Before we start creating the infrastructure and start provisioning the workloads to them we should consider the situation when out service may get a fault and we should have a backup plan for that. So in order to have the backup we should consider the following :

- The backup machines should be available in case there is any issue in the primary machines.
- How to deal with the unfortunate hardware and software failure.
- How is our infra going to handle the load
- Unplanned Hardware Maintenance Events - When Azure platform predicts that the hardware or any platform components associated to a physical machine is about to fail.
- An unexpected downtime -- rarely occurs
- Planned Maintenance events -- periodic updates made by Microsoft.

In this document we will see how we can solve the above issue and how to handle them. Let’s begin.

Availability Set

Availability Sets ensure that the Azure virtual machines are deployed across multiple isolated hardware nodes in a cluster.

By deploying your vms across multiple hardware nodes Azure ensures that if hardware or software failure happens within Azure, only a sub-set of your virtual machines are impacted and your overall solution is safe and in working condition.

Availability set spreads your virtual machines across multiple fault domains and update domains.

Now after knowing all this you should have a question in your mind. When should I put my VM in availability set? So here is the simple answer : If you want to leverage Microsoft’s 99.95% SLA from Microsoft you must place your VMs inside availability set except your VMs are having premium storage.

Before moving to further on creation and other details availability set Its important to understand two terms closely related to each other they are : Fault domain and Update domain.

- **Fault Domain**

  Fault domains define the group of virtual machines that share a common power source and network switch. Each and every fault domain contains some racks and each rack contains virtual machine.

  Each of these Fault domain shares a power supply and a network switch. If there is a failure in the fault domain then all the resources in the fault domain become unavailable. You should place your vms such a way that each fault domain get one web server, one database server and like that.

- **Update Domain**
- Virtual machines get update domains automatically once they are put inside availability set.
- All virtual machines within that update domain will reboot together.
- Update domains are used for patching of the virtual machines.
- Only one update domain would be updated at the time

Question : How many Fault Domains and How many Update Domains we can have?

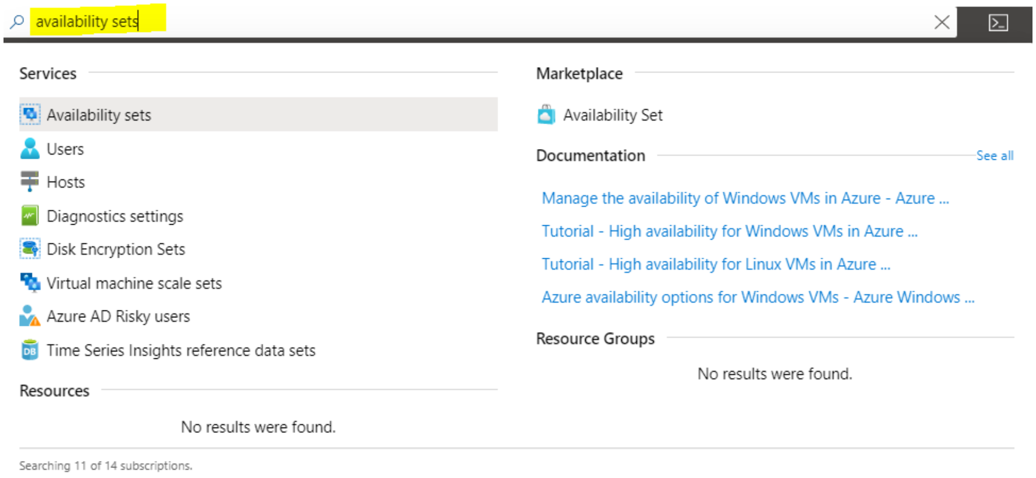
1. In Azure Service Management (ASM) portal, we have two Fault domains and 5 update domains.
2. In Azure Resource Manager(ARM) portal, we have three Fault domains and 5 update domains but we can upgrade our update domains from 5 to 20.
3. VMs are assigned sequentially in the update domains and fault domains.
4. Consider the below mentioned table to understand how virtual machines would be placed within Fault Domains and Update Domains inside an availability set.

Now we will go through the different ways of creating the availability set.

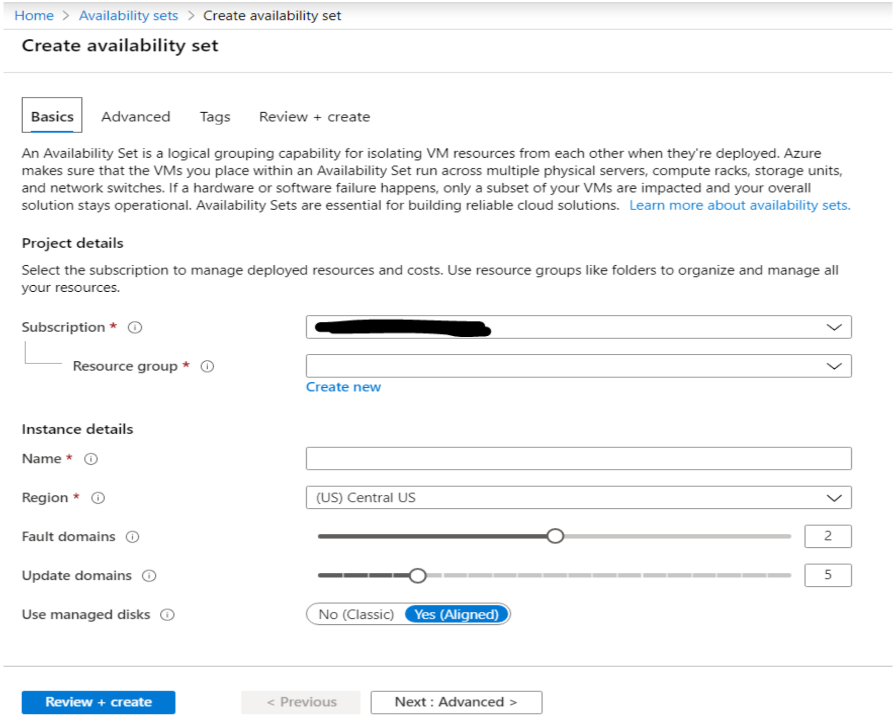
1) Through Azure Portal
2) Through PowerShell
3) Through C# code

Through Azure Portal

You can create an availability set using azure portal using the following steps. Go to azure portal and search for availability set.


availability sets
Services
Availability sets
Users
Hosts
Diagnostics settings
Disk Encryption Sets
Virtual machine scale sets
Azure AD Risky users
Time Series Insights reference data sets
Resources
No results were found.
Marketplace
Availability Set
Documentation
See all
Manage the availability of Windows VMs in Azure - Azure ...
Tutorial - High availability for Windows VMs in Azure ...
Tutorial - High availability for Linux VMs in Azure ...
Azure availability options for Windows VMs - Azure Windows ...
Resource Groups
No results were found.
Searching 11 of 14 subscriptions.

Once you are inside the availability sets page you can see an "+ Add" button in that. Upon clicking on the button you can see a form something like below.



