

# Hands-On Objective: Azure Virtual Machine Scale Set Deployment

## Objective

The objective of this hands-on exercise is to create an Azure Virtual Machine Scale Set (VMSS) and deploy a custom application automatically using a bootstrap (custom data) script. The task also includes configuring network security rules to allow HTTP and SSH access, verifying application accessibility through a web browser, and cleaning up all Azure resources after completion.

---

## Goals of the Hands-On

1. Create an Azure resource group
  2. Create and configure a Virtual Machine Scale Set
  3. Deploy a custom application using a startup script
  4. Configure firewall rules for ports 22 and 80
  5. Access the deployed application via a web browser
- 

## A) Creation of Resource Group

The Azure portal was accessed using a valid Azure account.

From the Azure search bar, **Resource Groups** was selected and a new resource group was created with the following details:

- **Subscription:** Free Trial / Pay-As-You-Go
- **Resource Group Name:** GL-VMSS-RG
- **Region:** Default

The resource group was reviewed and created successfully.

---

## B) Creation of Virtual Machine Scale Set – Basic Configuration

A new Virtual Machine Scale Set was created with the following configuration:

- **Subscription:** Free Trial
- **Resource Group:** GL-VMSS-RG
- **VM Scale Set Name:** User-defined
- **Region:** East US
- **Availability Zones:** All available zones
- **Image:** Ubuntu 20.04
- **Authentication Type:** SSH public key
- **Username:** azureuser
- **SSH Key:** New key pair generated

Default disk settings were used.

---

### C) Networking Configuration

The networking configuration was set up to enable external access and load balancing:

- A new virtual network was created
- Load balancer was enabled
- A new load balancer and backend pool were configured

This ensures incoming traffic is distributed across all VM instances.

---

### D) Scaling Configuration

The scale set was configured to support elasticity:

- **Initial Instance Count:** 2
- **Scaling Policy:** Custom
- **Minimum Instances:** 2
- **Maximum Instances:** 5

This configuration ensures availability while controlling costs.

---

## E) Health Monitoring Configuration

Application health monitoring was enabled to maintain high availability:

- **Protocol:** HTTP
- **Port:** 80
- **Health Probe Path:** /health.html

Only healthy VM instances receive traffic from the load balancer.

---

## F) Advanced Configuration – Custom Data Script

A bootstrap script was added under the **Custom Data** section to automate application deployment during VM startup.

### Custom Data Script Used

```
#!/bin/bash
```

```
APP_NAME=LiftShift-Application
```

```
apt update -y && apt -y install python3-pip zip
```

```
cd /opt
```

```
wget https://d6opu47qoi4ee.cloudfront.net/loadbalancer/simuapp-v1.zip
```

```
unzip simuapp-v1.zip
```

```
rm -f simuapp-v1.zip
```

```
sed -i "s=MOD_APPLICATION_NAME=$APP_NAME=g" templates/index.html
```

```
pip3 install -r requirements.txt
```

```
nohup python3 simu_app.py >> application.log 2>&1 &
```

### **Purpose of the Script**

- Updates system packages
- Installs required dependencies
- Downloads and extracts the application files
- Configures application settings
- Starts the application automatically in the background

This ensures consistent deployment across all VM instances.

---

### **G) Network Security Configuration**

Inbound security rules were configured to allow required traffic.

#### **Inbound Rule – HTTP**

- **Port:** 80
- **Action:** Allow
- **Priority:** 100
- **Name:** Port\_80

#### **Inbound Rule – SSH**

- **Port:** 22
- **Action:** Allow
- **Priority:** 110
- **Name:** Port\_22

These rules enable web access and secure remote administration.

---

## H) Application Access

The load balancer created with the VM Scale Set was accessed from the Azure portal. The public IP address associated with the load balancer was copied and pasted into a web browser.

The custom application was successfully displayed in the browser.

---

## Resource Cleanup

After verifying the application, all resources were deleted to avoid unnecessary costs:

- Navigated to **Resource Groups**
- Selected **GL-VMSS-RG**
- Deleted the resource group

This action removed all associated Azure resources.

---

## Result

The Azure Virtual Machine Scale Set was successfully created and configured. The custom application was deployed automatically using a bootstrap script, made accessible through a load balancer, and scaled across multiple VM instances.

---

## Conclusion

This hands-on exercise demonstrated the use of Azure Virtual Machine Scale Sets for deploying scalable, highly available applications using automation, load balancing, and health monitoring.

---