Module 346 - Mise en place d'une application Flask

Introduction

Ce document montre notre approche étape par étape pour préparer le déploiement de l'application Flask dans le Cloud.

Composants Azure de l'architecture

- **Groupe de ressources**: `pc04sej` (pré créée)
- **Region**: `switzerlandnorth`
- **Machines virtuelles**:
  - `vm-web`: Machine virtuelle contenant l'application flask (IP publique)
  - `vm-db`: Serveur MariaDB (IP privée uniquement)
- **Composants réseaux**:
  - **Réseau virtuel**: `vnet-main` (10.10.0.0/16)
  - **Subnets**: `subnet-web` (10.10.1.0/24), `subnet-db` (10.10.2.0/24)
  - **NSGs**: `nsg-web`, `nsg-db`
- **Authentication**:
  - **Username admin**: `azureuser`
  - **Clé SSH**: `~/.ssh/id_ed25519.pub` (doit être présente sur la machine host, ou peut être créée via la commande `ssh-keygen`)

---

1. Setup de l'infrastructure réseau

```
# Create virtual network and web subnet
az network vnet create \
  --resource-group pc04sej \
  --name vnet-main \
  --address-prefix 10.10.0.0/16 \
  --subnet-name subnet-web \
  --subnet-prefix 10.10.1.0/24 \
  --location switzerlandnorth

# Create database subnet
az network vnet subnet create \
  --resource-group pc04sej \
  --vnet-name vnet-main \
  --name subnet-db \
  --address-prefix 10.10.2.0/24

# Create Network Security Groups
az network nsg create --resource-group pc04sej --name nsg-web
az network nsg create --resource-group pc04sej --name nsg-db

# NSG rule: Allow SSH & HTTP to web subnet from internet
az network nsg rule create \
  --resource-group pc04sej \
  --nsg-name nsg-web \
  --name allow-ssh-http \
  --priority 1000 \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --source-address-prefix Internet \
  --destination-port-ranges 22 80
```

```
# NSG rule: Allow web subnet access to DB subnet (SSH & MariaDB)
az network nsg rule create \
  --resource-group pc04sej \
  --nsg-name nsg-db \
  --name allow-web-to-db \
  --priority 1000 \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --source-address-prefix 10.10.1.0/24 \
  --destination-port-ranges 22 3306

# Associate NSGs to subnets
az network vnet subnet update \
  --resource-group pc04sej \
  --vnet-name vnet-main \
  --name subnet-web \
  --network-security-group nsg-web

az network vnet subnet update \
  --resource-group pc04sej \
  --vnet-name vnet-main \
  --name subnet-db \
  --network-security-group nsg-db
```

2. Virtual Machine Deployment

```
# Create web VM (with public IP)
az vm create \
  --resource-group pc04sej \
  --name vm-web \
  --image Debian:debian-12:12:latest \
  --size Standard_B1ms \
  --subnet subnet-web \
  --vnet-name vnet-main \
  --public-ip-address pip-vm-web \
  --admin-username azureuser \
  --ssh-key-values ~/.ssh/id_ed25519.pub

# Create DB VM (private IP only)
az vm create \
  --resource-group pc04sej \
  --name vm-db \
  --image Debian:debian-12:12:latest \
  --size Standard_B1ms \
  --subnet subnet-db \
  --vnet-name vnet-main \
  --public-ip-address "" \
  --admin-username azureuser \
  --ssh-key-values ~/.ssh/id_ed25519.pub
```

3. SSH Access Configuration

```
# Direct SSH to web VM
ssh -i ~/.ssh/id_ed25519 azureuser@<web-vm-public-ip>
```

```
# SSH to database VM through web VM (jump host)
ssh -i ~/.ssh/id_ed25519 -J azureuser@<web-vm-public-ip> azureuser@10.10.2.4
```

4. Automated VM Backup Solution

This script creates daily snapshots of VM disks and maintains a 7-day retention policy.

```
#!/bin/bash

set -e

# Configuration variables
RESOURCE_GROUP="pc04sej"
LOCATION="switzerlandnorth"
VMS=("vm-web" "vm-db")
RETENTION_DAYS=7

# Current timestamp for snapshot naming
TIMESTAMP=$(date +%Y%m%d%H%M)

# Create snapshots for each VM
for VM_NAME in "${VMS[@]}"; do
  # Get the OS disk ID for the VM
  OS_DISK=$(az vm show \
    --resource-group "$RESOURCE_GROUP" \
    --name "$VM_NAME" \
    --query "storageProfile.osDisk.name" -o tsv)

  # Create a snapshot with timestamp
  SNAPSHOT_NAME="snap-${VM_NAME}-${TIMESTAMP}"

  az snapshot create \
    --resource-group "$RESOURCE_GROUP" \
    --name "$SNAPSHOT_NAME" \
    --source "$OS_DISK" \
    --location "$LOCATION" \
    --sku Standard_LRS

  echo "Snapshot created: $SNAPSHOT_NAME"
done

# Cleanup old snapshots (older than retention period)
EXPIRY_DATE=$(date -d "-${RETENTION_DAYS} days" +%Y%m%d%H%M)
for SNAPSHOT in $(az snapshot list --resource-group "$RESOURCE_GROUP" --query "[?
contains(name, 'snap-')].name" -o tsv); do
  DATE_PART=$(echo "$SNAPSHOT" | grep -oP '\d{12}')
  if [[ "$DATE_PART" < "$EXPIRY_DATE" ]]; then
    az snapshot delete --resource-group "$RESOURCE_GROUP" --name "$SNAPSHOT" --yes
    echo "Deleted old snapshot: $SNAPSHOT"
  fi
done
```

Setting Up Daily Backup Automation

```
# Make the script executable
chmod +x snapshot_vms.sh

# Edit crontab to run daily at 2 AM
crontab -e
```

```
# Add this line to crontab:
0 2 * * * /home/agunthe1/scrips/snapshot_vms.sh >> /var/log/azure_snapshot.log 2>&1
```

5. Performance Testing with JMeter

1. Download and install Apache JMeter
2. Create a test plan (`testplan.jmx`) with HTTP Requests targeting `http://<vm-web-public-ip>/`
3. Run tests in non-GUI mode for better performance:

```
jmeter -n -t testplan.jmx -l results.jtl -e -o jmeter-report
```

4. View detailed analysis by opening `jmeter-report/index.html` in your browser

6. Cost Analysis

Estimated monthly cost breakdown for the infrastructure in Switzerland North region:

Resource	Specification	Quantity	Price/Unit/Month	Monthly Cost (CHF)
Virtual Machines	Standard_B1ms (1 vCPU, 2 GB RAM)	2	CHF 19.27	CHF 38.54
Managed Disks	P4 Premium SSD (32 GB)	2	CHF 6.39	CHF 12.78
Public IP Address	Static	1	CHF 2.63	CHF 2.63
VNet	Data transfer (<100 GB)	1	Free	CHF 0.00
Snapshots	Standard LRS (32 GB × 7 days × 2 VMs)	~448 GB	CHF 0.0235/GB	CHF 10.528
Total Estimated Monthly Cost				**CHF 64.48**

Notes:

- Prices shown in Swiss Francs (CHF) for Switzerland North region
- Actual costs may vary based on Azure pricing changes and actual resource utilization