

Course-End Project 1:

Create Highly Available Architecture by Distributing Incoming Traffic among Healthy Service Instances in Cloud Services or Virtual Machines in a Load Balanced Set with the Help of Command-Line Interface

Project Agenda Create a highly available architecture by distributing incoming traffic among healthy service instances in cloud services or virtual machines in a load-balanced set with the help of a command-line interface.

Expected Deliverables:

- Identify Virtual machines and Networking
- Configure the load balancer
- Extend the load balancer with backend pool and frontend IP
- Define the Health probe

For this project, we will use Bash command through the Azure Command-Line Interface (CLI).

Step 1: Identify Virtual machines and Networking

1.1: Create Resource Group

First login to your Azure portal and be brought to the home page.

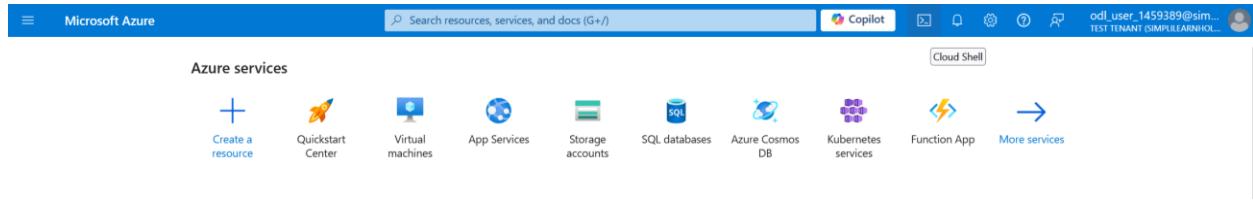
The screenshot shows the Microsoft Azure portal homepage. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for Copilot, notifications, and account management. Below the navigation bar, the main content area is divided into two main sections: "Azure services" and "Resources".

Azure services: This section contains icons for creating a resource, Quickstart Center, Virtual machines, App Services, Storage accounts, SQL databases, Azure Cosmos DB, Kubernetes services, Function App, and More services.

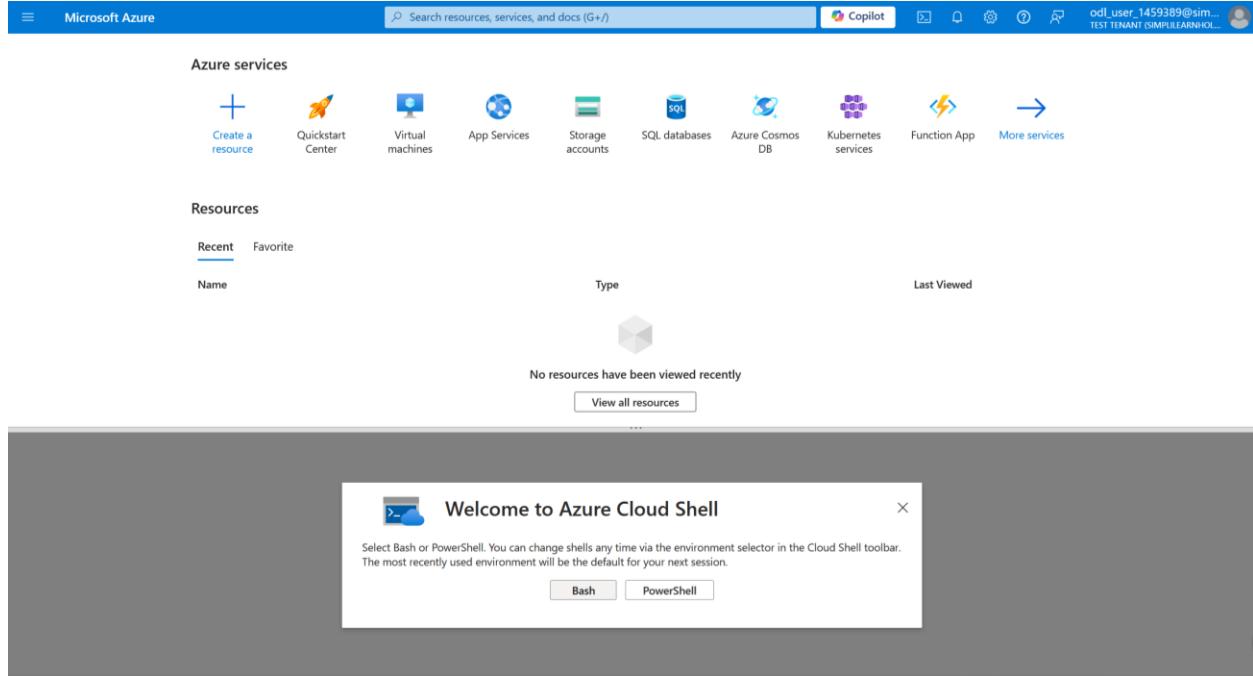
Resources: This section has tabs for "Recent" (which is underlined) and "Favorite". It includes columns for "Name", "Type", and "Last Viewed". A message at the bottom states "No resources have been viewed recently" and includes a "View all resources" button.

This project will require you to create a project with the command-line interface's help.

Go to the top right corner and click on the icon for “Cloud Shell.”

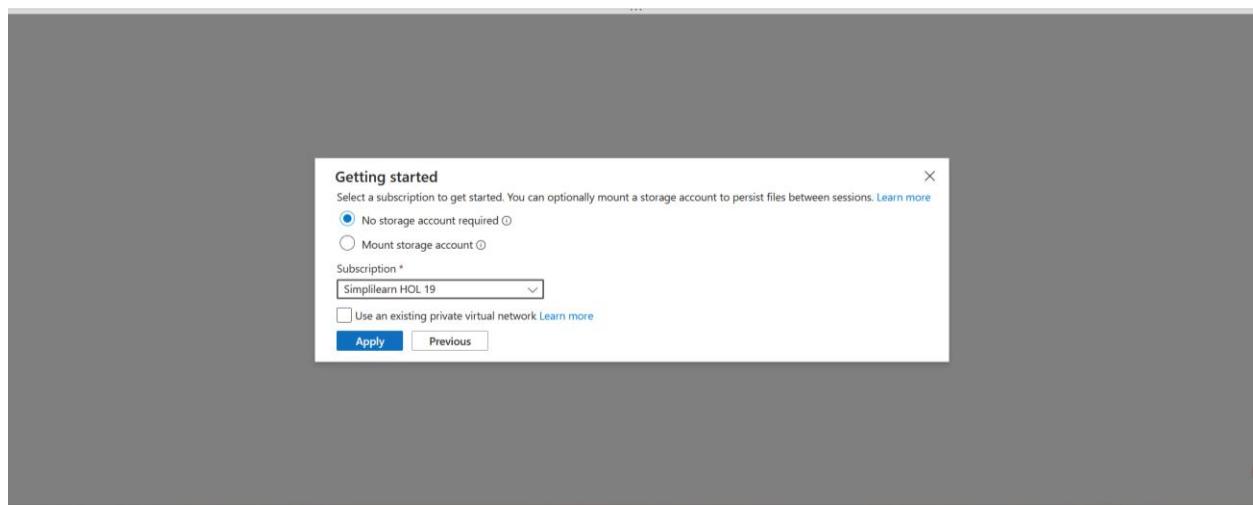


Click on the Cloud Shell icon.

