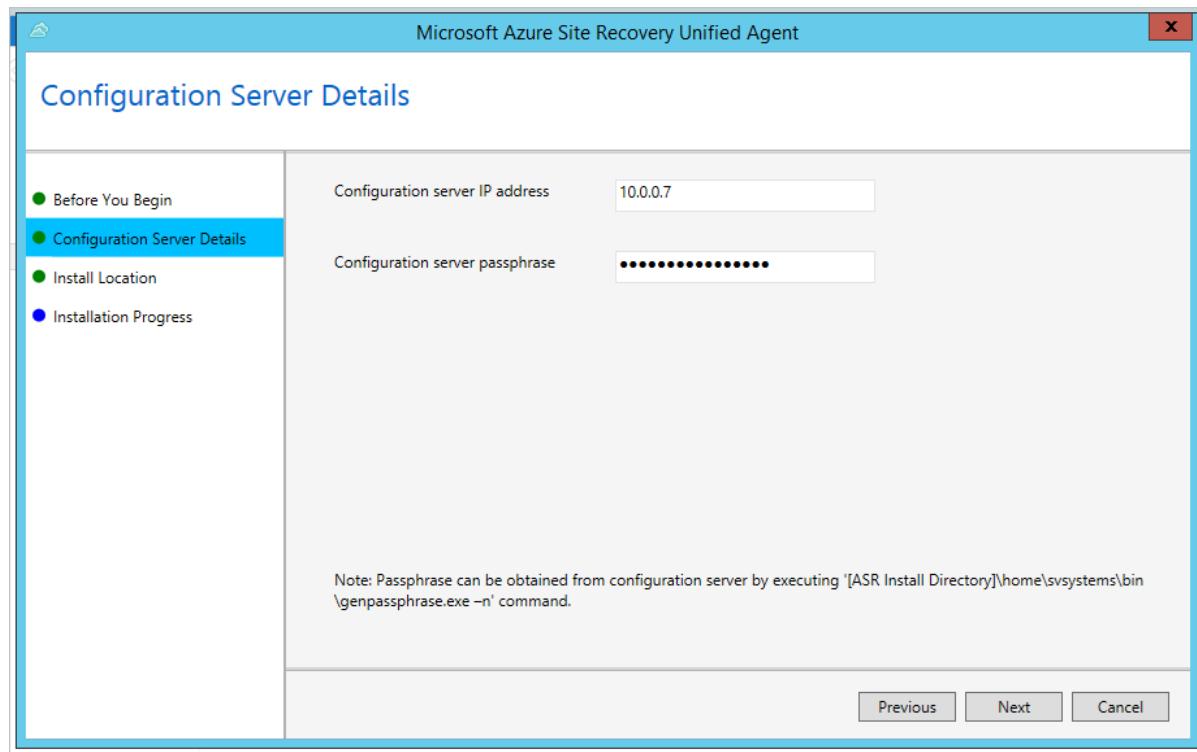
#### Step 6: Plan capacity

1. Use the [capacity planner](#) to accurately estimate network bandwidth, storage, and other requirements to meet your replication needs.
2. When you're done, select **Yes, I have done it** in **Have you completed capacity planning?**



## Step 7: Install the mobility service and enable replication

1. You can choose to [push installation](#) to your source VMs or to [manually install the mobility service](#) on your source VMs. You can find the requirement of pushing installation and the path of the manual installer in the provided link. If you're doing a manual installation, you might need to use an internal IP address to find the configuration server.



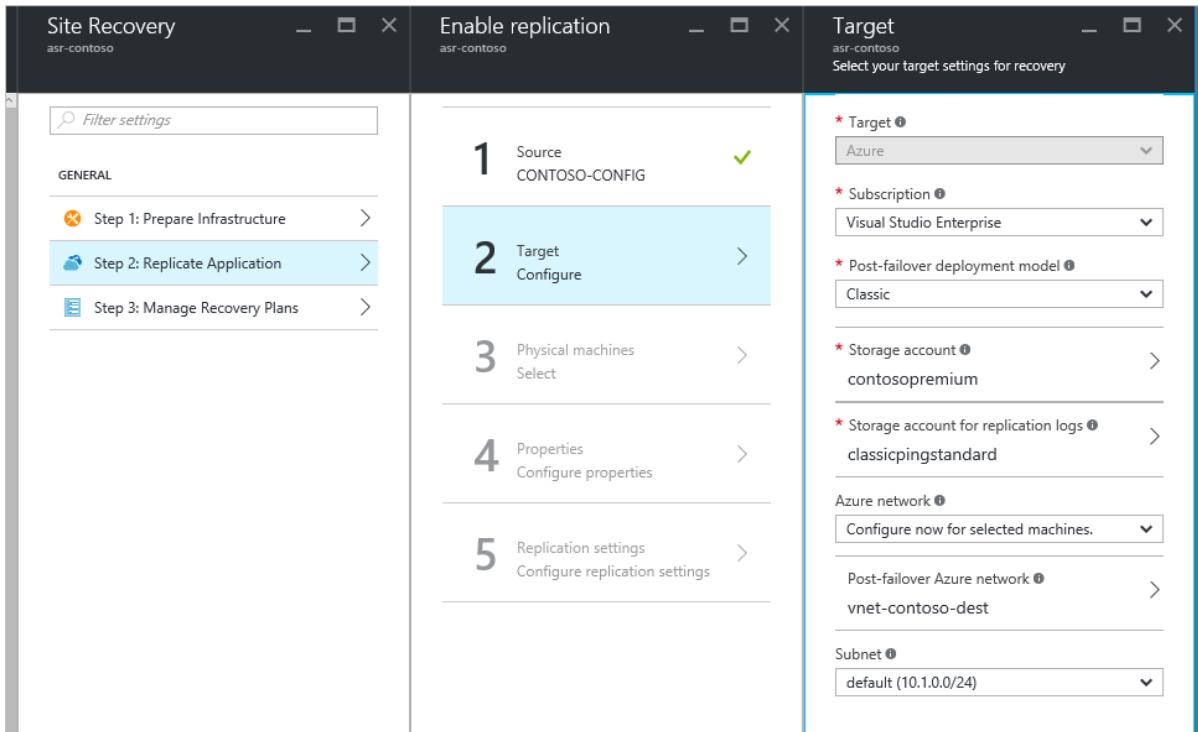
The failed-over VM will have two temporary disks: one from the primary VM and the other created during the provisioning of the VM in the recovery region. To exclude the temporary disk before replication, install the mobility service before you enable replication. To learn more about how to exclude the temporary disk, see [Exclude disks from replication](#).

2. Enable replication as follows:
  - a. Select **Replicate Application > Source**. After you've enabled replication for the first time, select **+Replicate** in the vault to enable replication for additional machines.

- b. In step 1, set up **Source** as your process server.
- c. In step 2, specify the post-failover deployment model, a premium storage account to migrate to, a standard storage account to save logs, and a virtual network to fail to.
- d. In step 3, add protected VMs by IP address. (You might need an internal IP address to find them.)
- e. In step 4, configure the properties by selecting the accounts that you set up previously on the process server.
- f. In step 5, choose the replication policy that you created previously in "Step 5: Set up replication settings."
- g. Select **OK**.

#### NOTE

When an Azure VM is deallocated and started again, there is no guarantee that it will get the same IP address. If the IP address of the configuration server/process server or the protected Azure VMs changes, the replication in this scenario might not work correctly.



When you design your Azure Storage environment, we recommend that you use separate storage accounts for each VM in an availability set. We recommend that you follow the best practice in the storage layer to [use multiple storage accounts for each availability set](#). Distributing VM disks to multiple storage accounts helps to improve storage availability and distributes the I/O across the Azure storage infrastructure.

If your VMs are in an availability set, instead of replicating disks of all VMs into one storage account, we highly recommend migrating multiple VMs multiple times. That way, the VMs in the same availability set do not share a single storage account. Use the **Enable Replication** pane to set up a destination storage account for each VM, one at a time.

You can choose a post-failover deployment model according to your need. If you choose Azure Resource Manager as your post-failover deployment model, you can fail over a VM (Resource Manager) to a VM (Resource Manager), or you can fail over a VM (classic) to a VM (Resource Manager).

#### Step 8: Run a test failover

To check whether your replication is complete, select your Site Recovery instance and then select **Settings > Replicated Items**. You will see the status and percentage of your replication process.

After initial replication is complete, run a test failover to validate your replication strategy. For detailed steps of a

test failover, see [Run a test failover in Site Recovery](#).

#### NOTE

Before you run any failover, make sure that your VMs and replication strategy meet the requirements. For more information about running a test failover, see [Test failover to Azure in Site Recovery](#).

You can see the status of your test failover in **Settings > Jobs > YOUR\_FAILOVER\_PLAN\_NAME**. In the pane, you can see a breakdown of the steps and success/failure results. If the test failover fails at any step, select the step to check the error message.

#### Step 9: Run a failover

After the test failover is completed, run a failover to migrate your disks to Premium Storage and replicate the VM instances. Follow the detailed steps in [Run a failover](#).

Be sure to select **Shut down VMs and synchronize the latest data**. This option specifies that Site Recovery should try to shut down the protected VMs and synchronize the data so that the latest version of the data will be failed over. If you don't select this option or the attempt doesn't succeed, the failover will be from the latest available recovery point for the VM.

Site Recovery will create a VM instance whose type is the same as or similar to a Premium Storage-capable VM. You can check the performance and price of various VM instances by going to [Windows Virtual Machines Pricing](#) or [Linux Virtual Machines Pricing](#).

### Post-migration steps

1. **Configure replicated VMs to the availability set if applicable.** Site Recovery does not support migrating VMs along with the availability set. Depending on the deployment of your replicated VM, do one of the following:
  - For a VM created through the classic deployment model: Add the VM to the availability set in the Azure portal. For detailed steps, go to [Add an existing virtual machine to an availability set](#).
  - For a VM created through the Resource Manager deployment model: Save your configuration of the VM and then delete and re-create the VMs in the availability set. To do so, use the script at [Set Azure Resource Manager VM Availability Set](#). Before you run this script, check its limitations and plan your downtime.
2. **Delete old VMs and disks.** Make sure that the Premium disks are consistent with source disks and that the new VMs perform the same function as the source VMs. Delete the VM and delete the disks from your source storage accounts in the Azure portal. If there's a problem in which the disk is not deleted even though you deleted the VM, see [Troubleshoot storage resource deletion errors](#).
3. **Clean the Azure Site Recovery infrastructure.** If Site Recovery is no longer needed, you can clean its infrastructure. Delete replicated items, the configuration server, and the recovery policy, and then delete the Azure Site Recovery vault.

### Troubleshooting

- [Monitor and troubleshoot protection for virtual machines and physical servers](#)
- [Microsoft Azure Site Recovery forum](#)

### Next steps

For specific scenarios for migrating virtual machines, see the following resources:

- [Migrate Azure Virtual Machines between Storage Accounts](#)
- [Create and upload a Windows Server VHD to Azure](#)
- [Migrating Virtual Machines from Amazon AWS to Microsoft Azure](#)

Also, see the following resources to learn more about Azure Storage and Azure Virtual Machines:

- [Azure Storage](#)
- [Azure Virtual Machines](#)

# Migrate Azure VMs to Managed Disks in Azure

2/28/2020 • 2 minutes to read • [Edit Online](#)

Azure Managed Disks simplifies your storage management by removing the need to separately manage storage accounts. You can also migrate your existing Azure VMs to Managed Disks to benefit from better reliability of VMs in an Availability Set. It ensures that the disks of different VMs in an Availability Set are sufficiently isolated from each other to avoid single point of failures. It automatically places disks of different VMs in an Availability Set in different Storage scale units (stamps) which limits the impact of single Storage scale unit failures caused due to hardware and software failures. Based on your needs, you can choose from four types of storage options. To learn about the available disk types, see our article [Select a disk type](#)

## Migration scenarios

You can migrate to Managed Disks in following scenarios:

| SCENARIO                                                                        | ARTICLE                                                                                                                                                           |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Convert stand alone VMs and VMs in an availability set to managed disks         | <a href="#">Convert VMs to use managed disks</a>                                                                                                                  |
| Convert a single VM from classic to Resource Manager on managed disks           | <a href="#">Create a VM from a classic VHD</a>                                                                                                                    |
| Convert all the VMs in a vNet from classic to Resource Manager on managed disks | <a href="#">Migrate IaaS resources from classic to Resource Manager</a> and then <a href="#">Convert a VM from unmanaged disks to managed disks</a>               |
| Upgrade VMs with standard unmanaged disks to VMs with managed premium disks     | First, <a href="#">Convert a Windows virtual machine from unmanaged disks to managed disks</a> . Then <a href="#">Update the storage type of a managed disk</a> . |

### IMPORTANT

Classic VMs will be retired on March 1, 2023.

If you use IaaS resources from ASM, please complete your migration by March 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.

For more information, see [Migrate your IaaS resources to Azure Resource Manager by March 1, 2023](#).

## Next steps

- Learn more about [Managed Disks](#)
- Review the [pricing for Managed Disks](#).

# Convert a Windows virtual machine from unmanaged disks to managed disks

12/10/2019 • 4 minutes to read • [Edit Online](#)

If you have existing Windows virtual machines (VMs) that use unmanaged disks, you can convert the VMs to use managed disks through the [Azure Managed Disks](#) service. This process converts both the OS disk and any attached data disks.

## Before you begin

- Review [Plan for the migration to Managed Disks](#).
- Review [the FAQ about migration to Managed Disks](#).
- The conversion requires a restart of the VM, so schedule the migration of your VMs during a pre-existing maintenance window.
- The conversion is not reversible.
- Be aware that any users with the [Virtual Machine Contributor](#) role will not be able to change the VM size (as they could pre-conversion). This is because VMs with managed disks require the user to have the Microsoft.Compute/disks/write permission on the OS disks.
- Be sure to test the conversion. Migrate a test virtual machine before you perform the migration in production.
- During the conversion, you deallocate the VM. The VM receives a new IP address when it is started after the conversion. If needed, you can [assign a static IP address](#) to the VM.
- Review the minimum version of the Azure VM agent required to support the conversion process. For information on how to check and update your agent version, see [Minimum version support for VM agents in Azure](#)
- The original VHDs and the storage account used by the VM before conversion are not deleted. They continue to incur charges. To avoid being billed for these artifacts, delete the original VHD blobs after you verify that the conversion is complete. If you need to find these unattached disks in order to delete them, see our article [Find and delete unattached Azure managed and unmanaged disks](#).

## Convert single-instance VMs

This section covers how to convert single-instance Azure VMs from unmanaged disks to managed disks. (If your VMs are in an availability set, see the next section.)

1. Deallocate the VM by using the [Stop-AzVM](#) cmdlet. The following example deallocates the VM named `myVM` in the resource group named `myResourceGroup` :

```
$rgName = "myResourceGroup"
$vmName = "myVM"
Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force
```

2. Convert the VM to managed disks by using the [ConvertTo-AzVMMManagedDisk](#) cmdlet. The following process converts the previous VM, including the OS disk and any data disks, and starts the Virtual

Machine:

```
ConvertTo-AzVMManagedDisk -ResourceGroupName $rgName -VMName $vmName
```

## Convert VMs in an availability set

If the VMs that you want to convert to managed disks are in an availability set, you first need to convert the availability set to a managed availability set.

1. Convert the availability set by using the [Update-AzAvailabilitySet](#) cmdlet. The following example updates the availability set named `myAvailabilitySet` in the resource group named `myResourceGroup`:

```
$rgName = 'myResourceGroup'
$avSetName = 'myAvailabilitySet'

$avSet = Get-AzAvailabilitySet -ResourceGroupName $rgName -Name $avSetName
Update-AzAvailabilitySet -AvailabilitySet $avSet -Sku Aligned
```

If the region where your availability set is located has only 2 managed fault domains but the number of unmanaged fault domains is 3, this command shows an error similar to "The specified fault domain count 3 must fall in the range 1 to 2." To resolve the error, update the fault domain to 2 and update `Sku` to `Aligned` as follows:

```
$avSet.PlatformFaultDomainCount = 2
Update-AzAvailabilitySet -AvailabilitySet $avSet -Sku Aligned
```

2. Deallocate and convert the VMs in the availability set. The following script deallocates each VM by using the [Stop-AzVM](#) cmdlet, converts it by using [ConvertTo-AzVMManagedDisk](#), and restarts it automatically as part of the conversion process:

```
$avSet = Get-AzAvailabilitySet -ResourceGroupName $rgName -Name $avSetName

foreach($vmInfo in $avSet.VirtualMachinesReferences)
{
 $vm = Get-AzVM -ResourceGroupName $rgName | Where-Object {$_.Id -eq $vmInfo.id}
 Stop-AzVM -ResourceGroupName $rgName -Name $vm.Name -Force
 ConvertTo-AzVMManagedDisk -ResourceGroupName $rgName -VMName $vm.Name
}
```

## Troubleshooting

If there is an error during conversion, or if a VM is in a failed state because of issues in a previous conversion, run the `ConvertTo-AzVMManagedDisk` cmdlet again. A simple retry usually unblocks the situation. Before converting, make sure all the VM extensions are in the 'Provisioning succeeded' state or the conversion will fail with the error code 409.

## Convert using the Azure portal

You can also convert unmanaged disks to managed disks using the Azure portal.

1. Sign in to the [Azure portal](#).
2. Select the VM from the list of VMs in the portal.
3. In the blade for the VM, select **Disks** from the menu.

4. At the top of the **Disks** blade, select **Migrate to managed disks**.
5. If your VM is in an availability set, there will be a warning on the **Migrate to managed disks** blade that you need to convert the availability set first. The warning should have a link you can click to convert the availability set. Once the availability set is converted or if your VM is not in an availability set, click **Migrate** to start the process of migrating your disks to managed disks.

The VM will be stopped and restarted after migration is complete.

## Next steps

[Convert standard managed disks to premium](#)

Take a read-only copy of a VM by using [snapshots](#).

# Enable Write Accelerator

11/13/2019 • 8 minutes to read • [Edit Online](#)

Write Accelerator is a disk capability for M-Series Virtual Machines (VMs) on Premium Storage with Azure Managed Disks exclusively. As the name states, the purpose of the functionality is to improve the I/O latency of writes against Azure Premium Storage. Write Accelerator is ideally suited where log file updates are required to persist to disk in a highly performant manner for modern databases.

Write Accelerator is generally available for M-series VMs in the Public Cloud.

## Planning for using Write Accelerator

Write Accelerator should be used for the volumes that contain the transaction log or redo logs of a DBMS. It is not recommended to use Write Accelerator for the data volumes of a DBMS as the feature has been optimized to be used against log disks.

Write Accelerator only works in conjunction with [Azure managed disks](#).

### IMPORTANT

Enabling Write Accelerator for the operating system disk of the VM will reboot the VM.

To enable Write Accelerator to an existing Azure disk that is NOT part of a volume build out of multiple disks with Windows disk or volume managers, Windows Storage Spaces, Windows Scale-out file server (SOFS), Linux LVM, or MDADM, the workload accessing the Azure disk needs to be shut down. Database applications using the Azure disk MUST be shut down.

If you want to enable or disable Write Accelerator for an existing volume that is built out of multiple Azure Premium Storage disks and striped using Windows disk or volume managers, Windows Storage Spaces, Windows Scale-out file server (SOFS), Linux LVM or MDADM, all disks building the volume must be enabled or disabled for Write Accelerator in separate steps.

**Before enabling or disabling Write Accelerator in such a configuration, shut down the Azure VM.**

Enabling Write Accelerator for OS disks should not be necessary for SAP-related VM configurations.

### Restrictions when using Write Accelerator

When using Write Accelerator for an Azure disk/VHD, these restrictions apply:

- The Premium disk caching must be set to 'None' or 'Read Only'. All other caching modes are not supported.
- Snapshot are not currently supported for Write Accelerator-enabled disks. During backup, the Azure Backup service automatically excludes Write Accelerator-enabled disks attached to the VM.
- Only smaller I/O sizes (<=512 KiB) are taking the accelerated path. In workload situations where data is getting bulk loaded or where the transaction log buffers of the different DBMS are filled to a larger degree before getting persisted to the storage, chances are that the I/O written to disk is not taking the accelerated path.

There are limits of Azure Premium Storage VHDs per VM that can be supported by Write Accelerator. The current limits are:

| VM SKU              | NUMBER OF WRITE ACCELERATOR DISKS | WRITE ACCELERATOR DISK IOPS PER VM |
|---------------------|-----------------------------------|------------------------------------|
| M416ms_v2, M416s_v2 | 16                                | 20000                              |
| M208ms_v2, M208s_v2 | 8                                 | 10000                              |

| VM SKU                    | NUMBER OF WRITE ACCELERATOR DISKS | WRITE ACCELERATOR DISK IOPS PER VM |
|---------------------------|-----------------------------------|------------------------------------|
| M128ms, M128s             | 16                                | 20000                              |
| M64ms, M64ls, M64s        | 8                                 | 10000                              |
| M32ms, M32ls, M32ts, M32s | 4                                 | 5000                               |
| M16ms, M16s               | 2                                 | 2500                               |
| M8ms, M8s                 | 1                                 | 1250                               |

The IOPS limits are per VM and *not* per disk. All Write Accelerator disks share the same IOPS limit per VM.

## Enabling Write Accelerator on a specific disk

The next few sections will describe how Write Accelerator can be enabled on Azure Premium Storage VHDs.

### Prerequisites

The following prerequisites apply to the usage of Write Accelerator at this point in time:

- The disks you want to apply Azure Write Accelerator against need to be [Azure managed disks](#) on Premium Storage.
- You must be using an M-series VM

## Enabling Azure Write Accelerator using Azure PowerShell

The Azure Power Shell module from version 5.5.0 include the changes to the relevant cmdlets to enable or disable Write Accelerator for specific Azure Premium Storage disks. In order to enable or deploy disks supported by Write Accelerator, the following Power Shell commands got changed, and extended to accept a parameter for Write Accelerator.

A new switch parameter, **-WriteAccelerator** has been added to the following cmdlets:

- [Set-AzVMOsDisk](#)
- [Add-AzVMDataDisk](#)
- [Set-AzVMDataDisk](#)
- [Add-AzVmssDataDisk](#)

Not giving the parameter sets the property to false and will deploy disks that have no support by Write Accelerator.

A new switch parameter, **-OsDiskWriteAccelerator** was added to the following cmdlets:

- [Set-AzVmssStorageProfile](#)

Not specifying the parameter sets the property to false by default, returning disks that don't leverage Write Accelerator.

A new optional Boolean (non-nullable) parameter, **-OsDiskWriteAccelerator** was added to the following cmdlets:

- [Update-AzVM](#)
- [Update-AzVmss](#)

Specify either \$true or \$false to control support of Azure Write Accelerator with the disks.

Examples of commands could look like:

```

New-AzVMConfig | Set-AzVMOsDisk | Add-AzVMDataDisk -Name "datadisk1" | Add-AzVMDataDisk -Name "logdisk1" -WriteAccelerator | New-AzVM

Get-AzVM | Update-AzVM -OsDiskWriteAccelerator $true

New-AzVmssConfig | Set-AzVmssStorageProfile -OsDiskWriteAccelerator | Add-AzVmssDataDisk -Name "datadisk1" -WriteAccelerator:$false | Add-AzVmssDataDisk -Name "logdisk1" -WriteAccelerator | New-AzVmss

Get-AzVmss | Update-AzVmss -OsDiskWriteAccelerator:$false

```

Two main scenarios can be scripted as shown in the following sections.

### **Adding a new disk supported by Write Accelerator using PowerShell**

You can use this script to add a new disk to your VM. The disk created with this script uses Write Accelerator.

Replace `myVM`, `myWAVMs`, `log001`, size of the disk, and LunID of the disk with values appropriate for your specific deployment.

```

Specify your VM Name
$vmName="myVM"
#Specify your Resource Group
$rgName = "myWAVMs"
#data disk name
$datadiskname = "log001"
#LUN Id
$lunid=8
#size
$size=1023
#Pulls the VM info for later
$vm=Get-AzVM -ResourceGroupName $rgname -Name $vmname
#add a new VM data disk
Add-AzVMDataDisk -CreateOption empty -DiskSizeInGB $size -Name $vmname-$datadiskname -VM $vm -Caching None -WriteAccelerator:$true -lun $lunid
#Updates the VM with the disk config - does not require a reboot
Update-AzVM -ResourceGroupName $rgname -VM $vm

```

### **Enabling Write Accelerator on an existing Azure disk using PowerShell**

You can use this script to enable Write Accelerator on an existing disk. Replace `myVM`, `myWAVMs`, and `test-log001` with values appropriate for your specific deployment. The script adds Write Accelerator to an existing disk where the value for `$newstatus` is set to '\$true'. Using the value '\$false' will disable Write Accelerator on a given disk.

```

#Specify your VM Name
$vmName="myVM"
#Specify your Resource Group
$rgName = "myWAVMs"
#data disk name
$datadiskname = "test-log001"
#new Write Accelerator status ($true for enabled, $false for disabled)
$newstatus = $true
#Pulls the VM info for later
$vm=Get-AzVM -ResourceGroupName $rgname -Name $vmname
#add a new VM data disk
Set-AzVMDataDisk -VM $vm -Name $datadiskname -Caching None -WriteAccelerator:$newstatus
#Updates the VM with the disk config - does not require a reboot
Update-AzVM -ResourceGroupName $rgname -VM $vm

```

## NOTE

Executing the script above will detach the disk specified, enable Write Accelerator against the disk, and then attach the disk again

## Enabling Write Accelerator using the Azure portal

You can enable Write Accelerator via the portal where you specify your disk caching settings:

| LUN | NAME    | SIZE     | STORAGE ACCOUNT TYPE | ENCRYPTION  | HOST CACHING                  |
|-----|---------|----------|----------------------|-------------|-------------------------------|
| 0   | WADisk1 | 1023 GiB | Premium_LRS          | Not enabled | Read-only + Write Accelerator |
| 1   | WADisk2 | 1023 GiB | Premium_LRS          | Not enabled | None + Write Accelerator      |

## Enabling Write Accelerator using the Azure CLI

You can use the [Azure CLI](#) to enable Write Accelerator.

To enable Write Accelerator on an existing disk, use [az vm update](#), you may use the following examples if you replace the diskName, VMName, and ResourceGroup with your own values:

```
az vm update -g group1 -n vm1 -write-accelerator 1=true
```

To attach a disk with Write Accelerator enabled use [az vm disk attach](#), you may use the following example if you substitute in your own values: `az vm disk attach -g group1 -vm-name vm1 -disk d1 --enable-write-accelerator`

To disable Write Accelerator, use [az vm update](#), setting the properties to false:

```
az vm update -g group1 -n vm1 -write-accelerator 0=false 1=false
```

## Enabling Write Accelerator using Rest APIs

To deploy through Azure Rest API, you need to install the Azure armclient.

