**Azure VM (Virtual Machine)**

**Table Of content: -**

**Introduction to Azure Virtual Machines**

1. **Introduction to Azure Virtual Machines (VMs)**
   * **Azure VMs as a compute resource**
   * **Core concepts**
2. **Benefits of Azure VMs**
   * **Scalability**
   * **Flexibility**
   * **Cost-Efficiency**
   * **Global Reach**

**Getting Started with Azure VMs**

1. **Creating an Azure Account**
2. **Logging into Azure Portal**
3. **Creating Azure Virtual Machines**
   * **Using Azure CLI**
   * **Using Azure Portal**
   * **Using PowerShell**

**Managing Azure Virtual Machines**

1. **Starting and Stopping Azure VMs**
2. **Scaling Azure VMs**
   * **Vertical Scaling (Scaling Up/Down)**
   * **Horizontal Scaling (Scaling Out/In)**
3. **Resource Group Management**
4. **Resource Cleanup**

**Azure VM Networking**

1. **Virtual Networks (VNETs)**
   * **Creating a Virtual Network**
2. **Network Security Groups (NSGs)**
   * **Creating an NSG**
   * **Associating NSGs with VMs**
3. **Azure VM Storage**
   * **Azure Disks**
     + **Creating and Managing Azure Disks**
   * **Azure Blob Storage**
     + **Creating Azure Blob Storage Containers**
   * **Accessing Azure Blob Storage from VMs**

**Backup and Disaster Recovery for Azure VMs**

1. **Azure Backup**
   * **Setting Up Azure Backup**
2. **Azure Site Recovery**
   * **Setting Up Azure Site Recovery**
   * **Testing Failover**

**Monitoring and Optimization for Azure VMs**

1. **Azure Monitor**
   * **Monitoring Azure VMs with Azure Monitor**
2. **Cost Management**
   * **Reserved Instances (RIs)**
   * **Azure Cost Management and Billing**
3. **Scaling and Automation**
   * **Automation Runbooks**
   * **Azure Logic Apps**
4. **Regular Maintenance and Patching**
   * **Operating System Updates**
   * **Security Best Practices**
5. **Performance Optimization**
   * **Vertical Scaling**
   * **Horizontal Scaling**

**Documentation and Knowledge Sharing**

1. **Documenting VM Configurations**
2. **Procedures and Best Practices**
3. **Troubleshooting**
4. **Knowledge Sharing**

* **Introduction to Azure Virtual Machines:**

**Introduction to Azure Virtual Machines (VMs)**

Azure Virtual Machines (VMs) are a fundamental and versatile compute resource offered by Microsoft Azure, the cloud computing platform provided by Microsoft. These virtual machines are a core building block for deploying a wide range of applications in the cloud. Azure VMs enable organizations to run virtualized instances of various operating systems, including Windows and Linux, in a highly scalable and customizable environment.

**Key Concepts**

1. **Virtual Machine (VM):**

A virtualized server that runs an operating system and applications in an isolated environment within Microsoft Azure. Azure VMs offer scalability and flexibility, enabling users to adjust computing resources to match their specific workloads. With a choice of operating systems and robust management tools, Azure VMs are a versatile solution for hosting a wide range of applications and services in the cloud.

1. **Azure Portal**:

Azure Portal is the centralized, user-friendly web interface for managing and monitoring your Azure resources, including Azure Virtual Machines (VMs). It provides a visually intuitive dashboard where you can create, configure, and control VM instances, networks, and associated services. With its role-based access control and rich visualization tools, Azure Portal simplifies the management of VMs and streamlines the process of optimizing your cloud infrastructure for performance and cost-efficiency.

1. **Region:**

A geographical area containing one or more Azure data centers, regions are strategically distributed worldwide to provide redundancy and resilience. When deploying Virtual Machines (VMs) in Azure, selecting an appropriate region is crucial for optimizing performance, reducing latency, and ensuring data compliance with local regulations. By strategically choosing the right region for your VM deployment, you can enhance reliability and responsiveness for your applications while adhering to regulatory requirements.

1. **Availability Zone**:

An Availability Zone (AZ) in Azure is a distinct, physically isolated data center within the same Azure region. These zones are designed to provide high availability and fault tolerance for your virtual machines (VMs) and other resources. By distributing your VMs across different Availability Zones, you can ensure resilience against hardware failures, power outages, and other unforeseen events, thereby enhancing the reliability and availability of your applications in the Azure cloud. Azure's service-level agreements (SLAs) often rely on the use of Availability Zones to meet uptime and availability targets.

1. **Resource Group**:

In Azure, a Resource Group serves as a logical container that helps you efficiently organize and manage Azure resources such as virtual machines, databases, and storage accounts. It acts as a fundamental unit for resource management, allowing you to easily group, deploy, monitor, and secure related resources together. By associating resources with a Resource Group, you gain simplified control over access permissions, cost tracking, and resource lifecycle management within your Azure environment.

* **Benefits of Azure VMs**

**Azure VMs offer several advantages:**

* **Scalability:**

Scalability in Azure Virtual Machines (VMs) allows you to dynamically adjust computing resources, such as CPU and RAM, to accommodate fluctuations in workload demands. This flexibility ensures optimal performance during peak usage periods and cost savings during quieter times, making Azure VMs an ideal choice for businesses with evolving computing needs.