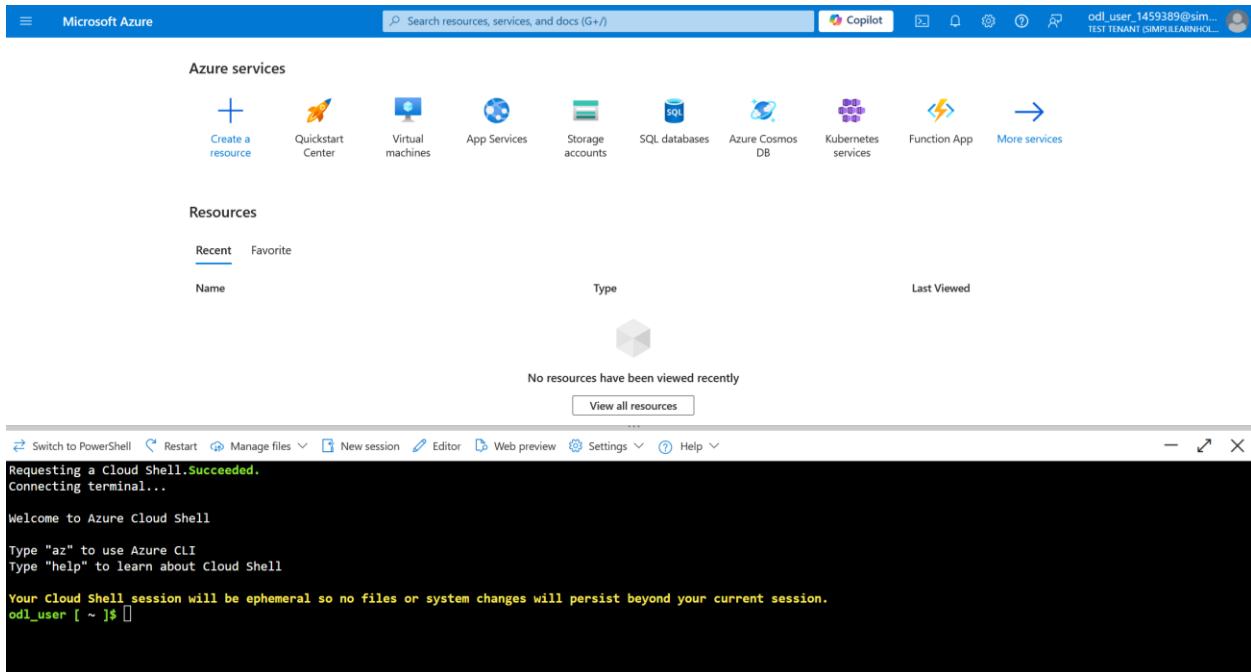


You are given two options for Azure Cloud Shell: Bash or Powershell.

For this project, we will use Bash.



Set to “No storage account required” and select your subscription. Then click Apply.



Your Cloud Shell screen should look like this for Bash. You can adjust the shell screen size.

Now create your resource group.

Use the command line:

```
az group create --name <name of resource group> --location <location name>
```

The resource group will be called “RG-01” and the location will be “eastus”

```
az group create --name RG-01 --location eastus
```

A screenshot of the Azure Cloud Shell terminal. The terminal window has a title bar with standard operating system icons. The main area of the terminal shows the output of the "az group create" command:

```
Switch to PowerShell  Restart  Manage files  New session  Editor  Web preview  Settings  Help
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell
Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
odl_user [ ~ ]$ az group create --name RG-01 --location eastus
{
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01",
  "location": "eastus",
  "managedBy": null,
  "name": "RG-01",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
odl_user [ ~ ]$
```

Check it has been created by going to the Azure Portal and type in “Resource Groups”.

Home >

Resource groups

Test tenant (simplilearnhol19.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags Group by none

You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

Filter for any field... Subscription equals all Location equals all + Add filter

Name	Subscription	Location
Cloudlabs-ACI-1459389-VM-1459389-cs209d05	Simplilearn HOL 19	East US
DefaultResourceGroup-EUS	Simplilearn HOL 19	East US
DefaultResourceGroup-EUS2	Simplilearn HOL 19	East US 2
NetworkWatcherRG	Simplilearn HOL 19	East US
ODL-azure-1459389	Simplilearn HOL 19	East US
RG-01	Simplilearn HOL 19	East US

Showing 1 - 6 of 6. Display count: 10

Give feedback

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

```
"managedBy": null,
"name": "RG-01",
"properties": {
  "provisioningState": "Succeeded"
},
"tags": null,
"type": "Microsoft.Resources/resourceGroups"
}
```

Under Resource Group, you will see your resource group has been created.

1.2 Create Virtual Network

Create a virtual network with a subnet through Azure Cloud Shell.

Use:

```
az network vnet create --name <name> vnet-1 --resource-group <resource group name> --address-prefix <CIDR IP address> --subnet-name <subnet name> --subnet-prefixes <CIDR IP address>
```

In this case, we create our own name and must use 10.0.0.0/16 to 10.0.0.0/28

Our code:

```
az network vnet create --name vnet-1 --resource-group RG-01 --address-prefix 10.20.0.0/16 --subnet-name subnet-1 --subnet-prefixes 10.20.10.0/24
```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

```
"type": "Microsoft.Resources/resourceGroups"
}
odl_user [ ~ ]$ az network vnet create \
--name vnet-1 \
--resource-group RG-01 \
--address-prefix 10.20.0.0/16 \
--subnet-name subnet-1 \
--subnet-prefixes 10.20.10.0/24
| Running ..
```

Your output in cloud shell.

```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
--address-prefix 10.20.0.0/16 \
--subnet-name subnet-1 \
--subnet-prefixes 10.20.10.0/24
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.20.0.0/16"
      ]
    },
    "enableDnsProtection": false,
    "etag": "W/\"62512422-6067-4fid-99ed-52c00fbe664c\"",
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1",
    "location": "eastus",
    "name": "vnet-1",
    "provisioningState": "Succeeded",
    "resourceGroup": "RG-01",
    "resourceGuid": "a1e17b8e-7fb0-4363-b497-5a97275de8f0",
    "subnets": [
      {
        "addressPrefix": "10.20.10.0/24",
        "delegations": [],
        "etag": "W/\"62512422-6067-4fid-99ed-52c00fbe664c\"",
        "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
        "name": "subnet-1",
        "privateEndpointNetworkPolicies": "Disabled",
        "privateLinkServiceNetworkPolicies": "Enabled",
        "provisioningState": "Succeeded",
        "resourceGroup": "RG-01",
        "type": "Microsoft.Network/virtualNetworks/subnets"
      }
    ],
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
odl_user [ ~ ]$ 

```

Check virtual network is set up by going to “Virtual Networks”.

The screenshot shows the Microsoft Azure portal interface for 'Virtual networks'. At the top, there's a search bar and a Copilot button. Below it, a table displays one item:

Name	Resource group	Location	Subscription
vnet-1	ODL-azure-1459389	East US	Simplilearn HOL 19

At the bottom, there are navigation links for 'Previous', 'Page 1 of 1', and 'Next >'. A 'Give feedback' link is also visible.

```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
--etag: "W/\"62512422-6067-4fid-99ed-52c00fbe664c\"",
--id: "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
--name: "subnet-1",
--privateEndpointNetworkPolicies: "Disabled",
--privateLinkServiceNetworkPolicies: "Enabled",
--provisioningState: "Succeeded",
--resourceGroup: "RG-01",
--type: "Microsoft.Network/virtualNetworks/subnets"
},
{
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": []
}
]
odl_user [ ~ ]$ 

```

Click on “vnet-1” and check the IP addresses of the virtual network and subnet.

