Implement, manage, and monitor on Azure environment

Problem Statement 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

Course-end Project 2

Description

To create high 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

Description: The Rand Enterprises Corporation wants to deploy a web application in a highly available environment so that only the healthy instances will be serving the traffic so end users will not be facing any downtime. They have decided to work on an Azure public load balancer to implement the functionality.

The operations team at Rand decides to define the entire architecture using the load balancer and its backend pool, once that's in place they intend to create the frontend IP and health probe along with virtual machines housing their application.

Rand Enterprises works extensively on delivering highly available web applications for their users in a secure way by avoiding directly exposing the virtual machines hosting the applications to the public internet. The communication from the application in the VM to the end-user must take place via the Load Balancer.

The expectation of the operation team is to create a reusable method that can be used for automation if in the future we need to deploy the same kind of infrastructure. So, rather than deploying resources in the Azure portal, they should leverage the command-line interface to deploy the resources so that in the future these commands can be used

As a security measure, you need to ensure that only the health instances of the virtual machine will be serving the traffic.

Tools required: Azure account with administrator access

Prerequisites: None

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

- Extend the security with the bastion Hosts

Solution

1. Login to VM and through VM use CLI to login to Azure
2. Create a Resource Group for RandEnterprise
   `az group create --name RandEnterprises-RG --location eastus`
3. Create a Virtual Network and Subnet

```
az network vnet create --resource-group RandEnterprises-RG --name RandVNet --address-prefix 10.0.0.0/16 --subnet-name AppSubnet --subnet-prefix 10.0.1.0/24
```

> **Note**
>
> This command creates a new Virtual Network (VNet) in Azure. Here's what each part means:
>
> 1. az network vnet create: This is the basic command to create a new VNet.
>
> 2. --resource-group RandEnterprises-RG: This specifies which resource group to create the VNet in.
>
> - --name RandVNet: This sets the name of the new VNet to "RandVNet".
> - --address-prefix 10.0.0.0/16: This defines the IP address range for the entire VNet. It allows for 65,536 possible IP addresses (10.0.0.0 to 10.0.255.255).
> - --subnet-name AppSubnet: This creates a subnet within the VNet named "AppSubnet".
> - --subnet-prefix 10.0.1.0/24: This defines the IP range for the subnet. It allows for 256 possible IP addresses (10.0.1.0 to 10.0.1.255) within the larger VNet range.
>
> In essence, this command sets up a networking space in Azure where you can deploy your virtual machines and other resources, with a specific range of IP addresses available for use

4. Create a Public IP for the Load Balancer

```
az network public-ip create --resource-group RandEnterprises-RG --name RandLB-PIP --sku Standard
```

5. Create the load balancer

```
az network lb create --resource-group RandEnterprises-RG --name RandLoadBalancer --sku Standard --public-ip-address RandLB-PIP --frontend-ip-name RandLBFrontend --backend-pool-name RandLBBackendPool
```

6. Create a Health Probe

```
az network lb probe create --resource-group RandEnterprises-RG --lb-name RandLoadBalancer --name RandHealthProbe --protocol tcp --port 80
```

7. Create a LB rule

```
az network lb rule create --resource-group RandEnterprises-RG --lb-name RandLoadBalancer --name RandLBRule --protocol tcp --frontend-port 80 --backend-port 80 --frontend-ip-name RandLBFrontend --backend-pool-name RandLBBackendPool --probe-name RandHealthProbe
```

8. Create a Network Security Group

```
az network nsg create --resource-group RandEnterprises-RG --name RandNSG
```

9. Create NSG Rule to allow traffic from Load Balancer

```
az network nsg rule create --resource-group RandEnterprises-RG --nsg-name RandNSG --name AllowHTTP --priority 100 --protocol tcp --destination-port-range 80 --access allow
```

10. Create VM

```
az vm create --resource-group RandEnterprises-RG --name RandVM1 --image UbuntuLTS --admin-username azureuser --generate-ssh-keys --vnet-name RandVNet --subnet AppSubnet --nsg RandNSG --public-ip-address "" --custom-data cloud-init.txt
```