* **Flexibility:**

Flexibility in Azure VMs extends to the ability to select from an extensive library of pre-configured OS images, encompassing various Windows and Linux distributions. Additionally, Azure provides a diverse array of VM sizes, allowing users to tailor their virtual machines to meet specific performance and resource requirements, ensuring optimal performance for their workloads.

* **Cost-Efficiency**:

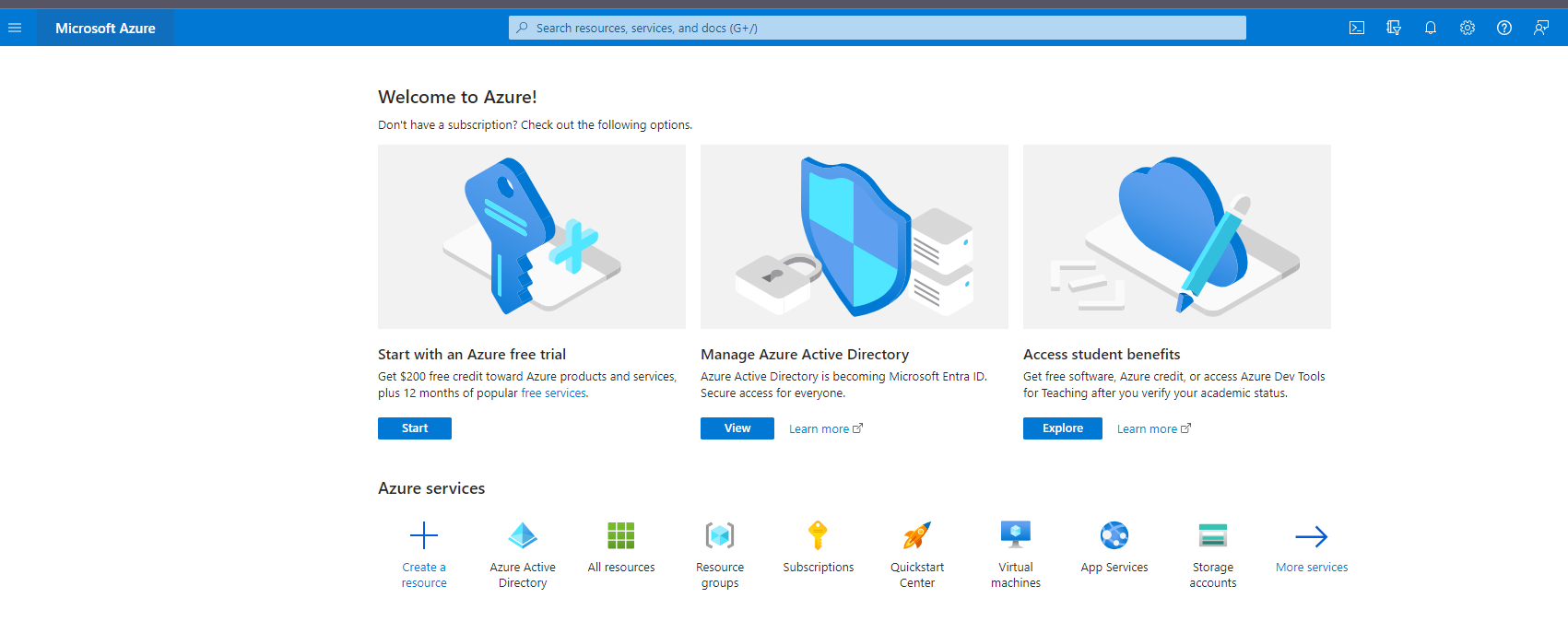
Azure VMs adhere to a pay-as-you-go model, ensuring that you are billed solely for the computing resources consumed. This cost-effective approach allows organizations to optimize their budgets by scaling resources up or down as needed, without incurring unnecessary expenses.

* **Global Reach**:

Azure Virtual Machines provide a global infrastructure that allows you to deploy VM instances in data centers across the world, ensuring low-latency access and improved performance for your applications, websites, or services, thus enabling you to serve customers worldwide seamlessly. This global reach also offers geographic redundancy, enhancing the availability and reliability of your applications in case of regional outages or high traffic demands.

*To begin your journey with Azure VMs:*

1. **Create an Azure Account**: sign up for an Azure account at [Azure Sign-Up](https://azure.com/free).
2. **Login to Azure Portal:** Visit [Azure Portal](https://portal.azure.com/) and log in with your Azure account credentials.  
     
   Below is the Dashboard after successfully log in with credentials:



**Creating an Azure Virtual Machine:**

* We can create Virtual Machine on windows using three environments:  
  1. CLI

2. Portal

3. PowerShell

1. ***Create a Windows virtual machine with the Azure CLI:***

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quick Start shows you how to use the Azure CLI 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.

create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) 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](https://learn.microsoft.com/en-us/cli/azure/group) 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 West US 3 location. Replace the value of the variables as needed.

**Azure CLI**

resourcegroup="myResourceGroupCLI"

location="westus3"

az group create --name $resourcegroup --location $location

## **Create virtual machine**

Create a VM with [az vm create](https://learn.microsoft.com/en-us/cli/azure/vm). The following example creates a VM named myVM. This example uses azureuser for an administrative user name. Replace the values of the variables as needed.

You'll be prompted to supply a password that meets the [password requirements for Azure VMs](https://learn.microsoft.com/en-us/azure/virtual-machines/windows/faq#what-are-the-password-requirements-when-creating-a-vm-).