### Install armclient

To run armclient, you need to install it through Chocolatey. You can install it through cmd.exe or powershell. Use elevated rights for these commands ("Run as Administrator").

Using cmd.exe, run the following command:

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))"
& SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```

Using Power Shell, run the following command:

```
Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Now you can install the armclient by using the following command in either cmd.exe or PowerShell

```
choco install armclient
```

## Getting your current VM configuration

To change the attributes of your disk configuration, you first need to get the current configuration in a JSON file.

You can get the current configuration by executing the following command:

```
armclient GET /subscriptions/<<subscription-ID>>/resourceGroups/<<ResourceGroup>>/providers/Microsoft.Compute/virtualMachines/<<virtualmachinename>>?api-version=2017-12-01 <<filename.json>>
```

Replace the terms within '<< >>' with your data, including the file name the JSON file should have.

The output could look like:

```
{
 "properties": {
 "vmId": "2444c93e-f8bb-4a20-af2d-1658d9dbbbc",
 "hardwareProfile": {
 "vmSize": "Standard_M64s"
 },
 "storageProfile": {
 "imageReference": {
 "publisher": "SUSE",
 "offer": "SLES-SAP",
 "sku": "12-SP3",
 "version": "latest"
 },
 "osDisk": {
 "osType": "Linux",
 "name": "mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a",
 "createOption": "FromImage",
 "caching": "ReadWrite",
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a"
 },
 "diskSizeGB": 30
 },
 "dataDisks": [
 {
 "lun": 0,
 "name": "data1",
 "createOption": "Attach",
 "caching": "None",
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/data1"
 },
 "diskSizeGB": 1023
 },
 {
 "lun": 1,
 "name": "log1",
 "createOption": "Attach",
 "caching": "None",
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/data2"
 }
 }
]
 }
 }
}
```

```

 },
 "diskSizeGB": 1023
 }
]
},
"osProfile": {
 "computerName": "mylittlesapVM",
 "adminUsername": "pl",
 "linuxConfiguration": {
 "disablePasswordAuthentication": false
 },
 "secrets": []
},
"networkProfile": {
 "networkInterfaces": [
 {
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Network/networkInterfaces/mylittlesap518"
 }
]
},
"diagnosticsProfile": {
 "bootDiagnostics": {
 "enabled": true,
 "storageUri": "https://mylittlesapdiag895.blob.core.windows.net/"
 }
},
"provisioningState": "Succeeded"
},
"type": "Microsoft.Compute/virtualMachines",
"location": "westeurope",
"id": "/subscriptions/XXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/virtualMachines/mylittlesapVM",
"name": "mylittlesapVM"

```

Next, update the JSON file and to enable Write Accelerator on the disk called 'log1'. This can be accomplished by adding this attribute into the JSON file after the cache entry of the disk.

```

{
 "lun": 1,
 "name": "log1",
 "createOption": "Attach",
 "caching": "None",
 "writeAcceleratorEnabled": true,
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/data2"
 },
 "diskSizeGB": 1023
}

```

Then update the existing deployment with this command:

```
armclient PUT /subscriptions/<<subscription-ID</>>/resourceGroups/<<ResourceGroup>>/providers/Microsoft.Compute/virtualMachines/<<virtualmachinename>>?api-version=2017-12-01 @<<filename.json>>
```

The output should look like the one below. You can see that Write Accelerator enabled for one disk.

```
{
 "properties": {
 "osProfile": {
 "computerName": "mylittlesapVM",
 "adminUsername": "pl",
 "linuxConfiguration": {
 "disablePasswordAuthentication": false
 },
 "secrets": []
 },
 "networkProfile": {
 "networkInterfaces": [
 {
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Network/networkInterfaces/mylittlesap518"
 }
]
 },
 "diagnosticsProfile": {
 "bootDiagnostics": {
 "enabled": true,
 "storageUri": "https://mylittlesapdiag895.blob.core.windows.net/"
 }
 },
 "provisioningState": "Succeeded"
 },
 "type": "Microsoft.Compute/virtualMachines",
 "location": "westeurope",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/virtualMachines/mylittlesapVM",
 "name": "mylittlesapVM"
}
```

```
 "vmSize": "Standard_M64s"
 },
 "storageProfile": {
 "imageReference": {
 "publisher": "SUSE",
 "offer": "SLES-SAP",
 "sku": "12-SP3",
 "version": "latest"
 },
 "osDisk": {
 "osType": "Linux",
 "name": "mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a",
 "createOption": "FromImage",
 "caching": "ReadWrite",
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a"
 },
 "diskSizeGB": 30
 },
 "dataDisks": [
 {
 "lun": 0,
 "name": "data1",
 "createOption": "Attach",
 "caching": "None",
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/data1"
 },
 "diskSizeGB": 1023
 },
 {
 "lun": 1,
 "name": "log1",
 "createOption": "Attach",
 "caching": "None",
 "writeAcceleratorEnabled": true,
 "managedDisk": {
 "storageAccountType": "Premium_LRS",
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/disks/data2"
 },
 "diskSizeGB": 1023
 }
]
 },
 "osProfile": {
 "computerName": "mylittlesapVM",
 "adminUsername": "pl",
 "linuxConfiguration": {
 "disablePasswordAuthentication": false
 },
 "secrets": []
 },
 "networkProfile": {
 "networkInterfaces": [
 {
 "id": "/subscriptions/XXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Network/networkInterfaces/mylittlesap518"
 }
]
 }
}
```

```
],
 },
 "diagnosticsProfile": {
 "bootDiagnostics": {
 "enabled": true,
 "storageUri": "https://mylittlesapdiag895.blob.core.windows.net/"
 }
 },
 "provisioningState": "Succeeded"
},
"type": "Microsoft.Compute/virtualMachines",
"location": "westeurope",
"id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittlesap/providers/Microsoft.Compute/vir
tualMachines/mylittlesapVM",
"name": "mylittlesapVM"
```

Once you've made this change, the drive should be supported by Write Accelerator.

# Using Azure ultra disks

11/15/2019 • 8 minutes to read • [Edit Online](#)

Azure ultra disks offer high throughput, high IOPS, and consistent low latency disk storage for Azure IaaS virtual machines (VMs). This new offering provides top of the line performance at the same availability levels as our existing disks offerings. One major benefit of ultra disks is the ability to dynamically change the performance of the SSD along with your workloads without the need to restart your VMs. Ultra disks are suited for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads.

## GA scope and limitations

For now, ultra disks have additional limitations, they are as follows:

- Are supported in the following regions, with a varying number of availability zones per region:
  - East US 2
  - East US
  - West US 2
  - SouthEast Asia
  - North Europe
  - West Europe
  - UK South
- Can only be used with availability zones (availability sets and single VM deployments outside of zones will not have the ability to attach an ultra disk)
- Are only supported on the following VM series:
  - [ESv3](#)
  - [DSv3](#)
  - FSv2
  - [M](#)
  - [Mv2](#)
- Not every VM size is available in every supported region with ultra disks
- Are only available as data disks and only support 4k physical sector size. Due to the 4K native sector size of Ultra Disk, there are some applications that won't be compatible with ultra disks. One example would be Oracle Database, which requires release 12.2 or later in order to support ultra disks.
- Can only be created as empty disks
- Do not yet support disk snapshots, VM images, availability sets, and Azure disk encryption
- Do not yet support integration with Azure Backup or Azure Site Recovery
- The current maximum limit for IOPS on GA VMs is 80,000.
- If you would like to participate in a limited preview of a VM that can accomplish 160,000 IOPS with ultra disks, please email [UltraDiskFeedback@microsoft.com](mailto:UltraDiskFeedback@microsoft.com)

## Determine VM size and region availability

To leverage ultra disks, you need to determine which availability zone you are in. Not every region supports every VM size with ultra disks. To determine if your region, zone, and VM size support ultra disks, run either of the following commands, make sure to replace the **region**, **vmSize**, and **subscription** values first:

CLI:

```

$subscription = "<yourSubID>"
example value is southeastasia
$region = "<yourLocation>"
example value is Standard_E64s_v3
$vmSize = "<yourVMSize>

az vm list-skus --resource-type virtualMachines --location $region --query "[?
name=='$vmSize'].locationInfo[0].zoneDetails[0].Name" --subscription $subscription

```

PowerShell:

```

$region = "southeastasia"
$vmSize = "Standard_E64s_v3"
(Get-AzComputeResourceSku | where {$_.Locations.Contains($region) -and ($_.Name -eq $vmSize) -and
$_.LocationInfo[0].ZoneDetails.Count -gt 0})[0].LocationInfo[0].ZoneDetails

```

The response will be similar to the form below, where X is the zone to use for deploying in your chosen region. X could be either 1, 2, or 3.

Preserve the **Zones** value, it represents your availability zone and you will need it in order to deploy an Ultra disk.

| RESOURCETYPE | NAME         | LOCATION | ZONES | RESTRICTION | CAPABILITY | VALUE |
|--------------|--------------|----------|-------|-------------|------------|-------|
| disks        | UltraSSD_LRS | eastus2  | X     |             |            |       |

#### NOTE

If there was no response from the command, then the selected VM size is not supported with ultra disks in the selected region.

Now that you know which zone to deploy to, follow the deployment steps in this article to either deploy a VM with an ultra disk attached or attach an ultra disk to an existing VM.

## Deploy an ultra disk using Azure Resource Manager

First, determine the VM size to deploy. For a list of supported VM sizes, see [GA scope and limitations](#) section.

If you would like to create a VM with multiple ultra disks, refer to the sample [Create a VM with multiple ultra disks](#).

If you intend to use your own template, make sure that **apiVersion** for `Microsoft.Compute/virtualMachines` and `Microsoft.Compute/Disks` is set as `2018-06-01` (or later).

Set the disk sku to **UltraSSD\_LRS**, then set the disk capacity, IOPS, availability zone, and throughput in MBps to create an ultra disk.

Once the VM is provisioned, you can partition and format the data disks and configure them for your workloads.

## Deploy an ultra disk using the Azure portal

This section covers deploying a virtual machine equipped with an ultra disk as a data disk. It assumes you have familiarity with deploying a virtual machine, if you do not, see our [Quickstart: Create a Windows virtual machine in the Azure portal](#).

- Sign in to the [Azure portal](#) and navigate to deploy a virtual machine (VM).
- Make sure to choose a [supported VM size and region](#).
- Select **Availability zone** in **Availability options**.

- Fill in the remaining entries with selections of your choice.
- Select **Disks**.

The screenshot shows the 'Create a virtual machine' blade in the Microsoft Azure portal. The 'Disks' tab is active. In the 'Instance details' section, the 'Virtual machine name' is 'myVMName', 'Region' is '(US) West US 2', 'Availability options' is 'Availability zone', and 'Availability zone' is set to '1'. The 'Image' dropdown shows 'Ubuntu Server 18.04 LTS' with a link to 'Browse all public and private images'. Under 'Azure Spot instance', the radio button for 'No' is selected. The 'Size' dropdown is set to 'Standard D2s v3' with a note about 2 vcpus and 8 GiB memory. A red box surrounds the 'Enable Ultra Disk compatibility' checkbox, which is checked ('Yes'). Another red box surrounds the 'Standard D2s v3' size selection.

- On the Disks blade, select **Yes** for **Enable Ultra Disk compatibility**.
- Select **Create and attach a new disk** to attach an ultra disk now.

### Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

**Disk options**

**OS disk type \***

**Enable Ultra Disk compatibility**  Yes  No

**Data disks**

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

| LUN                                          | Name | Size (GiB)                              | Disk type | Host caching |
|----------------------------------------------|------|-----------------------------------------|-----------|--------------|
|                                              |      |                                         |           |              |
| <a href="#">Create and attach a new disk</a> |      | <a href="#">Attach an existing disk</a> |           |              |

- On the **Create a new disk** blade, enter a name, then select **Change size**.
- Change the **Account type** to **Ultra Disk**.
- Change the values of **Custom disk size (GiB)**, **Disk IOPS**, and **Disk throughput** to ones of your choice.
- Select **OK** in both blades.
- Continue with the VM deployment, it will be the same as you would deploy any other VM.

## Attach an ultra disk using the Azure portal

Alternatively, if your existing VM is in a region/availability zone that is capable of using ultra disks, you can make use of ultra disks without having to create a new VM. By enabling ultra disks on your existing VM, then attaching them as data disks.

- Navigate to your VM and select **Disks**.
- Select **Edit**.

- Select **Yes** for **Enable Ultra Disk compatibility**.

- Select **Save**.
- Select **Add data disk** then in the dropdown for **Name** select **Create disk**.

- Fill in a name for your new disk, then select **Change size**.
- Change the **Account type** to **Ultra Disk**.
- Change the values of **Custom disk size (GiB)**, **Disk IOPS**, and **Disk throughput** to ones of your choice.

- Select **OK** then select **Create**.

Create managed disk

Create a new disk to store applications and data on your VM. Disk pricing varies based on storage type, and number of transactions.

Disk name \* ⓘ  
new-ultra

Resource group \* ⓘ  
exampleressourcegroup  
Create new

Location  
West US 2

Availability zone ⓘ  
1

Source type ⓘ  
None

Size \* ⓘ  
**1024 GiB**  
Ultra Disk, 2048 IOPS, 8 MB/s  
Change size

Account type ⓘ  
**Ultra Disk**

| Max size  | Disk tier | Max IOPS | Max throughput |
|-----------|-----------|----------|----------------|
| 65536 GiB | U         | 160000   | 2000           |

Create a custom size  
Enter the size of the disk you would like to create. You will be charged the same rate for your provisioned disk, regardless of how much of the disk space is being used For example, a 200 GiB disk is provisioned on a 256 GiB disk, so you would be billed for the 256 GiB provisioned.

Custom disk size (GiB) \*  
1024

Disk IOPS \*  
2048

Disk throughput (MB/s) \*  
8

- After you are returned to your disk's blade, select **Save**.

Save Discard Refresh Encryption Swap OS Disk

Managed disks created since June 10, 2017 are encrypted at rest with Storage Service Encryption (SSE). You may also want to enable Azure Disk Encryption.

Disk settings

Enable Ultra Disk compatibility ⓘ  
 Yes  No

OS disk

| Name                                                    | Size   | Storage account type | Encryption  | Host caching |
|---------------------------------------------------------|--------|----------------------|-------------|--------------|
| newfinalbuttontest_OsDisk_1_b2fd08ca503c41acb6a040ac... | 30 GiB | Premium SSD          | Not enabled | Read/write   |

Data disks

| LUN | Name             | Size     | Storage account type | Encryption  | Host caching |
|-----|------------------|----------|----------------------|-------------|--------------|
| 0   | <b>new-ultra</b> | 1024 GiB | Ultra Disk           | Not enabled | None         |
| 1   | ultra-new        | 1024 GiB | Ultra Disk           | Not enabled | None         |

+ Add data disk

## Adjust the performance of an ultra disk using the Azure portal

Ultra disks offer a unique capability that allows you to adjust their performance. You can make these adjustments from the Azure portal, on the disks themselves.

- Navigate to your VM and select **Disks**.
- Select the ultra disk you'd like to modify the performance of.

Search (Ctrl+ /) <<

Edit Refresh Encryption

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems

**Settings**

Networking  
Disks **(highlighted)**  
Size  
Security  
Extensions  
Continuous delivery (Preview)  
Availability + scaling

Managed disks created since June 10, 2023

The virtual machine must be stopped/

Disk settings

Enable Ultra Disk compatibility *i*  
 Yes  No

OS disk

Name: ultravm\_OsDisk\_1\_12f247eec85d4ee1a7

Data disks

| LUN | Name             |
|-----|------------------|
| 0   | <b>new-ultra</b> |
| 1   | ultra-new        |

- Select **Configuration** and then make your modifications.
- Select **Save**.

Search (Ctrl+ /) <<

Save Discard

Overview  
Activity log  
Access control (IAM)  
Tags  
**Settings**  
**Configuration** **(highlighted)**  
Disk Export  
Properties  
Locks  
Export template

Account type *i*  
Ultra Disk  
*i* Changing account type for Ultra Disks is not currently supported.

|                          |      |
|--------------------------|------|
| Size (GiB) *             | 1050 |
| Disk IOPS *              | 2548 |
| Disk throughput (MB/s) * | 10   |

Deploy an ultra disk using CLI

First, determine the VM size to deploy. See the [GA scope and limitations](#) section for a list of supported VM sizes.

You must create a VM that is capable of using ultra disks, in order to attach an ultra disk.

Replace or set the **\$vmname**, **\$rgname**, **\$diskname**, **\$location**, **\$password**, **\$user** variables with your own values. Set **\$zone** to the value of your availability zone that you got from the [start of this article](#). Then run the following CLI command to create an ultra enabled VM:

```
az vm create --subscription $subscription -n $vmname -g $rgname --image Win2016Datacenter --ultra-ssd-enabled true --zone $zone --authentication-type password --admin-password $password --admin-username $user --size Standard_D4s_V3 --location $location
```

## Create an ultra disk using CLI

Now that you have a VM that is capable of attaching ultra disks, you can create and attach an ultra disk to it.

```
$location="eastus2"
$subscription="xxx"
$rgname="ultraRG"
$diskname="ssd1"
$vmname="ultravm1"
$zone=123

#create an ultra disk
az disk create `
--subscription $subscription `
-n $diskname `
-g $rgname `
--size-gb 4 `
--location $location `
--zone $zone `
--sku UltraSSD_LRS `
--disk-iops-read-write 1000 `
--disk-mbps-read-write 50
```

## Attach an ultra disk to a VM using CLI

Alternatively, if your existing VM is in a region/availability zone that is capable of using ultra disks, you can make use of ultra disks without having to create a new VM.

```
$rgName = "<yourResourceGroupName>"
$vmName = "<yourVMName>"
$diskName = "<yourDiskName>"
$subscriptionId = "<yourSubscriptionID>"

az vm disk attach -g $rgName --vm-name $vmName --disk $diskName --subscription $subscriptionId
```

## Adjust the performance of an ultra disk using CLI

Ultra disks offer a unique capability that allows you to adjust their performance, the following command depicts how to use this feature:

```
az disk update `
--subscription $subscription `
--resource-group $rgname `
--name $diskName `
--set diskIopsReadWrite=80000 `
--set diskMbpsReadWrite=800
```

# Deploy an ultra disk using PowerShell

First, determine the VM size to deploy. See the [GA scope and limitations](#) section for a list of supported VM sizes.

To use ultra disks, you must create a VM that is capable of using ultra disks. Replace or set the **\$resourcegroup** and **\$vmName** variables with your own values. Set **\$zone** to the value of your availability zone that you got from the [start of this article](#). Then run the following `New-AzVm` command to create an ultra enabled VM:

```
New-AzVm `
 -ResourceGroupName $resourcegroup `
 -Name $vmName `
 -Location "eastus2" `
 -Image "Win2016Datacenter" `
 -EnableUltraSSD `
 -size "Standard_D4s_v3" `
 -zone $zone
```

## Create an ultra disk using PowerShell

Now that you have a VM that is capable of using ultra disks, you can create and attach an ultra disk to it:

```
$diskconfig = New-AzDiskConfig `
 -Location 'EastUS2' `
 -DiskSizeGB 8 `
 -DiskIOPSReadWrite 1000 `
 -DiskMBpsReadWrite 100 `
 -AccountType UltraSSD_LRS `
 -CreateOption Empty `
 -zone $zone;

New-AzDisk `
 -ResourceGroupName $resourceGroup `
 -DiskName 'Disk02' `
 -Disk $diskconfig;
```

## Attach an ultra disk to a VM using PowerShell

Alternatively, if your existing VM is in a region/availability zone that is capable of using ultra disks, you can make use of ultra disks without having to create a new VM.

```
add disk to VM
$subscription = "<yourSubscriptionID>"
$resourceGroup = "<yourResourceGroup>"
$vmName = "<yourVMName>"
$diskName = "<yourDiskName>"
$lun = 1
Login-AzureRMAccount -SubscriptionId $subscription
$vm = Get-AzVM -ResourceGroupName $resourceGroup -Name $vmName
$disk = Get-AzDisk -ResourceGroupName $resourceGroup -Name $diskName
$vm = Add-AzVMDataDisk -VM $vm -Name $diskName -CreateOption Attach -ManagedDiskId $disk.Id -Lun $lun
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
```

## Adjust the performance of an ultra disk using PowerShell

Ultra disks have a unique capability that allows you to adjust their performance, the following command is an example that adjusts the performance without having to detach the disk:

```
$diskupdateconfig = New-AzDiskUpdateConfig -DiskMBpsReadWrite 2000
Update-AzDisk -ResourceGroupName $resourceGroup -DiskName $diskName -DiskUpdate $diskupdateconfig
```

## Next steps

If you would like to try the new disk type [request access with this survey](#).

# Benchmarking a disk

1/7/2020 • 8 minutes to read • [Edit Online](#)

Benchmarking is the process of simulating different workloads on your application and measuring the application performance for each workload. Using the steps described in the [designing for high performance article](#), you have gathered the application performance requirements. By running benchmarking tools on the VMs hosting the application, you can determine the performance levels that your application can achieve with Premium Storage. In this article, we provide you examples of benchmarking a Standard DS14 VM provisioned with Azure Premium Storage disks.

We have used common benchmarking tools lometer and FIO, for Windows and Linux respectively. These tools spawn multiple threads simulating a production like workload, and measure the system performance. Using the tools you can also configure parameters like block size and queue depth, which you normally cannot change for an application. This gives you more flexibility to drive the maximum performance on a high scale VM provisioned with premium disks for different types of application workloads. To learn more about each benchmarking tool visit [lometer](#) and [FIO](#).

To follow the examples below, create a Standard DS14 VM and attach 11 Premium Storage disks to the VM. Of the 11 disks, configure 10 disks with host caching as "None" and stripe them into a volume called NoCacheWrites. Configure host caching as "ReadOnly" on the remaining disk and create a volume called CacheReads with this disk. Using this setup, you are able to see the maximum Read and Write performance from a Standard DS14 VM. For detailed steps about creating a DS14 VM with premium SSDs, go to [Designing for high performance](#).

## *Warming up the Cache*

The disk with ReadOnly host caching are able to give higher IOPS than the disk limit. To get this maximum read performance from the host cache, first you must warm up the cache of this disk. This ensures that the Read IOs that the benchmarking tool will drive on CacheReads volume, actually hits the cache, and not the disk directly. The cache hits result in additional IOPS from the single cache enabled disk.

### **IMPORTANT**

You must warm up the cache before running benchmarking, every time VM is rebooted.

## Tools

### **lometer**

[Download the lometer tool](#) on the VM.

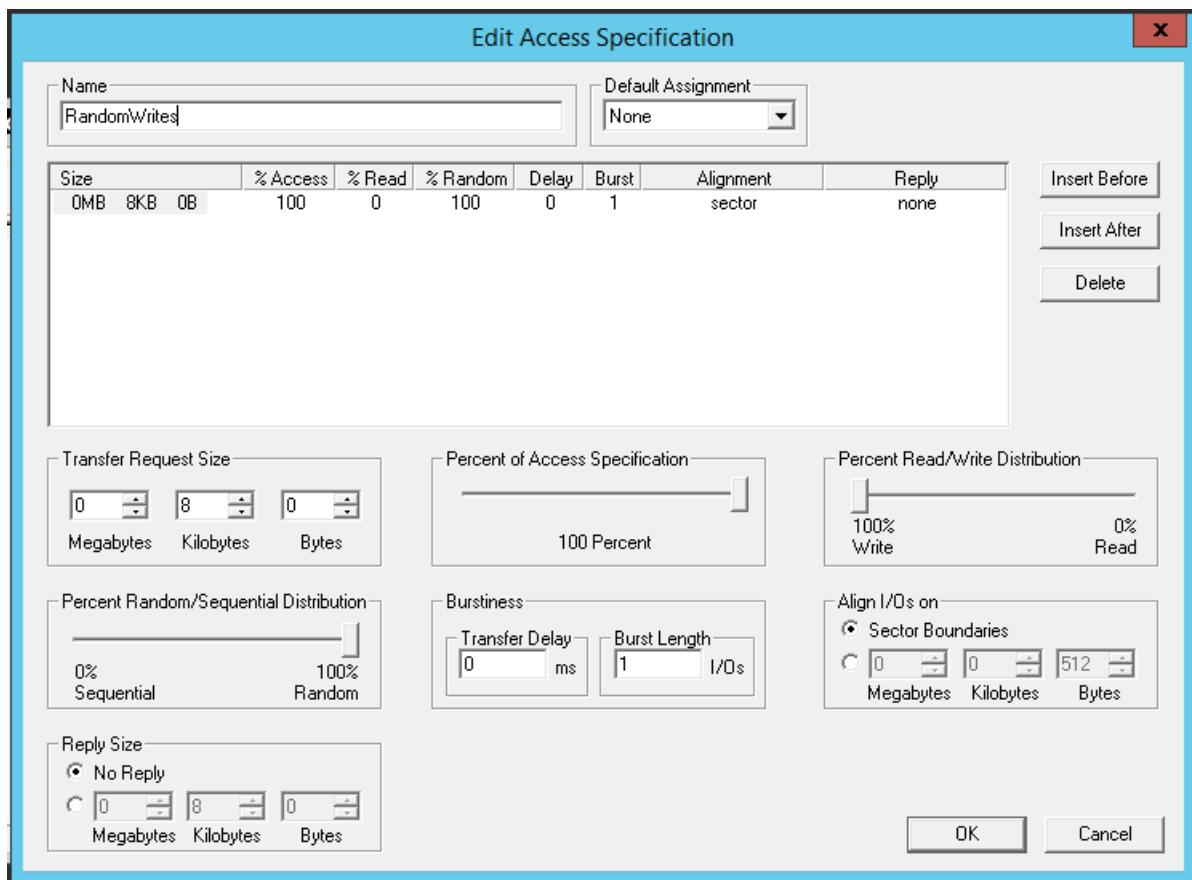
### **Test file**

lometer uses a test file that is stored on the volume on which you run the benchmarking test. It drives Reads and Writes on this test file to measure the disk IOPS and Throughput. lometer creates this test file if you have not provided one. Create a 200 GB test file called iobw.tst on the CacheReads and NoCacheWrites volumes.

### **Access specifications**

The specifications, request IO size, % read/write, % random/sequential are configured using the "Access Specifications" tab in lometer. Create an access specification for each of the scenarios described below. Create the access specifications and "Save" with an appropriate name like – RandomWrites\_8K, RandomReads\_8K. Select the corresponding specification when running the test scenario.

An example of access specifications for maximum Write IOPS scenario is shown below,



#### Maximum IOPS test specifications

To demonstrate maximum IOPs, use smaller request size. Use 8K request size and create specifications for Random Writes and Reads.

| ACCESS SPECIFICATION | REQUEST SIZE | RANDOM % | READ % |
|----------------------|--------------|----------|--------|
| RandomWrites_8K      | 8K           | 100      | 0      |
| RandomReads_8K       | 8K           | 100      | 100    |

#### Maximum throughput test specifications

To demonstrate maximum Throughput, use larger request size. Use 64 K request size and create specifications for Random Writes and Reads.

| ACCESS SPECIFICATION | REQUEST SIZE | RANDOM % | READ % |
|----------------------|--------------|----------|--------|
| RandomWrites_64K     | 64 K         | 100      | 0      |
| RandomReads_64K      | 64 K         | 100      | 100    |

#### Run the Iometer test

Perform the steps below to warm up cache

1. Create two access specifications with values shown below,

| NAME             | REQUEST SIZE | RANDOM % | READ % |
|------------------|--------------|----------|--------|
| RandomWrites_1MB | 1 MB         | 100      | 0      |
| RandomReads_1MB  | 1 MB         | 100      | 100    |

2. Run the lometer test for initializing cache disk with following parameters. Use three worker threads for the target volume and a queue depth of 128. Set the "Run time" duration of the test to 2 hrs on the "Test Setup" tab.

| SCENARIO              | TARGET VOLUME | NAME             | DURATION |
|-----------------------|---------------|------------------|----------|
| Initialize Cache Disk | CacheReads    | RandomWrites_1MB | 2 hrs    |

3. Run the lometer test for warming up cache disk with following parameters. Use three worker threads for the target volume and a queue depth of 128. Set the "Run time" duration of the test to 2 hrs on the "Test Setup" tab.

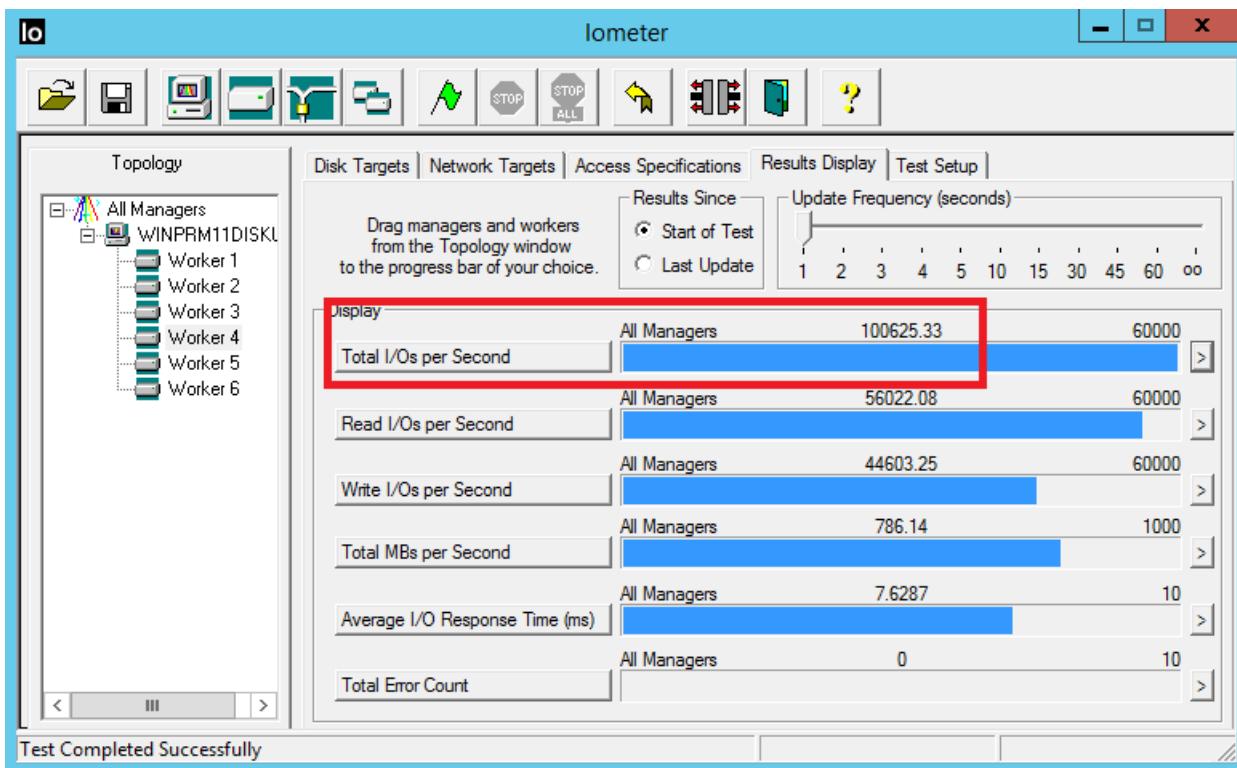
| SCENARIO           | TARGET VOLUME | NAME            | DURATION |
|--------------------|---------------|-----------------|----------|
| Warm up Cache Disk | CacheReads    | RandomReads_1MB | 2 hrs    |

After cache disk is warmed up, proceed with the test scenarios listed below. To run the lometer test, use at least three worker threads for **each** target volume. For each worker thread, select the target volume, set queue depth and select one of the saved test specifications, as shown in the table below, to run the corresponding test scenario. The table also shows expected results for IOPS and Throughput when running these tests. For all scenarios, a small IO size of 8 KB and a high queue depth of 128 is used.

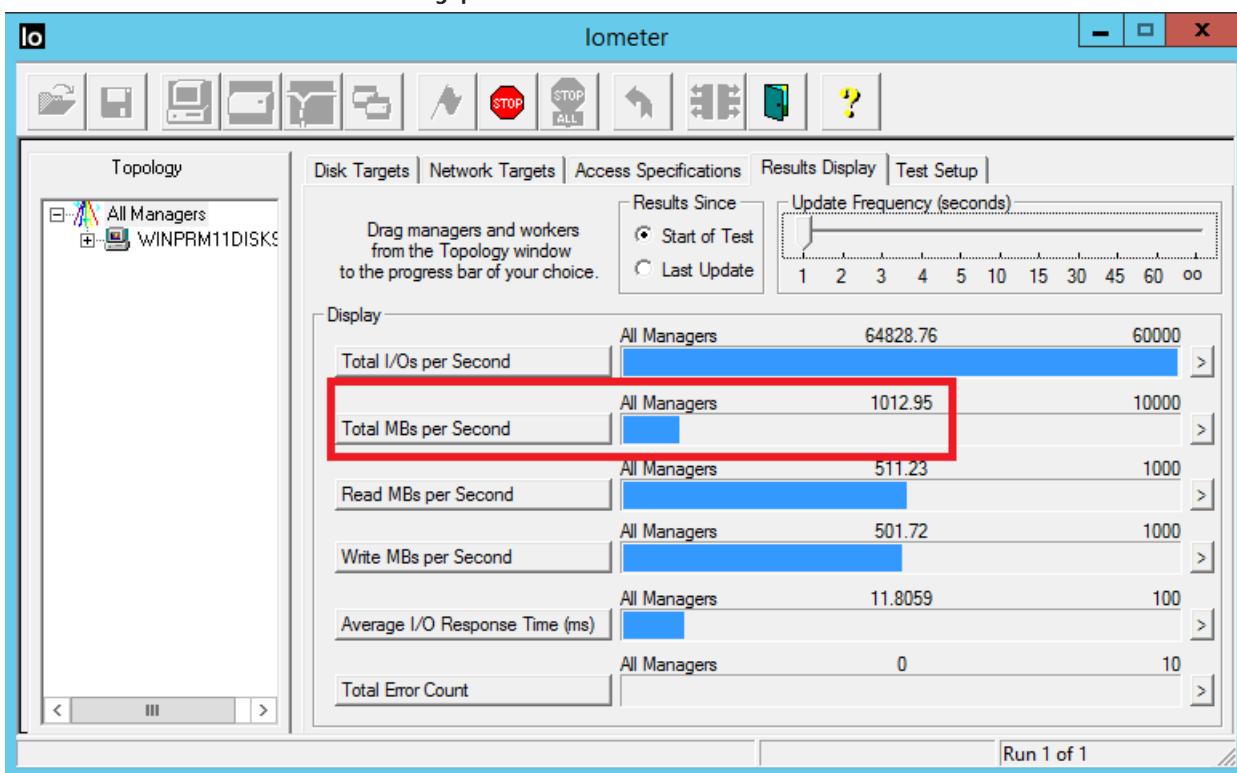
| TEST SCENARIO      | TARGET VOLUME   | NAME             | RESULT       |
|--------------------|-----------------|------------------|--------------|
| Max. Read IOPS     | CacheReads      | RandomWrites_8K  | 50,000 IOPS  |
| Max. Write IOPS    | NoCacheWrites   | RandomReads_8K   | 64,000 IOPS  |
| Max. Combined IOPS | CacheReads      | RandomWrites_8K  | 100,000 IOPS |
| NoCacheWrites      | RandomReads_8K  |                  |              |
| Max. Read MB/sec   | CacheReads      | RandomWrites_64K | 524 MB/sec   |
| Max. Write MB/sec  | NoCacheWrites   | RandomReads_64K  | 524 MB/sec   |
| Combined MB/sec    | CacheReads      | RandomWrites_64K | 1000 MB/sec  |
| NoCacheWrites      | RandomReads_64K |                  |              |

Below are screenshots of the lometer test results for combined IOPS and Throughput scenarios.

#### Combined reads and writes maximum IOPS



Combined reads and writes maximum throughput



## FIO

FIO is a popular tool to benchmark storage on the Linux VMs. It has the flexibility to select different IO sizes, sequential or random reads and writes. It spawns worker threads or processes to perform the specified I/O operations. You can specify the type of I/O operations each worker thread must perform using job files. We created one job file per scenario illustrated in the examples below. You can change the specifications in these job files to benchmark different workloads running on Premium Storage. In the examples, we are using a Standard DS 14 VM running **Ubuntu**. Use the same setup described in the beginning of the Benchmarking section and warm up the cache before running the benchmarking tests.

Before you begin, [download FIO](#) and install it on your virtual machine.

Run the following command for Ubuntu,

```
apt-get install fio
```

We use four worker threads for driving Write operations and four worker threads for driving Read operations on the disks. The Write workers are driving traffic on the "nocache" volume, which has 10 disks with cache set to "None". The Read workers are driving traffic on the "readcache" volume, which has one disk with cache set to "ReadOnly".

#### Maximum write IOPS

Create the job file with following specifications to get maximum Write IOPS. Name it "fiowrite.ini".

```
[global]
size=30g
direct=1
iodepth=256
ioengine=libaio
bs=8k

[writer1]
rw=randwrite
directory=/mnt/nocache
[writer2]
rw=randwrite
directory=/mnt/nocache
[writer3]
rw=randwrite
directory=/mnt/nocache
[writer4]
rw=randwrite
directory=/mnt/nocache
```

Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

- A high queue depth of 256.
- A small block size of 8 KB.
- Multiple threads performing random writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fiowrite.ini
```

While the test runs, you are able to see the number of write IOPS the VM and Premium disks are delivering. As shown in the sample below, the DS14 VM is delivering its maximum write IOPS limit of 50,000 IOPS.

```
demo@DS-VM-Linux-Demo:~$ sudo fio --runtime 30 fiowrite.ini
[sudo] password for demo:
writer1: (g=0): rw=randwrite, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
writer2: (g=0): rw=randwrite, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
writer3: (g=0): rw=randwrite, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
writer4: (g=0): rw=randwrite, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
fio-2.1.11
Starting 4 processes
Jobs: 4 (f=4): [w(4)] [63.3% done] [0KB/396.4MB/0KB /s] [0/50.8K/0 iops] [eta 00m:11s]
```

#### Maximum read IOPS

Create the job file with following specifications to get maximum Read IOPS. Name it "ioread.ini".

```

[global]
size=30g
direct=1
iodepth=256
ioengine=libaio
bs=8k

[reader1]
rw=randread
directory=/mnt/readcache
[reader2]
rw=randread
directory=/mnt/readcache
[reader3]
rw=randread
directory=/mnt/readcache
[reader4]
rw=randread
directory=/mnt/readcache

```

Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

- A high queue depth of 256.
- A small block size of 8 KB.
- Multiple threads performing random writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fioread.ini
```

While the test runs, you are able to see the number of read IOPS the VM and Premium disks are delivering. As shown in the sample below, the DS 14 VM is delivering more than 64,000 Read IOPS. This is a combination of the disk and the cache performance.

```

demo@DS-VM-Linux-Demo:~$ sudo fio --runtime 30 fioread.ini
[sudo] password for demo:
reader1: (g=0): rw=randread, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
reader2: (g=0): rw=randread, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
reader3: (g=0): rw=randread, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
reader4: (g=0): rw=randread, bs=8K-8K/8K-8K/8K-8K, ioengine=libaio, iodepth=256
fio-2.1.11
Starting 4 processes
Jobs: 4 (f=4): [r(4)] [70.0% done] [514.8MB/0KB/0KB /s] [65.9K/0/0 iops] [eta 00m:09s]
```

#### **Maximum read and write IOPS**

Create the job file with following specifications to get maximum combined Read and Write IOPS. Name it "fioreadwrite.ini".

```

[global]
size=30g
direct=1
iodepth=128
ioengine=libaio
bs=4k

[reader1]
rw=randread
directory=/mnt/readcache
[reader2]
rw=randread
directory=/mnt/readcache
[reader3]
rw=randread
directory=/mnt/readcache
[reader4]
rw=randread
directory=/mnt/readcache

[writer1]
rw=randwrite
directory=/mnt/nocache
rate_iops=12500
[writer2]
rw=randwrite
directory=/mnt/nocache
rate_iops=12500
[writer3]
rw=randwrite
directory=/mnt/nocache
rate_iops=12500
[writer4]
rw=randwrite
directory=/mnt/nocache
rate_iops=12500

```

Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

- A high queue depth of 128.
- A small block size of 4 KB.
- Multiple threads performing random reads and writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fioreadwrite.ini
```

While the test runs, you are able to see the number of combined read and write IOPS the VM and Premium disks are delivering. As shown in the sample below, the DS14 VM is delivering more than 100,000 combined Read and Write IOPS. This is a combination of the disk and the cache performance.

```

demo@DS-VM-Linux-Demo:~$ sudo fio --runtime 30 fioreadwrite.ini
reader1: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
reader2: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
reader3: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
reader4: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
writer1: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
writer2: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
writer3: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
writer4: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
fio-2.1.11
Starting 8 processes
Jobs: 8 (f=8), CR=50000/0 IOPS: [r(4),w(4)] [22.6% done] [251.2MB/183.3MB/0KB /s] [64.3K/46.1K/0 iops] [eta 00m:24s]
```

#### Maximum combined throughput

To get the maximum combined Read and Write Throughput, use a larger block size and large queue depth with

multiple threads performing reads and writes. You can use a block size of 64 KB and queue depth of 128.

## Next steps

Proceed to our article on [designing for high performance](#).

In that article, you create a checklist similar to your existing application for the prototype. Using Benchmarking tools you can simulate the workloads and measure performance on the prototype application. By doing so, you can determine which disk offering can match or surpass your application performance requirements. Then you can implement the same guidelines for your production application.

# Find and delete unattached Azure managed and unmanaged disks

11/13/2019 • 3 minutes to read • [Edit Online](#)

When you delete a virtual machine (VM) in Azure, by default, any disks that are attached to the VM aren't deleted. This feature helps to prevent data loss due to the unintentional deletion of VMs. After a VM is deleted, you will continue to pay for unattached disks. This article shows you how to find and delete any unattached disks and reduce unnecessary costs.

## Managed disks: Find and delete unattached disks

The following script looks for unattached [managed disks](#) by examining the value of the **ManagedBy** property. When a managed disk is attached to a VM, the **ManagedBy** property contains the resource ID of the VM. When a managed disk is unattached, the **ManagedBy** property is null. The script examines all the managed disks in an Azure subscription. When the script locates a managed disk with the **ManagedBy** property set to null, the script determines that the disk is unattached.

### IMPORTANT

First, run the script by setting the **deleteUnattachedDisks** variable to 0. This action lets you find and view all the unattached managed disks.

After you review all the unattached disks, run the script again and set the **deleteUnattachedDisks** variable to 1. This action lets you delete all the unattached managed disks.

```
Set deleteUnattachedDisks=1 if you want to delete unattached Managed Disks
Set deleteUnattachedDisks=0 if you want to see the Id of the unattached Managed Disks
$deleteUnattachedDisks=0
$managedDisks = Get-AzDisk
foreach ($md in $managedDisks) {
 # ManagedBy property stores the Id of the VM to which Managed Disk is attached to
 # If ManagedBy property is $null then it means that the Managed Disk is not attached to a VM
 if($md.ManagedBy -eq $null){
 if($deleteUnattachedDisks -eq 1){
 Write-Host "Deleting unattached Managed Disk with Id: $($md.Id)"
 $md | Remove-AzDisk -Force
 Write-Host "Deleted unattached Managed Disk with Id: $($md.Id) "
 }else{
 $md.Id
 }
 }
}
```

## Unmanaged disks: Find and delete unattached disks

Unmanaged disks are VHD files that are stored as [page blobs](#) in [Azure storage accounts](#). The following script looks for unattached unmanaged disks (page blobs) by examining the value of the **LeaseStatus** property. When an unmanaged disk is attached to a VM, the **LeaseStatus** property is set to **Locked**. When an unmanaged disk is unattached, the **LeaseStatus** property is set to **Unlocked**. The script examines all the unmanaged disks in all the Azure storage accounts in an Azure subscription. When the script locates an unmanaged disk with a **LeaseStatus** property set to **Unlocked**, the script determines that the disk is unattached.

## IMPORTANT

First, run the script by setting the **deleteUnattachedVHDs** variable to 0. This action lets you find and view all the unattached unmanaged VHDs.

After you review all the unattached disks, run the script again and set the **deleteUnattachedVHDs** variable to 1. This action lets you delete all the unattached unmanaged VHDs.

```
Set deleteUnattachedVHDs=1 if you want to delete unattached VHDs
Set deleteUnattachedVHDs=0 if you want to see the Uri of the unattached VHDs
$deleteUnattachedVHDs=0
$storageAccounts = Get-AzStorageAccount
foreach($storageAccount in $storageAccounts){
 $storageKey = (Get-AzStorageAccountKey -ResourceGroupName $storageAccount.ResourceGroupName -Name
$storageAccount.StorageAccountName)[0].Value
 $context = New-AzStorageContext -StorageAccountName $storageAccount.StorageAccountName -StorageAccountKey
$storageKey
 $containers = Get-AzStorageContainer -Context $context
 foreach($container in $containers){
 $blobs = Get-AzStorageBlob -Container $container.Name -Context $context
 #Fetch all the Page blobs with extension .vhd as only Page blobs can be attached as disk to Azure VMs
 $blobs | Where-Object {$_.BlobType -eq 'PageBlob' -and $_.Name.EndsWith('.vhd')} | ForEach-Object {
 #If a Page blob is not attached as disk then LeaseStatus will be unlocked
 if($_.ICloudBlob.Properties.LeaseStatus -eq 'Unlocked'){
 if($deleteUnattachedVHDs -eq 1){
 Write-Host "Deleting unattached VHD with Uri: $($_.ICloudBlob.Uri.AbsoluteUri)"
 $_ | Remove-AzStorageBlob -Force
 Write-Host "Deleted unattached VHD with Uri: $($_.ICloudBlob.Uri.AbsoluteUri)"
 }
 else{
 $_.ICloudBlob.Uri.AbsoluteUri
 }
 }
 }
 }
}
```

## Next steps

For more information, see [Delete storage account](#) and [Identify Orphaned Disks Using PowerShell](#)

# Frequently asked questions about Azure IaaS VM disks and managed and unmanaged premium disks

12/10/2019 • 21 minutes to read • [Edit Online](#)

This article answers some frequently asked questions about Azure Managed Disks and Azure Premium SSD disks.

## Managed Disks

### **What is Azure Managed Disks?**

Managed Disks is a feature that simplifies disk management for Azure IaaS VMs by handling storage account management for you. For more information, see the [Managed Disks overview](#).

### **If I create a standard managed disk from an existing VHD that's 80 GB, how much will that cost me?**

A standard managed disk created from an 80-GB VHD is treated as the next available standard disk size, which is an S10 disk. You're charged according to the S10 disk pricing. For more information, see the [pricing page](#).

### **Are there any transaction costs for standard managed disks?**

Yes. You're charged for each transaction. For more information, see the [pricing page](#).

### **For a standard managed disk, will I be charged for the actual size of the data on the disk or for the provisioned capacity of the disk?**

You're charged based on the provisioned capacity of the disk. For more information, see the [pricing page](#).

### **How is pricing of premium managed disks different from unmanaged disks?**

The pricing of premium managed disks is the same as unmanaged premium disks.

### **Can I change the storage account type (Standard or Premium) of my managed disks?**

Yes. You can change the storage account type of your managed disks by using the Azure portal, PowerShell, or the Azure CLI.

### **Can I use a VHD file in an Azure storage account to create a managed disk with a different subscription?**

Yes.

### **Can I use a VHD file in an Azure storage account to create a managed disk in a different region?**

No.

### **Are there any scale limitations for customers that use managed disks?**

Managed Disks eliminates the limits associated with storage accounts. However, the maximum limit is 50,000 managed disks per region and per disk type for a subscription.

### **Can I take an incremental snapshot of a managed disk?**

No. The current snapshot capability makes a full copy of a managed disk.

### **Can VMs in an availability set consist of a combination of managed and unmanaged disks?**

No. The VMs in an availability set must use either all managed disks or all unmanaged disks. When you create an availability set, you can choose which type of disks you want to use.

## **Is Managed Disks the default option in the Azure portal?**

Yes.

## **Can I create an empty managed disk?**

Yes. You can create an empty disk. A managed disk can be created independently of a VM, for example, without attaching it to a VM.

## **What is the supported fault domain count for an availability set that uses Managed Disks?**

Depending on the region where the availability set that uses Managed Disks is located, the supported fault domain count is 2 or 3.

## **How is the standard storage account for diagnostics set up?**

You set up a private storage account for VM diagnostics.

## **What kind of Role-Based Access Control support is available for Managed Disks?**

Managed Disks supports three key default roles:

- Owner: Can manage everything, including access
- Contributor: Can manage everything except access
- Reader: Can view everything, but can't make changes

## **Is there a way that I can copy or export a managed disk to a private storage account?**

You can generate a read-only shared access signature (SAS) URI for the managed disk and use it to copy the contents to a private storage account or on-premises storage. You can use the SAS URI using the Azure portal, Azure PowerShell, the Azure CLI, or [AzCopy](#)

## **Can I create a copy of my managed disk?**

Customers can take a snapshot of their managed disks and then use the snapshot to create another managed disk.

## **Are unmanaged disks still supported?**

Yes, both unmanaged and managed disks are supported. We recommend that you use managed disks for new workloads and migrate your current workloads to managed disks.

## **Can I co-locate unmanaged and managed disks on the same VM?**

No.

## **If I create a 128-GB disk and then increase the size to 130 gibabytes (GiB), will I be charged for the next disk size (256 GiB)?**

Yes.

## **Can I create locally redundant storage, geo-redundant storage, and zone-redundant storage managed disks?**

Azure Managed Disks currently supports only locally redundant storage managed disks.

## **Can I shrink or downsize my managed disks?**

No. This feature is not supported currently.

## **Can I break a lease on my disk?**

No. This is not supported currently as a lease is present to prevent accidental deletion when the disk is being used.

## **Can I change the computer name property when a specialized (not created by using the System Preparation tool or generalized) operating system disk is used to provision a VM?**

No. You can't update the computer name property. The new VM inherits it from the parent VM, which was used to create the operating system disk.

## **Where can I find sample Azure Resource Manager templates to create VMs with managed disks?**

- [List of templates using Managed Disks](#)
- <https://github.com/chagarw/MDPP>

## **When creating a disk from a blob, is there any continually existing relationship with that source blob?**

No, when the new disk is created it is a full standalone copy of that blob at that time and there is no connection between the two. If you like, once you've created the disk, the source blob may be deleted without affecting the newly created disk in any way.

## **Can I rename a managed or unmanaged disk after it has been created?**

For managed disks you cannot rename them. However, you may rename an unmanaged disk as long as it is not currently attached to a VHD or VM.

## **Can I use GPT partitioning on an Azure Disk?**

Generation 1 images can only use GPT partitioning on data disks, not OS disks. OS disks must use the MBR partition style.

[Generation 2 images](#) can use GPT partitioning on the OS disk as well as the data disks.

## **What disk types support snapshots?**

Premium SSD, standard SSD, and standard HDD support snapshots. For these three disk types, snapshots are supported for all disk sizes (including disks up to 32 TiB in size). Ultra disks do not support snapshots.

**What are Azure disk reservations?** Disk reservation is the option to purchase one year of disk storage in advance, reducing your total cost. For details regarding Azure disk reservations, see our article on the subject: [Understand how your reservation discount is applied to Azure Disk](#).

**What options does Azure disk reservation offer?** Azure disk reservation provides the option to purchase Premium SSDs in the specified SKUs from P30 (1 TiB) up to P80 (32 TiB) for a one-year term. There is no limitation on the minimum amount of disks necessary to purchase a disk reservation. Additionally, you can choose to pay with a single, upfront payment or monthly payments. There is no additional transactional cost applied for Premium SSD Managed Disks.

Reservations are made in the form of disks, not capacity. In other words, when you reserve a P80 (32 TiB) disk, you get a single P80 disk, you cannot then divide that specific reservation up into two smaller P70 (16 TiB) disks. You can, of course, reserve as many or as few disks as you like, including two separate P70 (16 TiB) disks.

**How is Azure disk reservation applied?** Disks reservation follows a model similar to reserved virtual machine (VM) instances. The difference being that a disk reservation cannot be applied to different SKUs, while a VM instance can. See [Save costs with Azure Reserved VM Instances](#) for more information on VM instances.

**Can I use my data storage purchased through Azure disks reservation across multiple regions?** Azure disks reservation are purchased for a specific region and SKU (like P30 in East US 2), and therefore cannot be used outside these constructs. You can always purchase an additional Azure Disks Reservation for your disk storage needs in other regions or SKUs.

**What happens when my Azure disks reservation expires?** You will receive email notifications 30 days prior to expiration and again on the expiration date. Once the reservation expires, deployed disks will continue to run and will be billed with the latest [pay-as-you-go rates](#).

## Azure shared disks

### Is the shared disks feature supported for unmanaged disks or page blobs?

No, it is only supported for premium SSD managed disks.

### What regions support shared disks?

Currently only West Central US.

### Can shared disks be used as an OS disk?

No, shared disks are only supported for data disks.

### What disk sizes support shared disks?

Only premium SSDs that are P15 or greater support shared disks.