The screenshot shows the Azure portal interface for managing virtual networks. On the left, a sidebar lists 'Virtual networks' and 'vnet-1'. The main area displays the 'Overview' of 'vnet-1', including details like Resource group (RG-01), Address space (10.20.0.0/16), and DNS servers. A 'Capabilities' tab is selected. At the bottom, a terminal window shows a JSON configuration snippet for creating a virtual network.

Click on the Settings blade and select “Subnet”.

The screenshot shows the 'Subnets' blade for 'vnet-1'. The 'Subnets' option is highlighted in the navigation menu. A table lists a single subnet named 'subnet-1' with an IPv4 range of 10.20.0.0/24 and 251 available IPs. The 'Subnets' blade also includes tabs for 'Gateway subnet' and 'Manage users'.

Your virtual network and subnet are ready.

1.3 Create Virtual Machines

You will create two virtual machines, a primary virtual machine and a secondary virtual machine for the backend pool.

Since you will be creating multiple VMs, use a “for” loop in bash to make more than one VM. As explained here: <https://learn.microsoft.com/en-us/cli/azure/use-azure-cli-successfully-bash>

Use the following command here.

```
for i in {1..2}
do
az vm create --resource-group RG-01 --name VM-$i --image Win2019Datacenter --admin-username
azureuser --admin-password '<Your Password>' --vnet-name vnet-1 --subnet subnet-1 --public-ip-
address-allocation static --public-ip-sku Standard --size Standard_DS1_v2 --security-type Standard --os-
disk-delete-option Delete --ultra-ssd-enabled false
done
```

Each line of code is designed to create your Virtual Machine (VM):

- --image provides the operating system for your Virtual Machine. In this case Windows 2019 Data Center.
- --admin-username and --admin-password are the administrator name and password for your VM. This part is optional as if you don't provide the line of code, it will automatically request you make one.
- --vnet-name and --subnet will link to the virtual network and subnet you created.
- --public-ip-address-allocation determines how your public IP address is allocated to your VM: static or dynamic. In this case, we choose static as the public IP address remains stable and unchanged until you release it.
- --public-ip-sku specifies the Stock Keeping Unit (SKU) of the public IP address resource. While Basic would have been fine since for most scenarios, we will use Standard to increase performance and high availability.
- --size determines the type and size of your disk for your VM. We will use Standard DS1 V2.
- --security-type sets for the security type of your Disk (Standard or Trusted Launch). In this case, we choose Standard.
- --os-disk-delete-option Delete ensures your public IP and NIC will be deleted when VM is as well.
- --enable-ultra-ssd is to enable or disable Ultra Disk compatibility. For us, we will have it set to "false" for disable.

Now enter your code into Azure CLI to create multiple VMs.

```
az user [ ~ ]$ for i in {1..2}
> do
> az vm create --resource-group RG-01 --name VM-$i --image Win2019Datacenter --admin-username azureuser --admin-password '4l+6Ordovician@' --vnet-name vnet-1 --subnet
subnet-1 --public-ip-address-allocation static --public-ip-sku Standard --size Standard_DS1_v2 --security-type Standard --os-disk-delete-option Delete --ultra-ssd-ena
bled false
> done
```

```

odl_user [ ~ ]$ for i in {1..2}
> do
> az vm create --resource-group RG-01 --name VM-$i --image Win2019Datacenter --admin-username azureuser --admin-password '41e6Ordovician@' --vnet-name vnet-1 --subnet
>   subnet-1 --public-ip-address-allocation static --public-ip-sku Standard --size Standard_DS1_v2 --security-type Standard --os-disk-delete-option Delete --ultra-ssd-ena
>   bled false
> done
{
  "fqdns": "",
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Compute/virtualMachines/VM-1",
  "location": "eastus",
  "macAddress": "00-0D-3A-8A-B0-9F",
  "powerState": "VM running",
  "privateIpAddress": "10.20.10.4",
  "publicIpAddress": "40.71.99.217",
  "resourceGroup": "RG-01",
  "zones": ""
}
{
  "fqdns": "",
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Compute/virtualMachines/VM-2",
  "location": "eastus",
  "macAddress": "00-0D-3A-17-84-12",
  "powerState": "VM running",
  "privateIpAddress": "10.20.10.5",
  "publicIpAddress": "13.92.186.34",
  "resourceGroup": "RG-01",
  "zones": ""
}
odl_user [ ~ ]$ 

```

Check your VMs are uploaded.

Home >

Virtual machines ⚡ ...

Rand Enterprises (simplilearnhol19.onmicrosoft.com)

+ Create ⚡ Switch to classic ⚡ Reservations ⚡ Manage view ⚡ Refresh ⚡ Export to CSV ⚡ Open query ⚡ Assign tags ⚡ Start ⚡ Restart ⚡ Stop ⚡ Delete ⚡ Services ⚡ Maintenance ⚡

Filter for any field... Subscription equals all Type equals all Resource group equals all Location equals all Add filter

No grouping List view

Name	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disks	...
VM-1	Simplilearn HOL 19	RG-01	East US	Running	Windows	Standard_DS1_v2	40.71.99.217	1	...
VM-1460338	Simplilearn HOL 19	ODL-azure-1460338	West US 2	Running	Windows	Standard_B4ms	40.65.106.149	1	...
VM-2	Simplilearn HOL 19	RG-01	East US	Running	Windows	Standard_DS1_v2	13.92.186.34	1	...

< Previous Page 1 of 1 Next > Give feedback

Switch to PowerShell ⚡ Restart ⚡ Manage files ⚡ New session ⚡ Editor ⚡ Web preview ⚡ Settings ⚡ Help ⚡

```

"privateIpAddress": "10.20.10.4",
"publicIpAddress": "40.71.99.217",
"resourceGroup": "RG-01",
"zones": ""
}
{
  "fqdns": "",
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Compute/virtualMachines/VM-2",
  "location": "eastus",
  "macAddress": "00-0D-3A-17-84-12",
  "powerState": "VM running",
  "privateIpAddress": "10.20.10.5",
  "publicIpAddress": "13.92.186.34",
  "resourceGroup": "RG-01",
  "zones": ""
}
odl_user [ ~ ]$ 

```

The screenshot shows the Azure portal's Virtual machines blade. A specific virtual machine, 'VM-2', is selected. The 'Overview' section displays various properties such as the resource group (RG-01), status (Running), location (East US), and public IP address (13.92.186.34). The 'JSON View' tab is open, showing the detailed configuration of the VM in JSON format.

```

{
  "resourceGroup": "RG-01",
  "zones": ""
}

{
  "fqdn": "",
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Compute/virtualMachines/VM-2",
  "location": "eastus",
  "macAddress": "00-0D-3A-17-84-12",
  "powerState": "VM running",
  "privateIpAddress": "10.20.10.5",
  "publicIpAddress": "13.92.186.34",
  "resourcegroup": "RG-01",
  "zones": ""
}
odl_user [ ~ ]$ 

```

Your Virtual Machines are now set.

Step 2: Create Load Balancer and Backend Pool

2.1 Create New Public IP address

Since we want to avoid exposing each VM and will need a public IP address for the load balancer and backend pool, remove current IPs from VMs.