Using the example below, you'll be prompted to enter a password at the command line. You could also add the --admin-password parameter with a value for your password. The user name and password will be used when you connect to the VM.

**Azure CLI**

vmname="myVM"

username="azureuser"

az vm create \

--resource-group $resourcegroup \

--name $vmname \

--image Win2022AzureEditionCore \

--public-ip-sku Standard \

--admin-username $username

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

**OutputCopy**

{

"fqdns": "",

"id": "/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",

"location": "westus3",

"macAddress": "00-0D-3A-23-9A-49",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "52.174.34.95",

"resourceGroup": "myResourceGroupCLI"

"zones": ""

}

Take a note your own publicIpAddress in the output when you create your VM. This IP address is used to access the VM later in this article.

## **Install web server**

To see your VM in action, install the IIS web server.

**Azure CLI**

az vm run-command invoke -g $resourcegroup \

-n $vmname \

--command-id RunPowerShellScript \

--scripts "Install-WindowsFeature -name Web-Server -IncludeManagementTools"

## **Open port 80 for web traffic**

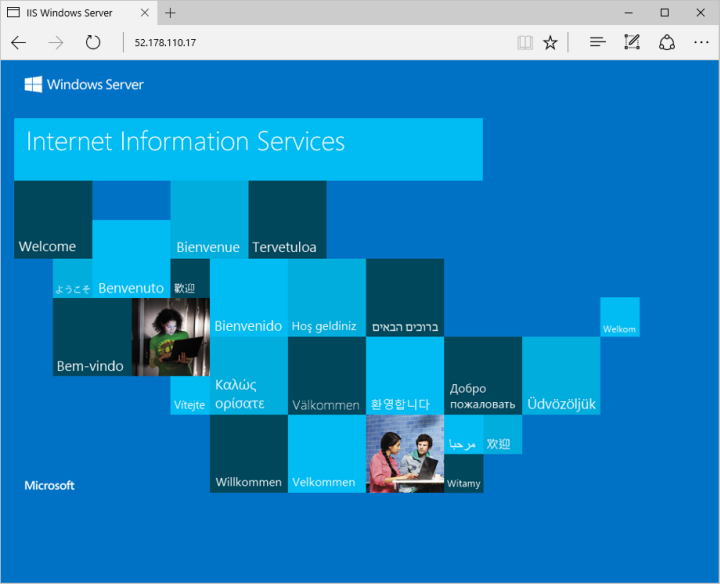
By default, only RDP connections are opened when you create a Windows VM in Azure. Use [az vm open-port](https://learn.microsoft.com/en-us/cli/azure/vm) to open TCP port 80 for use with the IIS web server:

**Azure CLI**

az vm open-port --port 80 --resource-group $resourcegroup --name $vmname

## **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](https://learn.microsoft.com/en-us/cli/azure/group) command to remove the resource group, VM, and all related resources:

**Azure CLI**

az group delete --name $resourcegroup

# *Create a Windows virtual machine in the Azure portal:*

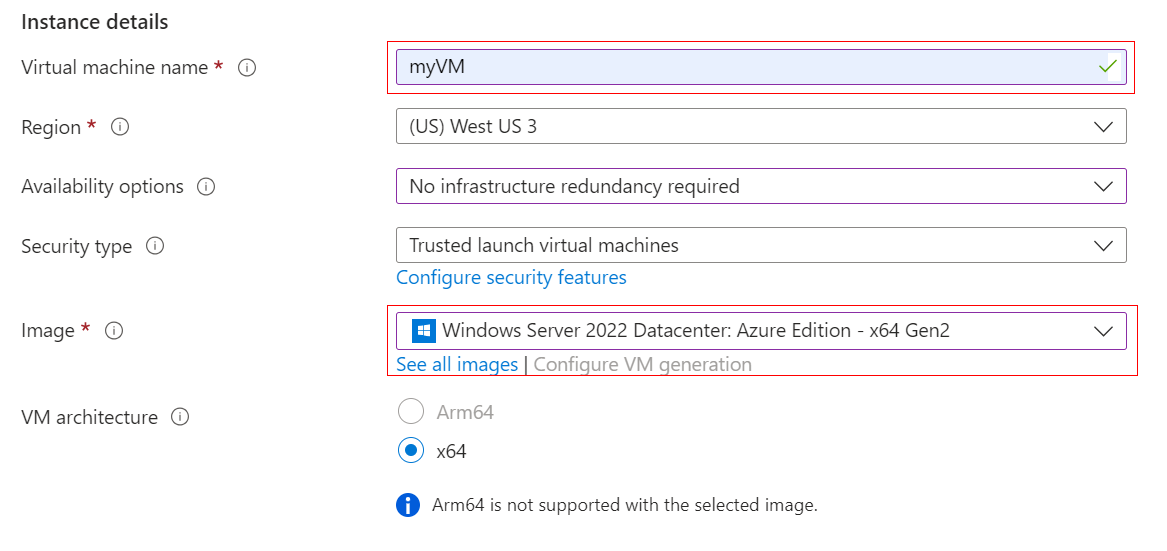
# Azure virtual machines (VMs) can be created through the Azure portal. This method provides a browser-based user interface to create VMs and their associated resources. This quick start shows you how to use the Azure portal to deploy a virtual machine (VM) in Azure that runs Windows Server 2019. To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

