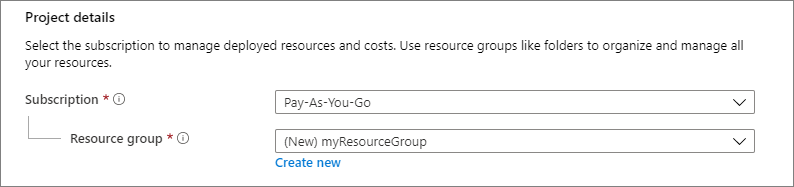
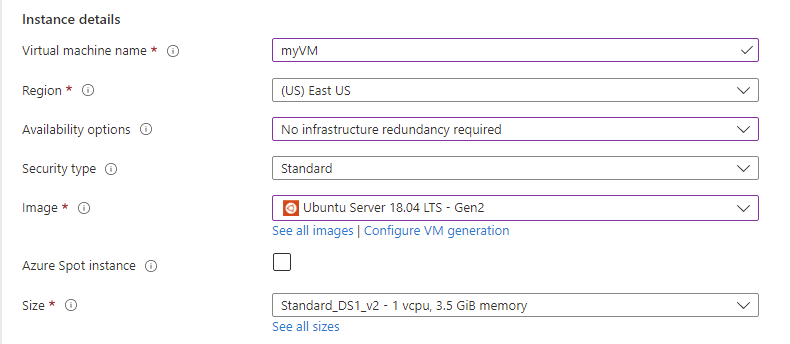
# **Azure virtual machine**

Resource group: A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization.

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 Virtual machine. The Create a virtual machine page opens.
4. In the Basics tab, under Project details, make sure the correct subscription is selected and then choose to Create new resource group. Enter myResourceGroup for the name.

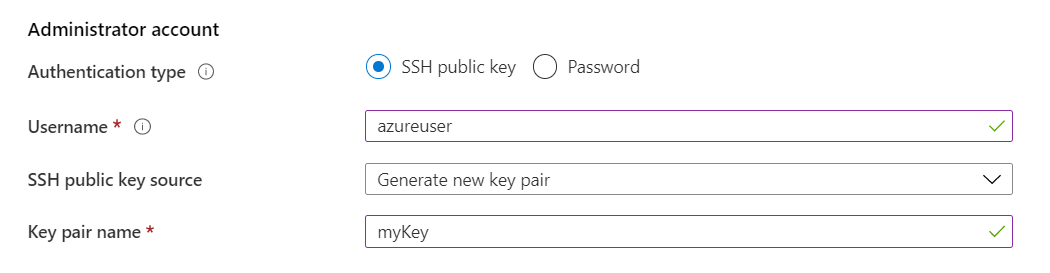


1. Under Instance details, enter myVM for the Virtual machine name, and choose Ubuntu 18.04 LTS - Gen2 for your Image. Leave the other defaults. The default size and pricing is only shown as an example. Size availability and pricing are dependent on your region and subscriptio

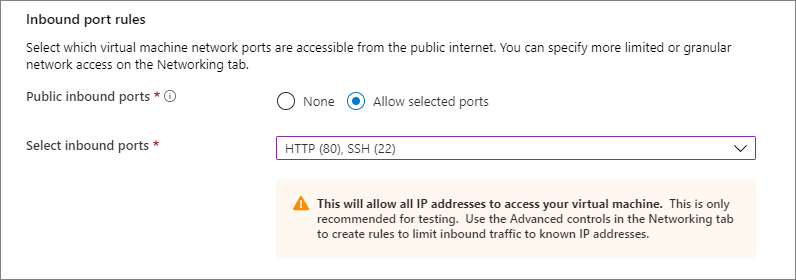
Under Administrator account, select SSH public key.

In Username enter azureuser.

For SSH public key source, leave the default of Generate new key pair, and then enter myKey for the Key pair name.



Under Inbound port rules > Public inbound ports, choose Allow selected ports and then select SSH (22) 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.
2. 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.
3. When the Generate new key pair window opens, select Download private key and create resource. Your key file will be download as myKey.pem. Make sure you know where the .pem file was downloaded; you will need the path to it in the next step.

**Connect to virtual machine**

1. Using putty we can connect to vm (for linux machine)

While creating vm in authentication type option give username and password and these credentials are to be entered in login time

In putty just give public ip of the vm and open

1. Using RDP we can connect to vm (for windows machine)

While creating vm in authentication type give username and password and these credentials are to be entered in login time using RDP

When we give connect option as RDP it will ask for to download and when we run that RDP application it will ask for username and password.

**Azure Storage**

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud. Azure Storage data objects are accessible from anywhere in the world over HTTP or HTTPS via a REST API.

## Azure Storage data services

The Azure Storage platform includes the following data services:

* [Azure Blobs](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction): It all has begun with Blob Storage in Microsoft Azure. BLOB is an acronym and means **B**inary **L**arge **OB**ject. Or, in plain English, the unstructured files, such as images, video, music files, backup files, etc.
* [Azure Files](https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction): Microsoft Azure File storage is the second storage type that was designed to support the needs of the Azure VM environment. That storage is, in essence, a network share. You can store files there that can be accessed from different Virtual Machines. It is similar to [Amazon EFS](https://www.msp360.com/resources/blog/amazon-s3-vs-ebs-vs-efs/) its direct competitor.
* [Azure Queues](https://docs.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction): When your services need to communicate with each other.

Queue Storage is designed to connect the components of your application. It allows you to build flexible applications with independent components that rely on asynchronous message queuing.

Let’s say you have an on-premises application interacting with a server somewhere in the cloud. Sometimes the server is down, which means that you can no longer send messages to it. If you try, that will normally result in an error. Here are some other issues concerning asynchronous communication that you have to deal with in such case:

The necessity to have both the receiver and the sender available simultaneously. Needless to say, if one of them goes down, the communication terminates.

Mandatory implementation of try/retry logic to provide for a possible outage.

Lack of proper scalability.

However, all of that can be avoided simply by using a mediator that will collect the messages while one of the communication partners is down. With Azure Queues, you have a third player that connects the two components and acts as both a buffer and a mediator.

Azure Queue storage is essentially a service for storing large numbers of messages that can be accessed from anywhere via authenticated HTTP or HTTPS requests. A single queue sizes up to 64 kb.

**Azure supports two types of queue mechanisms:**

**Storage queues.** Being part of the Azure storage infrastructure, they feature a simple REST-based GET/PUT/PEEK interface with reliable and persistent messaging within and between services.

**Service Bus queues** are part of a broader Azure messaging infrastructure that supports queuing as well as more advanced integration patterns.

[Azure Tables](https://docs.microsoft.com/en-us/azure/storage/tables/table-storage-overview): Cheaper, more scalable storage for your structured data and big data analysis

Microsoft Azure Table Storage was designed to store structured NoSQL data. The storage is hugely scalable and, at the same time, cheap to store data. However, it becomes more expensive if you access files frequently.

This storage is useful if you find Microsoft Azure SQL too expensive and can go without SQL structure and architecture.

[Azure Disks](https://docs.microsoft.com/en-us/azure/virtual-machines/managed-disks-overview): Microsoft Azure Disk Storage is based on Page Blobs. It is a service that allows you to create disks for your virtual machines. A disk created in Disk Storage can be accessed from only one virtual machine. In other words - it is your local drive. Yes, it’s that simple.

Here you can have two options for the speed of your disks:

* HDDs that are cheap but slow and called **standard** storage.
* SSDs that are fast but expensive and called **premium** storage.

And two options for disk management:

* **Unmanaged disk** - you should manage the disk storage and corresponding account yourself
* **Managed disk** - Azure does everything for you. You need to select only the size of the disk and the desired type - standard or premium

**Blobs types**

Azure Storage supports three types of blobs:

* **Block blobs:**Good for file storage, capable of 4.77 TB per file

When you store a file as a block blob, it arrives to the storage in small parts (or, you guessed it, blocks), and only after you complete the upload - the file/blob is merged in one piece. With this architecture, the file cannot be modified without a complete reload. This is the easiest and the cheapest way to store files in Azure.

* **Append blobs:** Good for storing logs or meta-data, can be updated constantly

As we’ve already stated, you cannot modify a Block Blob without re-uploading it. However, there are times when you need to perform many input/output operations, like in the databases. Append Blobs were created for this very purpose - they are structured in such a way that the user can upload parts of the files from the end.