First find out what your NIC name is, use:

```
az vm nic list --vm-name <VMname> --resource-group <ResourceGroup>
```

For us, use the 'for' loop method to get both:

```
for i in {1..2}
```

```
do
```

```
az vm nic list --vm-name VM-$i --resource-group RG-01
```

```
done
```

```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
[ ~ ]$ for i in {1..2}
> do
> az vm nic list --vm-name VM-$i --resource-group RG-01
> done
[
  {
    "deleteOption": null,
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-1VMNic",
    "primary": null,
    "resourcegroup": "RG-01"
  }
  [
    {
      "deleteOption": null,
      "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-2VMNic",
      "primary": null,
      "resourcegroup": "RG-01"
    }
  ]
[ ~ ]$ 

```

Find the name of the IP configuration of a network interface

`az network nic ip-config list --nic-name <VMNic> --resource-group <ResourceGroup> --out table`

Use “for” loop method again:

`for i in {1..2}`

`do`

`az network nic ip-config list --nic-name VM-${i}VMNic --resource-group RG-01 --out table`

`done`

NOTE: These steps will come in handy in Step 2.5.

Name	Primary	PrivateIPAddress	PrivateIPAddressVersion	PrivateIPAllocationMethod	ProvisioningState	ResourceGroup
ipconfigVM-1	True	10.20.10.4	IPv4	Dynamic	Succeeded	RG-01
ipconfigVM-2	True	10.20.10.5	IPv4	Dynamic	Succeeded	RG-01

Based on this information, use the following code based on the IP Name and NIC name you have received.

`for i in {1..2}`

`do`

`az network nic ip-config update --name ipconfigVM-$i --resource-group RG-01 --nic-name VM-${i}VMNic --public-ip-address null`

`done`

This will remove your current IP address for each VM.

```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
ipconfigVM-2 True 10.20.10.5 IPv4 Dynamic Succeeded RG-01
odl_user [ ~ ]$ for i in {1..2}
> do
> az network nic ip-config update --name ipconfigVM-$i --resource-group RG-01 --nic-name VM-$iVMNic --public-ip-address null
> done
{
    "etag": "W/\"895ac5ea-20b7-4d62-8e61-e547ele8aeF2\",
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-1VMNic/ipConfigurations/ipconfigVM-1",
    "name": "ipconfigVM-1",
    "primary": true,
    "privateIPAddress": "10.20.10.4",
    "privateIPAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "resourceGroup": "RG-01",
    "subnet": {
        "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
        "resourceGroup": "RG-01"
    },
    "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
{
    "etag": "W/\"07fc5785-20c7-406c-9591-cdf0528fa0e9\",
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-2VMNic/ipConfigurations/ipconfigVM-2",
    "name": "ipconfigVM-2",
    "primary": true,
    "privateIPAddress": "10.20.10.4",
    "privateIPAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "resourceGroup": "RG-01",
    "subnet": {
        "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
        "resourceGroup": "RG-01"
    },
    "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
odl_user [ ~ ]$ 

```

Double check IP addresses are removed

```

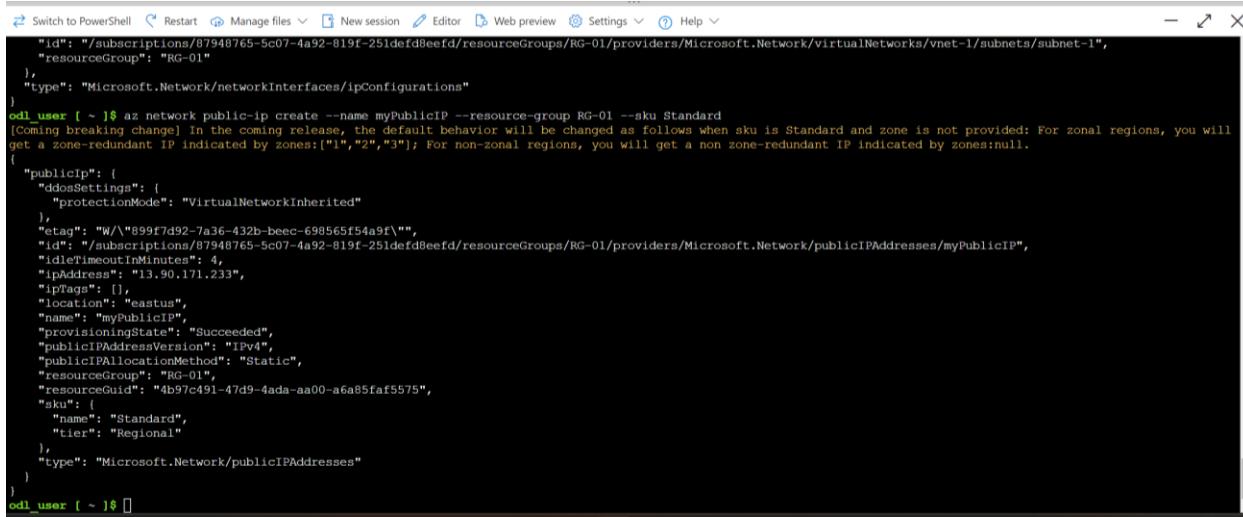
Home > Virtual machines >
Virtual machines
Rand Enterprises (simplelearnhol19.onmicrosoft.co...
+ Create < Switch to classic ...
Filter for any field...
Name ↑
VM-1 ...
VM-1460388 ...
VM-2 ...
Page 1 of 1
Search Connect Start Restart Stop Hibernate Capture Delete Refresh Open in mobile ...
VM-1
Virtual machine
Overview Activity log Access control (IAM) Tags Diagnose and solve problems
Connect Networking Network settings Load balancing Application security groups Network manager
Essentials
Resource group (move)
RG-01
Status Running
Location East US
Subscription (move)
Simplelearn HOL 19
Subscription ID 87948765-5c07-4a92-819f-251defd8eef
Tags (edit)
Operating system Windows (Windows Server 2019 Datacenter)
Size Standard DS1 v2 (1 vcpu, 3.5 GiB memory)
Public IP address
Virtual network/subnet vnet-1/subnet-1
DNS name
Health state
Time created 9/19/2024, 10:12 PM UTC
JSON View
odl_user [ ~ ]$ 

```

Now Create an Public IP address. This will be used for both your VMs and Load Balancer.

`az network public-ip create --name myPublicIP --resource-group RG-01 --sku Standard`

- Our public IP address will be named “myPublicIP”.
- --sku is the name of a public IP address SKU. Here we choose the default value Standard.



```
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
{id": "subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
},
"type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
odl_user [ ~ ]$ az network public-ip create --name myPublicIP --resource-group RG-01 --sku Standard
[Coming breaking change] In the coming release, the default behavior will be changed as follows when sku is Standard and zone is not provided: For zonal regions, you will get a zone-redundant IP indicated by zones:["1","2","3"]; For non-zonal regions, you will get a non zone-redundant IP indicated by zones:null.
{
"publicIp": {
"ddosSettings": {
"protectionMode": "VirtualNetworkInherited"
},
"etag": "W/\"899fd92-7a36-432b-becc-698565f54a9f\"",
"id": "subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/publicIPAddresses/myPublicIP",
"idleTimeoutInMinutes": 4,
"ipAddress": "13.90.171.233",
"ipTags": [],
"location": "eastus",
"name": "myPublicIP",
"provisioningState": "Succeeded",
"publicIPAddressVersion": "IPv4",
"publicIPAllocationMethod": "Static",
"resourceGroup": "RG-01",
"resourceGuid": "4b597c491-47d9-4ada-aa00-a6a85faf5575",
"sku": {
"name": "Standard",
"tier": "Regional"
},
"type": "Microsoft.Network/publicIPAddresses"
}
}
odl_user [ ~ ]$
```

You now have your own public IP address. By creating this, you can assign it to your load balancer and backend pool. And once connected to your VMs, you have your public IP attached to them as well.