Create availability set

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

Project details

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

Subscription *	[redacted]
Resource group *	 Create new

Instance details

Name *	
Region *	(US) Central US
Fault domains	2
Update domains	5
Use managed disks	No (Classic) [✓ Yes (Aligned)]

The definition of the parameters are below:

Subscription and resource group, subscription with be used for the cost evaluation and resource group will be your Resource group name in which you want to put your availability set.

Name: Name of your availability set.

Region: Region in which you want to put your availability set. Note that, the region or resource group in which you have created availability set we need to create the VM in same region and resource group.

Next you must put the number of fault domains and update domains you want to have in your availability set.

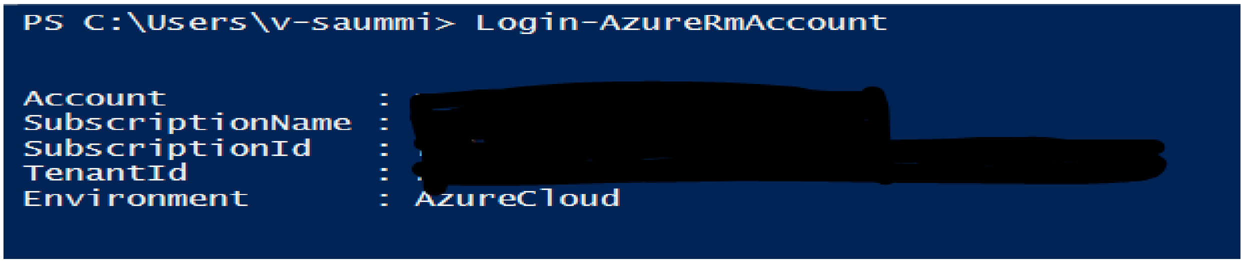
Through PowerShell

Creation of availability set through Portal may not be a best practice especially when you need to create multiple sets. In that case you may need to use PowerShell or C# code to do that.

Before you start running the commands to create the availability set in azure you need to login to the azure account. Based on the module you are using in powershell commands differs. Below are the module and commands:

i) Module: AzureRM.Compute

Login to azure account using Login-AzureRmAccount. Upon running the command it will ask you for the account and password. You can login using your Microsoft account. Once you login to the account it will show you the output as your account details.



```
PS C:\Users\v-saummi> Login-AzureRmAccount


Account          : [redacted]
SubscriptionName :
SubscriptionId   :
TenantId         :
Environment      : AzureCloud
```

Now once you logged in to your account you can use below command to create an availability set.

`SYNTAX and Command (New-AzureRmAvailabilitySet)`

New-AzureRmAvailabilitySet [-ResourceGroupName] <String> [-Name] <String> [-Location] <String> [[-PlatformUpdateDomainCount] <Int32>]
[[-PlatformFaultDomainCount] <Int32>] [[-Sku] <String>] [-AsJob] [-DefaultProfile <IAzureContextContainer>] [-Managed] [<CommonParameters>]

-AsJob parameter : Run cmdlet in the background and return a Job to track progress.
-Sku :

The Name of Sku. The acceptable values for this parameter are:

Aligned: For managed disks
Classic: For unmanaged disks

ii) Module: Az.Compute

Unlike AzureRm module the Az module will not be installed by default. To install that, Run the below command running the powershell in Administrator mode.

Command to install Az module : **install-module -name Az -Force**

Login to azure account using Login-AzAccount. Upon running the command it will ask you for the account and password. You can login using your Microsoft account. Once you login to the account it will show you the output as your account details.

Now once you logged in to your account you can use below command to create an availability set.

To create the availability set using az module use command New-AzAvailabilitySet. All the parameter will be same as of AzureRm module.

Creating the Availability set using C#

If you are planning to use some High-level coding language such as C# or Java, you can also create an availability set in azure. In this document we will only cover the code for C# and will provide the reference link for Java.

i) Using C#:

We can use following function / Constructor for creating the availability set in azure using C#.
To do so we first need to install the nuget package : "*Microsoft.Azure.Management.Compute*" into our solution and have to import the namespace :" *Microsoft.Azure.Management.Compute.Models*"

```
public AvailabilitySet (string location, string id = default, string name = default, string type = default,
System.Collections.Generic.IDictionary<string,string> tags = default, int? platformUpdateDomainCount = default, int?
platformFaultDomainCount = default,
System.Collections.Generic.IList<Microsoft.Azure.Management.Compute.Models.SubResource> virtualMachines = default,
Microsoft.Azure.Management.Compute.Models.SubResource proximityPlacementGroup = default,
System.Collections.Generic.IList<Microsoft.Azure.Management.Compute.Models.InstanceViewStatus> statuses = default,
Microsoft.Azure.Management.Compute.Models.Sku sku = default);
```

The Above function will create an instance of the Availability Set, once it is created we can use .Create() function to create the availability set.

Next to create an VM inside the availability set we can again use the following function :

```
var azure = Azure
   .Configure()
   .WithLogLevel(HttpLoggingDelegatingHandler.Level.Basic)
   .Authenticate(credentials)
   .WithDefaultSubscription();

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();
```

ii) Using Java
Reference Link : https://github.com/Azure-Samples/compute-java-manage-availability-sets/blob/master/src/main/java/com/microsoft/azure/management/compute/samples/ManageAvailabilitySet.java

But before using the function in java or C# we first need to create the auth file for azure authentication. You can refer below link to do so.
Create auth file using : https://github.com/Azure/azure-libraries-for-net/blob/master/AUTH.md

Load Balancer

Azure Load balancer is used in azure for efficient distribution of traffic thereby providing low latency. Before moving to the Load balancer its important to have some idea about the scale set.

First of all the key difference between the Scale set and availability set is that, the scale set will automatically delete & create the VMs where as, in Availability set we will be manually deleting and creating the VMs hence it is our responsibility to update the backend pools ourselves.

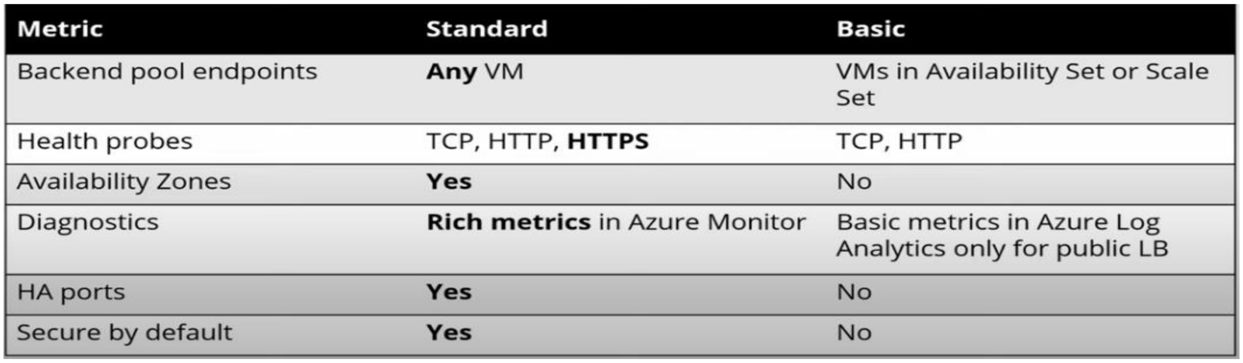
Scale Set:

- It is a collection of or we can say a set of identical machines, and inside a scale set we can have 100s of servers and in the following few pages in this tutorial we will see how to distribute the traffic within them using Load balancer.
- A scale set is always located inside the SubNet inside a VNet.
- When a request will be coming from the end User the request will be redirected to one of the machines in a backend pool through Load balancer.
- When a scale set will scale itself in / out it will update the Load balancer so that load balancer has the update how many machines are there in the backend pool.
- One more important point you should remember here is, Once our backend pool is set, it is upto us whether we want to make the load balancer a public one or a private one. When we choose it to give a public we can assign a public IP, allowing it application or user to use it, other wise a private IP will be assigned to make it a internal load balancer.