* **Page blobs:** Designed for storing disks

Page Blobs are the basis for the Microsoft Azure virtual machine environment. They have been specifically designed with disk limitations in mind - each Page Blob should be multiple of 512 bytes. The architecture of Page Blobs allows data to be written to each part of a blob.

In fact, if you are running any sort of disk on a VM in Microsoft Azure it uses Page Blobs. However, that is not the end with the blobs. They also have the so-called access tiers.

#### **Blobs Access Tiers:** Besides three Blob types, there are also Blob access tiers in Microsoft Azure. You may have already heard about hot, cool, and cold storage tiers:

* **The hot tier** is for the frequently needed data. It’s expensive to store but cheap to access.
* **The cool tier** is for the less frequently needed data. It’s less expensive to store files than in hot tier, but more expensive to access,
* **The cold tier** is for your archives. It’s dirt cheap to store files there but highly expensive to access files.

## Redundancy in the primary region

Data in an Azure Storage account is always replicated three times in the primary region. Azure Storage offers two options for how your data is replicated in the primary region:

* **Locally redundant storage (LRS)** copies your data synchronously three times within a single physical location in the primary region. LRS is the least expensive replication option but isn't recommended for applications requiring high availability or durability.
* **Zone-redundant storage (ZRS)** copies your data synchronously across three Azure availability zones in the primary region. For applications requiring high availability, Microsoft recommends using ZRS in the primary region and also replicating to a secondary region.

## Redundancy in a secondary region

For applications requiring high durability, you can choose to additionally copy the data in your storage account to a secondary region that is hundreds of miles away from the primary region. If your storage account is copied to a secondary region, then your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable.

* **Geo-redundant storage (GRS)** copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.
* **Geo-zone-redundant storage (GZRS)** copies your data synchronously across three Azure availability zones in the primary region using ZRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.

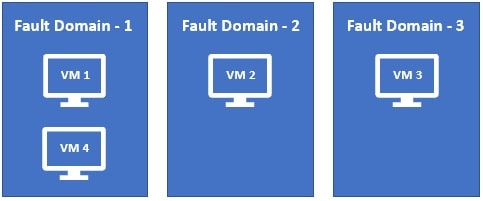
## Availability sets

To provide redundancy to your application, we recommend that you group two or more virtual machines in an availability set. This configuration ensures that during a planned or unplanned maintenance event, at least one virtual machine will be available and meet the 99.95% Azure SLA (Service-level agreements). The availability set of a virtual machine can't be changed after it is created.

**Fault domain:** Virtual machines in the same fault domain share a common power source and physical network switch.

The number of fault domains you want in the availability set. For example, if you set the fault domains to 3 and you create 3 virtual machines, each of them will be placed in 3 separate fault domains. If there is a fault like a power failure for example, only one of the server racks is affected. This means only one of you VM is down, but the other 2 vms from the other 2 fault domains are still available. This in turn means, your workload i.e., in this case your web application is still available to end users.

What happens if we create a fourth VM with 3 fault domains. Well, it will be placed in one of the 3 fault domains. This means, in one of the 3 fault domains, you will have 2 VMs and the rest 2 will have 1 each.



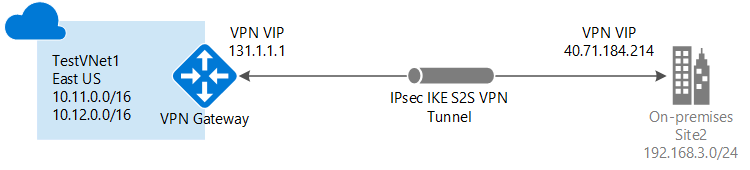
**Update domain:** Virtual machines in the same update domain will be restarted together during planned maintenance. Azure never restarts more than one update domain at a time.

The number of update domains you want. Let's say, you have 3 VMs deployed across 3 update domains. If an update is installed and a restart is required, only one update domain is restarted at any given time. This means you have the other 2 VMs available from the rest of the 2 update domains.

**VPN Types**

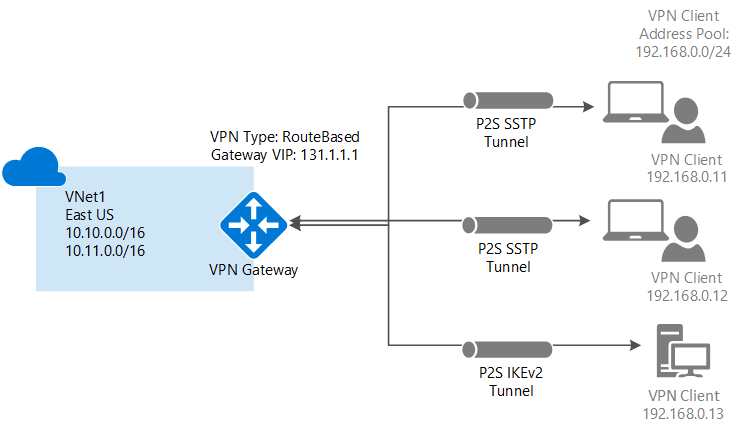
# **Site-to-Site connection**

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has an externally facing public IP address assigned to it.



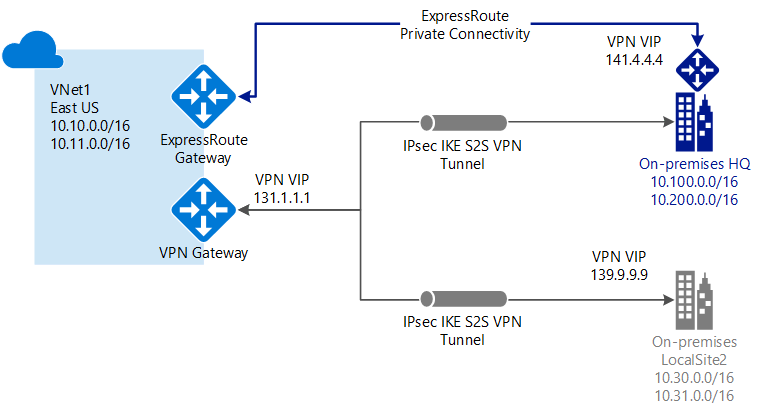
# **Point-to-Site connection**

A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer. A P2S connection is established by starting it from the client computer. This solution is useful for telecommuters who want to connect to Azure VNets from a remote location, such as from home or a conference. P2S VPN is also a useful solution to use instead of S2S VPN when you have only a few clients that need to connect to a VNet.



# **ExpressRoute connection**

ExpressRoute is an Azure service that lets you create private connections between Microsoft datacenters and infrastructure that's on your premises or in a colocation facility. ExpressRoute connections don't go over the public Internet, and offer higher security, reliability, and speeds with lower latencies than typical connections over the Internet.



**peering**

create 2 virtual networks in same resource group

1)in basic window give resource group name and instance details and region

2)in Ip addresses window it will take default ipv4 address and subnet also

3)and then review and create

4)create 2 virtual machines in the same region for two virtual networks (giving virtual network names in networking tab)

5)go for any of the virtual network and click on peering

6)give peering link name in "this virtual network"

7)in remote virtual network give the same name given in 6th step

8)in virtual network option select which network to connect

9)all other things should be same

10)open the virtual machine from which u want to connect and select connect (from RDP) login

11)open PowerShell and enter the command to open the port

To allow myVM2 to ping myVM1 in a later step, enter this command: New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4

12)in this virtual machine open RDC and connect to other vm using public Ip

**Load Balancing**

Load balancing refers to evenly distributing load (incoming network traffic) across a group of backend resources or servers.

Azure Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model. It's the single point of contact for clients. Load balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. These flows are according to configured load-balancing rules and health probes. The backend pool instances can be Azure Virtual Machines or instances in a virtual machine scale set.

A [**public load balancer**](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. Public Load Balancers are used to load balance internet traffic to your VMs.