2.2 Create Load Balancer

A simple procedure to create a load balancer that will distribute network traffic across multiple servers including your VMs. As we are creating a public load balancer, you will attach your public IP as your fronted IP address.

As well as that you will be creating your backend pool in the process.

Use code:

```
az network lb create --resource-group RG-01 --name LoadBalancerPrime --sku Standard --public-ip-address myPublicIP --frontend-ip-name FrontendIPPrime --backend-pool-name BackEndPoolPrime
```

- Our load balance will be named “LoadBalancerPrime” and will be attached to our public IP address “myPublicIP”.
- --frontend-ip-name comes in handing for later steps since you will know the name of your Fronted IP instead of Azure creating a default name.
- --backend-pool-name allows you to name your Backend Pool which will be created using the default settings. This will come in handy in later steps as well.

```

Switch to PowerShell ⌘ Restart Manage files New session Editor Web preview Settings Help
odl_user [ ~ ]$ az network lb create --resource-group RG-01 --name LoadBalancerPrime --sku Standard --public-ip-address myPublicIP --frontend-ip-name FrontendIPPrime --backend-pool-name BackEndPoolPrime
{
  "loadBalancer": {
    "backendAddressPools": [
      {
        "etag": "W/"b29c8967-8ac7-4af6-abe9-7c831dea094e"",
        "id": "subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackEndPoolPrime",
        "name": "BackEndPoolPrime",
        "properties": {
          "loadBalancerBackendAddresses": [],
          "provisioningState": "Succeeded"
        },
        "resourceGroup": "RG-01",
        "type": "Microsoft.Network/loadBalancers/backendAddressPools"
      }
    ],
    "frontendIPConfigurations": [
      {
        "etag": "W/"b29c8967-8ac7-4af6-abe9-7c831dea094e"",
        "id": "subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/frontendIPConfigurations/FrontendIPPrime",
        "name": "FrontendIPPrime",
        "properties": {
          "privateIPAllocationMethod": "Dynamic",
          "provisioningState": "Succeeded",
          "publicIPAddress": {
            "id": "subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/publicIPAddresses/myPublicIP",
            "resourceGroup": "RG-01"
          }
        },
        "resourceGroup": "RG-01",
        "type": "Microsoft.Network/loadBalancers/frontendIPConfigurations"
      }
    ],
    "inboundNatPools": [],
    "inboundNatRules": [],
    "loadBalancingRules": []
  }
}

```

Check if Load Balancer is up

Setting	Value
Resource group (move)	RG-01
Backend pool	BackEndPoolPrime
Location	East US
Subscription (move)	Simpleteam.HOL_19
Health probe	-
Subscriptions ID	87948765-5c07-4a92-819f-251defd8eef
NAT rules	0 inbound
SKU	Standard
Tier	Regional
Tags (edit)	
Add tags	

```

Switch to PowerShell ⌘ Restart Manage files New session Editor Web preview Settings Help
odl_user [ ~ ]$ 
  "resourceGroup": "RG-01",
  "type": "Microsoft.Network/loadBalancers/frontendIPConfigurations"
},
],
"inboundNatPools": [],
"inboundNatRules": [],
"loadBalancingRules": [],
"outboundRules": [],
"probes": [],
"provisioningState": "Succeeded",
"resourceGuid": "324fab67-b439-4f47-8692-70e7a3a3b130"
}
}
odl_user [ ~ ]$ 

```

Your load balancer is set and shows your Backend Pool as well.

2.3 Check for Fronted IP address

To check your Frontend IP address, use code:

```
az network lb frontend-ip list --resource-group RG-01 --lb-name LoadBalancerPrime --output table
```

```
odl_user [ ~ ]$ az network lb frontend-ip list --resource-group RG-01 --lb-name LoadBalancerPrime --output table
Name      PrivateIPAllocationMethod ProvisioningState ResourceGroup
FrontendIPPrime  Dynamic          Succeeded       RG-01
odl_user [ ~ ]$
```

You can check on the Azure Portal in Load Balancer.

Go to Settings > Fronted IP Configuration. Shows the IP address and indicates it's your public IP.

The screenshot shows the Azure portal interface for a load balancer named 'LoadBalancerPrime'. In the left sidebar, 'Frontend IP configuration' is selected. A table lists one item: 'FrontendIPPrime' with the IP address '13.90.171.233 (myPublicIP)' and '0 Rules count'. Below the table, a JSON representation of the resource is shown:

```
"primary": true,
"privateIPAddress": "10.20.10.5",
"privateIPAddressVersion": "IPv4",
"privateIPAllocationMethod": "Dynamic",
"provisioningState": "Succeeded",
"resourceGroup": "RG-01",
"subnet": {
  "id": "subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
  "resourceGroup": "RG-01"
},
"type": "Microsoft.Network/networkInterfaces/ipConfigurations"
```

2.4 Check for Backend Pool

Use code:

```
az network lb address-pool list -g RG-01 --lb-name LoadBalancerPrime -o table
```

```
odl_user [ ~ ]$ az network lb address-pool list -g RG-01 --lb-name LoadBalancerPrime -o table
Name      ProvisioningState ResourceGroup
BackEndPoolPrime  Succeeded       RG-01
odl_user [ ~ ]$
```

As shown before, you can check on Azure Portal as well by going to your Load Balancer and will appear in the Overview.

2.5 Attach Backend Pool to VMs

Now attach your backend pool to your VMs by managing the backend addresses of the backend address pool in the load balancer. Use the following code:

```
az network lb address-pool address add --pool-name <backend pool name> --ip-config-name <IP configuration name of VM> --nic-name <NIC name of VM> --resource-group <Resource Group> --lb-name <LoadBalancerName>
```

Considering we have two VMs and each has their own unique name for IP configuration and NIC, we will need to use an array and “for” loop in order to attach the backend pool to both.

