Here's a **project implementation guide** for **Azure Multi-VM Architecture with Load Balancer** along with a **GitHub-ready project description** including **modern emojis** for better engagement:

✅ Project: Azure Multi-VM Architecture with Load Balancer

📌 Overview:

This project demonstrates how to deploy a **high-availability architecture** on **Microsoft Azure** using:

- Multiple Virtual Machines (VMs) 🖥️
- Azure Load Balancer ⚖️
- Virtual Network (VNet) 🌐
- Network Security Groups (NSG) 🔐
  to balance traffic across the VMs for improved **scalability**, **performance**, and **fault tolerance**.

📝 Project Steps:

1️⃣ Create Resource Group

```
az group create --name MyResourceGroup --location eastus
```

2️⃣ Create Virtual Network & Subnet

```
az network vnet create \
  --resource-group MyResourceGroup \
  --name MyVNet \
  --subnet-name MySubnet
```

3️⃣ Create Network Security Group & Rules

```
az network nsg create --resource-group MyResourceGroup --name MyNSG


az network nsg rule create --resource-group MyResourceGroup --nsg-name MyNSG \
  --name AllowHTTP --protocol tcp --direction inbound --priority 100 \
  --source-address-prefix '*' --source-port-range '*' \
  --destination-address-prefix '*' --destination-port-range 80 \

  --access allow
```

4️⃣ Create Public IP for Load Balancer

```
az network public-ip create \

 --resource-group MyResourceGroup \

 --name MyPublicIP
```

5️⃣ Create Load Balancer

```
az network lb create --resource-group MyResourceGroup --name MyLoadBalancer \

 --public-ip-address MyPublicIP --frontend-ip-name MyFrontEndPool \

 --backend-pool-name MyBackEndPool
```

6️⃣ Create Health Probe & Load Balancing Rule

```
az network lb probe create --resource-group MyResourceGroup --lb-name
MyLoadBalancer \

 --name MyHealthProbe --protocol tcp --port 80


az network lb rule create --resource-group MyResourceGroup --lb-name
MyLoadBalancer \

 --name HTTPRule --protocol tcp --frontend-port 80 --backend-port 80 \

 --frontend-ip-name MyFrontEndPool --backend-pool-name MyBackEndPool \

 --probe-name MyHealthProbe
```

7️⃣ Create NICs & Associate with NSG & Subnet

```
az network nic create --resource-group MyResourceGroup --name MyNic1 \

 --vnet-name MyVNet --subnet MySubnet --network-security-group MyNSG


az network nic create --resource-group MyResourceGroup --name MyNic2 \

 --vnet-name MyVNet --subnet MySubnet --network-security-group MyNSG
```

8️⃣ Create Virtual Machines & Install Web Server

```
az vm create --resource-group MyResourceGroup --name VM1 --nics MyNic1 \
  --image UbuntuLTS --admin-username azureuser --generate-ssh-keys \
  --custom-data cloud-init.txt

az vm create --resource-group MyResourceGroup --name VM2 --nics MyNic2 \
  --image UbuntuLTS --admin-username azureuser --generate-ssh-keys \
  --custom-data cloud-init.txt
```

➕ ***Sample cloud-init.txt to install NGINX automatically:***

```
#cloud-config
package_upgrade: true
packages:
  - nginx
runcmd:
  - systemctl start nginx
  - systemctl enable nginx
```

9️⃣ Add VMs to Load Balancer Backend Pool

```
az network nic ip-config address-pool add --address-pool MyBackEndPool \
  --ip-config-name ipconfig1 --nic-name MyNic1 --resource-group MyResourceGroup \
  --lb-name MyLoadBalancer

az network nic ip-config address-pool add --address-pool MyBackEndPool \
  --ip-config-name ipconfig1 --nic-name MyNic2 --resource-group MyResourceGroup \
  --lb-name MyLoadBalancer
```

🔟 Test Load Balancer (Access via Public IP)

Visit the **Load Balancer's Public IP** in your browser:

```
az network public-ip show --resource-group MyResourceGroup --name MyPublicIP --query ipAddress -o tsv
```

📁 Folder Structure:

```
azure-multi-vm-load-balancer/
│
├── cloud-init.txt     # Startup script for web server
├── README.md          # Project Documentation
└── deploy.sh          # Optional Bash Automation Script (Steps Above)
```

🚀 **GitHub-Ready Project Description (For README.md):**

# ⚡ Azure Multi-VM Architecture with Load Balancer ⚙️

🚀 Deploy a high-availability architecture on Azure with:

- Multiple Virtual Machines (Ubuntu-based) 🖥️

- Azure Load Balancer ⚖️

- Virtual Network (VNet) 🌐

- Network Security Groups (NSG) 🔐

- Automated NGINX Installation via cloud-init 📦

## 📋 Features:

✅ Load-balanced NGINX web servers

✅ Fully automated Azure CLI-based deployment

✅ Secure inbound HTTP access (port 80)

## 📂 Folder Structure:

```
azure-multi-vm-load-balancer/
│
├── cloud-init.txt # Startup script for web server
├── README.md # Project Documentation
└── deploy.sh # Bash Script (Optional)
```

## 🛠️ Technologies:

- Azure CLI

- Virtual Machines

- Load Balancer

- NGINX

- Cloud-Init

## 🌐 Access:

Your load-balanced site will be available at the **Public IP** of the Load Balancer 🎉

### 📝 License:

MIT License