Key Terms of a Load Balancer :

- Front End IP configuration: Public or Private IP.
- Backend Pools: Group of servers.
- Health Probes: Monitoring the backend pool health. This will be periodically checking whether the machines or the servers within the backend pool is healthy to redirect the traffic to that.
- Load Balancing Rules: That will help us to set the rules for traffic distribution.
- Inbound NAT rule: This will help us in binding the specific IP/port to specific instance in backend pool.
- Outbound rule: helps us in controlling the outgoing traffic.

Difference between the two tiers of SKU in load balancer:



Metric	Standard	Basic
Backend pool endpoints	**Any** VM	VMs in Availability Set or Scale Set
Health probes	TCP, HTTP, **HTTPS**	TCP, HTTP
Availability Zones	**Yes**	No
Diagnostics	**Rich metrics** in Azure Monitor	Basic metrics in Azure Log Analytics only for public LB
HA ports	**Yes**	No
Secure by default	**Yes**	No

- The important point to remember here is, the load balancer has a regional scope that means the region in which you have created the load balancer you can only balance the traffic in that region for any other.

Now there is an interesting path from here in setting the load balancer. So we will see four different demos in configuring the load balancer.

- Creating the public load balancer
- Configuring the backend pools
- Adding health probes.
- Adding load balancing rules.

So to set up the Load balancer we will be using the availability set and to do that, we will be creating the VNet for first, inside the VNet we have to configure the Subnet and then we will add three machines to the availability set. So lets start…

1) Creating the VMs in Availability Set: In this tutorial we have already discussed how we can create the VM and add it to availability set.

   For the simplicity I am adding here the automation script that will help you do the same. The below script will be creating the VNEt, a availability set based on the VNet created and create three VMs within the availability set.

   To start please login to the azure account using "*Login-AzAccount*" cmdlet.

   Script to create availability set and add VMs in it.

```
$vnetconfig = New-AzVirtualNetwork -Name "saum-vnet" -Location "EastUS" -AddressPrefix 10.0.0.0/16 -
ResourceGroupName testRG

$subnetConfig = Add-AzVirtualNetworkSubnetConfig -Name "saum-subnet" -AddressPrefix 10.0.0.0/8 -
VirtualNetwork $vnetconfig

New-AzAvailabilitySet -Name "saum-avaiSet" -ResourceGroupName testRG -Location "EastUS" -
PlatformUpdateDomainCount 3 -PlatformFaultDomainCount 3 -Sku "Aligned"

for($i = 0; $i -lt 2 ; $i++)
{
    $name = "TEST-VM" + $i;
    New-AzVM -Name $name -ResourceGroupName testRG -Location "EastUS" -VirtualNetworkName "saum-vnet"
-SubnetName "saum-subnet" -Credential(get-credential) -AvailabilitySetName "saum-avaiSet"
}
```

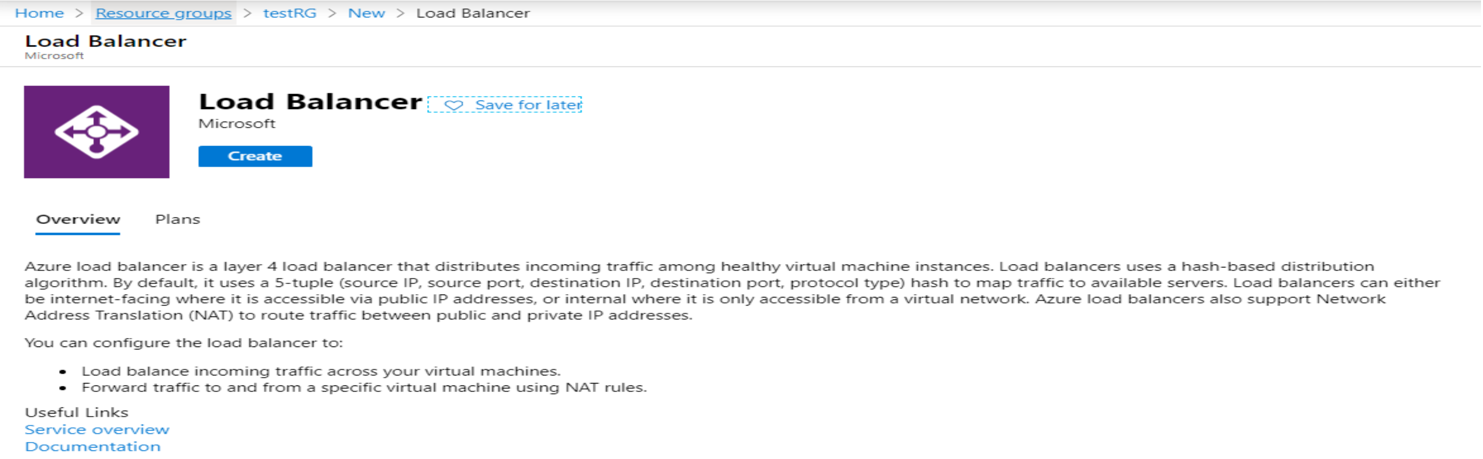
While running the script if you got below warning, you can suppress this using cmslet : "*Set-Item Env:\SuppressAzurePowerShellBreakingChangeWarnings "true"* "

```
WARNING: Breaking changes in the cmdlet 'Add-AzVirtualNetworkSubnetConfig' :
WARNING:  - Update Property Name


WARNING: Cmdlet invocation changes :
       Old Way : -ResourceId
       New Way : -NatGatewayId
WARNING:  - Update Property Name


WARNING: Cmdlet invocation changes :
       Old Way : -InputObject
       New Way : -NatGateway
WARNING: NOTE : Go to https://aka.ms/azps-changewarnings for steps to suppress this breaking change
warning, and other information on breaking changes in Azure Power
Shell.
```

2) Once the availability set is created and you added the VM in that. Login to the azure portal and search for Load balancer in the search box. You will find something like below. Click on the "create" button and proceed with the steps.