Use the following code:

```
NICarray=( VM-1VMNic VM-2VMNic)
```

```
ipConfarray=( ipconfigVM-1 ipconfigVM-2)
```

- Arrays for the following NIC and IP configurations of each VM as given in Step 2.1
- Then the “for loop”:

```
for index in "${!NICarray[@]}"
```

```
do
```

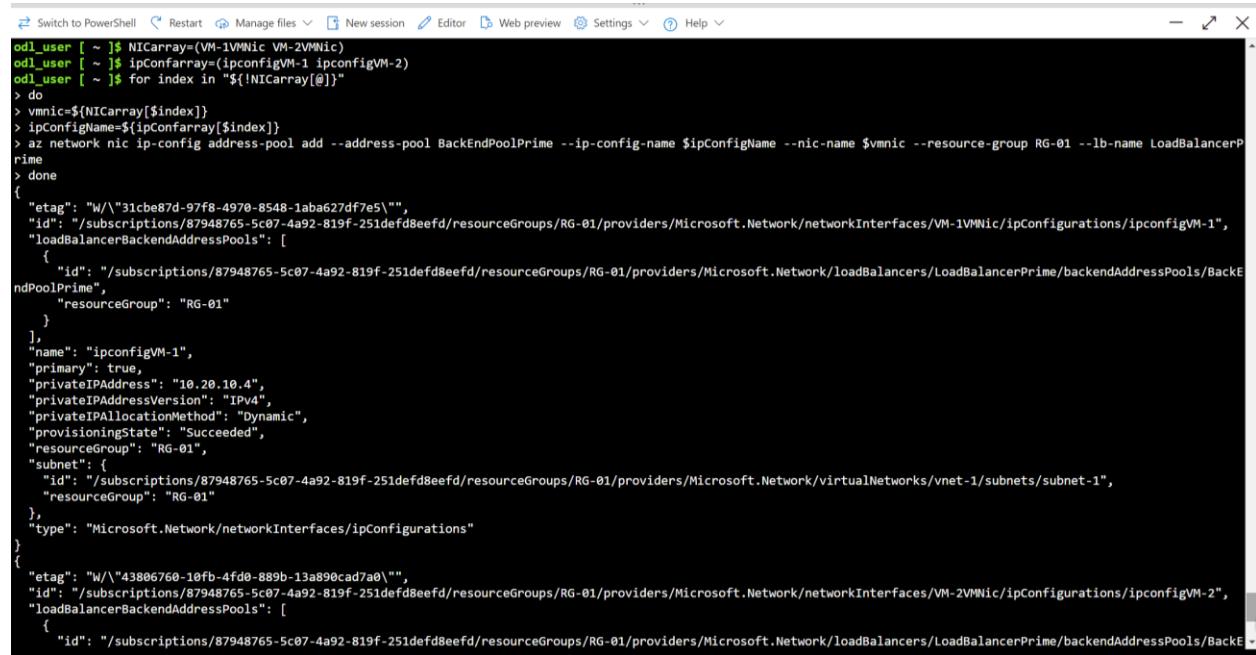
```
    vmnic=${NICarray[$index]}
```

```
    ipConfigName=${ipConfarray[$index]}
```

```
    az network nic ip-config address-pool add --address-pool BackEndPoolPrime --ip-config-name $ipConfigName --nic-name $vmnic --resource-group RG-01 --lb-name LoadBalancerPrime
```

```
done
```

The screenshot shows the order in which to follow and the output of the results.



```
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
odl_user [ ~ ]$ NICarray=( VM-1VMNic VM-2VMNic)
odl_user [ ~ ]$ ipConfarray=( ipconfigVM-1 ipconfigVM-2)
odl_user [ ~ ]$ for index in "${!NICarray[@]}"
> do
>     vmnic=${NICarray[$index]}
>     ipConfigName=${ipConfarray[$index]}
>     az network nic ip-config address-pool add --address-pool BackEndPoolPrime --ip-config-name $ipConfigName --nic-name $vmnic --resource-group RG-01 --lb-name LoadBalancerPrime
> done
{
    "etag": "W/\"31cbe87d-97f8-4970-8548-1aba627df7e5\"",
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-1VMNic/ipConfigurations/ipconfigVM-1",
    "loadBalancerBackendAddressPools": [
        {
            "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackEndPoolPrime",
            "resourceGroup": "RG-01"
        }
    ],
    "name": "ipconfigVM-1",
    "primary": true,
    "privateIPAddress": "10.20.10.4",
    "privateIPAddressVersion": "IPv4",
    "privateIPAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "resourceGroup": "RG-01",
    "subnet": {
        "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
        "resourceGroup": "RG-01"
    },
    "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
{
    "etag": "W/\"43806760-10fb-4fd0-889b-13a890cad7a0\"",
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/networkInterfaces/VM-2VMNic/ipConfigurations/ipconfigVM-2",
    "loadBalancerBackendAddressPools": [
        {
            "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackEndPoolPrime"
        }
    ]
}
```

```

    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef0/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackendPoolPrime",
    "resourceGroup": "RG-01"
  ],
  "name": "ipconfigVM-2",
  "primary": true,
  "privateIPAddress": "10.20.10.5",
  "privateIPAddressVersion": "IPv4",
  "privateIPAllocationMethod": "Dynamic",
  "provisioningState": "Succeeded",
  "resourceGroup": "RG-01",
  "subnet": {
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef0/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
    "resourceGroup": "RG-01"
  },
  "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
odl_user [ ~ ]$ 

```

Check on Azure Portal.

Select one of your Virtual Machines. In this case 'VM-1'.

In Overview, we see in Networking the public IP address and indicates from the load balancer.

Then go Networking>Load balancing.

VM-1 | Load balancing

Network interface / IP configuration
VM-1VMNic (primary) / ipconfigVM-1 (primary)

Name	Type	Frontend IP address	Frontend DNS address	Backend pool
LoadBalancerPrime	Load balancer	-	-	BackEndPoolPrime

```
"primary": true,
"privateIPAddress": "10.20.10.5",
"privateIPAddressVersion": "IPv4",
"privateIPAllocationMethod": "Dynamic",
"provisioningState": "Succeeded",
"resourceGroup": "RG-01",
"subnet": {
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eef/resourceGroups/RG-01/providers/Microsoft.Network/virtualNetworks/vnet-1/subnets/subnet-1",
  "resourceGroup": "RG-01"
},
"type": "Microsoft.Network/networkInterfaces/ipConfigurations"
```

We can see VM-1 is connected to the load balancer, public IP, and Fronted IP address

And Check on Load Balancer.

Go to Settings > Backend Pools

LoadBalancerPrime | Backend pools

The backend pool is a critical component of the load balancer. The backend pool defines the group of resources that will serve traffic for a given load-balancing rule. [Learn more](#)

Backend pool	Resource Name	IP address	Network interface	Availability zone	Rules count	Resource Status	Admin state
BackEndPoolPrime (2)	VM-1	10.20.10.4	VM-1VMNic	-	1	Running	None
BackEndPoolPrime	VM-2	10.20.10.5	VM-2VMNic	-	1	Running	None

```
{"resourceGroup": "RG-01",
"protocol": "Tcp",
"provisioningState": "Succeeded",
"resourceGroup": "RG-01",
"type": "Microsoft.Network/loadBalancers/loadBalancingRules"}
```

Shows backend pool is connected to two virtual machines.