11. Add VMs to the backend pool

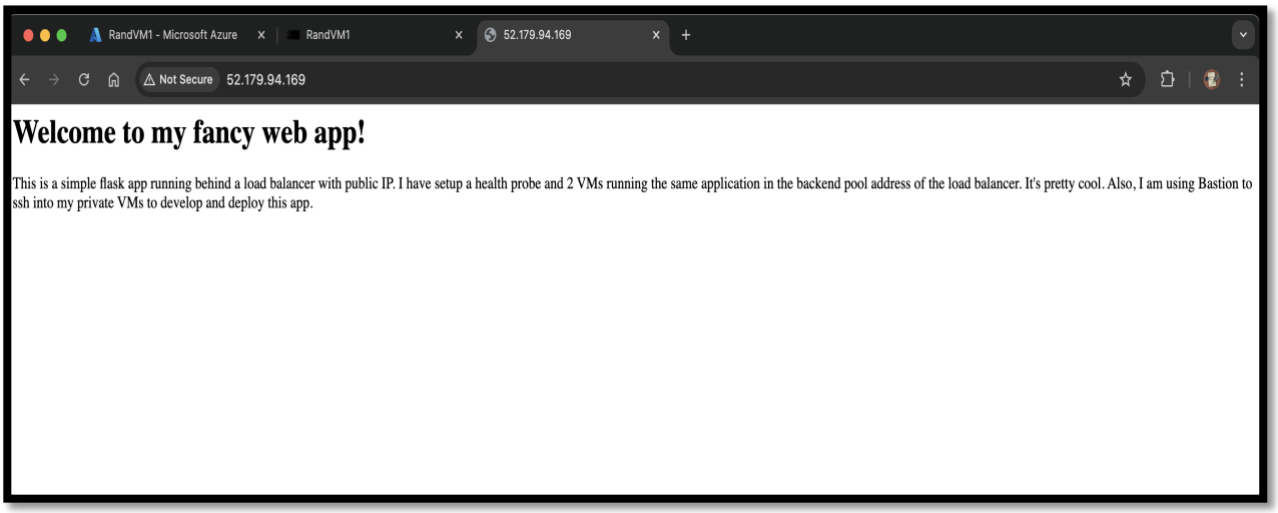
```
az network nic ip-config address-pool add --address-pool RandLBBackendPool --ip-config-name ipconfig1 --nic-name RandVM1VMNic --resource-group RandEnterprises-RG --lb-name RandLoadBalancer
```

12. Use the azure portal to connect to VM using bastion host

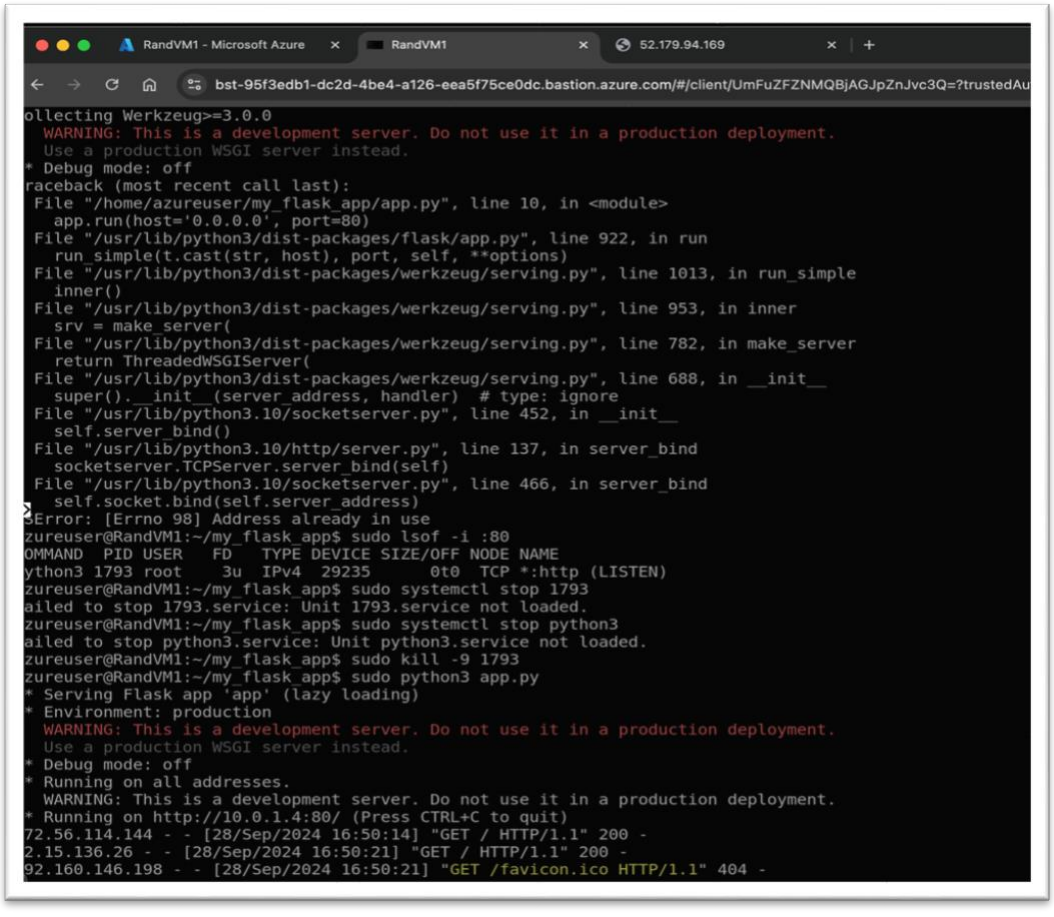
> **Note**
>
> During the process of configuring the Azure Bastion subnet, I encountered some challenges while utilizing the Azure Command-Line Interface (CLI). As a result, I opted to use the Azure portal to establish a connection to the virtual machine (VM).
>
> Once connected, I proceeded to install Python and the necessary libraries to support the application. Subsequently, I initiated a Flask application configured to listen on port 80.