## **Sign in to Azure**

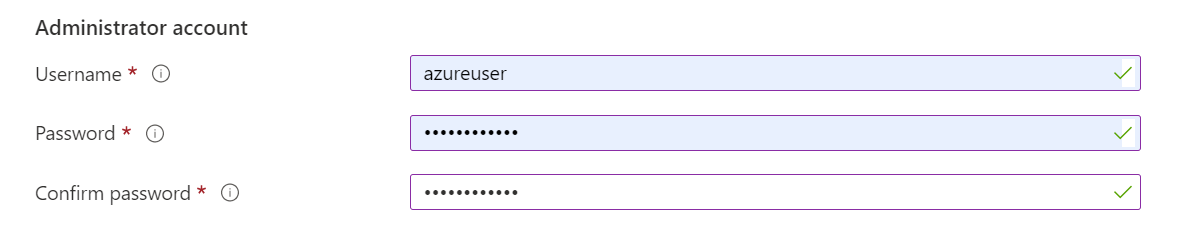
Sign in to the [Azure portal](https://portal.azure.com/).

## **Create virtual machine**

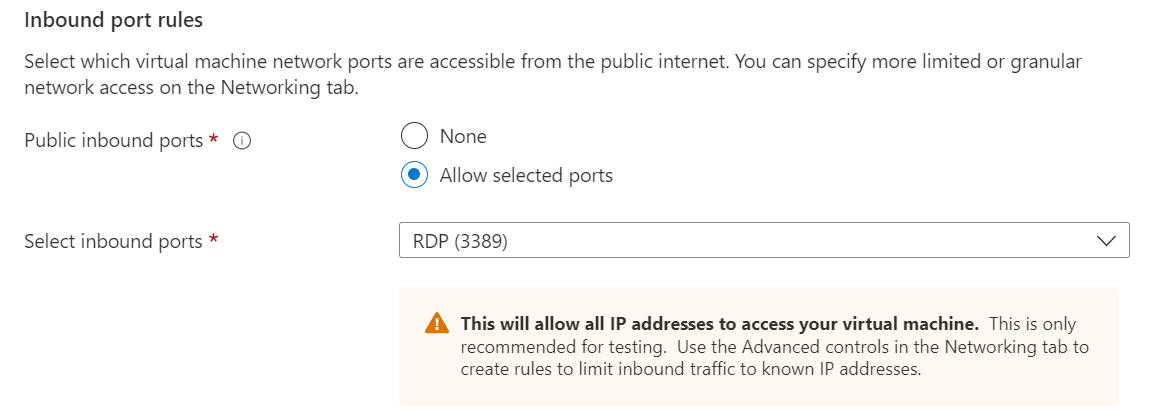
1. Enter virtual machines in the search.
2. Under **Services**, select **Virtual machines**.
3. In the **Virtual machines** page, select **Create** and then **Azure virtual machine**. The **Create a virtual machine** page opens.
4. Under **Instance details**, enter myVM for the **Virtual machine name** and choose Windows Server 2022 Datacenter: Azure Edition - x64 Gen 2 for the **Image**. Leave the other defaults.



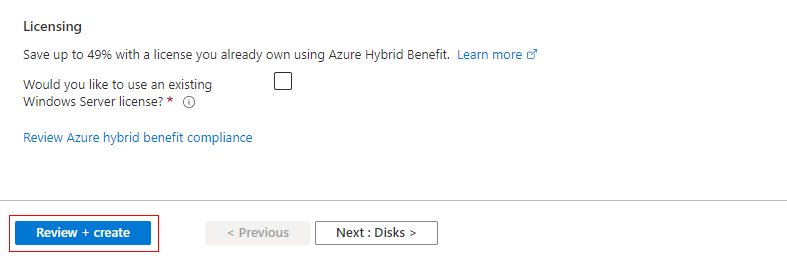
1. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the [defined complexity requirements](https://learn.microsoft.com/en-us/azure/virtual-machines/windows/faq#what-are-the-password-requirements-when-creating-a-vm-).