An [**internal (or private) load balancer**](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

**Steps to configure load balancer (public)**

Create virtual network

1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Setting** | **Value** | | **Project Details** |  | | Subscription | Select your Azure subscription | | Resource Group | Select **Create new**. In **Name** enter **CreatePubLBQS-rg**. Select **OK**. | | **Instance details** |  | | Name | Enter **myVNet** | | Region | Select -Southeast Asia | |  |
|  |  |

1. Select the **IP Addresses** tab or select **Next: IP Addresses** at the bottom of the page.
2. ISelect **West Europe**n the **IP Addresses** tab, enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| IPv4 address space | Enter **10.1.0.0/16** |

1. Under **Subnet name**, select the word **default**. If a subnet isn't present, select **+ Add subnet**.
2. In **Edit subnet**, enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Subnet name | Enter **myBackendSubnet** |
| Subnet address range | Enter **10.1.0.0/24** |

1. Select **Save** or **Add**.
2. Select the **Security** tab.
3. Under **BastionHost**, select **Enable**. Enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Bastion name | Enter **myBastionHost** |
| AzureBastionSubnet address space | Enter **10.1.1.0/27** |
| Public IP Address | Select **Create new**. For **Name**, enter **myBastionIP**. Select **OK**. |

1. Select the **Review + create** tab or select the **Review + create** button.
2. Select **Create**.

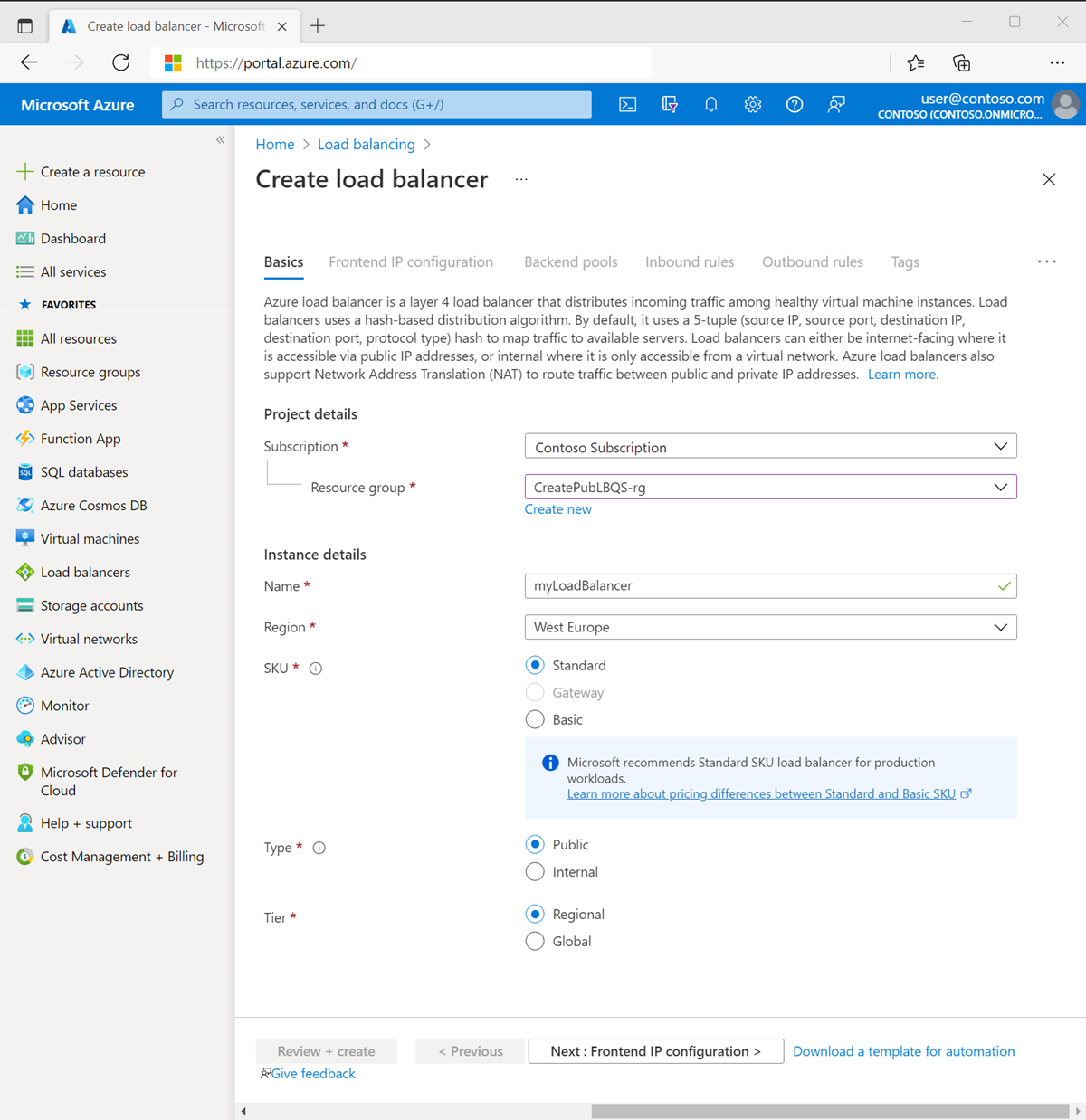
Create load balancer

During the creation of the load balancer, you'll configure:

* Frontend IP address
* Backend pool
* Inbound load-balancing rules
* Health probe

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project details** |  |
| Subscription | Select your subscription |
| Resource group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| Name | Enter **myLoadBalancer** |
| Region | Select **West Europe** |
| SKU | Leave the default **Standard** |
| Type | Select **Public** |
| Type | Leave the default **Regional** |



1. Select **Next: Frontend IP configuration** at the bottom of the page.
2. In **Frontend IP configuration**, select **+ Add a frontend IP configuration**.
3. Enter **myFrontend** in **Name**.
4. Select **IPv4** or **IPv6** for the **IP version**.
5. Select **IP address** for the **IP type**.
6. Select **Create new** in **Public IP address**.
7. In **Add a public IP address**, enter **myPublicIP** for **Name**.
8. Select **Zone-redundant** in **Availability zone**.
9. Leave the default of **Microsoft Network** for **Routing preference**.
10. Select **OK**.
11. Select **Add**.
12. Select **Next: Backend pools** at the bottom of the page.
13. In the **Backend pools** tab, select **+ Add a backend pool**.
14. Enter **myBackendPool** for **Name** in **Add backend pool**.
15. Select **myVNet** in **Virtual network**.
16. Select **NIC** or **IP Address** for **Backend Pool Configuration**.
17. Select **IPv4** or **IPv6** for **IP version**.
18. Select **Add**.
19. Select **Next: Inbound rules** at the bottom of the page.
20. In **Load balancing rule** in the **Inbound rules** tab, select **+ Add a load balancing rule**.
21. In **Add load balancing rule**, enter or select the following information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Name | Enter **myHTTPRule** |
| IP Version | Select **IPv4** or **IPv6** depending on your requirements |
| Frontend IP address | Select **myFrontend** |
| Backend pool | Select **myBackendPool** |
| Protocol | Select **TCP** |
| Port | Enter **80** |
| Backend port | Enter **80** |
| Health probe | Select **Create new**. In **Name**, enter **myHealthProbe**. Select **TCP** in **Protocol**. Leave the rest of the defaults, and select **OK** |
| Session persistence | Select **None** |
| Idle timeout (minutes) | Enter or select **15** |
| TCP reset | Select **Enabled** |
| Floating IP | Select **Disabled** |
| Outbound source network address translation (SNAT) | Leave the default of **(Recommended) Use outbound rules to provide backend pool members access to the internet** |

1. Select **Add**.
2. Select the blue **Review + create** button at the bottom of the page.
3. Select **Create**.

Create NAT gateway

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project details** |  |
| Subscription | Select your subscription |
| Resource group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| NAT gateway name | Enter **myNATgateway** |
| Region | Select **West Europe** |
| Availability zone | Select **None** |
| Idle timeout (minutes) | Enter **15** |

1. Select the **Outbound IP** tab or select **Next: Outbound IP** at the bottom of the page.
2. In **Outbound IP**, select **Create a new public IP address** next to **Public IP addresses**.
3. Enter **myNATgatewayIP** in **Name**.
4. Select **OK**.
5. Select the **Subnet** tab or select the **Next: Subnet** button at the bottom of the page.
6. In **Virtual network** in the **Subnet** tab, select **myVNet**.
7. Select **myBackendSubnet** under **Subnet name**.
8. Select the blue **Review + create** button at the bottom of the page, or select the **Review + create** tab.
9. Select **Create**.