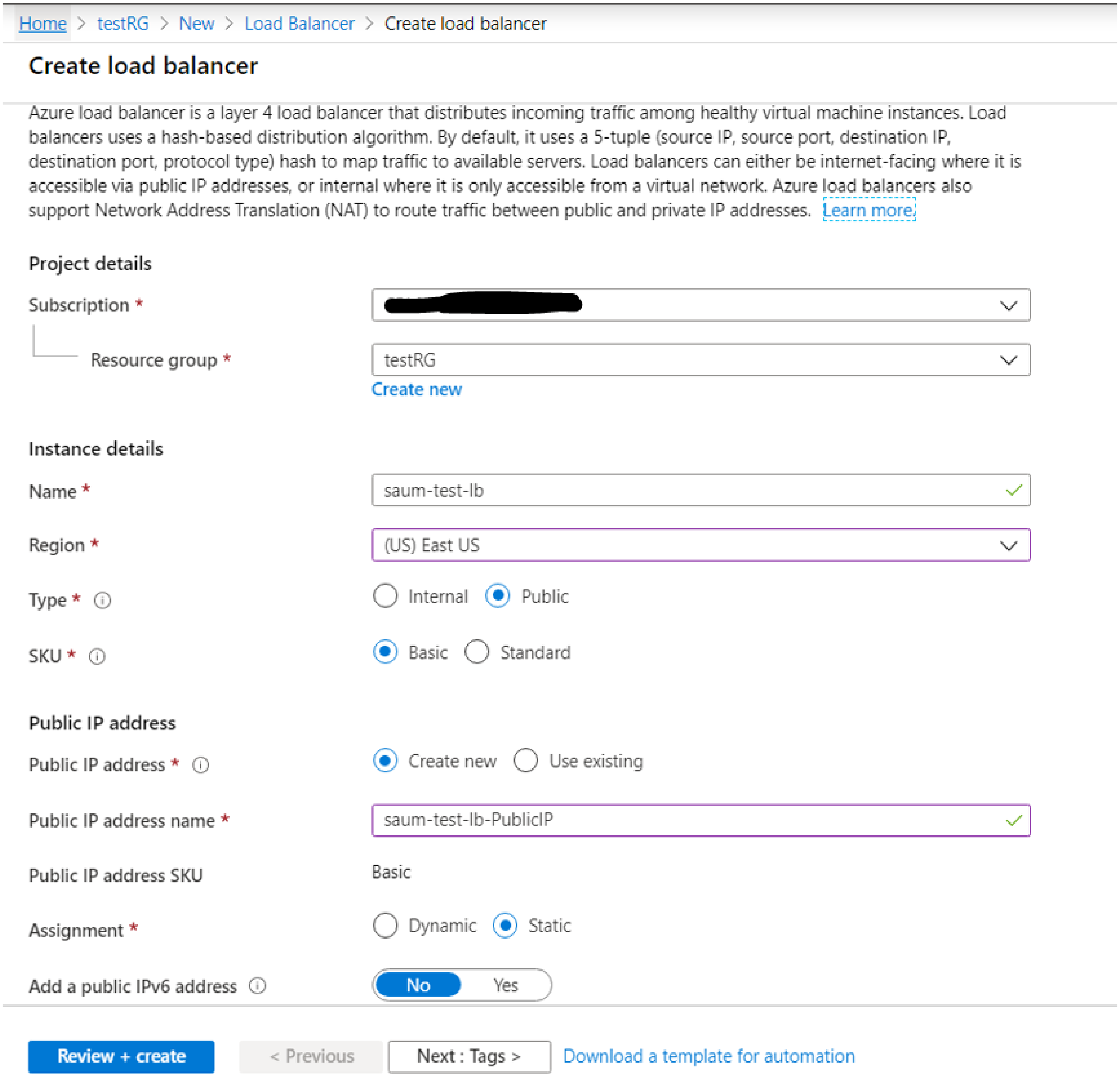
Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses.

You can configure the load balancer to:

- Load balance incoming traffic across your virtual machines.
- Forward traffic to and from a specific virtual machine using NAT rules.

Useful Links
Service overview
Documentation

3) Upon clicking on the create button we will get the form below. Enter the details as given below and click on the “review and create” button.



Home > testRG > New > Load Balancer > Create load balancer

Create load balancer

Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses. Learn more

Project details

Subscription *: [redacted]

Resource group *: testRG

Create new

Instance details

Name *: saum-test-lb ✓

Region *: (US) East US

Type *: ○ Internal ● Public

SKU *: ● Basic ○ Standard

Public IP address

Public IP address *: ● Create new ○ Use existing

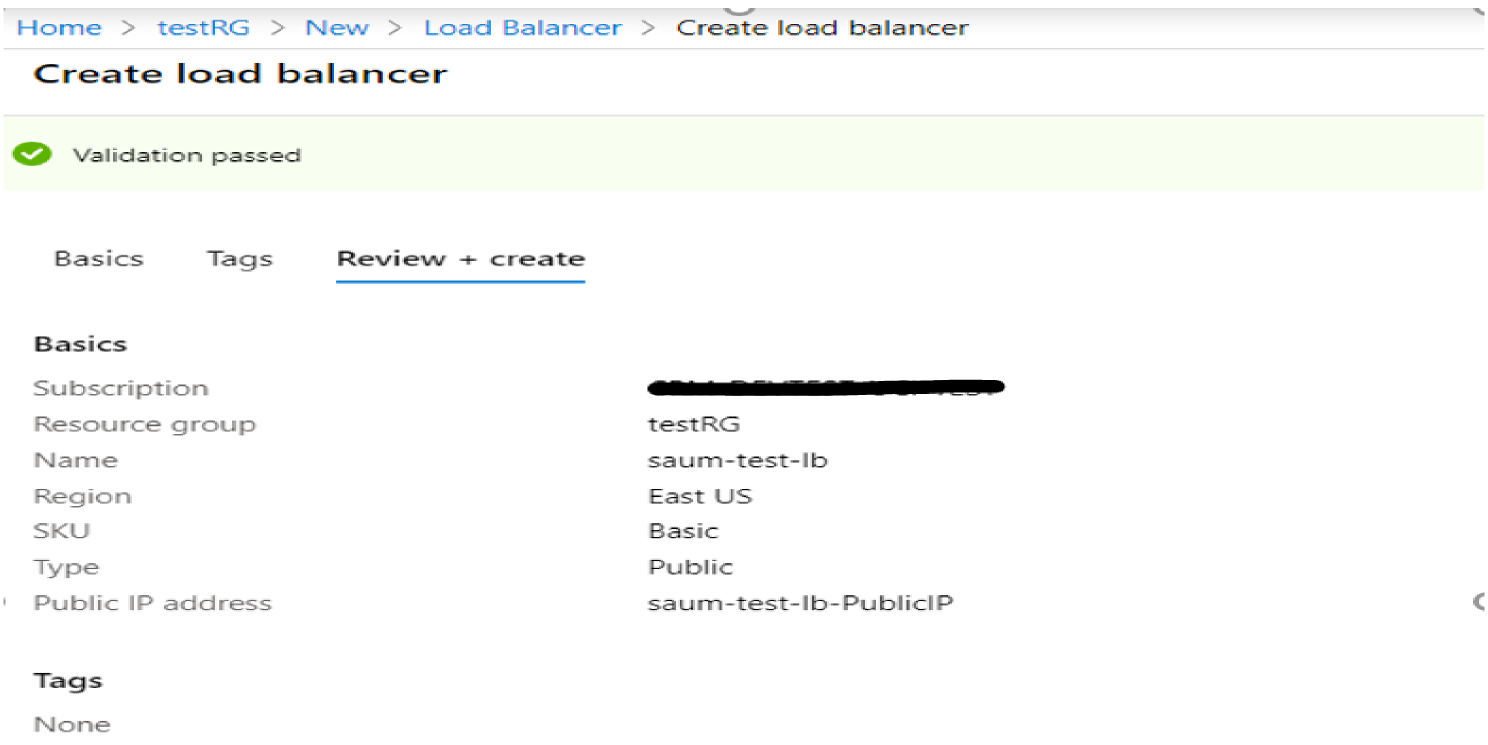
Public IP address name *: saum-test-lb-PublicIP ✓