### If I have an existing premium SSD, can I enable shared disks on it?

All managed disks created with API version 2019-07-01 or higher can enable shared disks. To do this, you need to unmount the disk from all VMs that it is attached to. Next, edit the `maxShares` property on the disk.

### If I no longer want to use a disk in shared mode, how do I disable it?

Unmount the disk from all VMs that it is attached to. Then edit the maxShare property on the disk to 1.

### Can you resize a shared disk?

Yes.

### Can I enable write accelerator on a disk that also has shared disks enabled?

No.

### Can I enable host caching for a disk that has shared disk enabled?

The only supported host caching option is 'None'.

## Ultra disks

**What should I set my ultra disk throughput to?** If you are unsure what to set your disk throughput to, we recommend you start by assuming an IO size of 16 KiB and adjust the performance from there as you monitor your application. The formula is: Throughput in MBps = # of IOPS \* 16 / 1000.

**I configured my disk to 40000 IOPS but I'm only seeing 12800 IOPS, why am I not seeing the performance of the disk?** In addition to the disk throttle, there is an IO throttle that gets imposed at the VM level. Please ensure that the VM size you are using can support the levels that are configured on your disks. For details regarding IO limits imposed by your VM, see [Sizes for Windows virtual machines in Azure](#).

**Can I use caching levels with an ultra disk?** No, ultra disks do not support the different caching methods that are supported on other disk types. Set the disk caching to None.

**Can I attach an ultra disk to my existing VM?** Maybe, your VM has to be in a region and availability zone pair that supports Ultra disks. See [getting started with ultra disks](#) for details.

**Can I use an ultra disk as the OS disk for my VM?** No, ultra Disks are only supported as data disks and are only supported as 4K native disks.

**Can I convert an existing disk to an ultra disk?** No, but you can migrate the data from an existing disk to an ultra disk. To migrate an existing disk to an ultra Disk, attach both disks to the same VM, and copy the disk's data from one disk to the other or leverage a 3rd party solution for data migration.

**Can I create snapshots for ultra disks?** No, snapshots are not yet available.

**Is Azure Backup available for ultra disks?** No, Azure Backup support is not yet available.

**Can I attach an ultra disk to a VM running in an availability set?** No, this is not yet supported.

**Can I enable Azure Site Recovery for VMs using ultra disks?** No, Azure Site Recovery is not yet supported for ultra disks.

## Uploading to a managed disk

**Can I upload data to an existing managed disk?**

No, upload can only be used during the creation of a new empty disk with the **ReadyToUpload** state.

**How do I upload to a managed disk?**

Create a managed disk with the `createOption` property of `creationData` set to "Upload", then you can upload data to it.

**Can I attach a disk to a VM while it is in an upload state?**

No.

**Can I take a snapshot of a manged disk in an upload state?**

No.

## Standard SSD disks

**What are Azure Standard SSD disks?** Standard SSD disks are standard disks backed by solid-state media, optimized as cost effective storage for workloads that need consistent performance at lower IOPS levels.

**What are the regions currently supported for Standard SSD disks?** All Azure regions now support Standard SSD disks.

**Is Azure Backup available when using Standard SSDs?** Yes, Azure Backup is now available.

**How do I create Standard SSD disks?** You can create Standard SSD disks using Azure Resource Manager templates, SDK, PowerShell, or CLI. Below are the parameters needed in the Resource Manager template to create Standard SSD Disks:

- `apiVersion` for Microsoft.Compute must be set as `2018-04-01` (or later)
- Specify `managedDisk.storageAccountType` as `StandardSSD_LRS`

The following example shows the `properties.storageProfile.osDisk` section for a VM that uses Standard SSD Disks:

```
"osDisk": {
 "osType": "Windows",
 "name": "myOsDisk",
 "caching": "ReadWrite",
 "createOption": "FromImage",
 "managedDisk": {
 "storageAccountType": "StandardSSD_LRS"
 }
}
```

For a complete template example of how to create a Standard SSD disk with a template, see [Create a VM from a Windows Image with Standard SSD Data Disks](#).

**Can I convert my existing disks to Standard SSD?** Yes, you can. Refer to [Convert Azure managed disks storage](#)

from standard to premium, and vice versa for the general guidelines for converting Managed Disks. And, use the following value to update the disk type to Standard SSD. -AccountType StandardSSD\_LRS

**What is the benefit of using Standard SSD disks instead of HDD?** Standard SSD disks deliver better latency, consistency, availability, and reliability compared to HDD disks. Application workloads run a lot more smoothly on Standard SSD because of that. Note, Premium SSD disks are the recommended solution for most IO-intensive production workloads.

**Can I use Standard SSDs as Unmanaged Disks?** No, Standard SSDs disks are only available as Managed Disks.

**Do Standard SSD Disks support "single instance VM SLA"?** No, Standard SSDs do not have single instance VM SLA. Use Premium SSD disks for single instance VM SLA.

## Migrate to Managed Disks

**Is there any impact of migration on the Managed Disks performance?**

Migration involves movement of the Disk from one Storage location to another. This is orchestrated via background copy of data, which can take several hours to complete, typically less than 24 Hrs depending on the amount of data in the disks. During that time your application can experience higher than usual read latency as some reads can get redirected to the original location, and can take longer to complete. There is no impact on write latency during this period.

**What changes are required in a pre-existing Azure Backup service configuration prior/after migration to Managed Disks?**

No changes are required.

**Will my VM backups created via Azure Backup service before the migration continue to work?**

Yes, backups work seamlessly.

**What changes are required in a pre-existing Azure Disks Encryption configuration prior/after migration to Managed Disks?**

No changes are required.

**Is automated migration of an existing virtual machine scale set from unmanaged disks to Managed Disks supported?**

No. You can create a new scale set with Managed Disks using the image from your old scale set with unmanaged disks.

**Can I create a Managed Disk from a page blob snapshot taken before migrating to Managed Disks?**

No. You can export a page blob snapshot as a page blob and then create a Managed Disk from the exported page blob.

**Can I fail over my on-premises machines protected by Azure Site Recovery to a VM with Managed Disks?**

Yes, you can choose to failover to a VM with Managed Disks.

**Is there any impact of migration on Azure VMs protected by Azure Site Recovery via Azure to Azure replication?**

No. Azure Site Recovery Azure to Azure protection for VMs with Managed Disks is available.

**Can I migrate VMs with unmanaged disks that are located on storage accounts that are or were previously encrypted to managed disks?**

Yes

## Managed Disks and Storage Service Encryption

### **Is Azure Storage Service Encryption enabled by default when I create a managed disk?**

Yes.

### **Is the boot volume encrypted by default on a managed disk?**

Yes. By default, all managed disks are encrypted, including the OS disk.

### **Who manages the encryption keys?**

Microsoft manages the encryption keys.

### **Can I disable Storage Service Encryption for my managed disks?**

No.

### **Is Storage Service Encryption only available in specific regions?**

No. It's available in all the regions where Managed Disks are available. Managed Disks is available in all public regions and Germany. It is also available in China, however, only for Microsoft managed keys, not customer managed keys.

### **How can I find out if my managed disk is encrypted?**

You can find out the time when a managed disk was created from the Azure portal, the Azure CLI, and PowerShell. If the time is after June 9, 2017, then your disk is encrypted.

### **How can I encrypt my existing disks that were created before June 10, 2017?**

As of June 10, 2017, new data written to existing managed disks is automatically encrypted. We are also planning to encrypt existing data, and the encryption will happen asynchronously in the background. If you must encrypt existing data now, create a copy of your disk. New disks will be encrypted.

- [Copy managed disks by using the Azure CLI](#)
- [Copy managed disks by using PowerShell](#)

### **Are managed snapshots and images encrypted?**

Yes. All managed snapshots and images created after June 9, 2017, are automatically encrypted.

### **Can I convert VMs with unmanaged disks that are located on storage accounts that are or were previously encrypted to managed disks?**

Yes

### **Will an exported VHD from a managed disk or a snapshot also be encrypted?**

No. But if you export a VHD to an encrypted storage account from an encrypted managed disk or snapshot, then it's encrypted.

## Premium disks: Managed and unmanaged

### **If a VM uses a size series that supports Premium SSD disks, such as a DSv2, can I attach both premium and standard data disks?**

Yes.

### **Can I attach both premium and standard data disks to a size series that doesn't support Premium SSD**

## **disks, such as D, Dv2, G, or F series?**

No. You can attach only standard data disks to VMs that don't use a size series that supports Premium SSD disks.

## **If I create a premium data disk from an existing VHD that was 80 GB, how much will that cost?**

A premium data disk created from an 80-GB VHD is treated as the next-available premium disk size, which is a P10 disk. You're charged according to the P10 disk pricing.

## **Are there transaction costs to use Premium SSD disks?**

There is a fixed cost for each disk size, which comes provisioned with specific limits on IOPS and throughput. The other costs are outbound bandwidth and snapshot capacity, if applicable. For more information, see the [pricing page](#).

## **What are the limits for IOPS and throughput that I can get from the disk cache?**

The combined limits for cache and local SSD for a DS series are 4,000 IOPS per core and 33 MiB per second per core. The GS series offers 5,000 IOPS per core and 50 MiB per second per core.

## **Is the local SSD supported for a Managed Disks VM?**

The local SSD is temporary storage that is included with a Managed Disks VM. There is no extra cost for this temporary storage. We recommend that you do not use this local SSD to store your application data because it isn't persisted in Azure Blob storage.

## **Are there any repercussions for the use of TRIM on premium disks?**

There is no downside to the use of TRIM on Azure disks on either premium or standard disks.

# New disk sizes: Managed and unmanaged

## **What regions support bursting capability for applicable premium SSD disk size?**

The bursting capability is currently supported in Azure West Central US.

## **What regions are 4/8/16 GiB Managed Disk sizes (P1/P2/P3, E1/E2/E3) supported in?**

These new disk sizes are currently supported in Azure West Central US.

## **Are P1/P2/P3 disk sizes supported for unmanaged disks or page blobs?**

No, it is only supported on premium SSD managed disks.

## **Are E1/E2/E3 disk sizes supported for unmanaged disks or page blobs?**

No, standard SSD managed disks of any size cannot be used with unmanaged disks or page blobs.

## **What is the largest Managed disk size supported for operating system and data disks?**

The partition type that Azure supports for an operating system disk is the master boot record (MBR). The MBR format supports a disk size up to 2 TiB. The largest size that Azure supports for an operating system disk is 2 TiB. Azure supports up to 32 TiB for managed data disks.

## **What is the largest Unmanaged Disk size supported for operating system and data disks?**

The partition type that Azure supports for an operating system disk is the master boot record (MBR). The MBR format supports a disk size up to 2 TiB. The largest size that Azure supports for an operating system Unmanaged disk is 2 TiB. Azure supports up to 4 TiB for data Unmanaged disks.

## **What is the largest page blob size that's supported?**

The largest page blob size that Azure supports is 8 TiB (8,191 GiB). The maximum page blob size when attached to

a VM as data or operating system disks is 4 TiB (4,095 GiB).

## **Do I need to use a new version of Azure tools to create, attach, resize, and upload disks larger than 1 TiB?**

You don't need to upgrade your existing Azure tools to create, attach, or resize disks larger than 1 TiB. To upload your VHD file from on-premises directly to Azure as a page blob or unmanaged disk, you need to use the latest tool sets listed below. We only support VHD uploads of up to 8 TiB.

| AZURE TOOLS      | SUPPORTED VERSIONS                                |
|------------------|---------------------------------------------------|
| Azure PowerShell | Version number 4.1.0: June 2017 release or later  |
| Azure CLI v1     | Version number 0.10.13: May 2017 release or later |
| Azure CLI v2     | Version number 2.0.12: July 2017 release or later |
| AzCopy           | Version number 6.1.0: June 2017 release or later  |

## **Are P4 and P6 disk sizes supported for unmanaged disks or page blobs?**

P4 (32 GiB) and P6 (64 GiB) disk sizes are not supported as the default disk tiers for unmanaged disks and page blobs. You need to explicitly [set the Blob Tier](#) to P4 and P6 to have your disk mapped to these tiers. If you deploy a unmanaged disk or page blob with the disk size or content length less than 32 GiB or between 32 GiB to 64 GiB without setting the Blob Tier, you will continue to land on P10 with 500 IOPS and 100 MiB/s and the mapped pricing tier.

## **If my existing premium managed disk less than 64 GiB was created before the small disk was enabled (around June 15, 2017), how is it billed?**

Existing small premium disks less than 64 GiB continue to be billed according to the P10 pricing tier.

## **How can I switch the disk tier of small premium disks less than 64 GiB from P10 to P4 or P6?**

You can take a snapshot of your small disks and then create a disk to automatically switch the pricing tier to P4 or P6 based on the provisioned size.

## **Can you resize existing Managed Disks from sizes fewer than 4 tebibytes (TiB) to new newly introduced disk sizes up to 32 TiB?**

Yes.

## **What are the largest disk sizes supported by Azure Backup and Azure Site Recovery service?**

The largest disk size supported by Azure Backup is 32 TiB (4 TiB for encrypted disks). The largest disk size supported by Azure Site Recovery is 8 TiB. Support for the larger disks up to 32 TiB is not yet available in Azure Site Recovery.

## **What are the recommended VM sizes for larger disk sizes (>4 TiB) for Standard SSD and Standard HDD disks to achieve optimized disk IOPS and Bandwidth?**

To achieve the disk throughput of Standard SSD and Standard HDD large disk sizes (>4 TiB) beyond 500 IOPS and 60 MiB/s, we recommend you deploy a new VM from one of the following VM sizes to optimize your performance: B-series, DSv2-series, Dsv3-Series, ESv3-Series, Fs-series, Fsv2-series, M-series, GS-series, NCv2-series, NCv3-series, or Ls-series VMs. Attaching large disks to existing VMs or VMs that are not using the recommended sizes above may experience lower performance.

## **How can I upgrade my disks (>4 TiB) which were deployed during the larger disk sizes preview in order to get the higher IOPS & bandwidth at GA?**

You can either stop and start the VM that the disk is attached to or, detach and re-attach your disk. The performance targets of larger disk sizes have been increased for both premium SSDs and standard SSDs at GA.

### **What regions are the managed disk sizes of 8 TiB, 16 TiB, and 32 TiB supported in?**

The 8 TiB, 16 TiB, and 32 TiB disk SKUs are supported in all regions under global Azure, Microsoft Azure Government, and Azure China 21Vianet.

### **Do we support enabling Host Caching on all disk sizes?**

We support Host Caching of ReadOnly and Read/Write on disk sizes less than 4 TiB. For disk sizes more than 4 TiB, we don't support setting caching option other than None. We recommend leveraging caching for smaller disk sizes where you can expect to observe better performance boost with data cached to the VM.

## **What if my question isn't answered here?**

If your question isn't listed here, let us know and we'll help you find an answer. You can post a question at the end of this article in the comments. To engage with the Azure Storage team and other community members about this article, use the MSDN [Azure Storage forum](#).

To request features, submit your requests and ideas to the [Azure Storage feedback forum](#).

# Common PowerShell commands for Azure Virtual Networks

2/28/2020 • 4 minutes to read • [Edit Online](#)

If you want to create a virtual machine, you need to create a [virtual network](#) or know about an existing virtual network in which the VM can be added. Typically, when you create a VM, you also need to consider creating the resources described in this article.

See [How to install and configure Azure PowerShell](#) for information about installing the latest version of Azure PowerShell, selecting your subscription, and signing in to your account.

Some variables might be useful for you if running more than one of the commands in this article:

- \$location - The location of the network resources. You can use [Get-AzLocation](#) to find a [geographical region](#) that works for you.
- \$myResourceGroup - The name of the resource group where the network resources are located.

## Create network resources

| TASK                          | COMMAND                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create subnet configurations  | \$subnet1 = <a href="#">New-AzVirtualNetworkSubnetConfig</a> -Name "mySubnet1" -AddressPrefix XX.X.X.X/XX<br>\$subnet2 = <a href="#">New-AzVirtualNetworkSubnetConfig</a> -Name "mySubnet2" -AddressPrefix XX.X.X.X/XX<br><br>A typical network might have a subnet for an <a href="#">internet facing load balancer</a> and a separate subnet for an <a href="#">internal load balancer</a> .                                                                                                                                                              |
| Create a virtual network      | \$vnet = <a href="#">New-AzVirtualNetwork</a> -Name "myVNet" -<br>ResourceGroupName \$myResourceGroup -Location \$location<br>-AddressPrefix XX.X.X.X/XX -Subnet \$subnet1, \$subnet2                                                                                                                                                                                                                                                                                                                                                                       |
| Test for a unique domain name | <a href="#">Test-AzDnsAvailability</a> -DomainNameLabel "myDNS" -Location<br>\$location<br><br>You can specify a DNS domain name for a <a href="#">public IP resource</a> , which creates a mapping for domainname.location.cloudapp.azure.com to the public IP address in the Azure-managed DNS servers. The name can contain only letters, numbers, and hyphens. The first and last character must be a letter or number and the domain name must be unique within its Azure location. If <b>True</b> is returned, your proposed name is globally unique. |
| Create a public IP address    | \$pip = <a href="#">New-AzPublicIpAddress</a> -Name "myPublicIp" -<br>ResourceGroupName \$myResourceGroup -DomainNameLabel<br>"myDNS" -Location \$location -AllocationMethod Dynamic<br><br>The public IP address uses the domain name that you previously tested and is used by the frontend configuration of the load balancer.                                                                                                                                                                                                                           |

| TASK                               | COMMAND                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a frontend IP configuration | <pre>\$frontendIP = <a href="#">New-AzLoadBalancerFrontendIpConfig</a> -Name "myFrontendIP" -PublicIpAddress \$pip</pre> <p>The frontend configuration includes the public IP address that you previously created for incoming network traffic.</p>                                                                                                                                                                                       |
| Create a backend address pool      | <pre>\$beAddressPool = <a href="#">New-AzLoadBalancerBackendAddressPoolConfig</a> -Name "myBackendAddressPool"</pre> <p>Provides internal addresses for the backend of the load balancer that are accessed through a network interface.</p>                                                                                                                                                                                               |
| Create a probe                     | <pre>\$healthProbe = <a href="#">New-AzLoadBalancerProbeConfig</a> -Name "myProbe" -RequestPath 'HealthProbe.aspx' -Protocol http -Port 80 -IntervalInSeconds 15 -ProbeCount 2</pre> <p>Contains health probes used to check availability of virtual machines instances in the backend address pool.</p>                                                                                                                                  |
| Create a load balancing rule       | <pre>\$lbRule = <a href="#">New-AzLoadBalancerRuleConfig</a> -Name HTTP -FrontendIpConfiguration \$frontendIP -BackendAddressPool \$beAddressPool -Probe \$healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80</pre> <p>Contains rules that assign a public port on the load balancer to a port in the backend address pool.</p>                                                                                                   |
| Create an inbound NAT rule         | <pre>\$inboundNATRule = <a href="#">New-AzLoadBalancerInboundNatRuleConfig</a> -Name "myInboundRule1" -FrontendIpConfiguration \$frontendIP -Protocol TCP -FrontendPort 3441 -BackendPort 3389</pre> <p>Contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the backend address pool.</p>                                                                                               |
| Create a load balancer             | <pre>\$loadBalancer = <a href="#">New-AzLoadBalancer</a> -ResourceGroupName \$myResourceGroup -Name "myLoadBalancer" -Location \$location -FrontendIpConfiguration \$frontendIP -InboundNatRule \$inboundNATRule -LoadBalancingRule \$lbRule -BackendAddressPool \$beAddressPool -Probe \$healthProbe</pre>                                                                                                                               |
| Create a network interface         | <pre>\$nic1 = <a href="#">New-AzNetworkInterface</a> -ResourceGroupName \$myResourceGroup -Name "myNIC" -Location \$location -PrivateIpAddress XX.X.X.X -Subnet \$subnet2 -LoadBalancerBackendAddressPool \$loadBalancer.BackendAddressPools[0] -LoadBalancerInboundNatRule \$loadBalancer.InboundNatRules[0]</pre> <p>Create a network interface using the public IP address and virtual network subnet that you previously created.</p> |

## Get information about network resources

| TASK                                            | COMMAND                                                                                                                                                                                                                                          |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| List virtual networks                           | <code>Get-AzVirtualNetwork -ResourceGroupName \$myResourceGroup</code><br>Lists all the virtual networks in the resource group.                                                                                                                  |
| Get information about a virtual network         | <code>Get-AzVirtualNetwork -Name "myVNet" -ResourceGroupName \$myResourceGroup</code>                                                                                                                                                            |
| List subnets in a virtual network               | <code>Get-AzVirtualNetwork -Name "myVNet" -ResourceGroupName \$myResourceGroup   Select Subnets</code>                                                                                                                                           |
| Get information about a subnet                  | <code>Get-AzVirtualNetworkSubnetConfig -Name "mySubnet1" -VirtualNetwork \$vnet</code><br>Gets information about the subnet in the specified virtual network. The \$vnet value represents the object returned by Get-AzVirtualNetwork.           |
| List IP addresses                               | <code>Get-AzPublicIpAddress -ResourceGroupName \$myResourceGroup</code><br>Lists the public IP addresses in the resource group.                                                                                                                  |
| List load balancers                             | <code>Get-AzLoadBalancer -ResourceGroupName \$myResourceGroup</code><br>Lists all the load balancers in the resource group.                                                                                                                      |
| List network interfaces                         | <code>Get-AzNetworkInterface -ResourceGroupName \$myResourceGroup</code><br>Lists all the network interfaces in the resource group.                                                                                                              |
| Get information about a network interface       | <code>Get-AzNetworkInterface -Name "myNIC" -ResourceGroupName \$myResourceGroup</code><br>Gets information about a specific network interface.                                                                                                   |
| Get the IP configuration of a network interface | <code>Get-AzNetworkInterfaceIPConfig -Name "myNICIP" -NetworkInterface \$nic</code><br>Gets information about the IP configuration of the specified network interface. The \$nic value represents the object returned by Get-AzNetworkInterface. |

## Manage network resources

| TASK                              | COMMAND                                                                                                                                                                                                                                    |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add a subnet to a virtual network | <code>Add-AzVirtualNetworkSubnetConfig -AddressPrefix XX.X.X.XXX -Name "mySubnet1" -VirtualNetwork \$vnet</code><br>Adds a subnet to an existing virtual network. The \$vnet value represents the object returned by Get-AzVirtualNetwork. |

| TASK                       | COMMAND                                                                                                                                                                       |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Delete a virtual network   | <p><code>Remove-AzVirtualNetwork -Name "myVNet" -ResourceGroupName \$myResourceGroup</code></p> <p>Removes the specified virtual network from the resource group.</p>         |
| Delete a network interface | <p><code>Remove-AzNetworkInterface -Name "myNIC" -ResourceGroupName \$myResourceGroup</code></p> <p>Removes the specified network interface from the resource group.</p>      |
| Delete a load balancer     | <p><code>Remove-AzLoadBalancer -Name "myLoadBalancer" -ResourceGroupName \$myResourceGroup</code></p> <p>Removes the specified load balancer from the resource group.</p>     |
| Delete a public IP address | <p><code>Remove-AzPublicIpAddress -Name "myIPAddress" -ResourceGroupName \$myResourceGroup</code></p> <p>Removes the specified public IP address from the resource group.</p> |

## Next Steps

Use the network interface that you just created when you [create a VM](#).

2 minutes to read

# How to open ports to a virtual machine with the Azure portal

12/23/2019 • 2 minutes to read • [Edit Online](#)

You open a port, or create an endpoint, to a virtual machine (VM) in Azure by creating a network filter on a subnet or a VM network interface. You place these filters, which control both inbound and outbound traffic, on a network security group attached to the resource that receives the traffic.

The example in this article demonstrates how to create a network filter that uses the standard TCP port 80 (it's assumed you've already started the appropriate services and opened any OS firewall rules on the VM).

After you've created a VM that's configured to serve web requests on the standard TCP port 80, you can:

1. Create a network security group.
2. Create an inbound security rule allowing traffic and assign values to the following settings:
  - **Destination port ranges:** 80
  - **Source port ranges:** \* (allows any source port)
  - **Priority value:** Enter a value that is less than 65,500 and higher in priority than the default catch-all deny inbound rule.
3. Associate the network security group with the VM network interface or subnet.

Although this example uses a simple rule to allow HTTP traffic, you can also use network security groups and rules to create more complex network configurations.

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create a network security group

1. Search for and select the resource group for the VM, choose **Add**, then search for and select **Network security group**.
2. Select **Create**.

The **Create network security group** window opens.

Create network security group

\* Name

\* Subscription  
<subscription name>

\* Resource group  
SELECT EXISTING...  
[Create new](#)

\* Location  
West US

**Create**   [Automation options](#)

3. Enter a name for your network security group.
4. Select or create a resource group, then select a location.
5. Select **Create** to create the network security group.

## Create an inbound security rule

1. Select your new network security group.
2. Select **Inbound security rules**, then select **Add**.

myNetworkSecurityGroup - Inbound security rules

Network security group

**+ Add**   [Default rules](#)

Search (Ctrl+/)

| PRIORITY    | NAME |
|-------------|------|
| No results. |      |

**Overview**

**Activity log**

**Access control (IAM)**

**Tags**

**Diagnose and solve problems**

**SETTINGS**

**Inbound security rules**

**Outbound security rules**

**Network interfaces**

3. Select **Advanced**.
4. Choose a common **Service** from the drop-down menu, such as **HTTP**. You can also select **Custom** if you want to provide a specific port to use.
5. Optionally, change the **Priority** or **Name**. The priority affects the order in which rules are applied: the lower

the numerical value, the earlier the rule is applied.

6. Select **Add** to create the rule.

## Associate your network security group with a subnet

Your final step is to associate your network security group with a subnet or a specific network interface. For this example, we'll associate the network security group with a subnet.

1. Select **Subnets**, then select **Associate**.

The screenshot shows the 'Subnets' blade for a Network Security Group named 'myNetworkSecurityGroup'. On the right, there's a large 'Associate' button with a plus sign. On the left, a sidebar lists various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Inbound security rules, Outbound security rules, Network interfaces, and Subnets. The 'Subnets' link is highlighted with a red box. Below the sidebar, there's a 'SETTINGS' section with Inbound and Outbound security rules, and Network interfaces.

2. Select your virtual network, and then select the appropriate subnet.

The screenshot shows two dialog boxes. The left one is titled 'Associate subnet' for 'myNetworkSecurityGroup' and lists 'Virtual network myVnet' with a checkmark. The right one is titled 'Choose subnet' and lists 'mySubnet' under 'myResourceGroup'.

Any VMs you connect to that subnet are now reachable on port 80.

## Additional information

You can also [perform the steps in this article by using Azure PowerShell](#).

The commands described in this article allow you to quickly get traffic flowing to your VM. Network security

groups provide many great features and granularity for controlling access to your resources. For more information, see [Filter network traffic with a network security group](#).