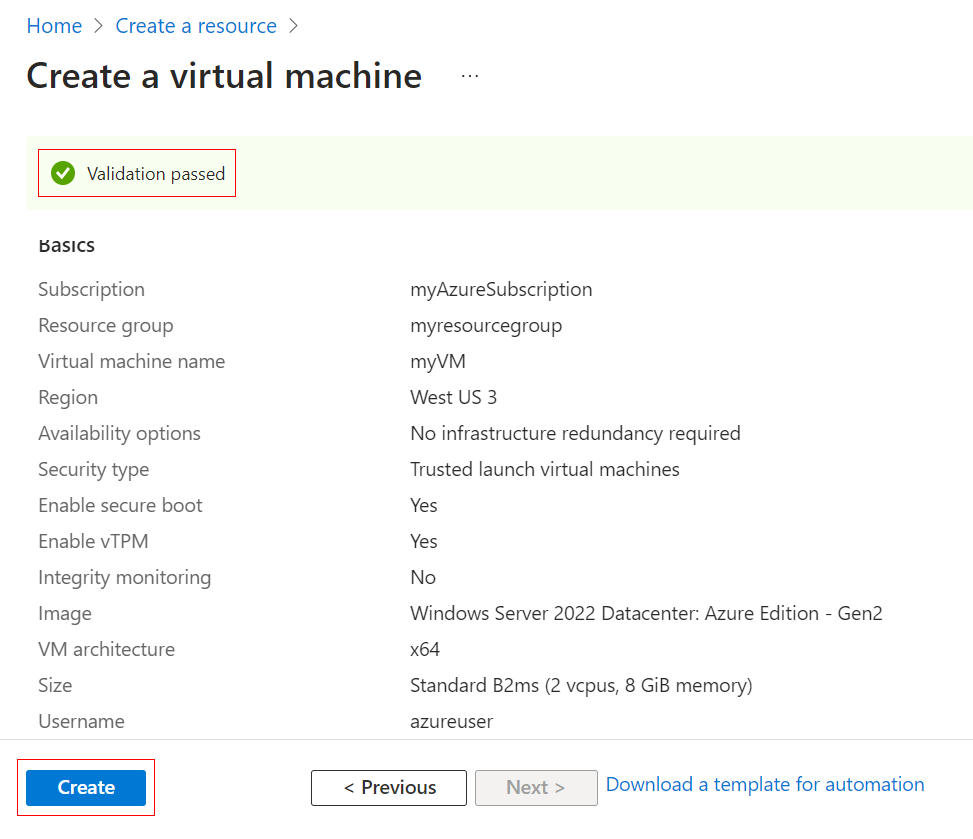
1. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP (80)** from the drop-down.



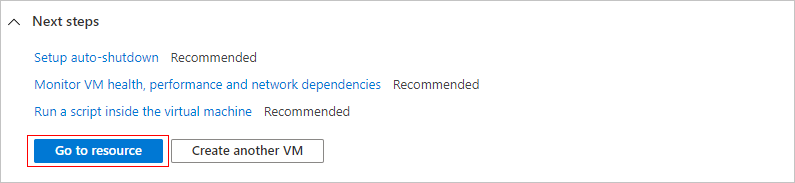
1. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.



1. After validation runs, select the **Create** button at the bottom of the page.



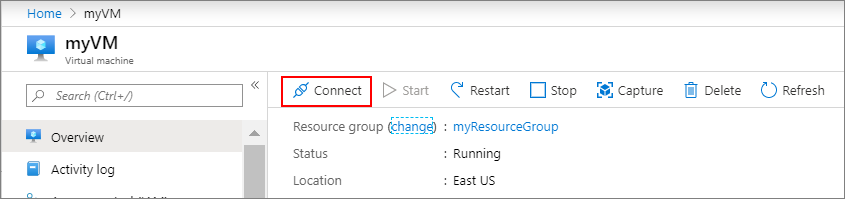
1. After deployment is complete, select **Go to resource**.



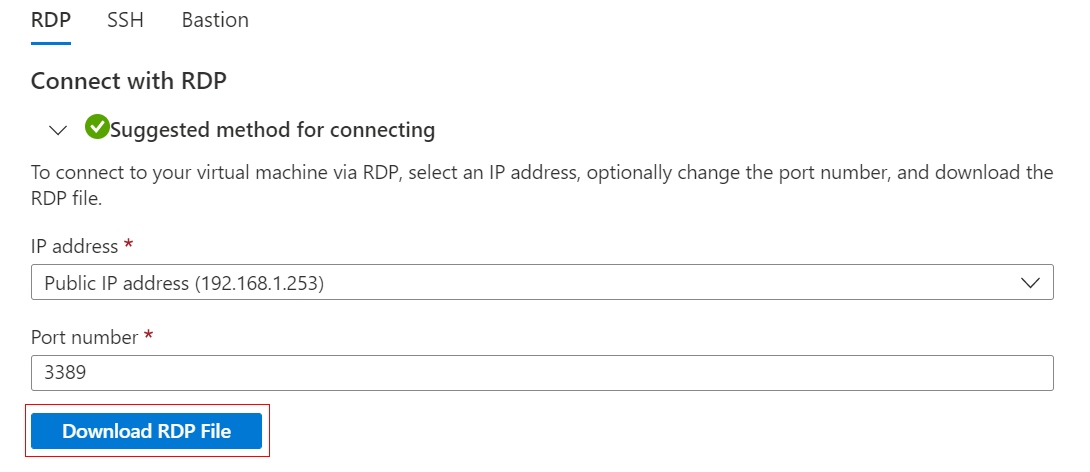
## **Connect to virtual machine**

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](https://apps.apple.com/app/microsoft-remote-desktop/id1295203466?mt=12) from the Mac App Store.

1. On the overview page for your virtual machine, select the **Connect** > **RDP**.



1. In the **Connect with RDP** tab, keep the default options to connect by IP address, over port 3389, and click **Download RDP file**.