Public IP address SKU: Basic

Assignment *: ○ Dynamic ● Static

Add a public IPv6 address: [✓ No] Yes

4) Upon clicking on the button it will validate the load balancer settings and again on clicking on the create button on this page, will create a load balancer.



Create load balancer

Validation passed

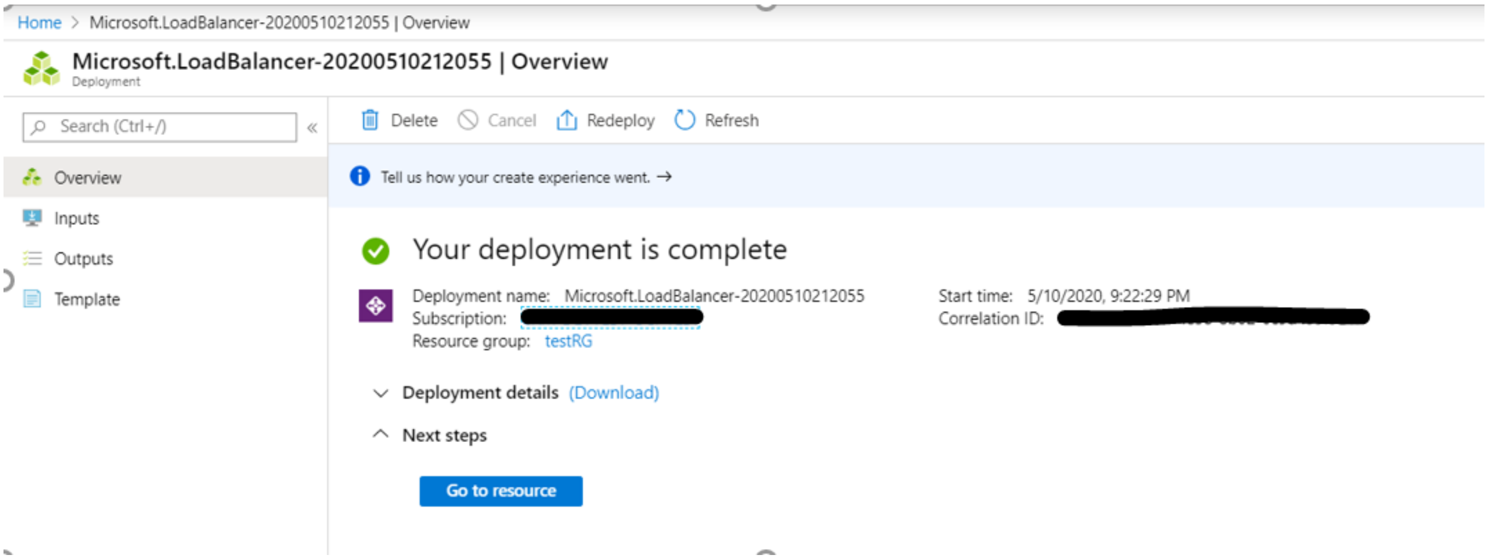
Basics

Subscription	[redacted]
Resource group	testRG
Name	saum-test-lb
Region	East US
SKU	Basic
Type	Public
Public IP address	saum-test-lb-PublicIP

Tags

None

Upon completing you can see the load balancer in load balancer resource. Something like below.


Home > Microsoft.LoadBalancer-20200510212055 | Overview
Search (Ctrl+/)
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Refresh
Tell us how your create experience went. →
Deployment details (Download)
Next steps
Go to resource

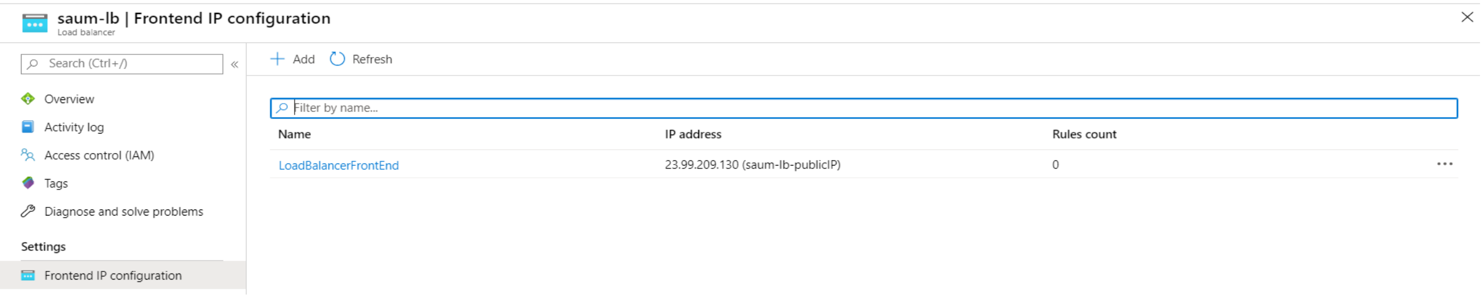
Microsoft.LoadBalancer-20200510212055 | Overview
Deployment

Your deployment is complete

Deployment name: Microsoft.LoadBalancer-20200510212055
Subscription: [redacted]
Resource group: testRG
Start time: 5/10/2020, 9:22:29 PM
Correlation ID: [redacted]

Once the load balancer is created you will have to now configure the IP configuration, backend pool, load Balancer rule and health probe.