For highly available web applications, consider placing your VMs behind an Azure load balancer. The load balancer distributes traffic to VMs, with a network security group that provides traffic filtering. For more information, see [Load balance Windows virtual machines in Azure to create a highly available application](#).

## Next steps

In this article, you created a network security group, created an inbound rule that allows HTTP traffic on port 80, and then associated that rule with a subnet.

You can find information on creating more detailed environments in the following articles:

- [Azure Resource Manager overview](#)
- [Security groups](#)

# How to open ports and endpoints to a VM in Azure using PowerShell

1/8/2020 • 2 minutes to read • [Edit Online](#)

You open a port, or create an endpoint, to a virtual machine (VM) in Azure by creating a network filter on a subnet or a VM network interface. You place these filters, which control both inbound and outbound traffic, on a network security group attached to the resource that receives the traffic.

The example in this article demonstrates how to create a network filter that uses the standard TCP port 80 (it's assumed you've already started the appropriate services and opened any OS firewall rules on the VM).

After you've created a VM that's configured to serve web requests on the standard TCP port 80, you can:

1. Create a network security group.
2. Create an inbound security rule allowing traffic and assign values to the following settings:

- **Destination port ranges:** 80
- **Source port ranges:** \* (allows any source port)
- **Priority value:** Enter a value that is less than 65,500 and higher in priority than the default catch-all deny inbound rule.

3. Associate the network security group with the VM network interface or subnet.

Although this example uses a simple rule to allow HTTP traffic, you can also use network security groups and rules to create more complex network configurations.

## Quick commands

To create a Network Security Group and ACL rules you need [the latest version of Azure PowerShell installed](#). You can also [perform these steps using the Azure portal](#).

Log in to your Azure account:

```
Connect-AzAccount
```

In the following examples, replace parameter names with your own values. Example parameter names included *myResourceGroup*, *myNetworkSecurityGroup*, and *myVnet*.

Create a rule with [New-AzNetworkSecurityRuleConfig](#). The following example creates a rule named *myNetworkSecurityGroupRule* to allow *tcp* traffic on port 80:

```
$httprule = New-AzNetworkSecurityRuleConfig `
 -Name "myNetworkSecurityGroupRule" `
 -Description "Allow HTTP" `
 -Access "Allow" `
 -Protocol "Tcp" `
 -Direction "Inbound" `
 -Priority "100" `
 -SourceAddressPrefix "Internet" `
 -SourcePortRange * `
 -DestinationAddressPrefix * `
 -DestinationPortRange 80
```

Next, create your Network Security group with [New-AzNetworkSecurityGroup](#) and assign the HTTP rule you just created as follows. The following example creates a Network Security Group named *myNetworkSecurityGroup*:

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName "myResourceGroup" `
 -Location "EastUS" `
 -Name "myNetworkSecurityGroup" `
 -SecurityRules $httprule
```

Now let's assign your Network Security Group to a subnet. The following example assigns an existing virtual network named *myVnet* to the variable *\$vnet* with [Get-AzVirtualNetwork](#):

```
$vnet = Get-AzVirtualNetwork `
 -ResourceGroupName "myResourceGroup" `
 -Name "myVnet"
```

Associate your Network Security Group with your subnet with [Set-AzVirtualNetworkSubnetConfig](#). The following example associates the subnet named *mySubnet* with your Network Security Group:

```
$subnetPrefix = $vnet.Subnets | ?{$_ . Name -eq 'mySubnet'}`

Set-AzVirtualNetworkSubnetConfig `
 -VirtualNetwork $vnet `
 -Name "mySubnet" `
 -AddressPrefix $subnetPrefix.AddressPrefix `
 -NetworkSecurityGroup $nsg
```

Finally, update your virtual network with [Set-AzVirtualNetwork](#) in order for your changes to take effect:

```
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

## More information on Network Security Groups

The quick commands here allow you to get up and running with traffic flowing to your VM. Network Security Groups provide many great features and granularity for controlling access to your resources. You can read more about [creating a Network Security Group and ACL rules here](#).

For highly available web applications, you should place your VMs behind an Azure Load Balancer. The load balancer distributes traffic to VMs, with a Network Security Group that provides traffic filtering. For more information, see [How to load balance Linux virtual machines in Azure to create a highly available application](#).

## Next steps

In this example, you created a simple rule to allow HTTP traffic. You can find information on creating more detailed environments in the following articles:

- [Azure Resource Manager overview](#)
- [What is a network security group?](#)
- [Azure Load Balancer Overview](#)

# Create a virtual machine with a static public IP address using the Azure portal

1/16/2020 • 3 minutes to read • [Edit Online](#)

You can create a virtual machine with a static public IP address. A public IP address enables you to communicate to a virtual machine from the internet. Assign a static public IP address, rather than a dynamic address, to ensure that the address never changes. Learn more about [static public IP addresses](#). To change a public IP address assigned to an existing virtual machine from dynamic to static, or to work with private IP addresses, see [Add, change, or remove IP addresses](#). Public IP addresses have a [nominal charge](#), and there is a [limit](#) to the number of public IP addresses that you can use per subscription.

## Sign in to Azure

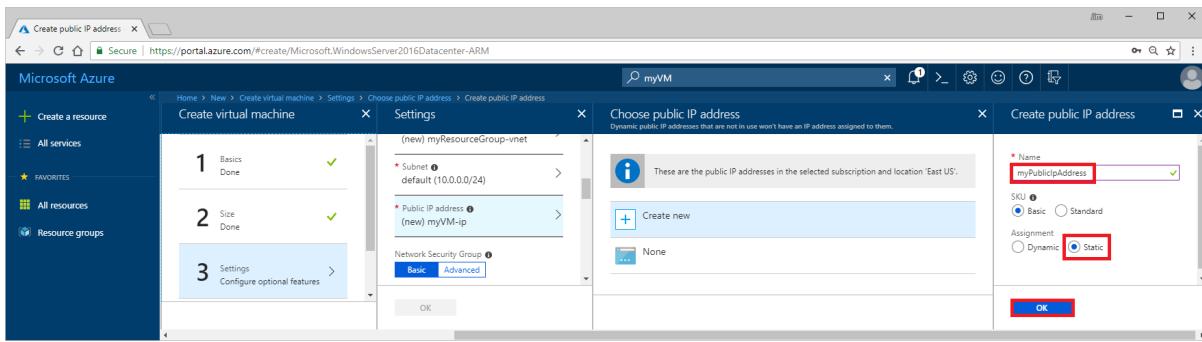
Sign in to the Azure portal at <https://portal.azure.com>.

## Create a virtual machine

1. Select + **Create a resource** found on the upper, left corner of the Azure portal.
2. Select **Compute**, and then select **Windows Server 2016 VM**, or another operating system of your choosing.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

| SETTING        | VALUE                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Name           | myVM                                                                                                                                               |
| User name      | Enter a user name of your choosing.                                                                                                                |
| Password       | Enter a password of your choosing. The password must be at least 12 characters long and meet the <a href="#">defined complexity requirements</a> . |
| Subscription   | Select your subscription.                                                                                                                          |
| Resource group | Select <b>Use existing</b> and select <b>myResourceGroup</b> .                                                                                     |
| Location       | Select <b>East US</b>                                                                                                                              |

4. Select a size for the VM and then select **Select**.
5. Under **Settings**, select **Public IP address**.
6. Enter *myPublicIpAddress*, select **Static**, and then select **OK**, as shown in the following picture:



If the public IP address must be a standard SKU, select **Standard** under **SKU**. Learn more about [Public IP address SKUs](#). If the virtual machine will be added to the back-end pool of a public Azure Load Balancer, the SKU of the virtual machine's public IP address must match the SKU of the load balancer's public IP address. For details, see [Azure Load Balancer](#).

7. Select a port, or no ports under **Select public inbound ports**. Portal 3389 is selected, to enable remote access to the Windows Server virtual machine from the internet. Opening port 3389 from the internet is not recommended for production workloads.

8. Accept the remaining default settings and select **OK**.
9. On the **Summary** page, select **Create**. The virtual machine takes a few minutes to deploy.
10. Once the virtual machine is deployed, enter *myPublicIpAddress* in the search box at the top of the portal. When **myPublicIpAddress** appears in the search results, select it.
11. You can view the public IP address that is assigned, and that the address is assigned to the **myVM** virtual machine, as shown in the following picture:

| Essentials                                   |                           |
|----------------------------------------------|---------------------------|
| Resource group ( <a href="#">change</a> )    | SKU                       |
| <a href="#">myResourceGroup</a>              | Basic                     |
| Location                                     | IP address                |
| East US                                      | 40.87.1.66                |
| Subscription name ( <a href="#">change</a> ) | DNS name                  |
| Subscription ID                              | Associated to             |
|                                              | <a href="#">myVMVMNic</a> |
|                                              | Virtual machine           |
|                                              | <a href="#">myVM</a>      |

Azure assigned a public IP address from addresses used in the region you created the virtual machine in. You can download the list of ranges (prefixes) for the Azure [Public](#), [US government](#), [China](#), and [Germany](#) clouds.

12. Select **Configuration** to confirm that the assignment is **Static**.

The screenshot shows the Azure portal interface for managing a public IP address. The left sidebar lists several options: Overview, Activity log, Access control (IAM), Tags, Configuration (which is highlighted with a red box), Properties, Locks, and Automation script. The main content area is titled "myPublicIpAddress - Configuration". It displays a message: "This static public IP address is associated to the IP configuration 'ipconfigmyVM', in the network interface 'myVMVMNic'. You must dissociate it from the network interface before changing its assignment." Below this, there is a section labeled "Assignment" with two radio button options: "Dynamic" and "Static", where "Static" is selected. Other settings shown include "IP address" (40.87.1.66), "Idle timeout (minutes)" (set to 4), and a "DNS name label (optional)" field containing ".eastus.cloudapp.azure.com". A note at the bottom suggests using Azure DNS with a link to "Try Azure DNS now".

**WARNING**

Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary, and not until after reading [Add a private IP address to an operating system](#).

## Clean up resources

When no longer needed, delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the **Search** box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

## Next steps

- Learn more about [public IP addresses](#) in Azure
- Learn more about all [public IP address settings](#)
- Learn more about [private IP addresses](#) and assigning a [static private IP address](#) to an Azure virtual machine
- Learn more about creating [Linux](#) and [Windows](#) virtual machines

# Create and manage a Windows virtual machine that has multiple NICs

12/23/2019 • 8 minutes to read • [Edit Online](#)

Virtual machines (VMs) in Azure can have multiple virtual network interface cards (NICs) attached to them. A common scenario is to have different subnets for front-end and back-end connectivity. You can associate multiple NICs on a VM to multiple subnets, but those subnets must all reside in the same virtual network (vNet). This article details how to create a VM that has multiple NICs attached to it. You also learn how to add or remove NICs from an existing VM. Different [VM sizes](#) support a varying number of NICs, so size your VM accordingly.

## Prerequisites

In the following examples, replace example parameter names with your own values. Example parameter names include *myResourceGroup*, *myVnet*, and *myVM*.

## Create a VM with multiple NICs

First, create a resource group. The following example creates a resource group named *myResourceGroup* in the *EastUs* location:

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

### Create virtual network and subnets

A common scenario is for a virtual network to have two or more subnets. One subnet may be for front-end traffic, the other for back-end traffic. To connect to both subnets, you then use multiple NICs on your VM.

1. Define two virtual network subnets with [New-AzVirtualNetworkSubnetConfig](#). The following example defines the subnets for *mySubnetFrontEnd* and *mySubnetBackEnd*:

```
$mySubnetFrontEnd = New-AzVirtualNetworkSubnetConfig -Name "mySubnetFrontEnd" `
 -AddressPrefix "192.168.1.0/24"
$mySubnetBackEnd = New-AzVirtualNetworkSubnetConfig -Name "mySubnetBackEnd" `
 -AddressPrefix "192.168.2.0/24"
```

2. Create your virtual network and subnets with [New-AzVirtualNetwork](#). The following example creates a virtual network named *myVnet*:

```
$myVnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
 -Location "EastUs" `
 -Name "myVnet" `
 -AddressPrefix "192.168.0.0/16" `
 -Subnet $mySubnetFrontEnd,$mySubnetBackEnd
```

### Create multiple NICs

Create two NICs with [New-AzNetworkInterface](#). Attach one NIC to the front-end subnet and one NIC to the back-end subnet. The following example creates NICs named *myNic1* and *myNic2*:

```

$frontEnd = $myVnet.Subnets | ?{$_ .Name -eq 'mySubnetFrontEnd'}
$myNic1 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic1" `
 -Location "EastUs" `
 -SubnetId $frontEnd.Id

$backEnd = $myVnet.Subnets | ?{$_ .Name -eq 'mySubnetBackEnd'}
$myNic2 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic2" `
 -Location "EastUs" `
 -SubnetId $backEnd.Id

```

Typically you also create a [network security group](#) to filter network traffic to the VM and a [load balancer](#) to distribute traffic across multiple VMs.

## Create the virtual machine

Now start to build your VM configuration. Each VM size has a limit for the total number of NICs that you can add to a VM. For more information, see [Windows VM sizes](#).

1. Set your VM credentials to the `$cred` variable as follows:

```
$cred = Get-Credential
```

2. Define your VM with [New-AzVMConfig](#). The following example defines a VM named *myVM* and uses a VM size that supports more than two NICs (*Standard\_DS3\_v2*):

```
$vmConfig = New-AzVMConfig -VMName "myVM" -VMSize "Standard_DS3_v2"
```

3. Create the rest of your VM configuration with [Set-AzVMOperatingSystem](#) and [Set-AzVMSourceImage](#). The following example creates a Windows Server 2016 VM:

```

$vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
 -Windows `
 -ComputerName "myVM" `
 -Credential $cred `
 -ProvisionVMAgent `
 -EnableAutoUpdate
$vmConfig = Set-AzVMSourceImage -VM $vmConfig `
 -PublisherName "MicrosoftWindowsServer" `
 -Offer "WindowsServer" `
 -Skus "2016-Datacenter" `
 -Version "latest"

```

4. Attach the two NICs that you previously created with [Add-AzVMNetworkInterface](#):

```

$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic1.Id -Primary
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic2.Id

```

5. Create your VM with [New-AzVM](#):

```
New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location "EastUs"
```

6. Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

# Add a NIC to an existing VM

To add a virtual NIC to an existing VM, you deallocate the VM, add the virtual NIC, then start the VM. Different [VM sizes](#) support a varying number of NICs, so size your VM accordingly. If needed, you can [resize a VM](#).

1. Deallocate the VM with [Stop-AzVM](#). The following example deallocates the VM named *myVM* in *myResourceGroup*:

```
Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with [Get-AzVm](#). The following example gets information for the VM named *myVM* in *myResourceGroup*:

```
$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. The following example creates a virtual NIC with [New-AzNetworkInterface](#) named *myNic3* that is attached to *mySubnetBackEnd*. The virtual NIC is then attached to the VM named *myVM* in *myResourceGroup* with [Add-AzVMNetworkInterface](#):

```
Get info for the back end subnet
$myVnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName "myResourceGroup"
$backEnd = $myVnet.Subnets | ?{$_._Name -eq 'mySubnetBackEnd'}`

Create a virtual NIC
$myNic3 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
 -Name "myNic3" `
 -Location "EastUs" `
 -SubnetId $backEnd.Id

Get the ID of the new virtual NIC and add to VM
$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -Name "MyNic3").Id
Add-AzVMNetworkInterface -VM $vm -Id $nicId | Update-AzVm -ResourceGroupName "myResourceGroup"
```

## Primary virtual NICs

One of the NICs on a multi-NIC VM needs to be primary. If one of the existing virtual NICs on the VM is already set as primary, you can skip this step. The following example assumes that two virtual NICs are now present on a VM and you wish to add the first NIC (`[0]`) as the primary:

```
List existing NICs on the VM and find which one is primary
$vm.NetworkProfile.NetworkInterfaces

Set NIC 0 to be primary
$vm.NetworkProfile.NetworkInterfaces[0].Primary = $true
$vm.NetworkProfile.NetworkInterfaces[1].Primary = $false

Update the VM state in Azure
Update-AzVM -VM $vm -ResourceGroupName "myResourceGroup"
```

4. Start the VM with [Start-AzVm](#):

```
Start-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM"
```

5. Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

## Remove a NIC from an existing VM

To remove a virtual NIC from an existing VM, you deallocate the VM, remove the virtual NIC, then start the VM.

1. Deallocation the VM with [Stop-AzVM](#). The following example deallocated the VM named *myVM* in *myResourceGroup*:

```
Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with [Get-AzVm](#). The following example gets information for the VM named *myVM* in *myResourceGroup*:

```
$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. Get information about the NIC remove with [Get-AzNetworkInterface](#). The following example gets information about *myNic3*:

```
List existing NICs on the VM if you need to determine NIC name
$vm.NetworkProfile.NetworkInterfaces

$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -Name "myNic3").Id
```

4. Remove the NIC with [Remove-AzVMNetworkInterface](#) and then update the VM with [Update-AzVm](#). The following example removes *myNic3* as obtained by `$nicId` in the preceding step:

```
Remove-AzVMNetworkInterface -VM $vm -NetworkInterfaceIDs $nicId | `
Update-AzVm -ResourceGroupName "myResourceGroup"
```

5. Start the VM with [Start-AzVm](#):

```
Start-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

## Create multiple NICs with templates

Azure Resource Manager templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. Resource Manager templates use declarative JSON files to define your environment. For more information, see [overview of Azure Resource Manager](#). You can use `copy` to specify the number of instances to create:

```
"copy": {
 "name": "multiplenics",
 "count": "[parameters('count')]"
}
```

For more information, see [creating multiple instances by using copy](#).

You can also use `copyIndex()` to append a number to a resource name. You can then create *myNic1*, *MyNic2* and so on. The following code shows an example of appending the index value:

```
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of [creating multiple NICs by using Resource Manager templates](#).

Add routes for secondary NICs to the OS by completing the steps in [Configure the operating system for multiple NICs](#).

## Configure guest OS for multiple NICs

Azure assigns a default gateway to the first (primary) network interface attached to the virtual machine. Azure does not assign a default gateway to additional (secondary) network interfaces attached to a virtual machine. Therefore, you are unable to communicate with resources outside the subnet that a secondary network interface is in, by default. Secondary network interfaces can, however, communicate with resources outside their subnet, though the steps to enable communication are different for different operating systems.

1. From a Windows command prompt, run the `route print` command, which returns output similar to the following output for a virtual machine with two attached network interfaces:

```
=====
Interface List
3...00 0d 3a 10 92 ceMicrosoft Hyper-V Network Adapter #3
7...00 0d 3a 10 9b 2aMicrosoft Hyper-V Network Adapter #4
=====
```

In this example, **Microsoft Hyper-V Network Adapter #4** (interface 7) is the secondary network interface that doesn't have a default gateway assigned to it.

2. From a command prompt, run the `ipconfig` command to see which IP address is assigned to the secondary network interface. In this example, 192.168.2.4 is assigned to interface 7. No default gateway address is returned for the secondary network interface.
3. To route all traffic destined for addresses outside the subnet of the secondary network interface to the gateway for the subnet, run the following command:

```
route add -p 0.0.0.0 MASK 0.0.0.0 192.168.2.1 METRIC 5015 IF 7
```

The gateway address for the subnet is the first IP address (ending in .1) in the address range defined for the subnet. If you don't want to route all traffic outside the subnet, you could add individual routes to specific destinations, instead. For example, if you only wanted to route traffic from the secondary network interface to the 192.168.3.0 network, you enter the command:

```
route add -p 192.168.3.0 MASK 255.255.255.0 192.168.2.1 METRIC 5015 IF 7
```

4. To confirm successful communication with a resource on the 192.168.3.0 network, for example, enter the following command to ping 192.168.3.4 using interface 7 (192.168.2.4):

```
ping 192.168.3.4 -S 192.168.2.4
```

You may need to open ICMP through the Windows firewall of the device you're pinging with the following command:

```
netsh advfirewall firewall add rule name=Allow-ping protocol=icmpv4 dir=in action=allow
```

5. To confirm the added route is in the route table, enter the `route print` command, which returns output similar to the following text:

| =====               |         |             |             |        |  |
|---------------------|---------|-------------|-------------|--------|--|
| Active Routes:      |         |             |             |        |  |
| Network Destination | Netmask | Gateway     | Interface   | Metric |  |
| 0.0.0.0             | 0.0.0.0 | 192.168.1.1 | 192.168.1.4 | 15     |  |
| 0.0.0.0             | 0.0.0.0 | 192.168.2.1 | 192.168.2.4 | 5015   |  |

The route listed with **192.168.1.1** under **Gateway**, is the route that is there by default for the primary network interface. The route with **192.168.2.1** under **Gateway**, is the route you added.

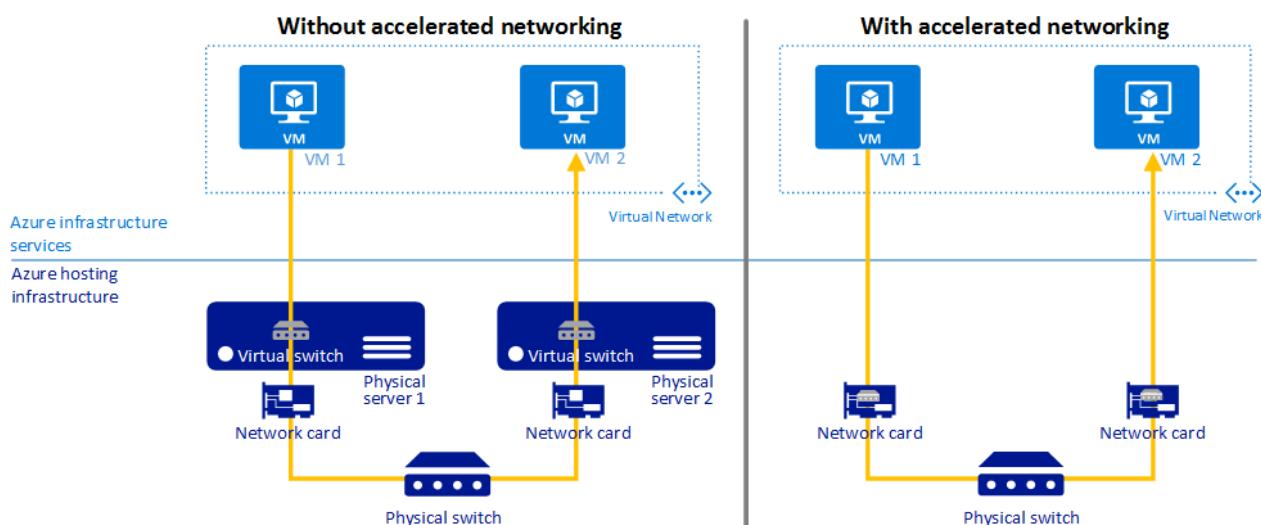
## Next steps

Review [Windows VM sizes](#) when you're trying to create a VM that has multiple NICs. Pay attention to the maximum number of NICs that each VM size supports.

# Create a Windows virtual machine with Accelerated Networking using Azure PowerShell

11/14/2019 • 10 minutes to read • [Edit Online](#)

In this tutorial, you learn how to create a Windows virtual machine (VM) with Accelerated Networking. To create a Linux VM with Accelerated Networking, see [Create a Linux VM with Accelerated Networking](#). Accelerated networking enables single root I/O virtualization (SR-IOV) to a VM, greatly improving its networking performance. This high-performance path bypasses the host from the datapath, reducing latency, jitter, and CPU utilization, for use with the most demanding network workloads on supported VM types. The following picture shows communication between two VMs with and without accelerated networking:



Without accelerated networking, all networking traffic in and out of the VM must traverse the host and the virtual switch. The virtual switch provides all policy enforcement, such as network security groups, access control lists, isolation, and other network virtualized services to network traffic. To learn more about virtual switches, see [Hyper-V network virtualization and virtual switch](#).

With accelerated networking, network traffic arrives at the VM's network interface (NIC), and is then forwarded to the VM. All network policies that the virtual switch applies are now offloaded and applied in hardware. Applying policy in hardware enables the NIC to forward network traffic directly to the VM, bypassing the host and the virtual switch, while maintaining all the policy it applied in the host.

The benefits of accelerated networking only apply to the VM that it is enabled on. For the best results, it is ideal to enable this feature on at least two VMs connected to the same Azure Virtual Network (VNet). When communicating across VNets or connecting on-premises, this feature has minimal impact to overall latency.

## Benefits

- **Lower Latency / Higher packets per second (pps):** Removing the virtual switch from the datapath removes the time packets spend in the host for policy processing and increases the number of packets that can be processed inside the VM.
- **Reduced jitter:** Virtual switch processing depends on the amount of policy that needs to be applied and the workload of the CPU that is doing the processing. Offloading the policy enforcement to the hardware removes that variability by delivering packets directly to the VM, removing the host to VM communication and all software interrupts and context switches.
- **Decreased CPU utilization:** Bypassing the virtual switch in the host leads to less CPU utilization for

processing network traffic.

## Limitations and Constraints

### Supported operating systems

The following distributions are supported out of the box from the Azure Gallery:

- **Windows Server 2016 Datacenter**
- **Windows Server 2012 R2 Datacenter**
- **Windows Server 2019 Datacenter**

### Supported VM instances

Accelerated Networking is supported on most general purpose and compute-optimized instance sizes with 2 or more vCPUs. These supported series are: D/DSv2 and F/Fs

On instances that support hyperthreading, Accelerated Networking is supported on VM instances with 4 or more vCPUs. Supported series are: D/Dsv3, E/Esv3, Fsv2, Lsv2, Ms/Mms and Ms/Mmsv2.

For more information on VM instances, see [Windows VM sizes](#).

### Regions

Available in all public Azure regions and Azure Government Cloud.

### Enabling Accelerated Networking on a running VM

A supported VM size without accelerated networking enabled can only have the feature enabled when it is stopped and deallocated.

### Deployment through Azure Resource Manager

Virtual machines (classic) cannot be deployed with Accelerated Networking.

## Create a Windows VM with Azure Accelerated Networking

### Portal creation

Though this article provides steps to create a virtual machine with accelerated networking using Azure Powershell, you can also [create a virtual machine with accelerated networking using the Azure portal](#). When creating a virtual machine in the portal, in the **Create a virtual machine** blade, choose the **Networking** tab. In this tab, there is an option for **Accelerated networking**. If you have chosen a [supported operating system](#) and [VM size](#), this option will automatically populate to "On." If not, it will populate the "Off" option for Accelerated Networking and give the user a reason why it is not be enabled.

- *Note:* Only supported operating systems can be enabled through the portal. If you are using a custom image, and your image supports Accelerated Networking, please create your VM using CLI or Powershell.

