Azure DevOps + Azure CLI + Terraform

End-to-End Execution Guide (Beginner → Production)

0️⃣ What this document helps you do

You will learn how to:

- Prepare **local environment**
- Authenticate to **Azure**
- Run **Python automation**
- Provision **Infrastructure using Terraform**
- Validate resources using **Azure CLI**
- Manage **VM lifecycle**
- Clean up resources safely

This is **exactly the same flow** used inside **Azure DevOps pipelines**, just executed manually first.

1️⃣ Environment Setup (One-time)

📌 Create Python Virtual Environment

```
python3 -m venv venv
```

Why?
Isolates Azure SDKs from your system Python (mandatory in real projects).

📌 Activate Virtual Environment

```
source venv/bin/activate
```

What happens?
Your terminal now uses `venv/bin/python`.

📌 Install Azure SDK for Authentication

```
pip install azure-identity
```

Why?
Allows Python code to securely authenticate using Azure AD.

2️⃣ Run Python Automation Script

📌 Run VM Creation / Automation Script

```
python /Users/venkatesh/Devops-GenAI_UST/Devops-Basics-L1/vmcreation1.py
```

OR (recommended single command):

```
source venv/bin/activate && python
/Users/venkatesh/Devops-GenAI_UST/Devops-Basics-L1/vmcreation1.py
```

What this does

- Uses Azure credentials
- Calls Azure APIs
- Automates infra tasks (VM / network / config)

3️⃣ Azure CLI Authentication (Mandatory)

📌 Login to Azure

```
az login
```

Why?
Authenticates your terminal with Azure Active Directory.

📌 Set Correct Subscription

az account set --subscription ab9b448f-07ad-4039-b26d-e74b90b60272

Critical Concept

> Azure DevOps pipelines **fail silently** if subscription is wrong.

📌 Verify Subscription

az account show -o table

4️⃣ Terraform Workflow (Infrastructure as Code)

📌 Check Terraform Installed

terraform -version

📌 Initialize Terraform

terraform init

What happens

- Downloads Azure provider
- Creates `.terraform/`
- Prepares backend (state)

📌 Validate Terraform Code

terraform validate

Why?
Checks syntax + configuration errors (no Azure calls yet).

📌 Preview Infrastructure Changes

```
terraform plan
```

Golden Rule

> `plan` shows **WHAT will change**
> `apply` actually **CHANGES Azure**

📌 Apply Infrastructure

```
terraform apply
```

Type `yes` when prompted.

5️⃣ Validate Resources via Azure CLI

📌 List Virtual Networks

```
az network vnet list \
  --resource-group example-resources \
  -o table
```

📌 List Subnets in VNet

```
az network vnet subnet list \
  --resource-group example-resources \
  --vnet-name example-vnet \
  -o table
```

Why this matters

- Confirms Terraform actually created infra
- Used heavily in debugging pipelines

6️⃣ VM Configuration & Lifecycle

📌 Enable Windows Automatic Updates

```
az vm update \
 --resource-group $resourceGroupName \
 --name $vmName \
 --set osProfile.windowsConfiguration.enableAutomaticUpdates=true
```

Used when

- Compliance
- Security hardening
- Enterprise baseline config

📌 Start VM

```
az vm start --resource-group $ResourceGroup --name $VMName
```

📌 Stop (Deallocate) VM (Cost Saving)

```
az vm deallocate --resource-group $ResourceGroup --name $VMName
```

7️⃣ PowerShell (Infra Engineers Use This Daily)

📌 List File System Drives

```
Get-PSDrive -PSProvider FileSystem
```

📌 Set Azure Subscription

```
az account set --subscription $SubscriptionId
```

8️⃣ Monitoring Setup (Production Mandatory)

📌 Create Resource Group

```
az group create --name $RESOURCE_GROUP --location $LOCATION
```

📌 Create Log Analytics Workspace

```
az monitor log-analytics workspace create \
 --resource-group $RESOURCE_GROUP \
 --workspace-name $WORKSPACE_NAME \
 --location $LOCATION
```

📌 View Workspace Details

```
az monitor log-analytics workspace show \
 --resource-group $RESOURCE_GROUP \
 --workspace-name $WORKSPACE_NAME
```

Why this is critical

- Logs
- Metrics
- Alerts
- Azure Monitor
- VM insights

9️⃣ Destroy Infrastructure (Clean Exit)

📌 Destroy All Terraform Resources

```
terraform destroy
```

Rule

> Always destroy in **non-production** to avoid billing leaks.

🔟 How this maps to Azure DevOps Pipelines

Local Command	Azure DevOps Stage
`az login`	Service Connection
`terraform init`	Init Stage
`terraform plan`	Validation
`terraform apply`	Release
`az vm start`	Ops Job
`terraform destroy`	Cleanup Job