To verify the deployment, I accessed the web application using the public IP address of the Load Balancer. The accompanying screenshots illustrate the complete setup, as well as the successful hosting of the web application on the Load Balancer's public IP.

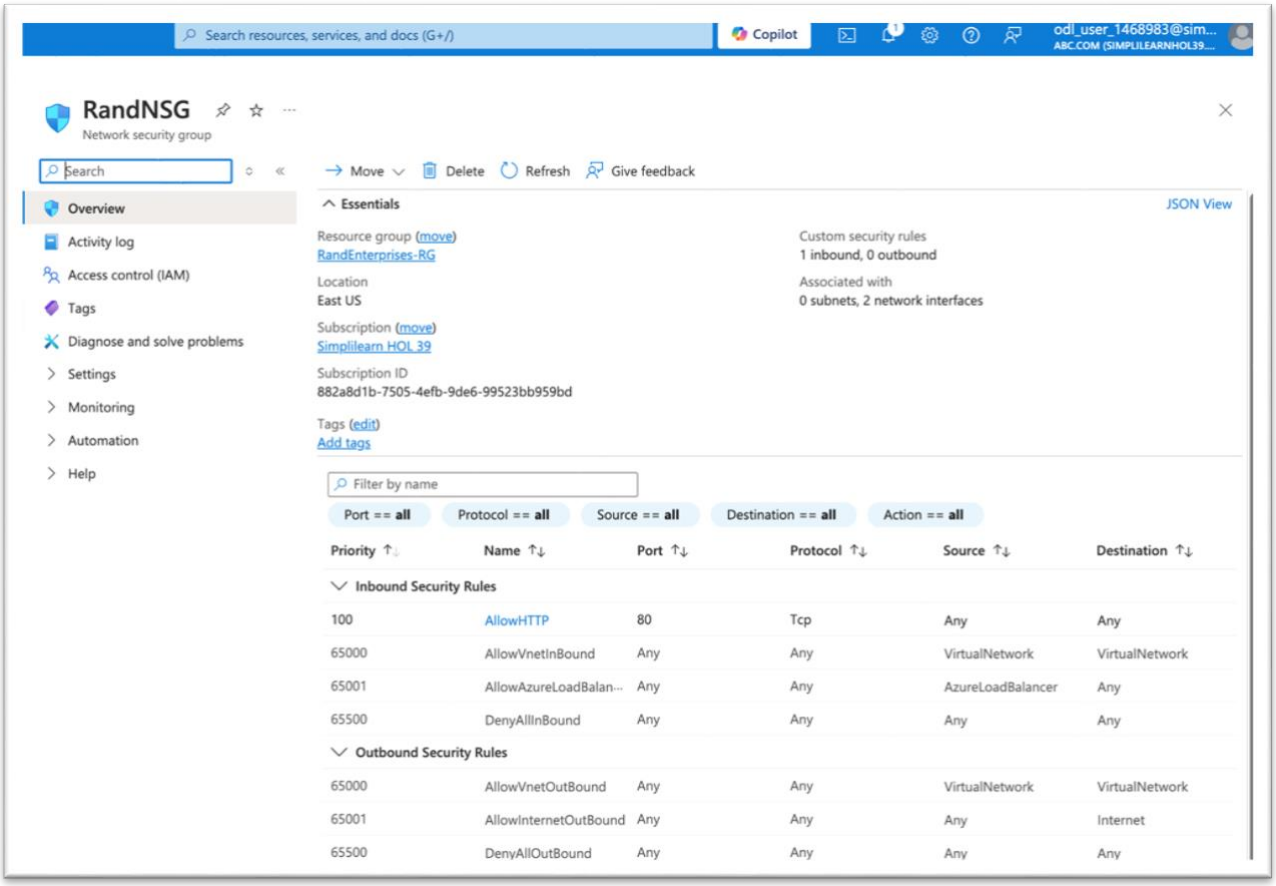
Final Web Application on the Public IP address of the Load Balancer