After the virtual machine is created, you can confirm Accelerated Networking is enabled by following the instructions in the Confirm that accelerated networking is enabled.

### Powershell creation

### Create a virtual network

## NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see [Introducing the new Azure PowerShell Az module](#). For Az module installation instructions, see [Install Azure PowerShell](#).

Install [Azure PowerShell](#) version 1.0.0 or later. To find your currently installed version, run

`Get-Module -ListAvailable Az`. If you need to install or upgrade, install the latest version of the Az module from the [PowerShell Gallery](#). In a PowerShell session, log in to an Azure account using `Connect-AzAccount`.

In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *myNic*, and *myVM*.

Create a resource group with [New-AzResourceGroup](#). The following example creates a resource group named *myResourceGroup* in the *centralus* location:

```
New-AzResourceGroup -Name "myResourceGroup" -Location "centralus"
```

First, create a subnet configuration with [New-AzVirtualNetworkSubnetConfig](#). The following example creates a subnet named *mySubnet*:

```
$subnet = New-AzVirtualNetworkSubnetConfig `
 -Name "mySubnet" `
 -AddressPrefix "192.168.1.0/24"
```

Create a virtual network with `New-AzVirtualNetwork`, with the *mySubnet* subnet.

```
$vnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
 -Location "centralus" `
 -Name "myVnet" `
 -AddressPrefix "192.168.0.0/16" `
 -Subnet $Subnet
```

## Create a network security group

First, create a network security group rule with [New-AzNetworkSecurityRuleConfig](#).

```
$rdp = New-AzNetworkSecurityRuleConfig `
```

- Name 'Allow-RDP-All' `
- Description 'Allow RDP' `
- Access Allow `
- Protocol Tcp `
- Direction Inbound `
- Priority 100 `
- SourceAddressPrefix \* `
- SourcePortRange \* `
- DestinationAddressPrefix \* `
- DestinationPortRange 3389

Create a network security group with [New-AzNetworkSecurityGroup](#) and assign the *Allow-RDP-All* security rule to it. In addition to the *Allow-RDP-All* rule, the network security group contains several default rules. One default rule disables all inbound access from the Internet, which is why the *Allow-RDP-All* rule is assigned to the network security group so that you can remotely connect to the virtual machine once it's created.

```
$nsg = New-AzNetworkSecurityGroup `
 -ResourceGroupName myResourceGroup `
 -Location centralus `
 -Name "myNsg" `
 -SecurityRules $rdp
```

Associate the network security group to the *mySubnet* subnet with [Set-AzVirtualNetworkSubnetConfig](#). The rule in the network security group is effective for all resources deployed in the subnet.

```
Set-AzVirtualNetworkSubnetConfig `
 -VirtualNetwork $vnet `
 -Name 'mySubnet' `
 -AddressPrefix "192.168.1.0/24" `
 -NetworkSecurityGroup $nsg
```

## Create a network interface with accelerated networking

Create a public IP address with [New-AzPublicIpAddress](#). A public IP address isn't required if you don't plan to access the virtual machine from the Internet, but to complete the steps in this article, it is required.

```
$publicIp = New-AzPublicIpAddress `
 -ResourceGroupName myResourceGroup `
 -Name 'myPublicIp' `
 -location centralus `
 -AllocationMethod Dynamic
```

Create a network interface with [New-AzNetworkInterface](#) with accelerated networking enabled and assign the public IP address to the network interface. The following example creates a network interface named *myNic* in the *mySubnet* subnet of the *myVnet* virtual network and assigns the *myPublicIp* public IP address to it:

```
$nic = New-AzNetworkInterface `
 -ResourceGroupName "myResourceGroup" `
 -Name "myNic" `
 -Location "centralus" `
 -SubnetId $vnet.Subnets[0].Id `
 -PublicIpAddressId $publicIp.Id `
 -EnableAcceleratedNetworking
```

## Create the virtual machine

Set your VM credentials to the `$cred` variable using [Get-Credential](#):

```
$cred = Get-Credential
```

First, define your VM with [New-AzVMConfig](#). The following example defines a VM named *myVM* with a VM size that supports Accelerated Networking (*Standard\_DS4\_v2*):

```
$vmConfig = New-AzVMConfig -VMName "myVm" -VMSize "Standard_DS4_v2"
```

For a list of all VM sizes and characteristics, see [Windows VM sizes](#).

Create the rest of your VM configuration with [Set-AzVMOperatingSystem](#) and [Set-AzVMSourceImage](#). The following example creates a Windows Server 2016 VM:

```
$vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
 -Windows `
 -ComputerName "myVM" `
 -Credential $cred `
 -ProvisionVMAgent `
 -EnableAutoUpdate

$vmConfig = Set-AzVMSourceImage -VM $vmConfig `
 -PublisherName "MicrosoftWindowsServer" `
 -Offer "WindowsServer" `
 -Skus "2016-Datacenter" `
 -Version "latest"
```

Attach the network interface that you previously created with [Add-AzVMNetworkInterface](#):

```
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id
```

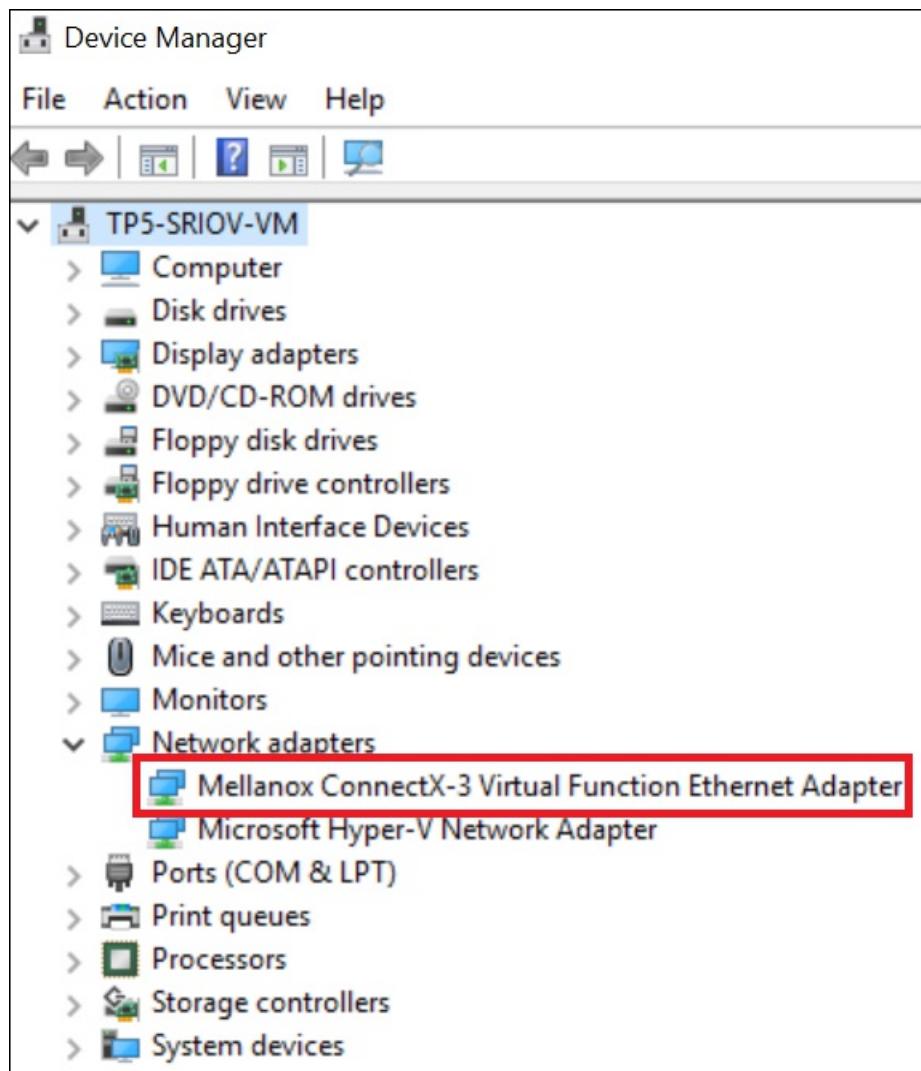
Finally, create your VM with [New-AzVM](#):

```
New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location "centralus"
```

## Confirm the driver is installed in the operating system

Once you create the VM in Azure, connect to the VM and confirm that the driver is installed in Windows.

1. From an Internet browser, open the Azure [portal](#) and sign in with your Azure account.
2. In the box that contains the text *Search resources* at the top of the Azure portal, type *myVm*. When **myVm** appears in the search results, click it. If **Creating** is visible under the **Connect** button, Azure has not yet finished creating the VM. Click **Connect** in the top left corner of the overview only after you no longer see **Creating** under the **Connect** button.
3. Enter the username and password you entered in [Create the virtual machine](#). If you've never connected to a Windows VM in Azure, see [Connect to virtual machine](#).
4. Right-click the Windows Start button and click **Device Manager**. Expand the **Network adapters** node. Confirm that the **Mellanox ConnectX-3 Virtual Function Ethernet Adapter** appears, as shown in the following picture:



Accelerated Networking is now enabled for your VM.

## Enable Accelerated Networking on existing VMs

If you have created a VM without Accelerated Networking, it is possible to enable this feature on an existing VM. The VM must support Accelerated Networking by meeting the following prerequisites that are also outlined above:

- The VM must be a supported size for Accelerated Networking
- The VM must be a supported Azure Gallery image (and kernel version for Linux)
- All VMs in an availability set or VMSS must be stopped/deallocated before enabling Accelerated Networking on any NIC

### Individual VMs & VMs in an availability set

First stop/deallocate the VM or, if an Availability Set, all the VMs in the Set:

```
Stop-AzVM -ResourceGroup "myResourceGroup" `
-Name "myVM"
```

Important, please note, if your VM was created individually, without an availability set, you only need to stop/deallocate the individual VM to enable Accelerated Networking. If your VM was created with an availability set, all VMs contained in the availability set will need to be stopped/deallocated before enabling Accelerated Networking on any of the NICs.

Once stopped, enable Accelerated Networking on the NIC of your VM:

```
$nic = Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
-Name "myNic"

$nic.EnableAcceleratedNetworking = $true

$nic | Set-AzNetworkInterface
```

Restart your VM or, if in an availability set, all the VMs in the set, and confirm that Accelerated Networking is enabled:

```
Start-AzVM -ResourceGroupName "myResourceGroup" `
-Name "myVM"
```

## VMSS

VMSS is slightly different but follows the same workflow. First, stop the VMs:

```
Stop-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet"
```

Once the VMs are stopped, update the Accelerated Networking property under the network interface:

```
$vmss = Get-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet"

$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations[0].EnableAcceleratedNetworking =
$true

Update-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet" `
-VirtualMachineScaleSet $vmss
```

Please note, a VMSS has VM upgrades that apply updates using three different settings, automatic, rolling and manual. In these instructions the policy is set to automatic so that the VMSS will pick up the changes immediately after restarting. To set it to automatic so that the changes are immediately picked up:

```
$vmss.UpgradePolicy.AutomaticOSUpgrade = $true

Update-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet" `
-VirtualMachineScaleSet $vmss
```

Finally, restart the VMSS:

```
Start-AzVmss -ResourceGroupName "myResourceGroup" `
-VMSScaleSetName "myScaleSet"
```

Once you restart, wait for the upgrades to finish but once completed, the VF will appear inside the VM. (Please make sure you are using a supported OS and VM size)

## Resizing existing VMs with Accelerated Networking

VMs with Accelerated Networking enabled can only be resized to VMs that support Accelerated Networking.

A VM with Accelerated Networking enabled cannot be resized to a VM instance that does not support Accelerated Networking using the resize operation. Instead, to resize one of these VMs:

- Stop/Deallocate the VM or if in an availability set/VMSS, stop/deallocate all the VMs in the set/VMSS.
- Accelerated Networking must be disabled on the NIC of the VM or if in an availability set/VMSS, all VMs in the set/VMSS.
- Once Accelerated Networking is disabled, the VM/availability set/VMSS can be moved to a new size that does not support Accelerated Networking and restarted.

# Create a fully qualified domain name in the Azure portal for a Windows VM

12/23/2019 • 2 minutes to read • [Edit Online](#)

When you create a virtual machine (VM) in the [Azure portal](#), a public IP resource for the virtual machine is automatically created. You use this IP address to remotely access the VM. Although the portal does not create a [fully qualified domain name](#), or FQDN, you can create one once the VM is created. This article demonstrates the steps to create a DNS name or FQDN.

## Create a FQDN

This article assumes that you have already created a VM. If needed, you can [create a VM in the portal](#) or [with Azure PowerShell](#). Follow these steps once your VM is up and running:

1. Select your VM in the portal. Under **DNS name**, click **Configure**.

The screenshot shows the Azure portal interface for a virtual machine named 'myVM'. The left sidebar lists navigation options like Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The main content area displays VM details under 'Resource group (change)' 'myResourceGroup'. It includes fields for Computer name ('myVM'), Operating system ('Linux'), Size ('Standard DS1 v2 (1 vcpus, 3.5 GB memory)'), Public IP address ('40.76.54.250'), and Virtual network/subnet ('myVMNET/myVMSubnet'). The 'DNS name' field is labeled 'Configure' and is highlighted with a red box. Below it, there's a 'Tags (change)' section with a link to 'Click here to add tags'.

2. Enter the desired DNS name and then select **Save**.

The screenshot shows the 'Configuration' dialog for a public IP address associated with 'myVM'. The dialog has tabs for 'Save' and 'Discard'. Under 'Assignment', 'Dynamic' is selected. The 'IP address' is listed as '40.76.54.250'. The 'Idle timeout (minutes)' is set to '4'. At the bottom, there's a field for 'DNS name label (optional)' containing '.eastus.cloudapp.azure.com', which is also highlighted with a red box. A note at the bottom encourages users to 'Prefer to use your own domain name? Try Azure DNS now'.

3. To return to the VM overview blade, close the *Public IP address* blade. Verify that the *DNS name* is now shown.

You can now connect remotely to the VM using this DNS name such as for Remote Desktop Protocol (RDP).

## Next steps

Now that your VM has a public IP and DNS name, you can deploy common application frameworks or services such as IIS, SQL, or SharePoint.

You can also read more about [using Resource Manager](#) for tips on building your Azure deployments.

# How Azure DNS works with other Azure services

2/1/2020 • 2 minutes to read • [Edit Online](#)

Azure DNS is a hosted DNS management and name resolution service. You can use it to create public DNS names for other applications and services that you deploy in Azure. Creating a name for an Azure service in your custom domain is simple. You just add a record of the correct type for your service.

- For dynamically allocated IP addresses, you can create a DNS CNAME record that maps to the DNS name that Azure created for your service. DNS standards prevent you from using a CNAME record for the zone apex. You can use an alias record instead. For more information, see [Tutorial: Configure an alias record to refer to an Azure Public IP address](#).
- For statically allocated IP addresses, you can create a DNS A record by using any name, which includes a *naked domain* name at the zone apex.

The following table outlines the supported record types you can use for various Azure services. As the table shows, Azure DNS supports only DNS records for Internet-facing network resources. Azure DNS can't be used for name resolution of internal, private addresses.

| AZURE SERVICE             | NETWORK INTERFACE   | DESCRIPTION                                                                                                                                                                                                                                         |
|---------------------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Azure Application Gateway | Front-end public IP | You can create a DNS A or CNAME record.                                                                                                                                                                                                             |
| Azure Load Balancer       | Front-end public IP | You can create a DNS A or CNAME record. Load Balancer can have an IPv6 public IP address that's dynamically assigned. Create a CNAME record for an IPv6 address.                                                                                    |
| Azure Traffic Manager     | Public name         | You can create an alias record that maps to the trafficmanager.net name assigned to your Traffic Manager profile. For more information, see <a href="#">Tutorial: Configure an alias record to support apex domain names with Traffic Manager</a> . |
| Azure Cloud Services      | Public IP           | For statically allocated IP addresses, you can create a DNS A record. For dynamically allocated IP addresses, you must create a CNAME record that maps to the cloudapp.net name.                                                                    |
| Azure App Service         | External IP         | For external IP addresses, you can create a DNS A record. Otherwise, you must create a CNAME record that maps to the azurewebsites.net name. For more information, see <a href="#">Map a custom domain name to an Azure app</a> .                   |

| AZURE SERVICE              | NETWORK INTERFACE         | DESCRIPTION                                                                                                                                                                                                                                                      |
|----------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Azure Resource Manager VMs | <a href="#">Public IP</a> | Resource Manager VMs can have public IP addresses. A VM with a public IP address also can be behind a load balancer. You can create a DNS A, CNAME, or alias record for the public address. You can use this custom name to bypass the VIP on the load balancer. |
| Classic VMs                | <a href="#">Public IP</a> | Classic VMs created by using PowerShell or CLI can be configured with a dynamic or static (reserved) virtual address. You can create a DNS CNAME or an A record, respectively.                                                                                   |

# Azure virtual machine extensions and features

11/13/2019 • 3 minutes to read • [Edit Online](#)

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs, you can use existing images and then customize them as part of your deployments, getting you out of the business of custom image building.

The Azure platform hosts many extensions that range from VM configuration, monitoring, security, and utility applications. Publishers take an application, then wrap it into an extension, and simplify the installation, so all you need to do is provide mandatory parameters.

There is a large choice of first and third party extensions, if the application in the extension repository does not exist, then you can use the Custom Script extension and configure your VM with your own scripts and commands.

Examples of key scenarios that extensions are used for:

- VM configuration, you can use Powershell DSC (Desired State Configuration), Chef, Puppet and Custom Script Extensions to install VM configuration agents and configure your VM.
- AV products, such as Symantec, ESET.
- VM vulnerability tool, such as Qualys, Rapid7, HPE.
- VM and App monitoring tooling, such as DynaTrace, Azure Network Watcher, Site24x7, and Stackify.

Extensions can be bundled with a new VM deployment. For example, they can be part of a larger deployment, configuring applications on VM provision, or run against any supported extension operated systems post deployment.

## How can I find What extensions are available?

You can view available extensions in the VM blade in the Portal, under extensions, this represents just a small amount, for the full list, you can use the CLI tools, see [Discovering VM Extensions for Linux](#) and [Discovering VM Extensions for Windows](#).

## How can I install an extension?

Azure VM extensions can be managed using either the Azure CLI, Azure PowerShell, Azure Resource Manager templates, and the Azure portal. To try an extension, you can go to the Azure portal, select the Custom Script Extension, then pass in a command / script and run the extensions.

If you want to same extension you added in the portal by CLI or Resource Manager template, see different extension documentation, such as [Windows Custom Script Extension](#) and [Linux Custom Script Extension](#).

## How do I manage extension application lifecycle?

You do not need to connect to a VM directly to install or delete the extension. As the Azure extension application lifecycle is managed outside of the VM and integrated into the Azure platform, you also get integrated status of the extension.

## Anything else I should be thinking about for extensions?

Extensions install applications, like any applications there are some requirements, for extensions there is a list of supported Windows and Linux OSes, and you need to have the Azure VM agents installed. Some individual VM

extension applications may have their own environmental prerequisites, such as access to an endpoint.

## Troubleshoot extensions

Troubleshooting information for each extension can be found in the **Troubleshoot and support** section in the overview for the extension. Here is a list of the troubleshooting information available:

| NAMESPACE                                                         | TROUBLESHOOTING                                         |
|-------------------------------------------------------------------|---------------------------------------------------------|
| microsoft.azure.monitoring.dependencyagent.dependencyagentlinux   | <a href="#">Azure Monitor Dependency for Linux</a>      |
| microsoft.azure.monitoring.dependencyagent.dependencyagentwindows | <a href="#">Azure Monitor Dependency for Windows</a>    |
| microsoft.azure.security.azurediskencryptionforlinux              | <a href="#">Azure Disk Encryption for Linux</a>         |
| microsoft.azure.security.azurediskencryption                      | <a href="#">Azure Disk Encryption for Windows</a>       |
| microsoft.compute.customscriptextension                           | <a href="#">Custom Script for Windows</a>               |
| microsoft.ostcextensions.customscriptforlinux                     | <a href="#">Desired State Configuration for Linux</a>   |
| microsoft.powershell.dsc                                          | <a href="#">Desired State Configuration for Windows</a> |
| microsoft.hpccompute.nvidiagpudriverlinux                         | <a href="#">NVIDIA GPU Driver Extension for Linux</a>   |
| microsoft.hpccompute.nvidiagpudriverwindows                       | <a href="#">NVIDIA GPU Driver Extension for Windows</a> |
| microsoft.azure.security.iaasantimalware                          | <a href="#">Antimalware Extension for Windows</a>       |
| microsoft.enterprisecloud.monitoring.omsagentforlinux             | <a href="#">Azure Monitor for Linux</a>                 |
| microsoft.enterprisecloud.monitoring.microsoftmonitoringagent     | <a href="#">Azure Monitor for Windows</a>               |
| stackify.linuxagent.extension.stackifylinuxagentextension         | <a href="#">Stackify Retrace for Linux</a>              |
| vmaccessforlinux.microsoft.ostcextensions                         | <a href="#">Reset password (VMAccess) for Linux</a>     |
| microsoft.recoveryservices.vmsnapshot                             | <a href="#">Snapshot for Linux</a>                      |
| microsoft.recoveryservices.vmsnapshot                             | <a href="#">Snapshot for Windows</a>                    |

## Next steps

- For more information about how the Linux Agent and Extensions work, see [Azure VM extensions and features for Linux](#).
- For more information about how the Windows Guest Agent and Extensions work, see [Azure VM extensions and features for Windows](#).
- To install the Windows Guest Agent, see [Azure Windows Virtual Machine Agent Overview](#).
- To install the Linux Agent, see [Azure Linux Virtual Machine Agent Overview](#).

# Move a Windows VM to another Azure subscription or resource group

2/28/2020 • 2 minutes to read • [Edit Online](#)

This article walks you through how to move a Windows virtual machine (VM) between resource groups or subscriptions. Moving between subscriptions can be handy if you originally created a VM in a personal subscription and now want to move it to your company's subscription to continue your work. You do not need to start the VM in order to move it and it should continue to run during the move.

## IMPORTANT

New resource IDs are created as part of the move. After the VM has been moved, you will need to update your tools and scripts to use the new resource IDs.

## Use the Azure portal to move a VM to a different subscription

You can move a VM and its associated resources to a different subscription by using the Azure portal.

1. Go to the [Azure portal](#) to manage the resource group containing the VM to move. Search for and select **Resource groups**.
2. Choose the resource group containing the VM that you would like to move.
3. At the top of the page for the resource group, select **Move** and then select **Move to another subscription**. The **Move resources** page opens.
4. Select each of the resources to move. In most cases, you should move all of the related resources that are listed.
5. Select the **Subscription** where you want the VM to be moved.
6. Select an existing **Resource group**, or enter a name to have a new resource group created.
7. When you are done, select that you understand that new resource IDs will be created and that the new IDs will need to be used with the VM after it is moved, and then select **OK**.

## Use the Azure portal to move a VM to another resource group

You can move a VM and its associated resources to another resource group by using the Azure portal.

1. Go to the [Azure portal](#) to manage the resource group containing the VM to move. Search for and select **Resource groups**.
2. Choose the resource group containing the VM that you would like to move.
3. At the top of the page for the resource group, select **Move** and then select **Move to another resource group**. The **Move resources** page opens.
4. Select each of the resources to move. In most cases, you should move all of the related resources that are listed.
5. Select an existing **Resource group**, or enter a name to have a new resource group created.
6. When you are done, select that you understand that new resource IDs will be created and that the new IDs will need to be used with the VM after it is moved, and then select **OK**.

## Use Powershell to move a VM

To move a virtual machine to another resource group, you need to make sure that you also move all of the dependent resources. To get a list with the resource ID of each of these resources, use the `Get-AzResource` cmdlet.

```
Get-AzResource -ResourceGroupName myResourceGroup | Format-table -wrap -Property ResourceId
```

You can use the output of the previous command to create a comma-separated list of resource IDs to [Move-AzResource](#) to move each resource to the destination.

```
Move-AzResource -DestinationResourceGroupName "myDestinationResourceGroup" `
-ResourceId <myResourceId,myResourceId,myResourceId>
```

To move the resources to different subscription, include the **-DestinationSubscriptionId** parameter.

```
Move-AzResource -DestinationSubscriptionId "<myDestinationSubscriptionID>" `
-DestinationResourceGroupName "<myDestinationResourceGroup>" `
-ResourceId <myResourceId,myResourceId,myResourceId>
```

When you are asked to confirm that you want to move the specified resources, enter **Y** to confirm.

## Next steps

You can move many different types of resources between resource groups and subscriptions. For more information, see [Move resources to a new resource group or subscription](#).

# Move Azure VMs to another region

11/12/2019 • 6 minutes to read • [Edit Online](#)

There are various scenarios in which you'd want to move your existing Azure IaaS virtual machines (VMs) from one region to another. For example, you want to improve reliability and availability of your existing VMs, to improve manageability, or to move for governance reasons. For more information, see the [Azure VM move overview](#).

You can use the [Azure Site Recovery](#) service to manage and orchestrate disaster recovery of on-premises machines and Azure VMs for business continuity and disaster recovery (BCDR). You can also use Site Recovery to manage the move of Azure VMs to a secondary region.

In this tutorial, you will:

- Verify prerequisites for the move
- Prepare the source VMs and the target region
- Copy the data and enable replication
- Test the configuration and perform the move
- Delete the resources in the source region

## NOTE

This tutorial shows you how to move Azure VMs from one region to another as is. If you need to improve availability by moving VMs in an availability set to zone pinned VMs in a different region, see the [Move Azure VMs into Availability Zones tutorial](#).

## Prerequisites

- Make sure that the Azure VMs are in the Azure region from which you want to move.
- Verify that your choice of [source region - target region combination is supported](#), and make an informed decision about the target region.
- Make sure that you understand the [scenario architecture and components](#).
- Review the [support limitations and requirements](#).
- Verify account permissions. If you created your free Azure account, you're the administrator of your subscription. If you're not the subscription administrator, work with the administrator to assign the permissions that you need. To enable replication for a VM and essentially copy data by using Azure Site Recovery, you must have:
  - Permissions to create a VM in Azure resources. The Virtual Machine Contributor built-in role has these permissions, which include:
  - Permission to create a VM in the selected resource group
  - Permission to create a VM in the selected virtual network
  - Permission to write to the selected storage account
  - Permissions to manage Azure Site Recovery operations. The Site Recovery Contributor role has all the permissions that are required to manage Site Recovery operations in a Recovery Services vault.