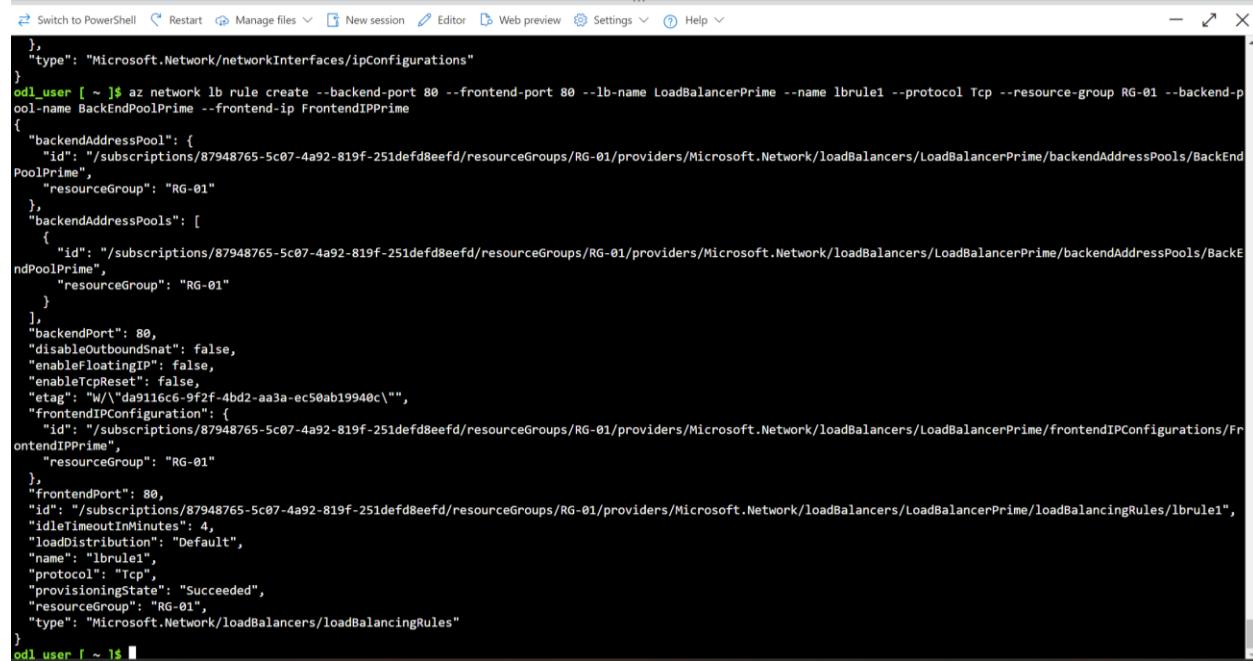
2.6 Create Load Balancer Rule

Create a load balancing rule in order to configure incoming traffic to the backend pool.

Use the code:

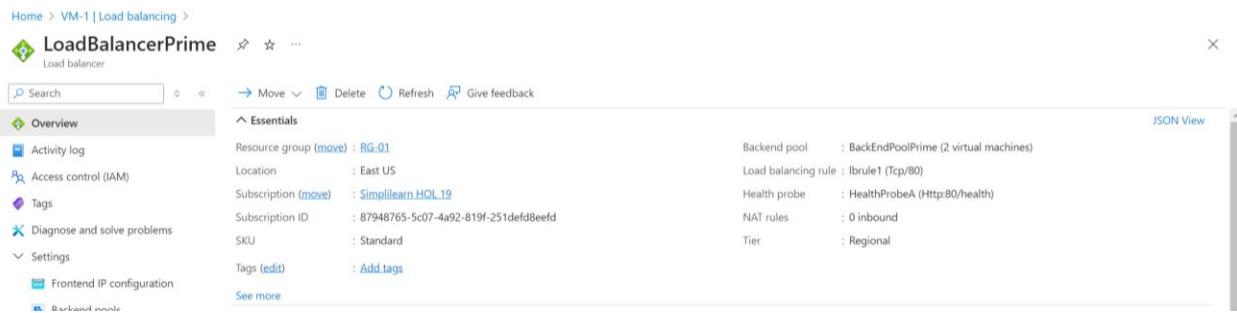
```
az network lb rule create --backend-port 80 --frontend-port 80 --lb-name LoadBalancerPrime --name lbrule1 --protocol Tcp --resource-group RG-01 --backend-pool-name BackEndPoolPrime --frontend-ip FrontendIPPrime
```

- `--backend-port` is used for internal connections on the endpoint.
- `--frontend-port` is for the external endpoint.
- `--protocol` refers to the transport protocol used by the load balancing rule. In this case, we will use `Tcp`.



```
az network lb rule create --backend-port 80 --frontend-port 80 --lb-name LoadBalancerPrime --name lbrule1 --protocol Tcp --resource-group RG-01 --backend-pool-name BackEndPoolPrime --frontend-ip FrontendIPPrime
```

On Azure Portal for Load Balancer.



LoadBalancerPrime

Resource group : RG-01

Location : East US

Backend pool : BackEndPoolPrime (2 virtual machines)

Load balancing rule : lbrule1 (Tcp/80)

Health probe : HealthProbeA (Http:80/health)

NAT rules : 0 inbound

Tier : Regional

Load Balancer Rule is set.

Step 3: Health Probe

A health probe detects the health status of your application instances such as your load balancer and VMs and detect application failures, manage load, and plan for downtime. The health probe can be

configured to use different protocols such as TCP, HTTP, or HTTPS. Azure Load Balancer rules require a health probe to detect the endpoint status where it determines which backend pool instances receive new connections and possible failures.

Creating a health probe will be important and the last step to the project.

3.1 Create Health Probe

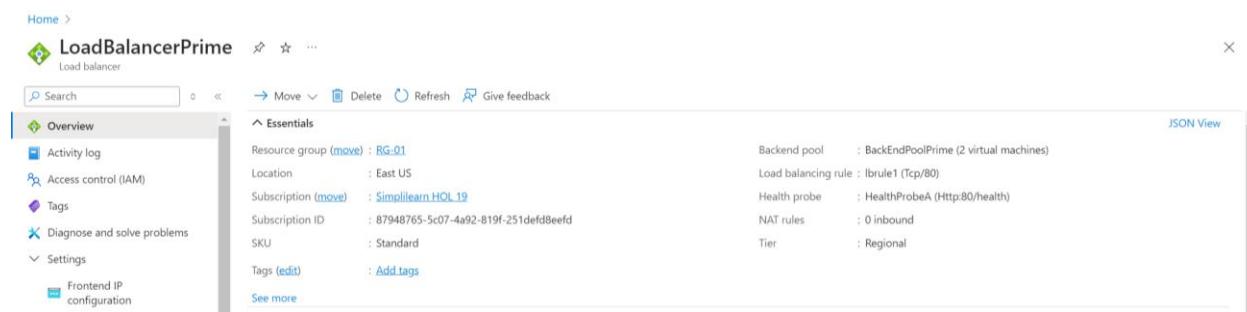
Create a probe for your load balancer and designate it as a health probe. Use the following code:

```
az network lb probe create --lb-name LoadBalancerPrime --name HealthProbeA --port 80 --protocol Http  
--resource-group RG-01 --interval 30 --path /health --probe-threshold 5
```

- --port is the port for communicating the probe.
- --protocol is the protocol of the end point. In this case, we choose Http.
- --interval is the interval of time on how frequently to probe the endpoint for health status. The default time unit is in seconds and here we have the interval set to 30 seconds.
- --path is the Uniform Resource Identifier (URI) used for requesting health status from the VM. Considering we have protocol set to http, the path is required and is given a location at "/health".
- --probe-threshold is the number of consecutive successful or failed probes to allow or deny traffic from being delivered to this endpoint. In this case, we give it 5.

```
odl_user [ ~ ]$ az network lb probe create --lb-name LoadBalancerPrime --name HealthProbeA --port 80 --protocol Http --resource-group RG-01 --interval 30 --path /health --probe-threshold 5
{
  "etag": "W/\"6a86c92e-4186-47a8-8636-2074ce3a5e5c\"",
  "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/probes/HealthProbeA",
  "intervalInSeconds": 30,
  "name": "HealthProbeA",
  "numberOfProbes": 2,
  "port": 80,
  "probeThreshold": 5,
  "protocol": "Http",
  "provisioningState": "Succeeded",
  "requestPath": "/health",
  "resourceGroup": "RG-01",
  "type": "Microsoft.Network/loadBalancers/probes"
}
odl_user [ ~ ]$
```

Check Health Probe is set up.