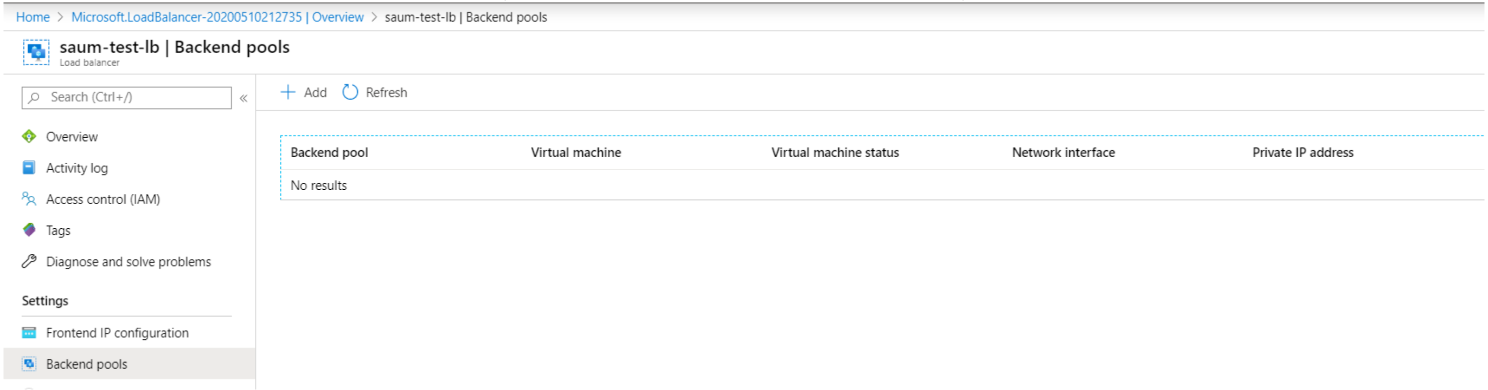
To configure the IP go to “FrontEnd IP configuration” and enter the details and create the IP configuration. Once you created the IP configuration you will be able to see something like below:



saum-lb | Frontend IP configuration
Load balancer

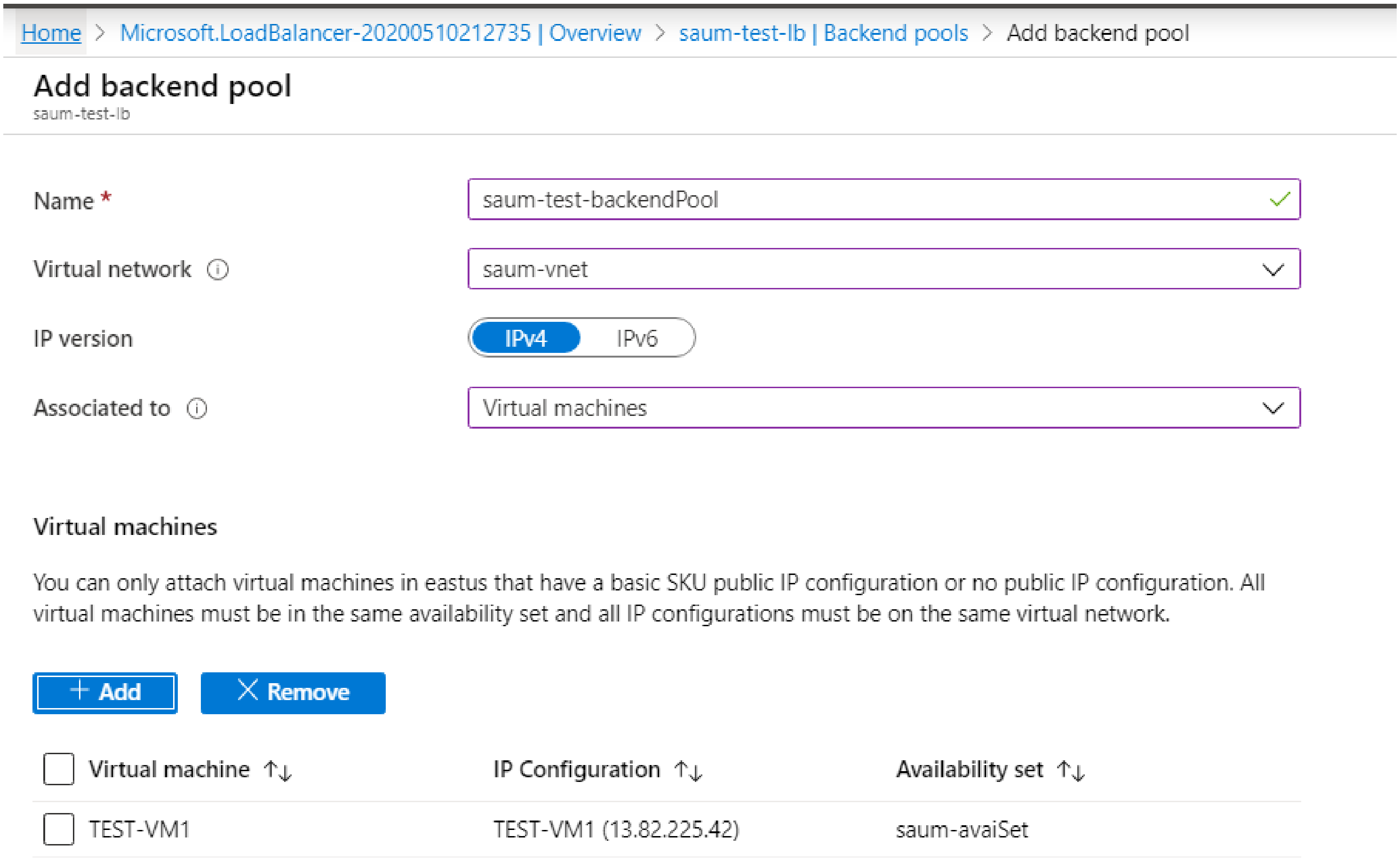
Name	IP address	Rules count
LoadBalancerFrontEnd	23.99.209.130 (saum-lb-publicIP)	0

- Now its time to add the Back end pools. To do that click on the Backend pools tab and click on add.


Home > Microsoft.LoadBalancer-20200510212735 | Overview > saum-test-lb | Backend pools
saum-test-lb | Backend pools
Load balancer
Search (Ctrl+/)
«
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
+ Add
Refresh

Backend pool	Virtual machine	Virtual machine status	Network interface	Private IP address
No results				

- Upon clicking on the add button you will get something like below asking few info regarding the backend pool settings.
- You can fill the form, keep in mind, you have to give the IP created above for this purpose only. That is in the same region as of Load balancer and Availability set.
- Another point to note here is, choose the “associated to” field as Virtual machines when using availability set.
- After that you can add the VMs in your availability set using the Add button.