- Make sure that all the latest root certificates are on the Azure VMs that you want to move. If the latest root certificates aren't on the VM, security constraints will prevent the data copy to the target region.
- For Windows VMs, install all the latest Windows updates on the VM, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows Update and certificate update processes for your organization.
- For Linux VMs, follow the guidance provided by your Linux distributor to get the latest trusted root certificates and certificate revocation list on the VM.
- Make sure that you're not using an authentication proxy to control network connectivity for VMs that you want to move.
- If the VM that you're trying to move doesn't have access to the internet, or it's using a firewall proxy to control outbound access, [check the requirements](#).
- Identify the source networking layout and all the resources that you're currently using. This includes but isn't limited to load balancers, network security groups (NSGs), and public IPs.
- Verify that your Azure subscription allows you to create VMs in the target region that's used for disaster recovery. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support VMs with sizes that match your source VMs. If you're using Site Recovery to copy data to the target, Site Recovery chooses the same size or the closest possible size for the target VM.
- Make sure that you create a target resource for every component that's identified in the source networking layout. This step is important to ensure that your VMs have all the functionality and features in the target region that you had in the source region.

**NOTE**

Azure Site Recovery automatically discovers and creates a virtual network when you enable replication for the source VM. You can also pre-create a network and assign it to the VM in the user flow for enable replication. As mentioned later, you need to manually create any other resources in the target region.

To create the most commonly used network resources that are relevant for you based on the source VM configuration, see the following documentation:

- [Network security groups](#)
- [Load balancers](#)
- [Public IP](#)
- For any other networking components, see the [networking documentation](#).

## Prepare

The following steps shows how to prepare the virtual machine for the move using Azure Site Recovery as a solution.

### Create the vault in any region, except the source region

1. Sign in to the [Azure portal](#) > **Recovery Services**.
2. Select **Create a resource** > **Management Tools** > **Backup and Site Recovery**.
3. In **Name**, specify the friendly name **ContosoVMVault**. If you have more than one subscription, select the appropriate one.
4. Create the resource group **ContosoRG**.

- Specify an Azure region. To check supported regions, see geographic availability in [Azure Site Recovery pricing details](#).
- In **Recovery Services vaults**, select **Overview > ContosoVMVault > +Replicate**.
- In **Source**, select **Azure**.
- In **Source location**, select the source Azure region where your VMs are currently running.
- Select the Resource Manager deployment model. Then select the **Source subscription** and **Source resource group**.
- Select **OK** to save the settings.

#### **Enable replication for Azure VMs and start copying the data**

Site Recovery retrieves a list of the VMs that are associated with the subscription and resource group.

- In the next step, select the VM that you want to move, then select **OK**.
- In **Settings**, select **Disaster recovery**.
- In **Configure disaster recovery > Target region**, select the target region to which you'll replicate.
- For this tutorial, accept the other default settings.
- Select **Enable replication**. This step starts a job to enable replication for the VM.

The screenshot shows the 'Configure settings' dialog box with the following configuration:

- Resource group, Network, Storage and Availability sets**: Includes a 'Customize' link. Description: By default Azure Site Recovery(ASR) will mirror the source site configuration to target site by creating/using the required resource groups, storage accounts, virtual network and availability sets as below. Click 'Customize' above to change the configuration. The resources created by ASR are appended with "asr" suffix.
- Target resource group**: ContosoRG
- Target virtual network**: A2ATest2-vnet-asr(new)
- Cache storage accounts**: a2atest2disks86cacheasr(new)
- Target storage accounts**: a2atest2disks864asr(new)
- Target availability sets**: (empty)
- Replication Policy**: Includes a 'Customize' link. Details: Name: 24-hour-retention-policy, Recovery point retention: 24 hour(s), App consistent snapshot frequency: 4 hour(s).

## Move

The following steps shows how to perform the move to the target region.

- Go to the vault. In **Settings > Replicated items**, select the VM, and then select **Failover**.
- In **Failover**, select **Latest**.

3. Select **Shut down machine before beginning failover**. Site Recovery attempts to shut down the source VM before triggering the failover. Failover continues even if shutdown fails. You can follow the failover progress on the **Jobs** page.
4. After the job is finished, check that the VM appears in the target Azure region as expected.

## Discard

In case you checked the moved VM and need to make changes to point of failover or want to go back to a previous point, in the **Replicated items**, right-select the VM > **Change recovery point**. This step provides you the option to specify a different recovery point and failover to that one.

## Commit

Once you have checked the moved VM and are ready to commit the change, in the **Replicated items**, right-select the VM > **Commit**. This step finishes the move process to the target region. Wait until the commit job finishes.

## Clean up

The following steps will guide you through how to clean up the source region as well as related resources that were used for the move.

For all resources that were used for the move:

- Go to the VM. Select **Disable Replication**. This step stops the process from copying the data for the VM.

### IMPORTANT

It's important to perform this step to avoid being charged for Azure Site Recovery replication.

If you have no plans to reuse any of the source resources, complete these additional steps:

1. Delete all the relevant network resources in the source region that you identified in [prerequisites](#).
2. Delete the corresponding storage account in the source region.

## Next steps

In this tutorial, you moved an Azure VM to a different Azure region. Now you can configure disaster recovery for the VM that you moved.

[Set up disaster recovery after migration](#)

# Move Azure VMs into Availability Zones

11/6/2019 • 7 minutes to read • [Edit Online](#)

Availability Zones in Azure help protect your applications and data from datacenter failures. Each Availability Zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions. The physical separation of Availability Zones within a region helps protect applications and data from datacenter failures. With Availability Zones, Azure offers a service-level agreement (SLA) of 99.99% for uptime of virtual machines (VMs). Availability Zones are supported in select regions, as mentioned in [What are Availability Zones in Azure?](#).

In a scenario where your VMs are deployed as *single instance* into a specific region, and you want to improve your availability by moving these VMs into an Availability Zone, you can do so by using Azure Site Recovery. This action can further be categorized into:

- Move single-instance VMs into Availability Zones in a target region
- Move VMs in an availability set into Availability Zones in a target region

## IMPORTANT

Currently, Azure Site Recovery supports moving VMs from one region to another but doesn't support moving within a region.

## Check prerequisites

- Check whether the target region has [support for Availability Zones](#). Check that your choice of [source region/target region combination is supported](#). Make an informed decision on the target region.
- Make sure that you understand the [scenario architecture and components](#).
- Review the [support limitations and requirements](#).
- Check account permissions. If you just created your free Azure account, you're the admin of your subscription. If you aren't the subscription admin, work with the admin to assign the permissions you need. To enable replication for a VM and eventually copy data to the target by using Azure Site Recovery, you must have:
  1. Permission to create a VM in Azure resources. The *Virtual Machine Contributor* built-in role has these permissions, which include:
    - Permission to create a VM in the selected resource group
    - Permission to create a VM in the selected virtual network
    - Permission to write to the selected storage account
  2. Permission to manage Azure Site Recovery tasks. The *Site Recovery Contributor* role has all permissions required to manage Site Recovery actions in a Recovery Services vault.

## Prepare the source VMs

1. Your VMs should use managed disks if you want to move them to an Availability Zone by using Site Recovery. You can convert existing Windows VMs that use unmanaged disks to use managed disks. Follow the steps at [Convert a Windows virtual machine from unmanaged disks to managed disks](#). Ensure that the availability set is configured as *managed*.

2. Check that all the latest root certificates are present on the Azure VMs you want to move. If the latest root certificates aren't present, the data copy to the target region can't be enabled because of security constraints.
3. For Windows VMs, install all the latest Windows updates on the VM, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows update and certificate update processes for your organization.
4. For Linux VMs, follow the guidance provided by your Linux distributor to get the latest trusted root certificates and certificate revocation list on the VM.
5. Make sure you don't use an authentication proxy to control network connectivity for VMs that you want to move.
6. If the VM you're trying to move doesn't have access to the internet and uses a firewall proxy to control outbound access, check the requirements at [Configure outbound network connectivity](#).
7. Identify the source networking layout and the resources you currently use for verification, including load balancers, NSGs, and public IP.

## Prepare the target region

1. Check that your Azure subscription lets you create VMs in the target region used for disaster recovery. If necessary, contact support to enable the required quota.
2. Make sure your subscription has enough resources to support VMs with sizes that match your source VMs. If you use Site Recovery to copy data to the target, it picks the same size or the closest possible size for the target VM.
3. Create a target resource for every component identified in the source networking layout. This action ensures that after you cut over to the target region, your VMs have all the functionality and features that you had in the source.

### NOTE

Azure Site Recovery automatically discovers and creates a virtual network and storage account when you enable replication for the source VM. You can also pre-create these resources and assign to the VM as part of the enable replication step. But for any other resources, as mentioned later, you need to manually create them in the target region.

The following documents tell how to create the most commonly used network resources that are relevant to you, based on the source VM configuration.

- [Network security groups](#)
- [Load balancers](#)
- [Public IP](#)

For any other networking components, refer to the networking [documentation](#).

### IMPORTANT

Ensure that you use a zone-redundant load balancer in the target. You can read more at [Standard Load Balancer and Availability Zones](#).

4. Manually [create a non-production network](#) in the target region if you want to test the configuration before you cut over to the target region. We recommend this approach because it causes minimal interference with the production environment.

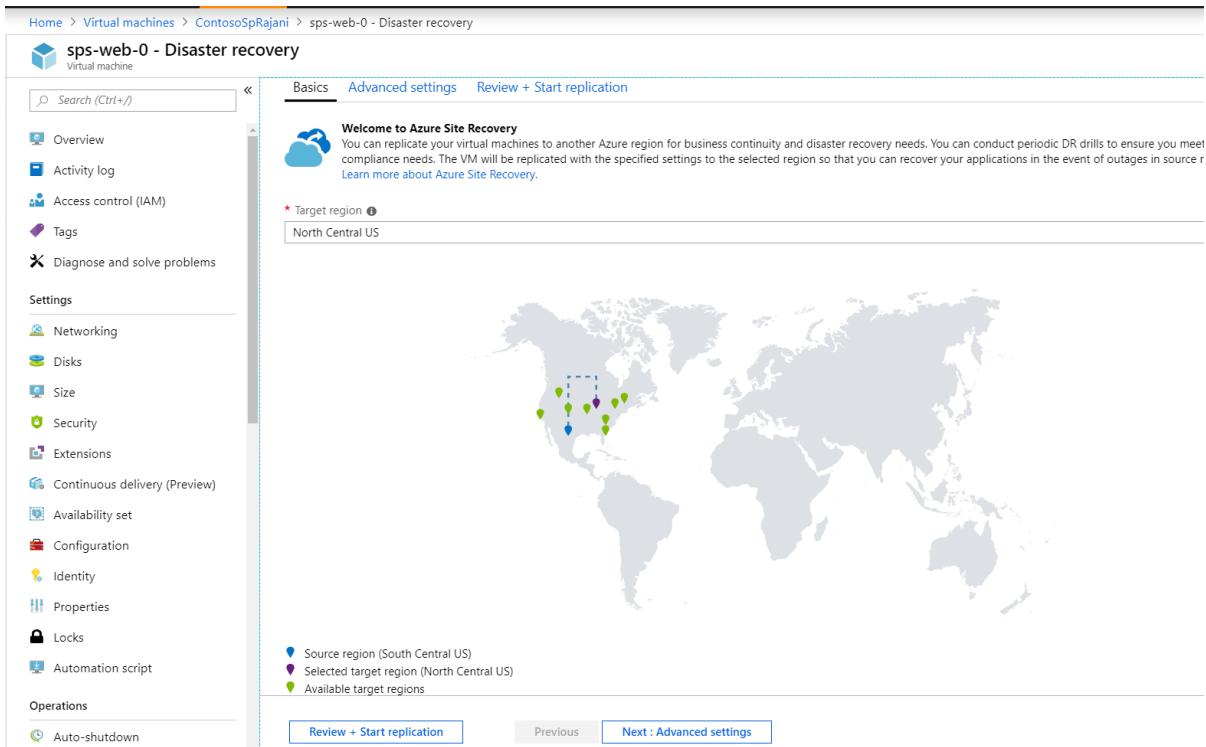
# Enable replication

The following steps will guide you when using Azure Site Recovery to enable replication of data to the target region, before you eventually move them into Availability Zones.

## NOTE

These steps are for a single VM. You can extend the same to multiple VMs. Go to the Recovery Services vault, select + **Replicate**, and select the relevant VMs together.

1. In the Azure portal, select **Virtual machines**, and select the VM you want to move into Availability Zones.
2. In **Operations**, select **Disaster recovery**.
3. In **Configure disaster recovery > Target region**, select the target region to which you'll replicate. Ensure this region [supports](#) Availability Zones.



4. Select **Next: Advanced settings**.
5. Choose the appropriate values for the target subscription, target VM resource group, and virtual network.
6. In the **Availability** section, choose the Availability Zone into which you want to move the VM.

## NOTE

If you don't see the option for availability set or Availability Zone, ensure that the [prerequisites](#) are met and the [preparation](#) of source VMs is complete.

The screenshot shows the 'Review + Start replication' step in the Azure portal. Under the 'Availability' section, a red box highlights the 'Availability zone' dropdown which lists three options: 1, 2, and 3. Other sections like 'Storage settings', 'Replication settings', and 'Extension settings' are also visible.

- Select **Enable Replication**. This action starts a job to enable replication for the VM.

## Check settings

After the replication job has finished, you can check the replication status, modify replication settings, and test the deployment.

- In the VM menu, select **Disaster recovery**.
- You can check replication health, the recovery points that have been created and the source, and target regions on the map.

The screenshot shows the 'Disaster recovery' blade for a specific VM. The left sidebar has 'Disaster recovery' selected. The main area shows replication settings (Active location: East US, Target location: South Central US), replication health (Healthy), and latest recovery points (Crash-consistent: 8/21/2018, 3:23:56 PM; App-consistent: 8/21/2018, 3:23:56 PM). A world map on the right shows the geographical distribution of recovery points.

## Test the configuration

- In the virtual machine menu, select **Disaster recovery**.

2. Select the **Test Failover** icon.
3. In **Test Failover**, select a recovery point to use for the failover:
  - **Latest processed:** Fails the VM over to the latest recovery point that was processed by the Site Recovery service. The time stamp is shown. With this option, no time is spent processing data, so it provides a low recovery time objective (RTO).
  - **Latest app-consistent:** This option fails over all VMs to the latest app-consistent recovery point. The time stamp is shown.
  - **Custom:** Select any recovery point.
4. Select the test target Azure virtual network to which you want to move the Azure VMs to test the configuration.

#### IMPORTANT

We recommend that you use a separate Azure VM network for the test failure, and not the production network in the target region into which you want to move your VMs.

5. To start testing the move, select **OK**. To track progress, select the VM to open its properties. Or, you can select the **Test Failover** job in the vault name > **Settings** > **Jobs** > **Site Recovery jobs**.
6. After the failover finishes, the replica Azure VM appears in the Azure portal > **Virtual Machines**. Make sure that the VM is running, sized appropriately, and connected to the appropriate network.
7. If you want to delete the VM created as part of testing the move, select **Cleanup test failover** on the replicated item. In **Notes**, record and save any observations associated with the test.

## Move to the target region and confirm

1. In the virtual machine menu, select **Disaster recovery**.
2. Select the **Failover** icon.
3. In **Failover**, select **Latest**.
4. Select **Shut down machine before beginning failover**. Site Recovery attempts to shut down the source VM before triggering the failover. Failover continues even if shutdown fails. You can follow the failover progress on the **Jobs** page.
5. After the job is finished, check that the VM appears in the target Azure region as expected.
6. In **Replicated items**, right-click the VM > **Commit**. This finishes the move process to the target region. Wait until the commit job is finished.

## Discard the resource in the source region

Go to the VM. Select **Disable Replication**. This action stops the process of copying the data for the VM.

#### IMPORTANT

Do the preceding step to avoid getting charged for Site Recovery replication after the move. The source replication settings are cleaned up automatically. Note that the Site Recovery extension that is installed as part of the replication isn't removed and needs to be removed manually.

## Next steps

In this tutorial, you increased the availability of an Azure VM by moving into an availability set or Availability Zone. Now you can set disaster recovery for the moved VM.

[Set up disaster recovery after migration](#)

# Migrate from Amazon Web Services (AWS) and other platforms to Managed Disks in Azure

11/13/2019 • 4 minutes to read • [Edit Online](#)

You can upload VHD files from AWS or on-premises virtualization solutions to Azure to create VMs that take advantage of Managed Disks. Azure Managed Disks removes the need to manage storage accounts for Azure IaaS VMs. You have to only specify the type (Premium or Standard) and size of disk you need, and Azure creates and manages the disk for you.

You can upload either generalized and specialized VHDs.

- **Generalized VHD** - has had all of your personal account information removed using Sysprep.
- **Specialized VHD** - maintains the user accounts, applications, and other state data from your original VM.

## IMPORTANT

Before uploading any VHD to Azure, you should follow [Prepare a Windows VHD or VHDX to upload to Azure](#)

| SCENARIO                                                                                                                | DOCUMENTATION                                                                   |
|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| You have existing AWS EC2 instances that you would like to migrate to Azure VMs using managed disks                     | <a href="#">Move a VM from Amazon Web Services (AWS) to Azure</a>               |
| You have a VM from another virtualization platform that you would like to use as an image to create multiple Azure VMs. | <a href="#">Upload a generalized VHD and use it to create a new VM in Azure</a> |
| You have a uniquely customized VM that you would like to recreate in Azure.                                             | <a href="#">Upload a specialized VHD to Azure and create a new VM</a>           |

## Overview of Managed Disks

Azure Managed Disks simplifies VM management by removing the need to manage storage accounts. Managed Disks also benefit from better reliability of VMs in an Availability Set. It ensures that the disks of different VMs in an Availability Set are sufficiently isolated from each other to avoid a single point of failure. It automatically places disks of different VMs in an Availability Set in different Storage scale units (stamps) which limits the impact of single Storage scale unit failures caused due to hardware and software failures. Based on your needs, you can choose from four types of storage options. To learn about the available disk types, see our article [Select a disk type](#).

## Plan for the migration to Managed Disks

This section helps you to make the best decision on VM and disk types.

If you are planning on migrating from unmanaged disks to managed disks, you should be aware that users with the [Virtual Machine Contributor](#) role will not be able to change the VM size (as they could pre-conversion). This is because VMs with managed disks require the user to have the Microsoft.Compute/disks/write permission on the OS disks.

### Location

Pick a location where Azure Managed Disks are available. If you are migrating to Premium Managed Disks, also

ensure that Premium storage is available in the region where you are planning to migrate to. See [Azure Services by Region](#) for up-to-date information on available locations.

## VM sizes

If you are migrating to Premium Managed Disks, you have to update the size of the VM to Premium Storage capable size available in the region where VM is located. Review the VM sizes that are Premium Storage capable. The Azure VM size specifications are listed in [Sizes for virtual machines](#). Review the performance characteristics of virtual machines that work with Premium Storage and choose the most appropriate VM size that best suits your workload. Make sure that there is sufficient bandwidth available on your VM to drive the disk traffic.

## Disk sizes

### Premium Managed Disks

There are seven types of premium managed disks that can be used with your VM and each has specific IOPs and throughput limits. Take into consideration these limits when choosing the Premium disk type for your VM based on the needs of your application in terms of capacity, performance, scalability, and peak loads.

| PREMIUM DISKS TYPE  | P4               | P6               | P10               | P15               | P20               | P30               | P40               | P50               |
|---------------------|------------------|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Disk size           | 32 GB            | 64 GB            | 128 GB            | 256 GB            | 512 GB            | 1024 GB (1 TB)    | 2048 GB (2 TB)    | 4095 GB (4 TB)    |
| IOPS per disk       | 120              | 240              | 500               | 1100              | 2300              | 5000              | 7500              | 7500              |
| Throughput per disk | 25 MB per second | 50 MB per second | 100 MB per second | 125 MB per second | 150 MB per second | 200 MB per second | 250 MB per second | 250 MB per second |

### Standard Managed Disks

There are seven types of standard managed disks that can be used with your VM. Each of them have different capacity but have same IOPS and throughput limits. Choose the type of Standard Managed disks based on the capacity needs of your application.

| STANDARD DISK TYPE  | S4               | S6               | S10              | S15              | S20              | S30              | S40              | S50              |
|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Disk size           | 30 GB            | 64 GB            | 128 GB           | 256 GB           | 512 GB           | 1024 GB (1 TB)   | 2048 GB (2TB)    | 4095 GB (4 TB)   |
| IOPS per disk       | 500              | 500              | 500              | 500              | 500              | 500              | 500              | 500              |
| Throughput per disk | 60 MB per second |

## Disk caching policy

### Premium Managed Disks

By default, disk caching policy is *Read-Only* for all the Premium data disks, and *Read-Write* for the Premium operating system disk attached to the VM. This configuration setting is recommended to achieve the optimal performance for your application's IOs. For write-heavy or write-only data disks (such as SQL Server log files),

disable disk caching so that you can achieve better application performance.

## Pricing

Review the [pricing for Managed Disks](#). Pricing of Premium Managed Disks is same as the Premium Unmanaged Disks. But pricing for Standard Managed Disks is different than Standard Unmanaged Disks.

## Next Steps

- Before uploading any VHD to Azure, you should follow [Prepare a Windows VHD or VHDX to upload to Azure](#)

# Upload a generalized VHD and use it to create new VMs in Azure

12/13/2019 • 2 minutes to read • [Edit Online](#)

This article walks you through using PowerShell to upload a VHD of a generalized VM to Azure, create an image from the VHD, and create a new VM from that image. You can upload a VHD exported from an on-premises virtualization tool or from another cloud. Using [Managed Disks](#) for the new VM simplifies the VM management and provides better availability when the VM is placed in an availability set.

For a sample script, see [Sample script to upload a VHD to Azure and create a new VM](#).

## Before you begin

- Before uploading any VHD to Azure, you should follow [Prepare a Windows VHD or VHDX to upload to Azure](#).
- Review [Plan for the migration to Managed Disks](#) before starting your migration to Managed Disks.

## Generalize the source VM by using Sysprep

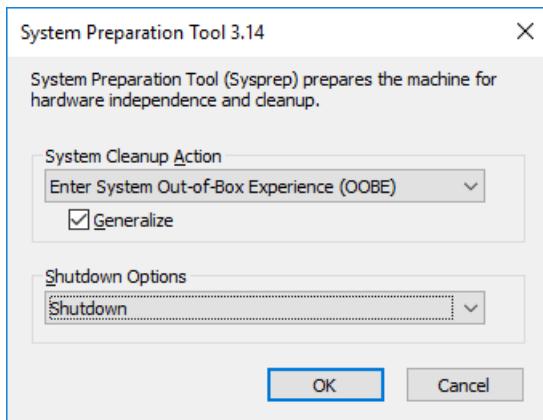
If you haven't already, you need to Sysprep the VM before uploading the VHD to Azure. Sysprep removes all your personal account information, among other things, and prepares the machine to be used as an image. For details about Sysprep, see the [Sysprep Overview](#).

Make sure the server roles running on the machine are supported by Sysprep. For more information, see [Sysprep Support for Server Roles](#).

### IMPORTANT

If you plan to run Sysprep before uploading your VHD to Azure for the first time, make sure you have [prepared your VM](#).

1. Sign in to the Windows virtual machine.
2. Open the Command Prompt window as an administrator. Change the directory to %windir%\system32\sysprep, and then run `sysprep.exe`.
3. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that the **Generalize** check box is enabled.
4. For **Shutdown Options**, select **Shutdown**.
5. Select **OK**.



- When Sysprep finishes, it shuts down the virtual machine. Do not restart the VM.

## Upload the VHD

You can now upload a VHD straight into a managed disk. For instructions, see [Upload a VHD to Azure using Azure PowerShell](#).

Once the VHD is uploaded to the managed disk, you need to use [Get-AzDisk](#) to get the managed disk.

```
$disk = Get-AzDisk -ResourceGroupName 'myResourceGroup' -DiskName 'myDiskName'
```

## Create the image

Create a managed image from your generalized OS managed disk. Replace the following values with your own information.

First, set some variables:

```
$location = 'East US'
$imageName = 'myImage'
$rgName = 'myResourceGroup'
```

Create the image using your managed disk.

```
$imageConfig = New-AzImageConfig `
-Location $location
$imageConfig = Set-AzImageOsDisk `
-Image $imageConfig `
-OsState Generalized `
-OsType Windows `
-ManagedDiskId $disk.Id
```

Create the image.

```
$image = New-AzImage `
-ImageName $imageName `
-ResourceGroupName $rgName `
-Image $imageConfig
```

## Create the VM

Now that you have an image, you can create one or more new VMs from the image. This example creates a VM

named *myVM* from *myImage*, in *myResourceGroup*.

```
New-AzVm `‐
-ResourceGroupName $rgName `‐
-Name "myVM" `‐
-Image $image.Id `‐
-Location $location `‐
-VirtualNetworkName "myVnet" `‐
-SubnetName "mySubnet" `‐
-SecurityGroupName "myNSG" `‐
-PublicIpAddressName "myPIP" `‐
-OpenPorts 3389
```

## Next steps

Sign in to your new virtual machine. For more information, see [How to connect and log on to an Azure virtual machine running Windows](#).

# Move a Windows VM from Amazon Web Services (AWS) to an Azure virtual machine

11/13/2019 • 2 minutes to read • [Edit Online](#)

If you are evaluating Azure virtual machines for hosting your workloads, you can export an existing Amazon Web Services (AWS) EC2 Windows VM instance then upload the virtual hard disk (VHD) to Azure. Once the VHD is uploaded, you can create a new VM in Azure from the VHD.

This article covers moving a single VM from AWS to Azure. If you want to move VMs from AWS to Azure at scale, see [Migrate virtual machines in Amazon Web Services \(AWS\) to Azure with Azure Site Recovery](#).

## Prepare the VM

You can upload both generalized and specialized VHDs to Azure. Each type requires that you prepare the VM before exporting from AWS.

- **Generalized VHD** - a generalized VHD has had all of your personal account information removed using Sysprep. If you intend to use the VHD as an image to create new VMs from, you should:
  - [Prepare a Windows VM](#).
  - Generalize the virtual machine using Sysprep.
- **Specialized VHD** - a specialized VHD maintains the user accounts, applications and other state data from your original VM. If you intend to use the VHD as-is to create a new VM, ensure the following steps are completed.
  - [Prepare a Windows VHD to upload to Azure](#). **Do not** generalize the VM using Sysprep.
  - Remove any guest virtualization tools and agents that are installed on the VM (i.e. VMware tools).
  - Ensure the VM is configured to pull its IP address and DNS settings via DHCP. This ensures that the server obtains an IP address within the VNet when it starts up.

## Export and download the VHD

Export the EC2 instance to a VHD in an Amazon S3 bucket. Follow the steps in the Amazon documentation article [Exporting an Instance as a VM Using VM Import/Export](#) and run the `create-instance-export-task` command to export the EC2 instance to a VHD file.