Setting	Value
Resource group	RG-01
Location	East US
Subscription	Simplilearn HOL_19
Subscription ID	87948765-5c07-4a92-819f-251defd8eefd
SKU	Standard
Tags (edit)	Add.tags
Backend pool	BackEndPoolPrime (2 virtual machines)
Load balancing rule	lbrule1 (Tcp:80)
Health probe	HealthProbeA (Http:80/health)
NAT rules	0 inbound
Tier	Regional

Health probe appears on the Overview of your Load Balancer.

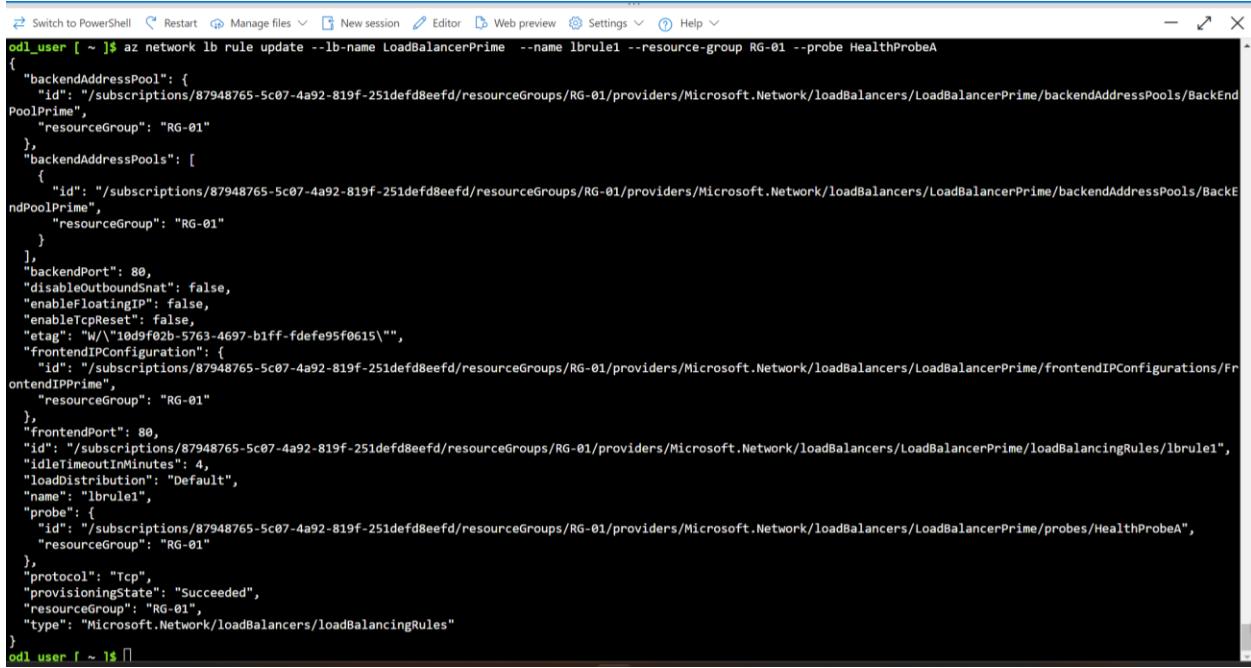
3.2 Connect Health Probe to Load Balancer Rule

Update your load balancer rule to connect your health probe.

Use code:

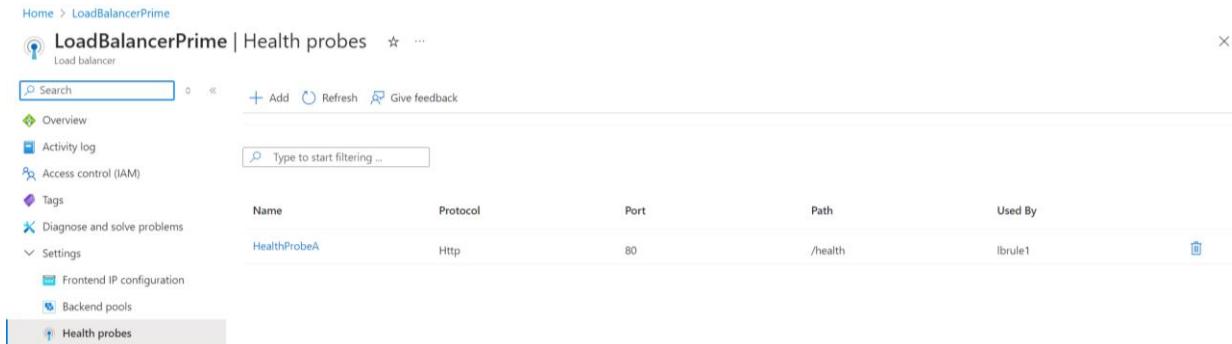
```
az network lb rule update --lb-name LoadBalancerPrime --name lbrule1 --resource-group RG-01 --probe HealthProbeA
```

- --probe is the name of an existing probe to associate with this rule, in this case “HealthProbeA”.



```
Switch to PowerShell ⌘ Restart Manage files New session Editor Web preview Settings Help
odl_user [ ~ ]$ az network lb rule update --lb-name LoadBalancerPrime --name lbrule1 --resource-group RG-01 --probe HealthProbeA
{
  "backendAddressPool": {
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackendPoolPrime",
    "resourceGroup": "RG-01"
  },
  "backendAddressPools": [
    {
      "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/backendAddressPools/BackendPoolPrime",
      "resourceGroup": "RG-01"
    }
  ],
  "backendPort": 80,
  "disableOutboundSnat": false,
  "enableFloatingIP": false,
  "enableTcpReset": false,
  "etag": "W/"10d9f02b-5763-4697-b1ff-fdefe95f0615\"",
  "frontendIPConfiguration": {
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/frontendIPConfigurations/FrontendIPPrime",
    "resourceGroup": "RG-01"
  },
  "frontendPort": 80,
  "idleTimeoutInMinutes": 4,
  "loadDistribution": "Default",
  "name": "lbrule1",
  "probe": {
    "id": "/subscriptions/87948765-5c07-4a92-819f-251defd8eefd/resourceGroups/RG-01/providers/Microsoft.Network/loadBalancers/LoadBalancerPrime/probes/HealthProbeA",
    "resourceGroup": "RG-01"
  },
  "protocol": "Tcp",
  "provisioningState": "Succeeded",
  "resourceGroup": "RG-01",
  "type": "Microsoft.Network/loadBalancers/loadBalancingRules"
}
odl_user [ ~ ]$
```

Check on Azure Portal. In your Load Balancer, go to Settings > Health Probes.



The screenshot shows the Azure portal interface for managing a Load Balancer named 'LoadBalancerPrime'. The 'Health probes' section is selected in the navigation menu. A table lists one health probe:

Name	Protocol	Port	Path	Used By
HealthProbeA	Http	80	/health	lbrule1

The Health Probe is fully set.

Your entire project is all set to provide highly available architecture and healthy service instances in virtual machines in a load-balanced set.

Finish