1. Open the downloaded RDP file and click **Connect** when prompted.
2. In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username as **localhost**\username, enter the password you created for the virtual machine, and then click **OK**.
3. 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:

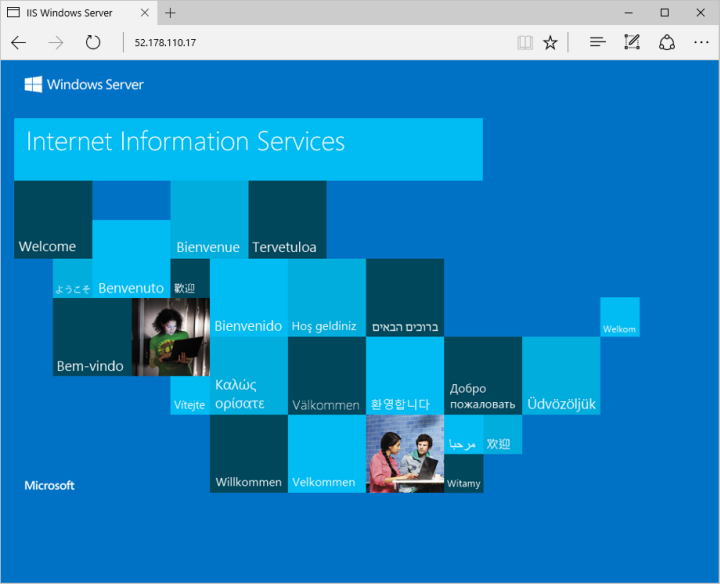
**PowerShell**

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, hover over the IP address to show **Copy to clipboard**. Copy the IP address 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.

1. On the Overview page for the VM, select the **Resource group** link.
2. At the top of the page for the resource group, select **Delete resource group**.
3. A page will open warning you that you are about to delete resources. Type the name of the resource group and select **Delete** to finish deleting the resources and the resource group.

# *Create a Windows virtual machine in Azure with PowerShell:*

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

## **Launch Azure Cloud Shell**

The Azure Cloud Shell is a free interactive shell 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 **Open Cloudshell** 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 code blocks, paste them into the Cloud Shell, and press enter to run the them.

## **Create resource group**

Create an Azure resource group with [New-AzResourceGroup](https://learn.microsoft.com/en-us/powershell/module/az.resources/new-azresourcegroup). A resource group is a logical container into which Azure resources are deployed and managed.

**Azure PowerShell**

New-AzResourceGroup -Name 'myResourceGroup' -Location 'EastUS'

## **Create virtual machine**

Create a VM with [New-AzVM](https://learn.microsoft.com/en-us/powershell/module/az.compute/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:

**Azure PowerShell**

New-AzVm `

-ResourceGroupName 'myResourceGroup' `

-Name 'myVM' `

-Location 'East US' `

-Image 'MicrosoftWindowsServer:WindowsServer:2022-datacenter-azure-edition:latest' `

-VirtualNetworkName 'myVnet' `

-SubnetName 'mySubnet' `

-SecurityGroupName 'myNetworkSecurityGroup' `

-PublicIpAddressName 'myPublicIpAddress' `

-OpenPorts 80,3389

## **Install web server**

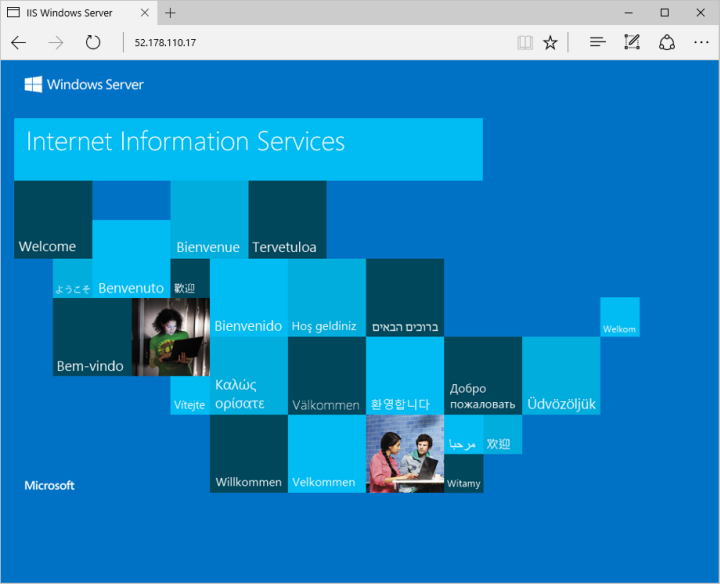
To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

**Azure PowerShell**

Invoke-AzVMRunCommand -ResourceGroupName 'myResourceGroup' -VMName 'myVM' -CommandId 'RunPowerShellScript' -ScriptString 'Install-WindowsFeature -Name Web-Server -IncludeManagementTools'

## **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 the VM that you created. The following example shows the default IIS web site:



## **Clean up resources**

When no longer needed, you can use the [Remove-AzResourceGroup](https://learn.microsoft.com/en-us/powershell/module/az.resources/remove-azresourcegroup) cmdlet to remove the resource group, VM, and all related resources:

**Azure PowerShell**

Remove-AzResourceGroup -Name 'myResourceGroup'

**Managing Azure Virtual Machines:**

**Starting and Stopping Azure VMs**

One of the key benefits of Azure VMs is their flexibility in starting and stopping based on your needs, which can help optimize costs.

**Starting a VM**:

* To start a VM, you can use the Azure Portal or PowerShell:
  + In the Azure Portal, navigate to your VM, and click the "Start" button.
  + In PowerShell, use the **Start-AzVM** cmdlet:

**PowerShell:**

Start-AzVM -ResourceGroupName 'myResourceGroup' -Name 'myVM'

**Stopping a VM**:

* To stop a VM, you can also use the Azure Portal or PowerShell:
  + In the Azure Portal, navigate to your VM, and click the "Stop" button.
  + In PowerShell, use the **Stop-AzVM** cmdlet:

**PowerShell:**

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

**Scaling Azure VMs:**

Azure provides options for scaling VMs based on your resource requirements:

* **Vertical Scaling (Scaling Up/Down)**:
  + Increase or decrease the VM size to adjust CPU, memory, and other resources.
  + Use the Azure Portal or PowerShell to resize a VM.