The exported VHD file is saved in the Amazon S3 bucket you specify. The basic syntax for exporting the VHD is below, just replace the placeholder text in <brackets> with your information.

```
aws ec2 create-instance-export-task --instance-id <instanceID> --target-environment Microsoft \
--export-to-s3-task DiskImageFormat=VHD,ContainerFormat=ova,S3Bucket=<bucket>,S3Prefix=<prefix>
```

Once the VHD has been exported, follow the instructions in [How Do I Download an Object from an S3 Bucket?](#) to download the VHD file from the S3 bucket.

### IMPORTANT

AWS charges data transfer fees for downloading the VHD. See [Amazon S3 Pricing](#) for more information.

## Next steps

Now you can upload the VHD to Azure and create a new VM.

- If you ran Sysprep on your source to **generalize** it before exporting, see [Upload a generalized VHD and use it to create a new VMs in Azure](#)
- If you did not run Sysprep before exporting, the VHD is considered **specialized**, see [Upload a specialized VHD to Azure and create a new VM](#)

2 minutes to read

# Understand the structure and syntax of Azure Resource Manager templates

2/26/2020 • 13 minutes to read • [Edit Online](#)

This article describes the structure of an Azure Resource Manager template. It presents the different sections of a template and the properties that are available in those sections.

This article is intended for users who have some familiarity with Resource Manager templates. It provides detailed information about the structure of the template. For a step-by-step tutorial that guides you through the process of creating a template, see [Tutorial: Create and deploy your first Azure Resource Manager template](#).

## Template format

In its simplest structure, a template has the following elements:

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "",
 "apiProfile": "",
 "parameters": { },
 "variables": { },
 "functions": [],
 "resources": [],
 "outputs": { }
}
```

| ELEMENT NAME   | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$schema       | Yes      | <p>Location of the JSON schema file that describes the version of the template language.</p> <p>For resource group deployments, use:<br/><a href="https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#">https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#</a></p> <p>For subscription deployments, use:<br/><a href="https://schema.management.azure.com/schemas/2015-01-01/subscriptionDeploymentTemplate.json#">https://schema.management.azure.com/schemas/2015-01-01/subscriptionDeploymentTemplate.json#</a></p> |
| contentVersion | Yes      | <p>Version of the template (such as 1.0.0.0). You can provide any value for this element. Use this value to document significant changes in your template. When deploying resources using the template, this value can be used to make sure that the right template is being used.</p>                                                                                                                                                                                                                                                                                         |

| ELEMENT NAME | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| apiProfile   | No       | <p>An API version that serves as a collection of API versions for resource types. Use this value to avoid having to specify API versions for each resource in the template. When you specify an API profile version and don't specify an API version for the resource type, Resource Manager uses the API version for that resource type that is defined in the profile.</p> <p>The API profile property is especially helpful when deploying a template to different environments, such as Azure Stack and global Azure. Use the API profile version to make sure your template automatically uses versions that are supported in both environments. For a list of the current API profile versions and the resources API versions defined in the profile, see <a href="#">API Profile</a>.</p> <p>For more information, see <a href="#">Track versions using API profiles</a>.</p> |
| parameters   | No       | Values that are provided when deployment is executed to customize resource deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| variables    | No       | Values that are used as JSON fragments in the template to simplify template language expressions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| functions    | No       | User-defined functions that are available within the template.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| resources    | Yes      | Resource types that are deployed or updated in a resource group or subscription.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| outputs      | No       | Values that are returned after deployment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Each element has properties you can set. This article describes the sections of the template in greater detail.

## Parameters

In the parameters section of the template, you specify which values you can input when deploying the resources. You're limited to 256 parameters in a template. You can reduce the number of parameters by using objects that contain multiple properties.

The available properties for a parameter are:

```

"parameters": {
 "<parameter-name>" : {
 "type" : "<type-of-parameter-value>",
 "defaultValue": "<default-value-of-parameter>",
 "allowedValues": ["<array-of-allowed-values>"],
 "minValue": <minimum-value-for-int>,
 "maxValue": <maximum-value-for-int>,
 "minLength": <minimum-length-for-string-or-array>,
 "maxLength": <maximum-length-for-string-or-array-parameters>,
 "metadata": {
 "description": "<description-of-the parameter>"
 }
 }
}

```

| ELEMENT NAME   | REQUIRED | DESCRIPTION                                                                                                                                                                                                              |
|----------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| parameter-name | Yes      | Name of the parameter. Must be a valid JavaScript identifier.                                                                                                                                                            |
| type           | Yes      | Type of the parameter value. The allowed types and values are <b>string</b> , <b>securestring</b> , <b>int</b> , <b>bool</b> , <b>object</b> , <b>secureObject</b> , and <b>array</b> . See <a href="#">Data types</a> . |
| defaultValue   | No       | Default value for the parameter, if no value is provided for the parameter.                                                                                                                                              |
| allowedValues  | No       | Array of allowed values for the parameter to make sure that the right value is provided.                                                                                                                                 |
| minValue       | No       | The minimum value for int type parameters, this value is inclusive.                                                                                                                                                      |
| maxValue       | No       | The maximum value for int type parameters, this value is inclusive.                                                                                                                                                      |
| minLength      | No       | The minimum length for string, secure string, and array type parameters, this value is inclusive.                                                                                                                        |
| maxLength      | No       | The maximum length for string, secure string, and array type parameters, this value is inclusive.                                                                                                                        |
| description    | No       | Description of the parameter that is displayed to users through the portal. For more information, see <a href="#">Comments in templates</a> .                                                                            |

For examples of how to use parameters, see [Parameters in Azure Resource Manager templates](#).

## Data types

For integers passed as inline parameters, the range of values may be limited by the SDK or command-line tool you use for deployment. For example, when using PowerShell to deploy a template, integer types can range from -2147483648 to 2147483647. To avoid this limitation, specify large integer values in a [parameter file](#). Resource types apply their own limits for integer properties.

When specifying boolean and integer values in your template, don't surround the value with quotation marks. Start and end string values with double quotation marks.

Objects start with a left brace and end with a right brace. Arrays start with a left bracket and end with a right bracket.

Secure strings and secure objects can't be read after resource deployment.

For samples of formatting data types, see [Parameter type formats](#).

## Variables

In the variables section, you construct values that can be used throughout your template. You don't need to define variables, but they often simplify your template by reducing complex expressions.

The following example shows the available options for defining a variable:

```
"variables": {
 "<variable-name>": "<variable-value>",
 "<variable-name>": {
 <variable-complex-type-value>
 },
 "<variable-object-name>": {
 "copy": [
 {
 "name": "<name-of-array-property>",
 "count": <number-of-iterations>,
 "input": <object-or-value-to-repeat>
 }
]
 },
 "copy": [
 {
 "name": "<variable-array-name>",
 "count": <number-of-iterations>,
 "input": <object-or-value-to-repeat>
 }
]
}
```

For information about using `copy` to create several values for a variable, see [Variable iteration](#).

For examples of how to use variables, see [Variables in Azure Resource Manager template](#).

## Functions

Within your template, you can create your own functions. These functions are available for use in your template. Typically, you define complicated expressions that you don't want to repeat throughout your template. You create the user-defined functions from expressions and [functions](#) that are supported in templates.

When defining a user function, there are some restrictions:

- The function can't access variables.
- The function can only use parameters that are defined in the function. When you use the [parameters function](#) within a user-defined function, you're restricted to the parameters for that function.
- The function can't call other user-defined functions.
- The function can't use the [reference function](#).
- Parameters for the function can't have default values.

```

"functions": [
 {
 "namespace": "<namespace-for-functions>",
 "members": {
 "<function-name>": {
 "parameters": [
 {
 "name": "<parameter-name>",
 "type": "<type-of-parameter-value>"
 }
],
 "output": {
 "type": "<type-of-output-value>",
 "value": "<function-return-value>"
 }
 }
 }
],
],

```

| ELEMENT NAME    | REQUIRED | DESCRIPTION                                                                                                                                                                                                                  |
|-----------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| namespace       | Yes      | Namespace for the custom functions. Use to avoid naming conflicts with template functions.                                                                                                                                   |
| function-name   | Yes      | Name of the custom function. When calling the function, combine the function name with the namespace. For example, to call a function named uniqueName in the namespace contoso, use <code>"[contoso.uniqueName()]"</code> . |
| parameter-name  | No       | Name of the parameter to be used within the custom function.                                                                                                                                                                 |
| parameter-value | No       | Type of the parameter value. The allowed types and values are <b>string</b> , <b>securestring</b> , <b>int</b> , <b>bool</b> , <b>object</b> , <b>secureObject</b> , and <b>array</b> .                                      |
| output-type     | Yes      | Type of the output value. Output values support the same types as function input parameters.                                                                                                                                 |
| output-value    | Yes      | Template language expression that is evaluated and returned from the function.                                                                                                                                               |

For examples of how to use custom functions, see [User-defined functions in Azure Resource Manager template](#).

## Resources

In the resources section, you define the resources that are deployed or updated.

You define resources with the following structure:

```

"resources": [
 {
 "condition": "<true-to-deploy-this-resource>",
 "type": "<resource-provider-namespace/resource-type-name>",
 "apiVersion": "<api-version-of-resource>",
 "name": "<name-of-the-resource>",
 "comments": "<your-reference-notes>",
 "location": "<location-of-resource>",
 "dependsOn": [
 "<array-of-related-resource-names>"
],
 "tags": {
 "<tag-name1>": "<tag-value1>",
 "<tag-name2>": "<tag-value2>"
 },
 "sku": {
 "name": "<sku-name>",
 "tier": "<sku-tier>",
 "size": "<sku-size>",
 "family": "<sku-family>",
 "capacity": <sku-capacity>
 },
 "kind": "<type-of-resource>",
 "copy": {
 "name": "<name-of-copy-loop>",
 "count": <number-of-iterations>,
 "mode": "<serial-or-parallel>",
 "batchSize": <number-to-deploy-serially>
 },
 "plan": {
 "name": "<plan-name>",
 "promotionCode": "<plan-promotion-code>",
 "publisher": "<plan-publisher>",
 "product": "<plan-product>",
 "version": "<plan-version>"
 },
 "properties": {
 "<settings-for-the-resource>",
 "copy": [
 {
 "name": ,
 "count": ,
 "input": {}
 }
]
 },
 "resources": [
 "<array-of-child-resources>"
]
 }
]

```

| ELEMENT NAME | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                              |
|--------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| condition    | No       | Boolean value that indicates whether the resource will be provisioned during this deployment. When <code>true</code> , the resource is created during deployment. When <code>false</code> , the resource is skipped for this deployment. See <a href="#">condition</a> . |

| ELEMENT NAME | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type         | Yes      | <p>Type of the resource. This value is a combination of the namespace of the resource provider and the resource type (such as <b>Microsoft.Storage/storageAccounts</b>). To determine available values, see <a href="#">template reference</a>. For a child resource, the format of the type depends on whether it's nested within the parent resource or defined outside of the parent resource. See <a href="#">Set name and type for child resources</a>.</p>                                                                                                                                                                                                                                                |
| apiVersion   | Yes      | <p>Version of the REST API to use for creating the resource. To determine available values, see <a href="#">template reference</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| name         | Yes      | <p>Name of the resource. The name must follow URI component restrictions defined in RFC3986. Azure services that expose the resource name to outside parties validate the name to make sure it isn't an attempt to spoof another identity. For a child resource, the format of the name depends on whether it's nested within the parent resource or defined outside of the parent resource. See <a href="#">Set name and type for child resources</a>.</p>                                                                                                                                                                                                                                                     |
| comments     | No       | <p>Your notes for documenting the resources in your template. For more information, see <a href="#">Comments in templates</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| location     | Varies   | <p>Supported geo-locations of the provided resource. You can select any of the available locations, but typically it makes sense to pick one that is close to your users. Usually, it also makes sense to place resources that interact with each other in the same region. Most resource types require a location, but some types (such as a role assignment) don't require a location. See <a href="#">Set resource location</a>.</p>                                                                                                                                                                                                                                                                         |
| dependsOn    | No       | <p>Resources that must be deployed before this resource is deployed. Resource Manager evaluates the dependencies between resources and deploys them in the correct order. When resources aren't dependent on each other, they're deployed in parallel. The value can be a comma-separated list of a resource names or resource unique identifiers. Only list resources that are deployed in this template. Resources that aren't defined in this template must already exist. Avoid adding unnecessary dependencies as they can slow your deployment and create circular dependencies. For guidance on setting dependencies, see <a href="#">Defining dependencies in Azure Resource Manager templates</a>.</p> |

| ELEMENT NAME | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                                                                                                             |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tags         | No       | Tags that are associated with the resource. Apply tags to logically organize resources across your subscription.                                                                                                                                                                                                                                        |
| sku          | No       | Some resources allow values that define the SKU to deploy. For example, you can specify the type of redundancy for a storage account.                                                                                                                                                                                                                   |
| kind         | No       | Some resources allow a value that defines the type of resource you deploy. For example, you can specify the type of Cosmos DB to create.                                                                                                                                                                                                                |
| copy         | No       | If more than one instance is needed, the number of resources to create. The default mode is parallel. Specify serial mode when you don't want all or the resources to deploy at the same time. For more information, see <a href="#">Create several instances of resources in Azure Resource Manager</a> .                                              |
| plan         | No       | Some resources allow values that define the plan to deploy. For example, you can specify the marketplace image for a virtual machine.                                                                                                                                                                                                                   |
| properties   | No       | Resource-specific configuration settings. The values for the properties are the same as the values you provide in the request body for the REST API operation (PUT method) to create the resource. You can also specify a copy array to create several instances of a property. To determine available values, see <a href="#">template reference</a> . |
| resources    | No       | Child resources that depend on the resource being defined. Only provide resource types that are permitted by the schema of the parent resource. Dependency on the parent resource isn't implied. You must explicitly define that dependency. See <a href="#">Set name and type for child resources</a> .                                                |

## Outputs

In the Outputs section, you specify values that are returned from deployment. Typically, you return values from resources that were deployed.

The following example shows the structure of an output definition:

```

"outputs": {
 "<output-name>": {
 "condition": "<boolean-value-whether-to-output-value>",
 "type": "<type-of-output-value>",
 "value": "<output-value-expression>",
 "copy": {
 "count": <number-of-iterations>,
 "input": <values-for-the-variable>
 }
 }
}

```

| ELEMENT NAME | REQUIRED | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| output-name  | Yes      | Name of the output value. Must be a valid JavaScript identifier.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| condition    | No       | Boolean value that indicates whether this output value is returned. When <code>true</code> , the value is included in the output for the deployment. When <code>false</code> , the output value is skipped for this deployment. When not specified, the default value is <code>true</code> .                                                                                                                                                                                                             |
| type         | Yes      | Type of the output value. Output values support the same types as template input parameters. If you specify <b>securestring</b> for the output type, the value isn't displayed in the deployment history and can't be retrieved from another template. To use a secret value in more than one template, store the secret in a Key Vault and reference the secret in the parameter file. For more information, see <a href="#">Use Azure Key Vault to pass secure parameter value during deployment</a> . |
| value        | No       | Template language expression that is evaluated and returned as output value. Specify either <b>value</b> or <b>copy</b> .                                                                                                                                                                                                                                                                                                                                                                                |
| copy         | No       | Used to return more than one value for an output. Specify <b>value</b> or <b>copy</b> . For more information, see <a href="#">Output iteration in Azure Resource Manager templates</a> .                                                                                                                                                                                                                                                                                                                 |

For examples of how to use outputs, see [Outputs in Azure Resource Manager template](#).

## Comments and metadata

You have a few options for adding comments and metadata to your template.

### Comments

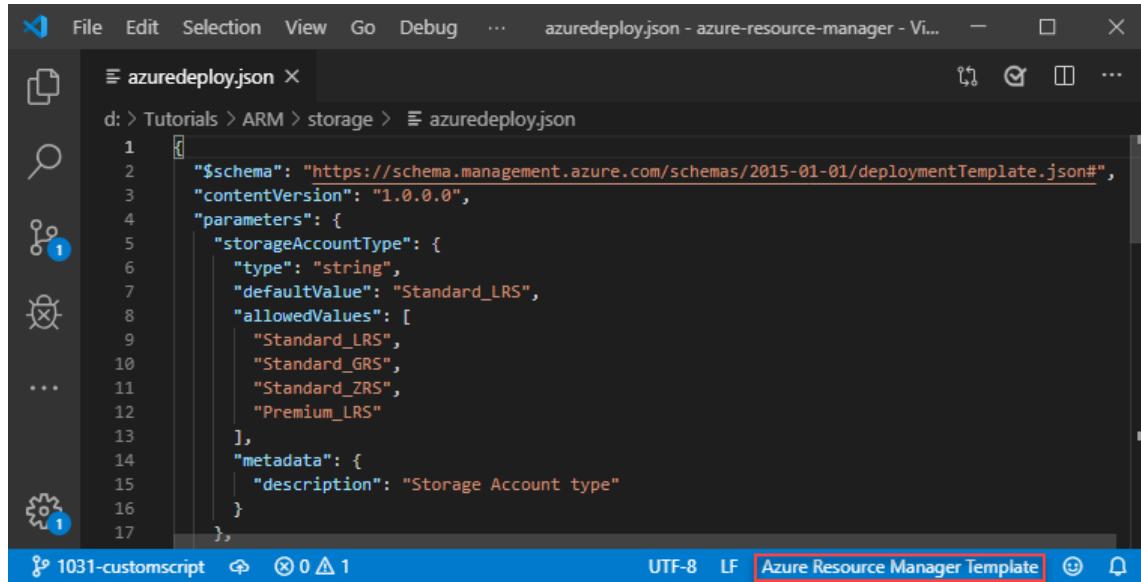
For inline comments, you can use either `//` or `/* ... */` but this syntax doesn't work with all tools. You can't use the portal template editor to work on templates with inline comments. If you add this style of comment, be sure the tools you use support inline JSON comments.

#### NOTE

To deploy templates with comments by using Azure CLI, you must use the `--handle-extended-json-format` switch.

```
{
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2018-10-01",
 "name": "[variables('vmName')]", // to customize name, change it in variables
 "location": "[parameters('location')]", //defaults to resource group location
 "dependsOn": [/* storage account and network interface must be deployed first */
 "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
 "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
],
}
```

In Visual Studio Code, the [Azure Resource Manager Tools extension](#) can automatically detect Resource Manager template and change the language mode accordingly. If you see **Azure Resource Manager Template** at the bottom-right corner of VS Code, you can use the inline comments. The inline comments are no longer marked as invalid.



## Metadata

You can add a `metadata` object almost anywhere in your template. Resource Manager ignores the object, but your JSON editor may warn you that the property isn't valid. In the object, define the properties you need.

```
{
 "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "metadata": {
 "comments": "This template was developed for demonstration purposes.",
 "author": "Example Name"
 },
}
```

For **parameters**, add a `metadata` object with a `description` property.

```
"parameters": {
 "adminUsername": {
 "type": "string",
 "metadata": {
 "description": "User name for the Virtual Machine."
 }
},
```

When deploying the template through the portal, the text you provide in the description is automatically used as a tip for that parameter.

| SETTINGS                          | User name for the Virtual Machine. |
|-----------------------------------|------------------------------------|
| * Admin Username <small>i</small> | <input type="text"/>               |
| * Admin Password <small>i</small> | <input type="password"/>           |

For **resources**, add a `comments` element or a metadata object. The following example shows both a comments element and a metadata object.

```
"resources": [
 {
 "type": "Microsoft.Storage/storageAccounts",
 "apiVersion": "2018-07-01",
 "name": "[concat('storage', uniqueString(resourceGroup().id))]",
 "comments": "Storage account used to store VM disks",
 "location": "[parameters('location')]",
 "metadata": {
 "comments": "These tags are needed for policy compliance."
 },
 "tags": {
 "Dept": "[parameters('deptName')]",
 "Environment": "[parameters('environment')]"
 },
 "sku": {
 "name": "Standard_LRS"
 },
 "kind": "Storage",
 "properties": {}
 }
]
```

For **outputs**, add a metadata object to the output value.

```
"outputs": {
 "hostname": {
 "type": "string",
 "value": "[reference(variables('publicIPAddressName')).dnsSettings.fqdn]",
 "metadata": {
 "comments": "Return the fully qualified domain name"
 }
},
```

You can't add a metadata object to user-defined functions.

## Multi-line strings

You can break a string into multiple lines. For example, see the location property and one of the comments in the following JSON example.

```
{
 "type": "Microsoft.Compute/virtualMachines",
 "apiVersion": "2018-10-01",
 "name": "[variables('vmName')]", // to customize name, change it in variables
 "location": "[
 parameters('location')
]", //defaults to resource group location
 /*
 storage account and network interface
 must be deployed first
 */
 "dependsOn": [
 "[resourceId('Microsoft.Storage/storageAccounts/', variables('storageAccountName'))]",
 "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
],
```

To deploy templates with multi-line strings by using Azure CLI, you must use the `--handle-extended-json-format` switch.

## Next steps

- To view complete templates for many different types of solutions, see the [Azure Quickstart Templates](#).
- For details about the functions you can use from within a template, see [Azure Resource Manager Template Functions](#).
- To combine several templates during deployment, see [Using linked templates with Azure Resource Manager](#).
- For recommendations about creating templates, see [Azure Resource Manager template best practices](#).
- For recommendations on creating Resource Manager templates that you can use across all Azure environments and Azure Stack, see [Develop Azure Resource Manager templates for cloud consistency](#).

# Frequently asked question about Windows Virtual Machines

2/12/2020 • 4 minutes to read • [Edit Online](#)

This article addresses some common questions about Windows virtual machines created in Azure using the Resource Manager deployment model. For the Linux version of this topic, see [Frequently asked question about Linux Virtual Machines](#).

## What can I run on an Azure VM?

All subscribers can run server software on an Azure virtual machine. For information about the support policy for running Microsoft server software in Azure, see [Microsoft server software support for Azure Virtual Machines](#).

Certain versions of Windows 7, Windows 8.1, and Windows 10 are available to MSDN Azure benefit subscribers and MSDN Dev and Test Pay-As-You-Go subscribers, for development and test tasks. For details, including instructions and limitations, see [Windows Client images for MSDN subscribers](#).

## How much storage can I use with a virtual machine?

Each data disk can be up to 32,767 GiB. The number of data disks you can use depends on the size of the virtual machine. For details, see [Sizes for Virtual Machines](#).

Azure Managed Disks are the recommended disk storage offerings for use with Azure Virtual Machines for persistent storage of data. You can use multiple Managed Disks with each Virtual Machine. Managed Disks offer two types of durable storage options: Premium and Standard Managed Disks. For pricing information, see [Managed Disks Pricing](#).

Azure storage accounts can also provide storage for the operating system disk and any data disks. Each disk is a .vhdx file stored as a page blob. For pricing details, see [Storage Pricing Details](#).

## How can I access my virtual machine?

Establish a remote connection using Remote Desktop Connection (RDP) for a Windows VM. For instructions, see [How to connect and sign on to an Azure virtual machine running Windows](#). A maximum of two concurrent connections are supported, unless the server is configured as a Remote Desktop Services session host.

If you're having problems with Remote Desktop, see [Troubleshoot Remote Desktop connections to a Windows-based Azure Virtual Machine](#).

If you're familiar with Hyper-V, you might be looking for a tool similar to VMConnect. Azure doesn't offer a similar tool because console access to a virtual machine isn't supported.

## Can I use the temporary disk (the D: drive by default) to store data?

Don't use the temporary disk to store data. It is only temporary storage, so you would risk losing data that can't be recovered. Data loss can occur when the virtual machine moves to a different host. Resizing a virtual machine, updating the host, or a hardware failure on the host are some of the reasons a virtual machine might move.

If you have an application that needs to use the D: drive letter, you can reassign drive letters so that the temporary disk uses something other than D:. For instructions, see [Change the drive letter of the Windows temporary disk](#).

## How can I change the drive letter of the temporary disk?

You can change the drive letter by moving the page file and reassigning drive letters, but you need to make sure you do the steps in a specific order. For instructions, see [Change the drive letter of the Windows temporary disk](#).

## Can I add an existing VM to an availability set?

No. If you want your VM to be part of an availability set, you need to create the VM within the set. There currently isn't a way to add a VM to an availability set after it has been created.

## Can I upload a virtual machine to Azure?

Yes. For instructions, see [Migrating on-premises VMs to Azure](#).

## Can I resize the OS disk?

Yes. For instructions, see [How to expand the OS drive of a Virtual Machine in an Azure Resource Group](#).

## Can I copy or clone an existing Azure VM?

Yes. Using managed images, you can create an image of a virtual machine and then use the image to build multiple new VMs. For instructions, see [Create a custom image of a VM](#).

## Why am I not seeing Canada Central and Canada East regions through Azure Resource Manager?

The two new regions of Canada Central and Canada East are not automatically registered for virtual machine creation for existing Azure subscriptions. This registration is done automatically when a virtual machine is deployed through the Azure portal to any other region using Azure Resource Manager. After a virtual machine is deployed to any other Azure region, the new regions should be available for subsequent virtual machines.

## Does Azure support Linux VMs?

Yes. To quickly create a Linux VM to try out, see [Create a Linux VM on Azure using the Portal](#).

## Can I add a NIC to my VM after it's created?

Yes, this is now possible. The VM first needs to be stopped deallocated. Then you can add or remove a NIC (unless it's the last NIC on the VM).

## Are there any computer name requirements?

Yes. The computer name can be a maximum of 15 characters in length. See [Naming conventions rules and restrictions](#) for more information around naming your resources.

## Are there any resource group name requirements?

Yes. The resource group name can be a maximum of 90 characters in length. See [Naming conventions rules and restrictions](#) for more information about resource groups.

## What are the username requirements when creating a VM?

Usernames can be a maximum of 20 characters in length and cannot end in a period (".").

The following usernames are not allowed:

|               |         |                  |        |
|---------------|---------|------------------|--------|
| administrator | admin   | user             | user1  |
| test          | user2   | test1            | user3  |
| admin1        | 1       | 123              | a      |
| actuser       | adm     | admin2           | aspnet |
| backup        | console | david            | guest  |
| john          | owner   | root             | server |
| sql           | support | support_388945a0 | sys    |
| test2         | test3   | user4            | user5  |

## What are the password requirements when creating a VM?

There are varying password length requirements, depending on the tool you are using:

- Portal - between 12 - 72 characters
- PowerShell - between 8 - 123 characters
- CLI - between 12 - 123
- Have lower characters
- Have upper characters
- Have a digit
- Have a special character (Regex match [\W\_])

The following passwords are not allowed:

|            |            |            |           |             |
|------------|------------|------------|-----------|-------------|
| abc@123    | iloveyou!  | P@\$\$w0rd | P@ssw0rd  | P@ssword123 |
| Pa\$\$word | pass@word1 | Password!  | Password1 | Password22  |