Home > Microsoft.LoadBalancer-20200510212735 | Overview > saum-test-lb | Backend pools > Add backend pool

Add backend pool

saum-test-lb

Name * saum-test-backendPool ✓

Virtual network ⓘ saum-vnet

IP version [✓ IPv4] IPv6

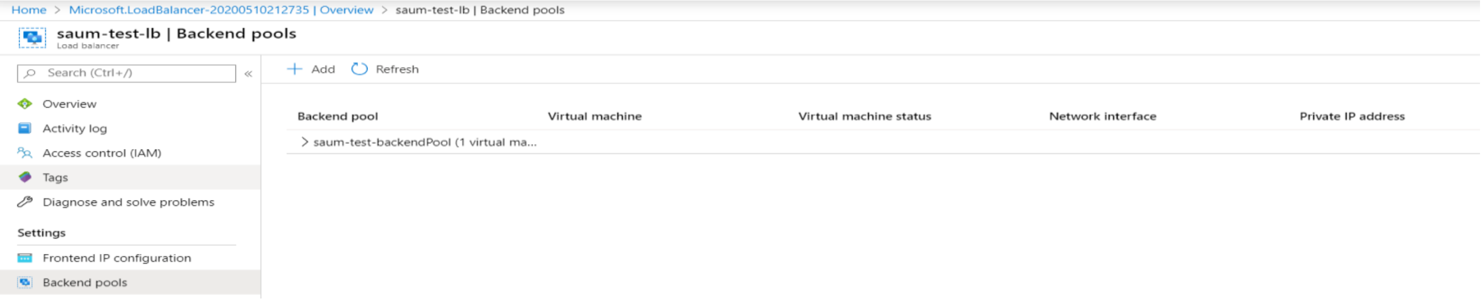
Associated to ⓘ Virtual machines

Virtual machines

You can only attach virtual machines in eastus that have a basic SKU public IP configuration or no public IP configuration. All virtual machines must be in the same availability set and all IP configurations must be on the same virtual network.

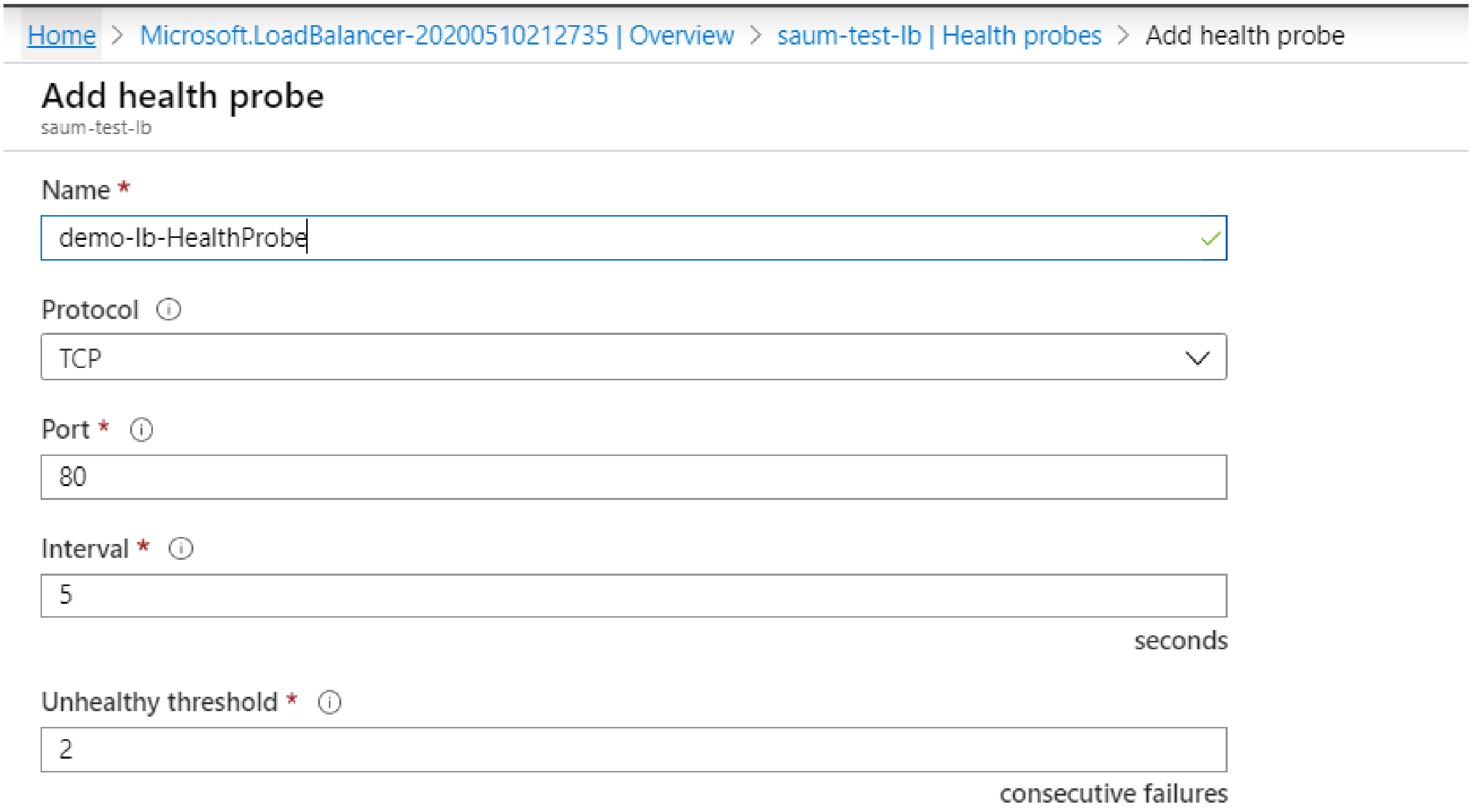
□ Virtual machine ↑↓	IP Configuration ↑↓	Availability set ↑↓
□ TEST-VM1	TEST-VM1 (13.82.225.42)	saum-avaiSet

Once backend pool is configured, we can see those in the backend pools tab.


Home > Microsoft.LoadBalancer-20200510212735 | Overview > saum-test-lb | Backend pools
saum-test-lb | Backend pools
Load balancer
Search (Ctrl+/)
«
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
+ Add
Refresh

Backend pool	Virtual machine	Virtual machine status	Network interface	Private IP address
> saum-test-backendPool (1 virtual ma...				

- Now adding the health probe, health probe is an very useful thing to configure and health probe will decide whether to route the traffic to the particular server or not. To do that, it will continuously check whether the machines are up and running every time. Again to do that, It will be sending a signal to all the machines in a regular interval (will be setting in the "Interval" tab of the form below) and in case it didn't receive a good signal from anyone of the machines for the number of times (will be setting to the "Unhealthy threshold") in below form, it will make the server as unavailable and will stop rerouting the traffic to the server.



Home > Microsoft.LoadBalancer-20200510212735 | Overview > saum-test-lb | Health probes > Add health probe