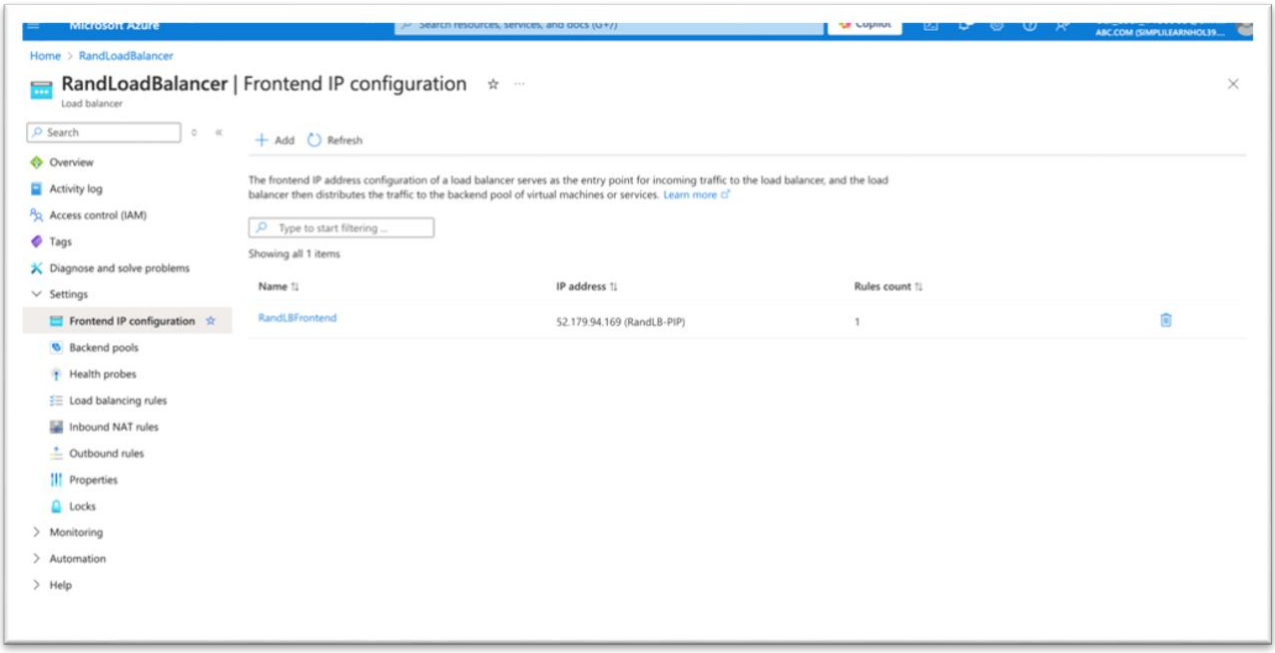


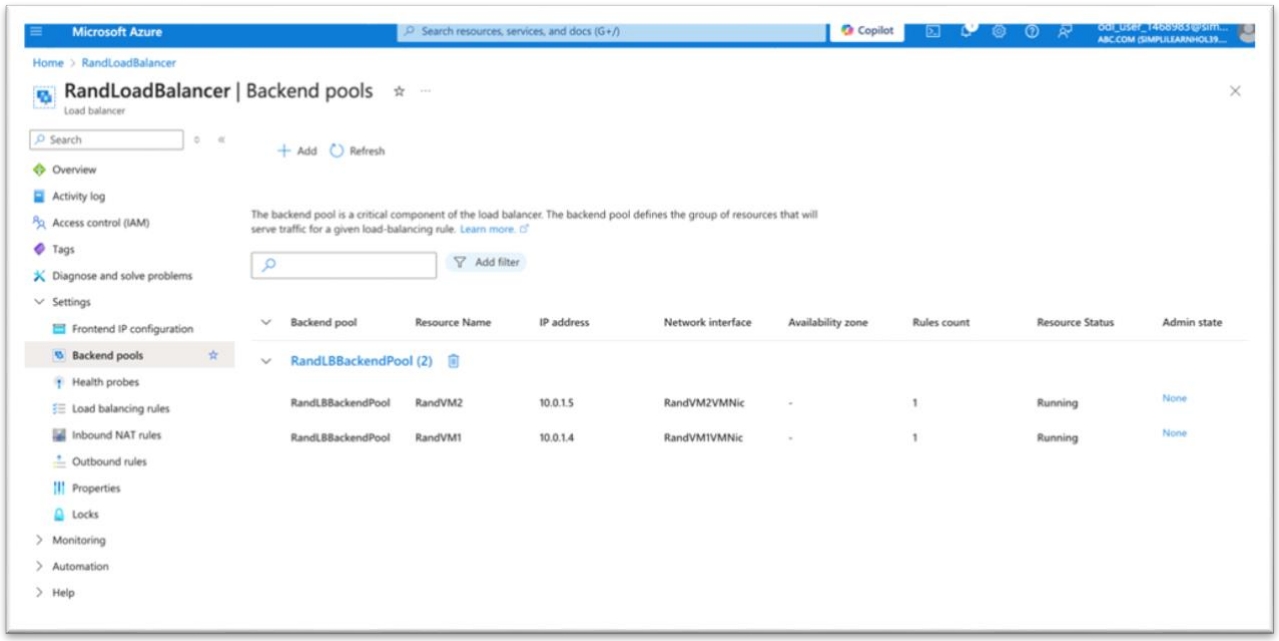
SSH into VMs and deploy a flask app