Create virtual machines

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project Details** |  |
| Subscription | Select your Azure subscription |
| Resource Group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| Virtual machine name | Enter **myVM1** |
| Region | Select **(Europe) West Europe** |
| Availability Options | Select **Availability zones** |
| Availability zone | Select **Zone 1** |
| Security type | Select **Standard** |
| Image | Select **Windows Server 2022 Datacenter: Azure Edition - Gen2** |
| Azure Spot instance | Leave the default of unchecked |
| Size | Choose VM size or take default setting |
| **Administrator account** |  |
| Username | Enter a username |
| Password | Enter a password |
| Confirm password | Reenter password |
| **Inbound port rules** |  |
| Public inbound ports | Select **None** |

1. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.
2. In the Networking tab, select or enter the following information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Network interface** |  |
| Virtual network | Select **myVNet** |
| Subnet | Select **myBackendSubnet** |
| Public IP | Select **None** |
| NIC network security group | Select **Advanced** |
| Configure network security group | Select **Create new**. In the **Create network security group**, enter **myNSG** in **Name**. Under **Inbound rules**, select **+Add an inbound rule**. Under **Service**, select **HTTP**. Under **Priority**, enter **100**. In **Name**, enter **myNSGRule** Select **Add** Select **OK** |
| Delete NIC when VM is deleted | Leave the default of **unselected** |
| Accelerated networking | Leave the default of **selected** |
| **Load balancing** |  |
| Place this virtual machine behind an existing load-balancing solution? | Select the check box |
| **Load balancing settings** |  |
| Load-balancing options | Select a backend pool |
| Select a load balancer | Select **myLoadBalancer** |
| Select a backend pool | Select **myBackendPool** |

1. Select **Review + create**.
2. Review the settings, and then select **Create**.
3. Follow the steps 1 through 7 to create another VM with the following values and all the other settings the same as **myVM1**:

|  |  |
| --- | --- |
| **Setting** | **VM 2** |
| Name | **myVM2** |
| Availability zone | **Zone 2** |
| Network security group | Select the existing **myNSG** |

Install IIS

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **myVM1**.
3. On the **Overview** page, select **Connect**, then **Bastion**.
4. Enter the username and password entered during VM creation.
5. Select **Connect**.
6. On the server desktop, navigate to **Windows Administrative Tools** > **Windows PowerShell**.
7. In the PowerShell Window, run the following commands to:
   * Install the IIS server
   * Remove the default iisstart.htm file
   * Add a new iisstart.htm file that displays the name of the VM:

# Install IIS server role

Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file

Remove-Item C:\inetpub\wwwroot\iisstart.htm

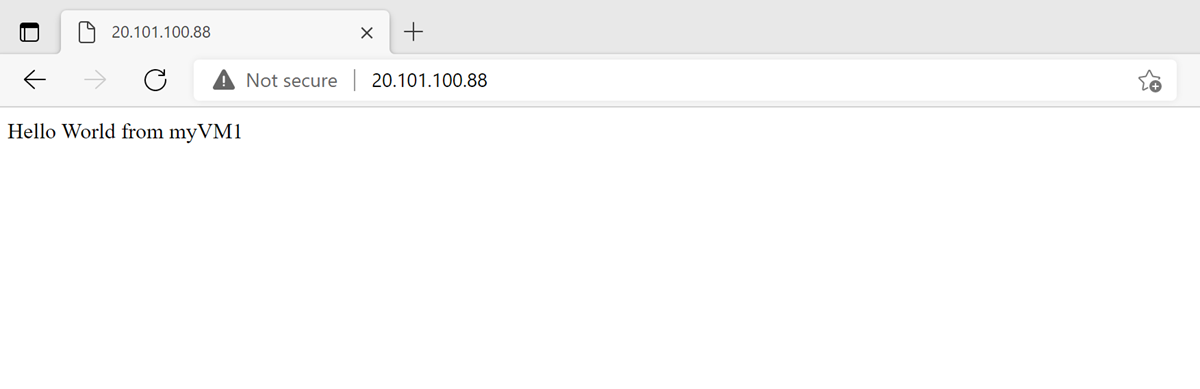
# Add a new htm file that displays server name

Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)

1. Close the Bastion session with **myVM1**.
2. Repeat steps 1 to 8 to install IIS and the updated iisstart.htm file on **myVM2**.

Test the load balancer

1. In the search box at the top of the page, enter **Public IP**. Select **Public IP addresses** in the search results.
2. In **Public IP addresses**, select **myPublicIP**.
3. Copy the item in **IP address**. Paste the public IP into the address bar of your browser. The custom VM page of the IIS Web server is displayed in the browser.
4. **Steps to configure load balancer (public)**



**Steps to configure load balancer (private)**

Create virtual network

1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Setting** | **Value** | | **Project Details** |  | | Subscription | Select your Azure subscription | | Resource Group | Select **Create new**. In **Name** enter **CreatePubLBQS-rg**. Select **OK**. | | **Instance details** |  | | Name | Enter **myVNet** | | Region | Select -Southeast Asia | |  |

1. Select the **IP Addresses** tab or select **Next: IP Addresses** at the bottom of the page.
2. ISelect **West Europe**n the **IP Addresses** tab, enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| IPv4 address space | Enter **10.1.0.0/16** |

1. Under **Subnet name**, select the word **default**. If a subnet isn't present, select **+ Add subnet**.
2. In **Edit subnet**, enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Subnet name | Enter **myBackendSubnet** |
| Subnet address range | Enter **10.1.0.0/24** |

1. Select **Save** or **Add**.
2. Select the **Security** tab.
3. Under **BastionHost**, select **Enable**. Enter this information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Bastion name | Enter **myBastionHost** |
| AzureBastionSubnet address space | Enter **10.1.1.0/27** |
| Public IP Address | Select **Create new**. For **Name**, enter **myBastionIP**. Select **OK**. |

1. Select the **Review + create** tab or select the **Review + create** button.
2. Select **Create**.

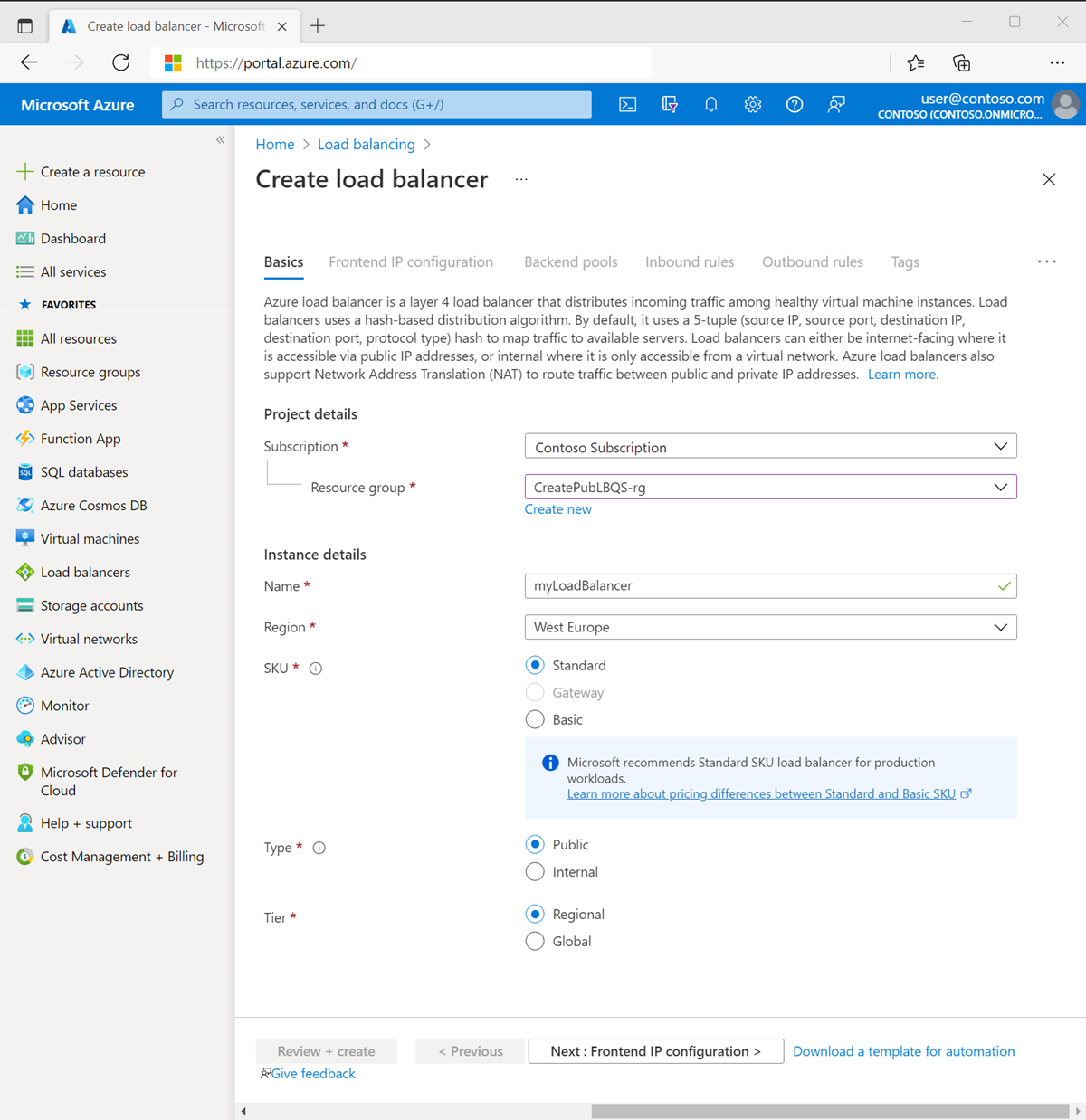
Create load balancer

During the creation of the load balancer, you'll configure:

* Frontend IP address
* Backend pool
* Inbound load-balancing rules
* Health probe

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project details** |  |
| Subscription | Select your subscription |
| Resource group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| Name | Enter **myLoadBalancer** |
| Region | Select **West Europe** |
| SKU | Leave the default **Standard** |
| Type | Select **Internal** |
| Type | Leave the default **Regional** |



1. Select **Next: Frontend IP configuration** at the bottom of the page.
2. In **Frontend IP configuration**, select **+ Add a frontend IP configuration**.
3. Enter **myFrontend** in **Name**.
4. Select **myBackendSubnet** in **Subnet**.
5. Select **Dynamic** for **Assignment**.
6. Select **Zone-redundant** in **Availability zone**.
7. Select **Add**.
8. Select **Next: Backend pools** at the bottom of the page.
9. In the **Backend pools** tab, select **+ Add a backend pool**.
10. Enter **myBackendPool** for **Name** in **Add backend pool**.
11. Select **NIC** or **IP Address** for **Backend Pool Configuration**.
12. Select **IPv4** or **IPv6** for **IP version**.
13. Select **Add**.
14. Select **Next: Inbound rules** at the bottom of the page.
15. In **Load balancing rule** in the **Inbound rules** tab, select **+ Add a load balancing rule**.
16. In **Add load balancing rule**, enter or select the following information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| Name | Enter **myHTTPRule** |
| IP Version | Select **IPv4** or **IPv6** depending on your requirements |
| Frontend IP address | Select **myFrontend** |
| Backend pool | Select **myBackendPool** |
| Protocol | Select **TCP** |
| Port | Enter **80** |
| Backend port | Enter **80** |
| Health probe | Select **Create new**. In **Name**, enter **myHealthProbe**. Select **TCP** in **Protocol**. Leave the rest of the defaults, and select **OK** |
| Session persistence | Select **None** |
| Idle timeout (minutes) | Enter or select **15** |
| TCP reset | Select **Enabled** |
| Floating IP | Select **Disabled** |
|  |  |

1. Select **Add**.
2. Select the blue **Review + create** button at the bottom of the page.
3. Select **Create**.

Create NAT gateway

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project details** |  |
| Subscription | Select your subscription |
| Resource group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| NAT gateway name | Enter **myNATgateway** |
| Region | Select **West Europe** |
| Availability zone | Select **None** |
| Idle timeout (minutes) | Enter **15** |

1. Select the **Outbound IP** tab or select **Next: Outbound IP** at the bottom of the page.
2. In **Outbound IP**, select **Create a new public IP address** next to **Public IP addresses**.
3. Enter **myNATgatewayIP** in **Name**.
4. Select **OK**.
5. Select the **Subnet** tab or select the **Next: Subnet** button at the bottom of the page.
6. In **Virtual network** in the **Subnet** tab, select **myVNet**.
7. Select **myBackendSubnet** under **Subnet name**.
8. Select the blue **Review + create** button at the bottom of the page, or select the **Review + create** tab.
9. Select **Create**.

Create virtual machines

1)

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project Details** |  |
| Subscription | Select your Azure subscription |
| Resource Group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| Virtual machine name | Enter **myVM1** |
| Region | Select **(Europe) West Europe** |
| Availability Options | Select **Availability zones** |
| Availability zone | Select **Zone 1** |
| Security type | Select **Standard** |
| Image | Select **Windows Server 2022 Datacenter: Azure Edition - Gen2** |
| Azure Spot instance | Leave the default of unchecked |
| Size | Choose VM size or take default setting |
| **Administrator account** |  |
| Username | Enter a username |
| Password | Enter a password |
| Confirm password | Reenter password |
| **Inbound port rules** |  |
| Public inbound ports | Select **None** |

1. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.
2. In the Networking tab, select or enter the following information:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Network interface** |  |
| Virtual network | Select **myVNet** |
| Subnet | Select **myBackendSubnet** |
| Public IP | Select **None** |
| NIC network security group | Select **Advanced** |
| Configure network security group | Select **Create new**. In the **Create network security group**, enter **myNSG** in **Name**. Under **Inbound rules**, select **+Add an inbound rule**. Under **Service**, select **HTTP**. Under **Priority**, enter **100**. In **Name**, enter **myNSGRule** Select **Add** Select **OK** |
|  |  |
|  |  |
| **Load balancing** |  |
| Place this virtual machine behind an existing load-balancing solution? | Select the check box |
| **Load balancing settings** |  |
| Load-balancing options | Select **Azure load balancing** |
| Select a load balancer | Select **myLoadBalancer** |
| Select a backend pool | Select **myBackendPool** |

1. Select **Review + create**.
2. Review the settings, and then select **Create**.
3. Follow the steps 1 through 7 to create another VM with the following values and all the other settings the same as **myVM1**:

|  |  |
| --- | --- |
| **Setting** | **VM 2** |
| Name | **myVM2** |
| Availability zone | **Zone 2** |
| Network security group | Select the existing **myNSG** |

## Create test virtual machine

## In this section, you'll create a VM named ****myTestVM****. This VM will be used to test the load balancer configuration.

1. In **Create a virtual machine**, type or select the values in the **Basics** tab:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Project Details** |  |
| Subscription | Select your Azure subscription |
| Resource Group | Select **CreatePubLBQS-rg** |
| **Instance details** |  |
| Virtual machine name | Enter **myTestVM** |
| Region | Select **Southeast Asia** |
| Availability Options | Select **No infrastructure redundancy required** |
|  |  |
| Security type | Select **Standard** |
| Image | Select **Windows Server 2022 Datacenter: Azure Edition - Gen2** |
| Azure Spot instance | Leave the default of unchecked |
| Size | Choose VM size or take default setting |
| **Administrator account** |  |
| Username | Enter a username |
| Password | Enter a password |
| Confirm password | Reenter password |
| **Inbound port rules** |  |
| Public inbound ports | Select **None** |

1. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.
2. In the **Networking** tab, select or enter:

|  |  |
| --- | --- |
| **Setting** | **Value** |
| **Network interface** |  |
| Virtual network | **myVNet** |
| Subnet | **myBackendSubnet** |
| Public IP | Select **None**. |
| NIC network security group | Select **Advanced** |
| Configure network security group | Select **MyNSG** created in the previous step. |

1. Select **Review + create**.
2. Review the settings, and then select **Create**.

**Install IIS**

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **myVM1**.
3. In the **Overview** page, select **Connect**, then **Bastion**.
4. Enter the username and password entered during VM creation.
5. Select **Connect**.
6. On the server desktop, navigate to **Windows Administrative Tools** > **Windows PowerShell** > **Windows PowerShell**.
7. In the PowerShell Window, execute the following commands to:
   * Install the IIS server.
   * Remove the default iisstart.htm file.
   * Add a new iisstart.htm file that displays the name of the VM.
8. # Install IIS server role

Install-WindowsFeature -name Web-Server -IncludeManagementTools

1. # Remove default htm file

Remove-Item C:\inetpub\wwwroot\iisstart.htm

1. # Add a new htm file that displays server name

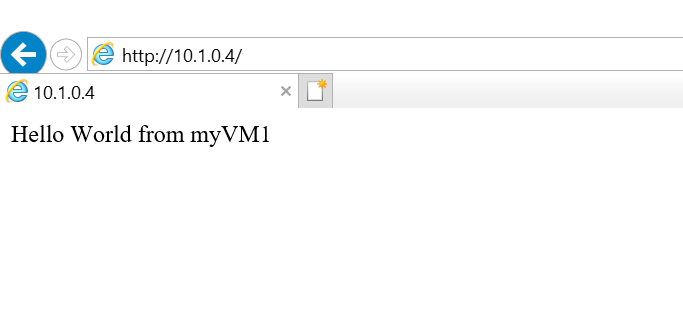
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)

1. Close the Bastion session with **myVM1**.
2. Repeat steps 1 through 8 to install IIS and the updated iisstart.htm file on **myVM2**.