Add health probe
saum-test-lb

Name * demo-lb-HealthProbe

Protocol TCP

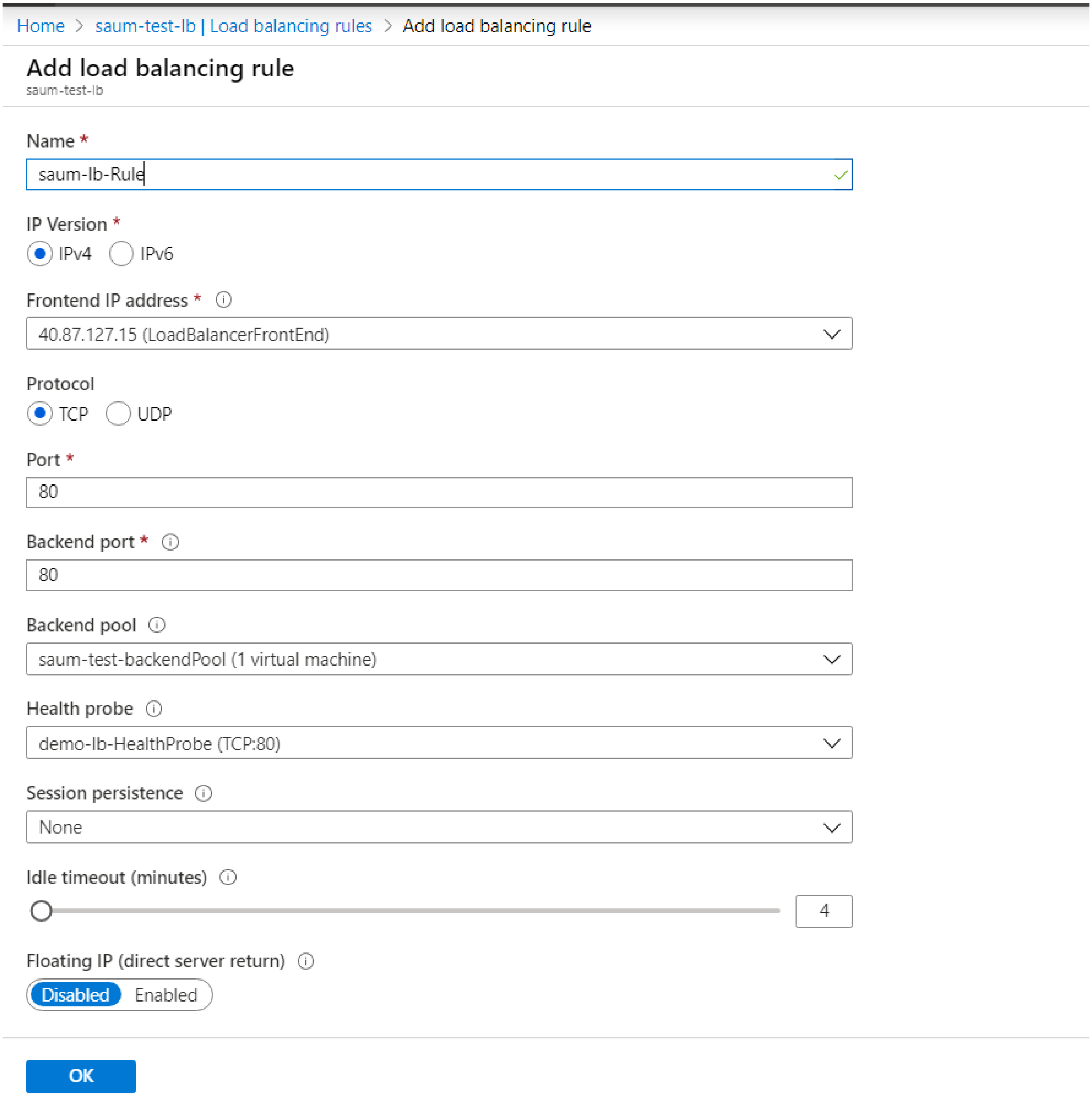
Port * 80

Interval * 5 seconds

Unhealthy threshold * 2 consecutive failures

Now again its time for the full and final steps is to add the routing rules. To do that click on the routing rules tab and fill the form as below. It will be completely based on the previous settings you have done, like pointing to the backend pool, ip configuration and health probe.

A sample form is given below with the required parameters filled.



Home > saum-test-lb | Load balancing rules > Add load balancing rule

Add load balancing rule

saum-test-lb

Name *

saum-lb-Rule

IP Version *

● IPv4 ○ IPv6

Frontend IP address * ⓘ

40.87.127.15 (LoadBalancerFrontEnd)

Protocol

● TCP ○ UDP

Port *

80

Backend port * ⓘ

80

Backend pool ⓘ

saum-test-backendPool (1 virtual machine)

Health probe ⓘ

demo-lb-HealthProbe (TCP:80)

Session persistence ⓘ

None

Idle timeout (minutes) ⓘ

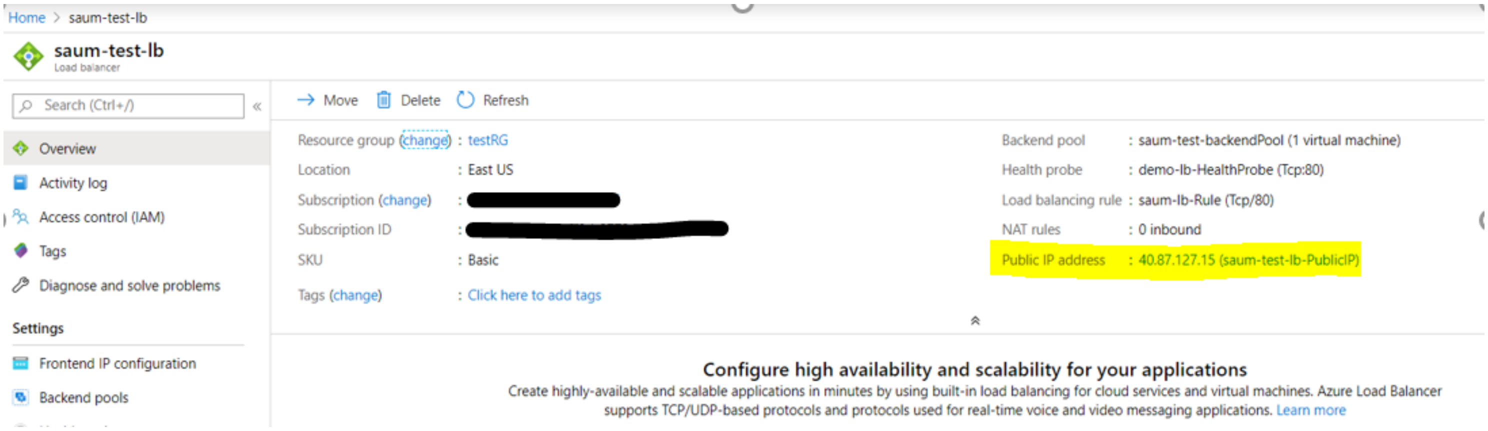
4

Floating IP (direct server return) ⓘ

[✓ Disabled] Enabled

OK

Once you are done you can now use the IP of the load balancer to balance the public traffic.



Resource group (change) : testRG

Location : East US

Subscription (change) : [redacted]

Subscription ID : [redacted]

SKU : Basic

Tags (change) : Click here to add tags

Backend pool : saum-test-backendPool (1 virtual machine)

Health probe : demo-lb-HealthProbe (Tcp:80)

Load balancing rule : saum-lb-Rule (Tcp/80)

NAT rules : 0 inbound

Public IP address : 40.87.127.15 (saum-test-lb-PublicIP)

Configure high availability and scalability for your applications

Create highly-available and scalable applications in minutes by using built-in load balancing for cloud services and virtual machines. Azure Load Balancer supports TCP/UDP-based protocols and protocols used for real-time voice and video messaging applications. Learn more

Application Gateway:

Link : https://docs.microsoft.com/en-us/azure/application-gateway/quick-create-portal