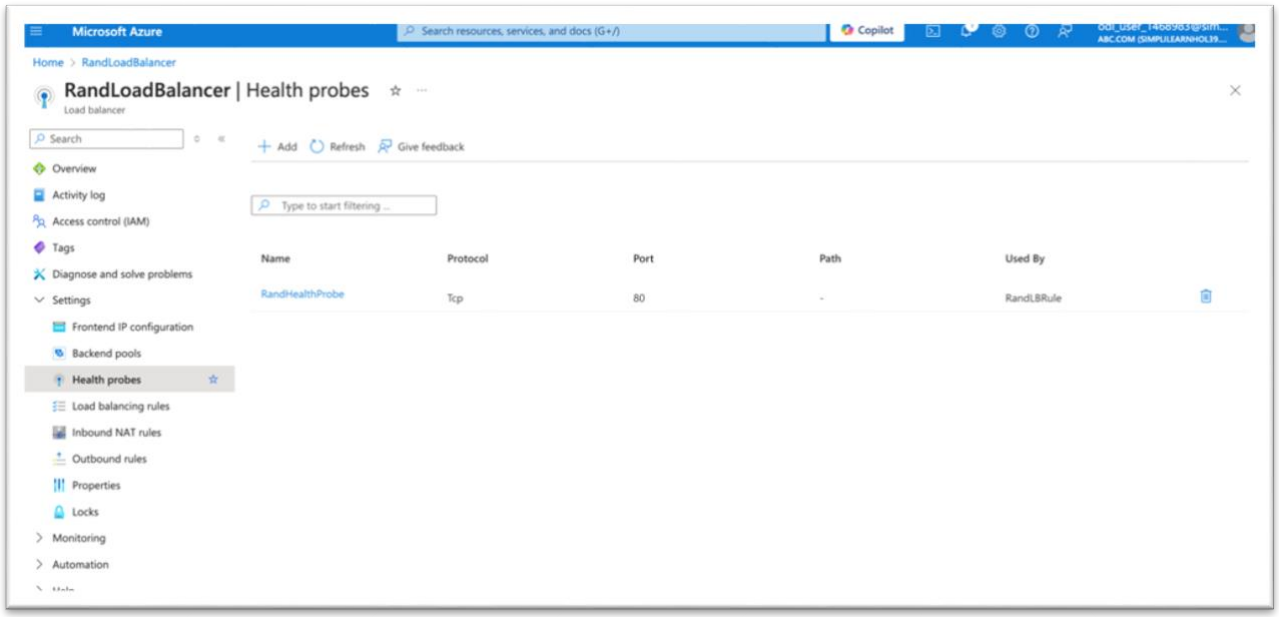


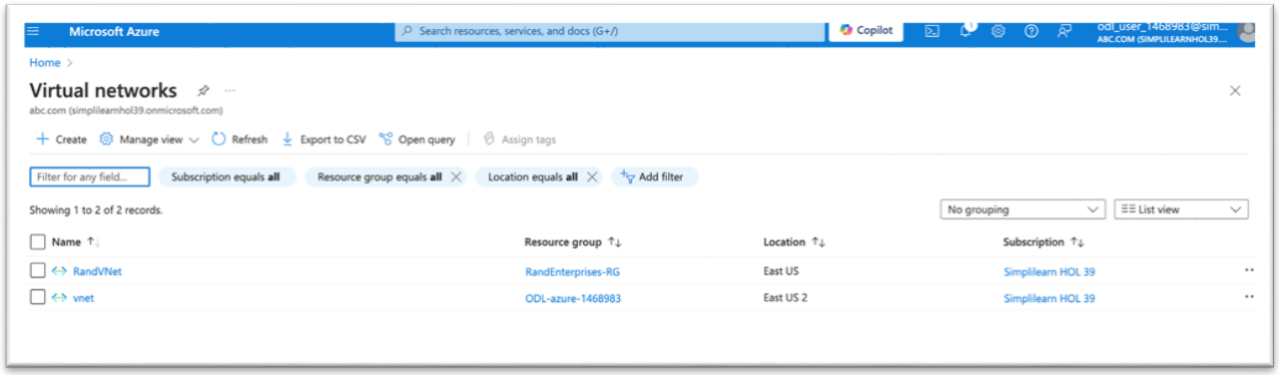
Creation of VNET, NSG, LB, VMs, Pool, Health Probe, RG