**PowerShell:**

Resize-AzVM -ResourceGroupName 'myResourceGroup' -VMName 'myVM' -Size 'Standard\_DS2\_v2'

* **Horizontal Scaling (Scaling Out/In):**
  + Add or remove VM instances to distribute the workload.
  + Consider using Azure Virtual Machine Scale Sets for automated scaling based on demand.

**Resource Group Management**

Resource groups are containers for organizing and managing Azure resources, including VMs. You can:

* Create a new resource group:

**PowerShell:**

New-AzResourceGroup -Name 'newResourceGroup' -Location 'EastUS'

* List resource groups:

Get-AzResourceGroup

* Delete a resource group (and all contained resources):

**PowerShell:**

Remove-AzResourceGroup -Name 'oldResourceGroup' -Force

**Resource Cleanup:**

Remember to remove or deallocate VMs and associated resources when they are no longer needed to avoid unnecessary costs. Use the **Remove-AzResourceGroup** command as mentioned earlier to delete the entire resource group and its contents.

**Azure VM Networking**

**Virtual Networks (VNETs)**

Azure Virtual Networks (VNETs) are the foundation of your Azure network infrastructure. They allow you to isolate, segment, and control network traffic for your Azure resources, including VMs.

**Creating a Virtual Network**:

1. **Azure Portal**:
   * Navigate to the Azure Portal.
   * Click on "Create a resource" > "Networking" > "Virtual network."
   * Follow the steps to create a VNET, specifying its name, address space, and subnet configurations.
2. **PowerShell**:
   * Use the **New-AzVirtualNetwork** cmdlet to create a VNET:

New-AzVirtualNetwork -ResourceGroupName 'myResourceGroup' -Name 'myVnet' -AddressPrefix '10.0.0.0/16'

**Network Security Groups (NSGs)**

Network Security Groups (NSGs) are Azure firewall rules that control inbound and outbound traffic to network interfaces (including VMs). You can define rules to permit or deny traffic based on source/destination IP, port, and protocol.

**Creating an NSG**:

1. **Azure Portal**:
   * Navigate to the Azure Portal.
   * Click on "Create a resource" > "Security + Identity" > "Network security group."
   * Define the NSG's name, resource group, and region.
   * Create inbound and outbound security rules as needed.
2. **PowerShell**:
   * Use the **New-AzNetworkSecurityGroup** cmdlet to create an NSG:

$nsg = New-AzNetworkSecurityGroup -ResourceGroupName 'myResourceGroup' -Name 'myNSG' -Location 'EastUS'

Add security rules to the NSG:

$rule1 = New-AzNetworkSecurityRuleConfig -Name 'AllowHTTP' -Protocol Tcp -Direction Inbound -Priority 100 -SourceAddressPrefix '\*' -SourcePortRange '\*' -DestinationAddressPrefix '\*' -DestinationPortRange 80 -Access Allow

$rule2 = New-AzNetworkSecurityRuleConfig -Name 'DenyAll' -Protocol '\*' -Direction Inbound -Priority 200 -SourceAddressPrefix '\*' -SourcePortRange '\*' -DestinationAddressPrefix '\*' -DestinationPortRange '\*' -Access Deny

Add-AzNetworkSecurityRuleConfig -NetworkSecurityGroup $nsg -SecurityRule $rule1

Add-AzNetworkSecurityRuleConfig -NetworkSecurityGroup $nsg -SecurityRule $rule2

Set-AzNetworkSecurityGroup -NetworkSecurityGroup $nsg

**Associating NSGs with VMs**:

You can associate NSGs with VMs to control their network traffic. In the Azure Portal, go to your VM's settings, navigate to the "Networking" tab, and select the NSG you created. In PowerShell, you can use the **Set-AzNetworkInterface** cmdlet.

**Azure VM Storage**

**Azure Disks**

Azure Disks are block storage devices used to store the operating system, applications, and data of your Azure Virtual Machines (VMs). There are different types of Azure Disks, each suited for specific use cases:

1. **Standard HDD**: Standard Hard Disk Drives provide cost-effective storage for workloads with low I/O requirements.
2. **Standard SSD**: Standard Solid-State Drives offer better performance than HDDs and are suitable for a wide range of workloads.
3. **Premium SSD**: Premium Solid-State Drives provide high-performance storage, ideal for I/O-intensive workloads.
4. **Ultra SSD**: Ultra Solid-State Drives offer the highest performance and low latency for mission-critical applications.

**Creating and Managing Azure Disks**

To create and manage Azure Disks, you can use the Azure Portal or Azure PowerShell:

1. **Azure Portal**:
   * Navigate to your VM in the Azure Portal.
   * In the VM's settings, go to "Disks" to add or manage disks.
2. **Azure PowerShell**:
   * Use the **New-AzDisk** cmdlet to create a new disk:

New-AzDisk -ResourceGroupName 'myResourceGroup' -DiskName 'myDisk' -DiskSizeGB 128 -Location 'EastUS' -DiskType 'Premium\_LRS'

Attach a disk to a VM:

Add-AzVMDataDisk -VMName 'myVM' -ResourceGroupName 'myResourceGroup' -DiskName 'myDisk' -Caching ReadWrite -CreateOption Attach -Lun 1

**Azure Blob Storage**

Azure Blob Storage is a service for storing unstructured data, such as documents, images, videos, and backups. You can use Azure Blob Storage with your Azure VMs for data storage and backups.