## Test the load balancer

In this section, you'll test the load balancer by connecting to the **myTestVM** and verifying the webpage.

1. In the search box at the top of the portal, enter **Load balancer**. Select **Load balancers** in the search results.
2. Select **myLoadBalancer**.
3. Make note or copy the address next to **Private IP address** in the **Overview** of **myLoadBalancer**. If you can't see the **Private IP address** field, select **See more** in the information window.
4. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
5. Select **myTestVM**.
6. In the **Overview** page, select **Connect**, then **Bastion**.
7. Enter the username and password entered during VM creation.
8. Open **Internet Explorer** on **myTestVM**.
9. Enter the IP address from the previous step into the address bar of the browser(inside the mytestvm browser). The custom page displaying one of the backend server names is displayed on the browser. In this example, it's **10.1.0.4**.



To see the load balancer, distribute traffic across both VMs, you can force-refresh your web browser from the client machine.

## virtual machine 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 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?

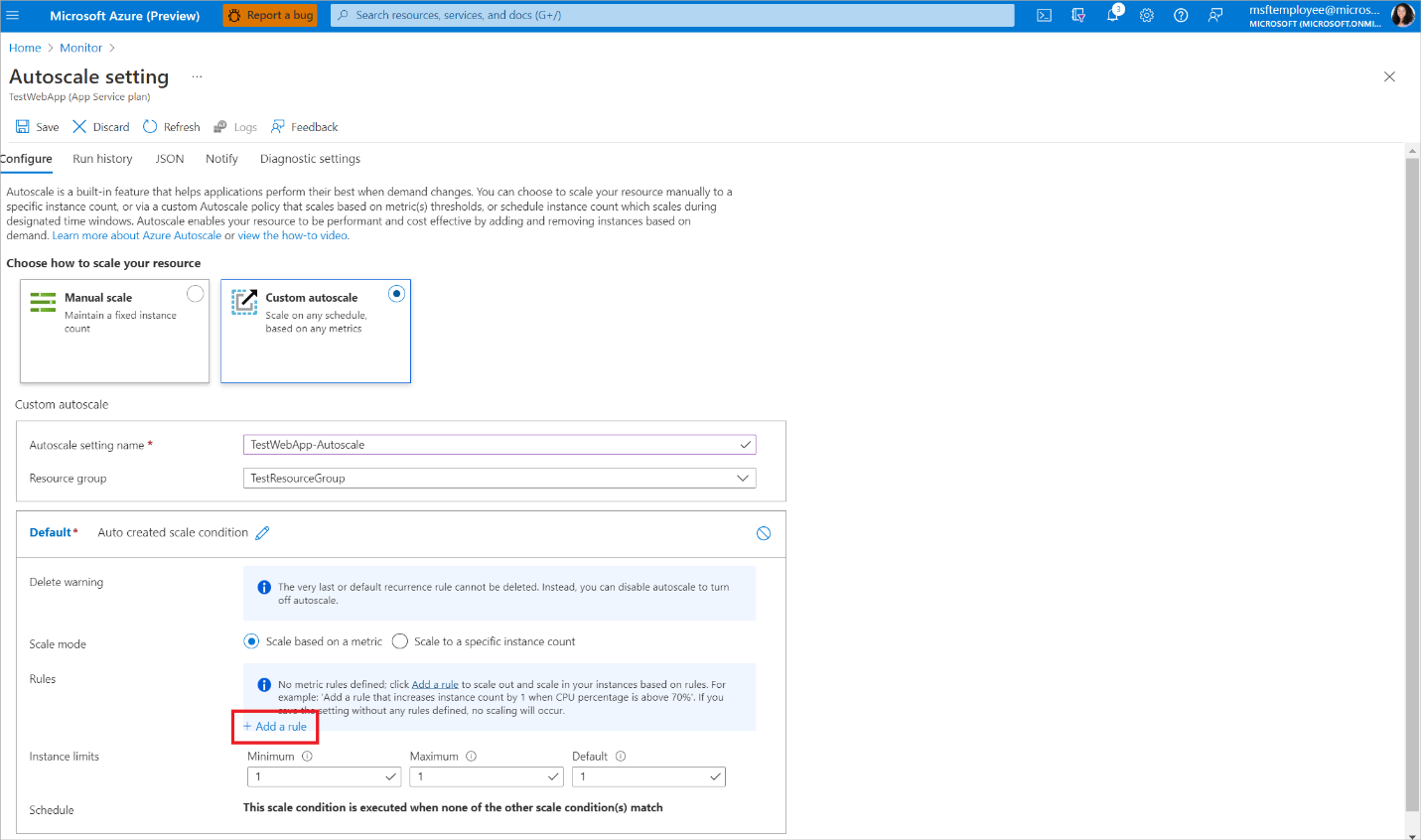
To provide redundancy and improved performance, applications are typically distributed across multiple instances. Customers may access your application through a load balancer that distributes requests to one of the application instances. If you need to perform maintenance or update an application instance, your customers must be distributed to another available application instance. To keep up with extra customer demand, you may need to increase the number of application instances that run your application.

## Create your first Autoscale setting

## Open the ****Autoscale**** blade in Azure Monitor and select a resource that you want to scale.

## Note that the current instance count is 1. Click Custom autoscale. [Scale setting for new web app.](https://docs.microsoft.com/en-us/azure/azure-monitor/autoscale/media/autoscale-get-started/manual-scale-04.png#lightbox)

Provide a name for the scale setting, and then click **Add a rule**. This opens as a context pane on the right side. By default, this sets the option to scale your instance count by 1 if the CPU percentage of the resource exceeds 70 percent. Leave it at its default values and click **Add**.

[](https://docs.microsoft.com/en-us/azure/azure-monitor/autoscale/media/autoscale-get-started/custom-scale-add-rule-05.png#lightbox)

You've now created your first scale rule. Note that the UX recommends best practices and states that "It is recommended to have at least one scale in rule." To do so:

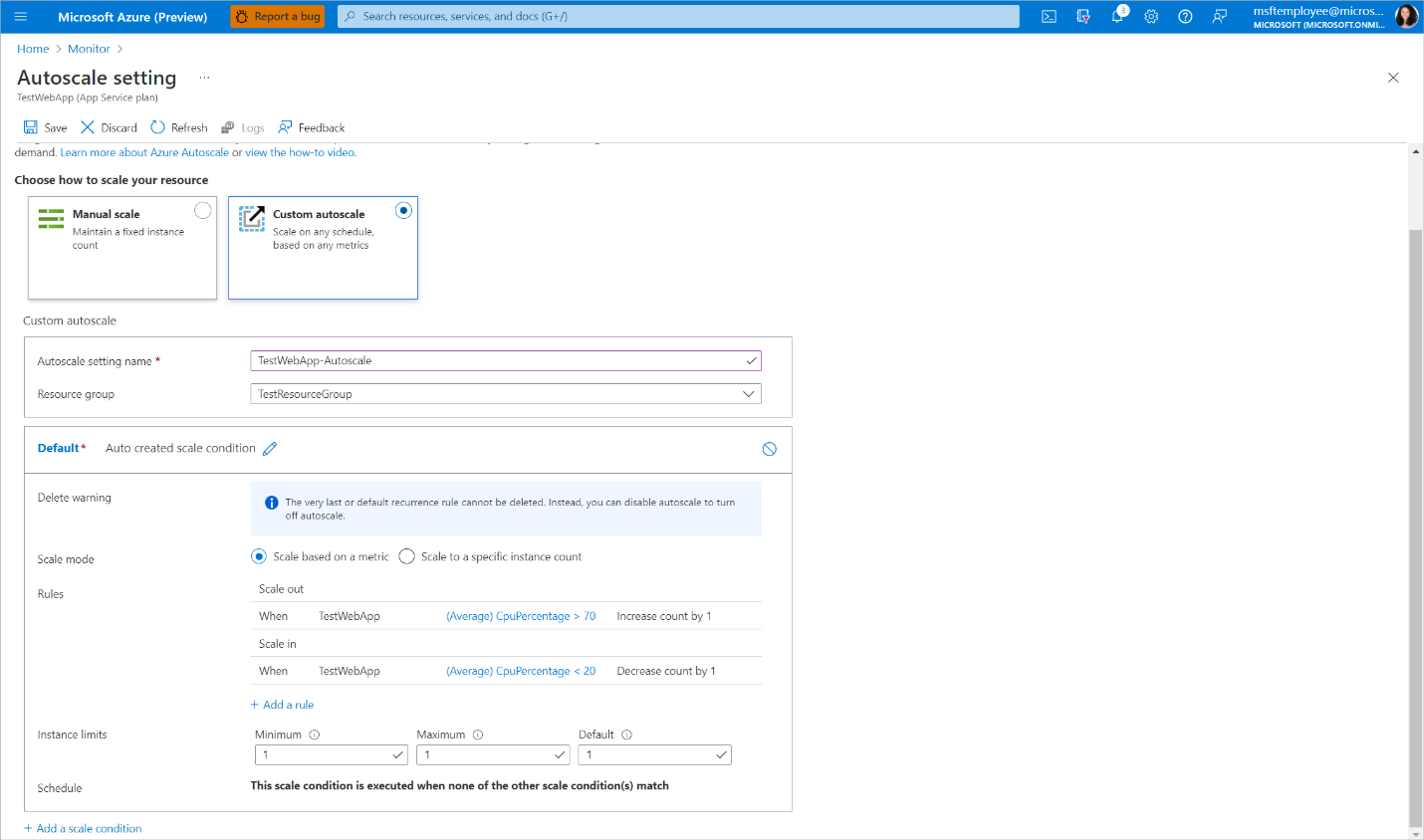
a. Click **Add a rule**.