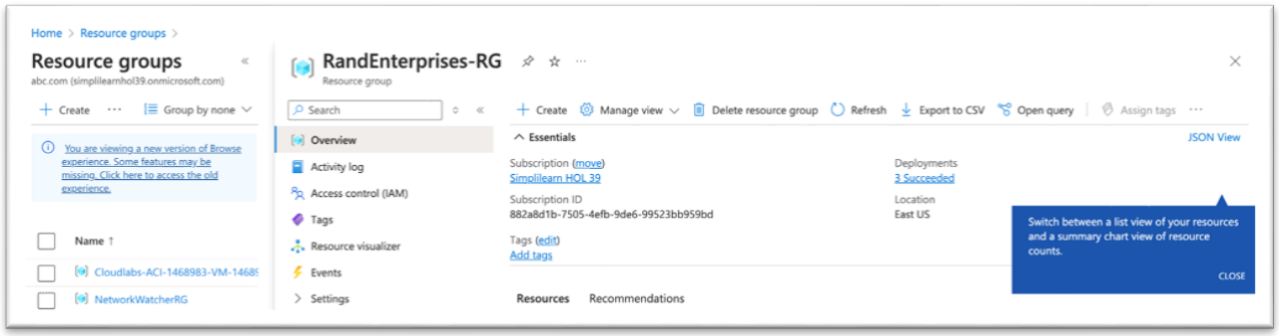


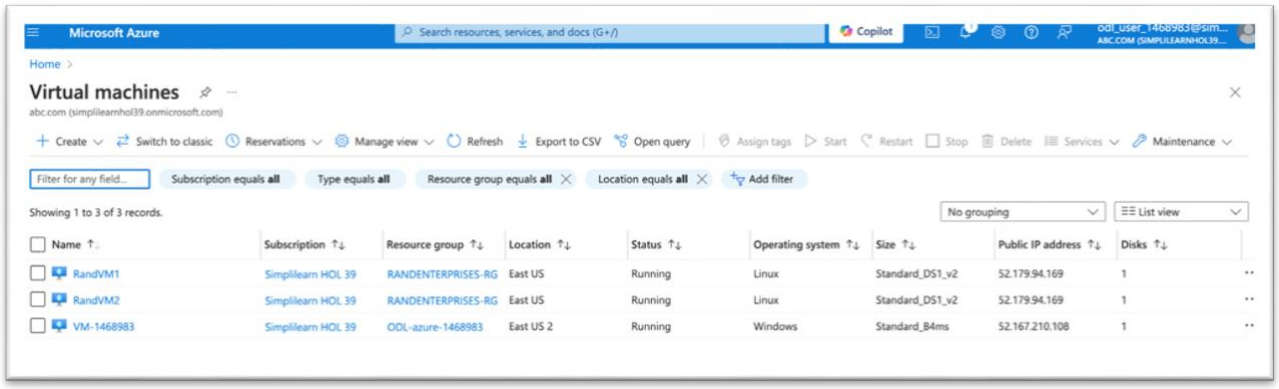



Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > RandLoadBalancer
RandLoadBalancer | Backend pools
Load balancer
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
Health probes
Load balancing rules
Inbound NAT rules
Outbound rules
Properties
Locks
Monitoring
Automation
Help
Add
Refresh
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.
Add filter
Backend pool
Resource Name
IP address
Network interface
Availability zone
Rules count
Resource Status
Admin state
RandLBBackendPool (2)
RandLBBackendPool RandVM2 10.0.1.5 RandVM2VMNic - 1 Running None
RandLBBackendPool RandVM1 10.0.1.4 RandVM1VMNic - 1 Running None


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > RandLoadBalancer
RandLoadBalancer | Health probes
Load balancer
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
Health probes
Load balancing rules
Inbound NAT rules
Outbound rules
Properties
Locks
Monitoring
Automation
Add
Refresh
Give feedback
Type to start filtering ...
Name
Protocol
Port
Path
Used By
RandHealthProbe Tcp 80 - RandLBRule


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home >
Virtual networks
abc.com (simplilearnhol39.onmicrosoft.com)
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Filter for any field...
Subscription equals all
Resource group equals all
Location equals all
Add filter
Showing 1 to 2 of 2 records.
No grouping
List view
Name
Resource group
Location
Subscription
RandVNet RandEnterprises-RG East US Simplilearn HOL 39
vnet ODL-azure-1468983 East US 2 Simplilearn HOL 39


Home > Resource groups >
Resource groups
abc.com (simplilearnhol39.onmicrosoft.com)
Create
Group by none
You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.
Name ↑
Cloudlabs-ACI-1468983-VM-1468
NetworkWatcherRG
RandEnterprises-RG
Resource group
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Events
Settings
Create
Manage view
Delete resource group
Refresh
Export to CSV
Open query
Assign tags
Essentials
JSON View
Subscription (move)
Simplilearn HOL 39
Subscription ID
882a8d1b-7505-4efb-9de6-99523bb959bd
Tags (edit)
Add tags
Deployments
3 Succeeded
Location
East US
Switch between a list view of your resources and a summary chart view of resource counts.
CLOSE
Resources
Recommendations


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home >
Virtual machines
abc.com (simplilearnhol39.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
Showing 1 to 3 of 3 records.
No grouping
List view
Name
Subscription
Resource group
Location
Status
Operating system
Size
Public IP address
Disks
RandVM1 Simplilearn HOL 39 RANDENTERPRISES-RG East US Running Linux Standard_DS1_v2 52.179.94.169 1
RandVM2 Simplilearn HOL 39 RANDENTERPRISES-RG East US Running Linux Standard_DS1_v2 52.179.94.169 1
VM-1468983 Simplilearn HOL 39 ODL-azure-1468983 East US 2 Running Windows Standard_B4ms 52.167.210.108 1