**Creating Azure Blob Storage Containers**

1. **Azure Portal**:
   * Navigate to your Azure Storage Account in the Azure Portal.
   * In the Storage Account settings, go to "Containers" to create and manage containers.
2. **Azure PowerShell**:
   * Use the **New-AzStorageContainer** **New-AzStorageContainer -Name 'mycontainer' -Context $ctx** cmdlet to create a new container:

New-AzStorageContainer -Name 'mycontainer' -Context $ctx

**Accessing Azure Blob Storage from VMs**

To access Azure Blob Storage from your VMs, you can use various methods, including Azure SDKs, Azure CLI, or even PowerShell scripts to upload and download data to and from your storage containers.

**Backup and Disaster Recovery for Azure VMs**

**Azure Backup**

Azure Backup is a service that allows you to protect your Azure VMs by creating and managing backups. It's essential to have a reliable backup strategy in place to recover from data loss or system failures.

**Setting Up Azure Backup**

1. **Azure Portal**:
   * Navigate to your VM in the Azure Portal.
   * In the VM's settings, go to "Backup" to configure backup settings.
   * Create a new backup policy specifying the retention period, frequency, and backup time.
2. **Azure PowerShell**:
   * Use the **Set-AzRecoveryServicesBackupProtectionPolicy** cmdlet to configure backup policies:

Set-AzRecoveryServicesBackupProtectionPolicy -Policy $policy -WorkloadType 'AzureVM'

**Azure Site Recovery**

Azure Site Recovery is a service that provides disaster recovery solutions for your Azure VMs. It helps ensure business continuity in case of site failures or disasters.

**Setting Up Azure Site Recovery**

1. **Azure Portal**:
   * Navigate to your VM in the Azure Portal.
   * In the VM's settings, go to "Disaster recovery."
   * Follow the steps to set up a Site Recovery vault, replication, and failover.
2. **Azure PowerShell**:
   * Use the **New-AzRecoveryServicesAsrVaultSettings** and **New-AzRecoveryServicesAsrProtectableItem** cmdlets to configure Site Recovery:

$vault = New-AzRecoveryServicesAsrVaultSettings -Name 'myRecoveryVault' -ResourceGroupName 'myResourceGroup' -Location 'EastUS'

$item = New-AzRecoveryServicesAsrProtectableItem -ProtectableItem (Get-AzRecoveryServicesAsrProtectableItem -ResourceGroupName 'myResourceGroup' -ResourceName 'myVM') -ProtectionContainer $container

**Testing Failover**

It's crucial to periodically test your disaster recovery solution. Azure Site Recovery allows you to perform test failovers to ensure that your VMs can be successfully recovered in case of a disaster.

**Monitoring and Optimization for Azure VMs**

**Azure Monitor**

Azure Monitor provides a comprehensive solution for collecting, analyzing, and acting on telemetry data from your Azure VMs. It helps you ensure the health, performance, and availability of your VMs and applications.

**Monitoring Azure VMs with Azure Monitor**

1. **Azure Portal**:
   * Navigate to your VM in the Azure Portal.
   * In the VM's settings, go to "Monitoring" to access monitoring data, performance metrics, and logs.
2. **Azure Metrics**:
   * Azure Metrics Explorer allows you to visualize and analyze performance metrics such as CPU usage, memory, and network traffic.
3. **Azure Log Analytics**:
   * Set up Azure Log Analytics to collect and analyze logs from your VMs and applications.
   * Create custom queries to gain insights into your VMs' behavior.

**Cost Management**

Optimizing costs is essential when running VMs in Azure. Here are some strategies:

1. **Reserved Instances (RIs)**:
   * Consider purchasing Azure Reserved Instances to save costs on VM usage.
2. **Azure Cost Management and Billing**:
   * Use Azure Cost Management and Billing to track and manage your VM spending.
   * Set up budgets and alerts to get notified when spending exceeds defined thresholds.

**Scaling and Automation**

Efficiently manage resources by automating tasks and scaling VMs as needed:

1. **Automation Runbooks**:
   * Create Azure Automation Runbooks to automate common tasks like starting and stopping VMs on a schedule.
2. **Azure Logic Apps**:
   * Use Azure Logic Apps to trigger actions based on events, such as scaling up VMs when resource usage increases.

**Regular Maintenance and Patching**

Keep your VMs up to date and secure:

1. **Operating System Updates**:
   * Regularly apply OS updates and security patches to your VMs.
2. **Security Best Practices**:
   * Follow Azure's security best practices to protect your VMs from threats.

**Performance Optimization**

Optimize VM performance for your workloads:

1. **Vertical Scaling**:
   * If your VMs experience increased resource demands, consider resizing to a higher VM size.
2. **Horizontal Scaling**:
   * Implement load balancing and auto-scaling using Azure Virtual Machine Scale Sets for high availability and performance.

**Documentation and Knowledge Sharing**

Maintain documentation for your VM configurations, procedures, and best practices. This documentation helps in troubleshooting and knowledge sharing among your team members.