b. Set Operator to Less than.

c. Set Threshold to 20.

d. Set Operation to Decrease count by.

You should now have a scale setting that scales out/scales in based on CPU usage.

[](https://docs.microsoft.com/en-us/azure/azure-monitor/autoscale/media/autoscale-get-started/custom-scale-results-06.png#lightbox)

Click **Save**.

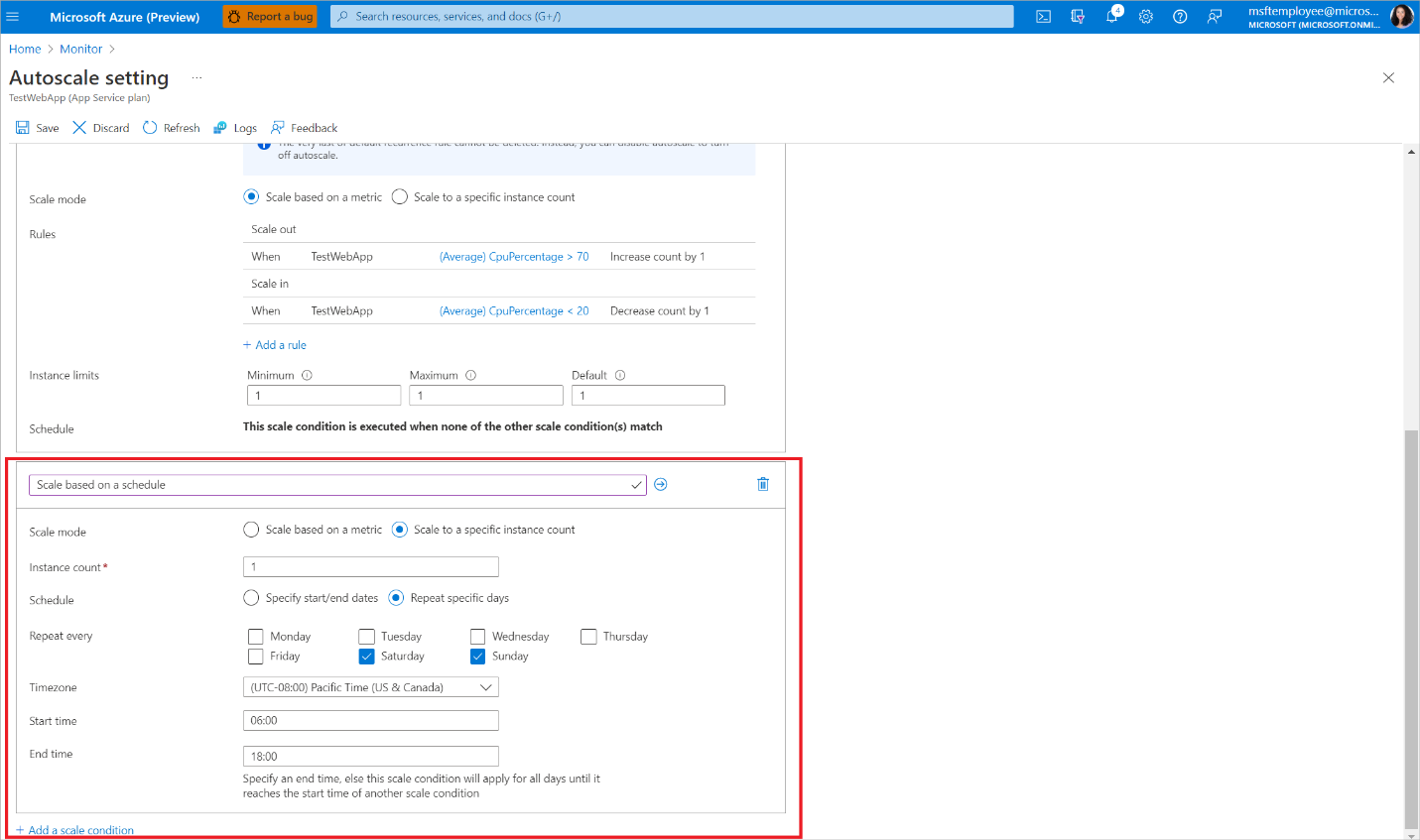
## Other considerations

### **Scale based on a schedule**

In addition to scale based on CPU, you can set your scale differently for specific days of the week.

1. Click **Add a scale condition**.
2. Setting the scale mode and the rules is the same as the default condition.
3. Select **Repeat specific days** for the schedule.
4. Select the days and the start/end time for when the scale condition should be applied.

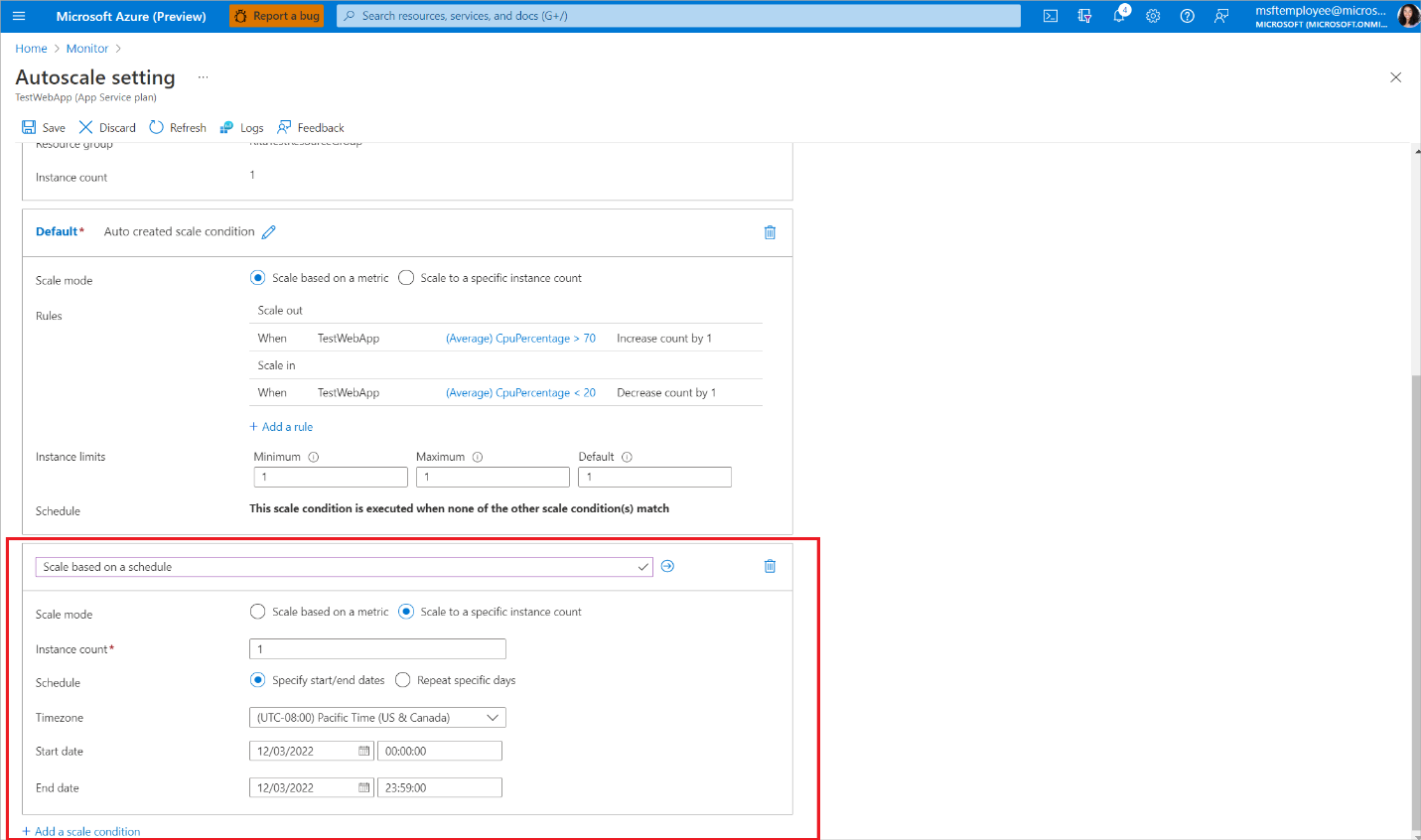
### 

[](https://docs.microsoft.com/en-us/azure/azure-monitor/autoscale/media/autoscale-get-started/scale-same-based-on-condition-07.png#lightbox)

### **Scale differently on specific dates**

In addition to scale based on CPU, you can set your scale differently for specific dates.

1. Click **Add a scale condition**.
2. Setting the scale mode and the rules is the same as the default condition.
3. Select **Specify start/end dates** for the schedule.
4. Select the start/end dates and the start/end time for when the scale condition should be applied.

[](https://docs.microsoft.com/en-us/azure/azure-monitor/autoscale/media/autoscale-get-started/scale-different-based-on-time-08.png#lightbox)

**Azure Devops**

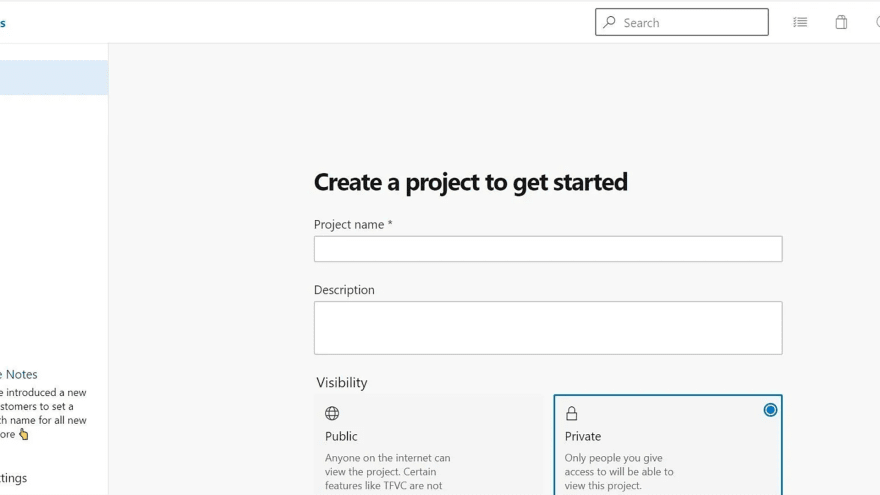
### **Azure Boards**

So, what is Azure Boards? In simple words, Azure Boards is a tool that allows you to **plan, organize, and track the work** of your team and your organization in a simple, easy way.

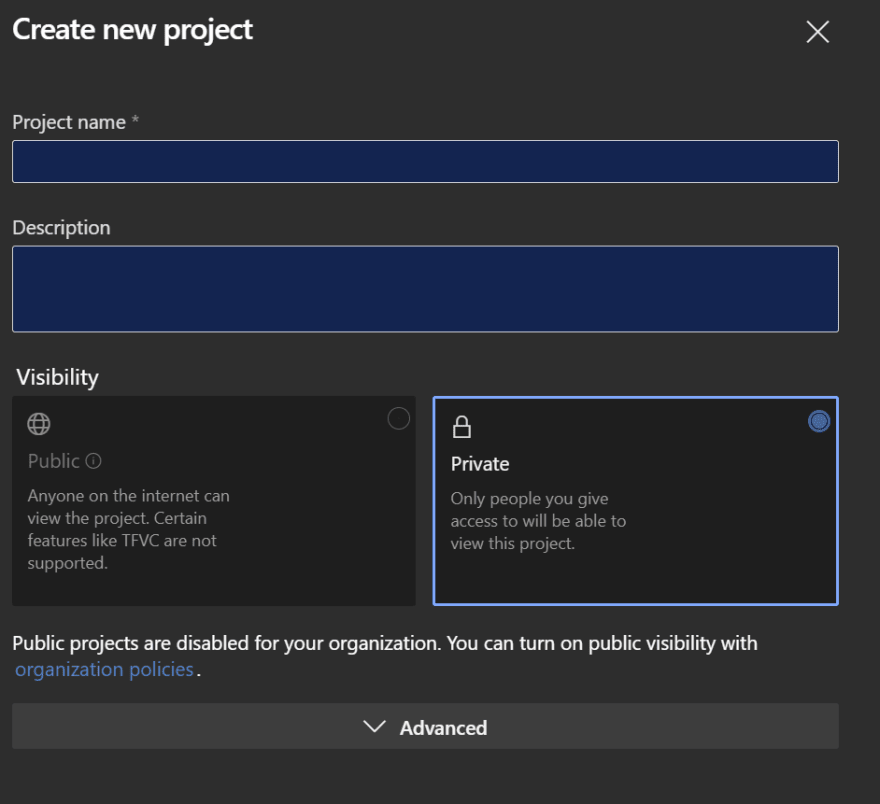
### **Create a new Project**

Alright, let’s see how we can get started and create a new project.

Once you login into Azure DevOps, if you don’t have any other project, you’ll have a prompt that says "*Create a project to get started*".

[](https://res.cloudinary.com/practicaldev/image/fetch/s--SaHc4_go--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/9dh7njpr7iu8edxoppjl.png)

If instead other projects are present, you can click on the "*New Project*" button in the top right and the *Create New Project* dialog will open.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--xIlX_dzc--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/ga0eyazv7g3gzhlt7s39.png)

In either case, here you can specify a name for your project and provide a brief description for it.

You’ll then be given two options on the type of project you can create.

* Public visibility permits users anywhere on the internet to see your board. That word public means exactly that! This is ideal for open-source projects that may need collaborators that are not part of a unified group that requires authentication.
* Private visibility is just that, it allows you to lock your Azure DevOps and only permit those you choose to have access. This is great for your personal projects or smaller projects that do not have a large team within your team.

### **Work Items**

### Work Items is just a flat list of all the work items you have created, regardless of the type, status, area, etc. In fact, you can see I have Tasks, Product Backlog Items, Features and even Bugs in here**.**

### **Backlogs and Boards**

### Backlogs and Boards are strictly correlated, in the sense that they show the same work items but in a different representation.

**Boards** show Epics, Features, and Backlog Items (in Scrum, or Users Stories if you are using the Agile template) in a Kanban-stye view.

#### **Sprints**

#### Sprints, on the other side, let you visualize and plan your iterations in a Kanban-like view, for tasks and bugs, keeping the relationship with their parent backlog items visible.

#### Here you can move the items via drag-and-drop to track the work done by your team.

#### **Queries**

#### Finally, the Queries tab allows you to navigate the data about the work items using either predefined queries or custom ones. You can specify conditions and clauses and manipulate the data for your own use.

### **Azure Repos**

Azure Repos is a set of version control tools that you can use to manage your code. Whether your software project is large or small, using version control as soon as possible is a good idea.

Version control systems are software that help you track changes you make in your code over time. As you edit your code, you tell the version control system to take a snapshot of your files. The version control system saves that snapshot permanently so you can recall it later if you need it. Use version control to save your work and coordinate code changes across your team.

Even if you're just a single developer, version control helps you stay organized as you fix bugs and develop new features. Version control keeps a history of your development so that you can review and even roll back to any version of your code with ease.

Azure Repos provides two types of version control:

* [Git](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos?view=azure-devops#git): distributed version control
* [Team Foundation Version Control (TFVC)](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos?view=azure-devops#tfvc): centralized version control