📘 Azure CLI (az CLI) — Detailed Notes

🔹 1. What is Azure CLI?

- **Azure CLI (Command-Line Interface)** is a cross-platform tool to **create, manage, and monitor Azure resources**.
- Written in **Python**.
- Works on **Windows, Linux, macOS**, and also in **Azure Cloud Shell** (browser).
- Command format always starts with `az`.

👉 Example:

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

✅ Creates a Resource Group in East US.

🔹 2. Why Use Azure CLI?

✨ **Advantages**:

- ✅ **Cross-platform** (works everywhere).
- ✅ **Automation-friendly** (Bash, PowerShell, scripts).
- ✅ **Faster than portal** (no clicking, just commands).

- ✅ **Idempotent** → safe to re-run (won't break if already exists).
- ✅ **Great for CI/CD pipelines**.

🔹 3. Azure CLI Installation

🔧 Windows

- **MSI Installer** → Download here.

winget (faster):

```
winget install -e --id Microsoft.AzureCLI
```

-

🔧 Linux

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

🔧 macOS

```
brew update && brew install azure-cli
```

🔧 Cloud Shell

- Open **Azure Portal** → **Cloud Shell** (no install needed).

🔹 4. Basic Syntax

Format:

az <resource> <subcommand> [parameters]

📌 Examples:

Create Resource Group:

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

-

List VMs:

```
az vm list -o table
```

-

Start VM:

```
az vm start -g MyRG -n MyVM
```

-

🔹 5. Authentication in Azure CLI

Interactive Login

```
az login
```

1. → Opens browser, login with Azure credentials.

Service Principal (for automation)

```
az login --service-principal \
 --username <APP_ID> \
 --password <PASSWORD> \
 --tenant <TENANT_ID>
```

2.

Managed Identity (inside Azure resources)

```
az login --identity
```

3.

🔹 6. Output Formats

Azure CLI supports different output types:

- `table` → Human-readable
- `json` → Default, machine-readable
- `yaml` → YAML format
- `tsv` → Tab-separated values

📌 Example:

```
az vm list --output table
```

🔹 7. Commonly Used Commands

🗂️ Resource Groups

```
az group create -n MyRG -l eastus
az group list -o table
az group delete -n MyRG --yes
```

💻 Virtual Machines

```
az vm create -g MyRG -n MyVM --image UbuntuLTS \
  --admin-username azureuser --generate-ssh-keys

az vm list -o table
az vm start -g MyRG -n MyVM
az vm stop -g MyRG -n MyVM
```

☁️ Storage

```
az storage account create -n mystorage123 -g MyRG -l eastus --sku Standard_LRS
az storage account list -g MyRG -o table
```

🌐 Networking

```
az network vnet create -g MyRG -n MyVnet --subnet-name MySubnet
az network public-ip create -g MyRG -n MyPublicIP
```

🌍 App Services

```
az appservice plan create -g MyRG -n MyPlan --sku B1 --location eastus
az webapp create -g MyRG -p MyPlan -n MyWebApp --runtime "PYTHON:3.9"
```

🔹 8. Using CLI in Scripts

Bash Script Example

```
#!/bin/bash
az group create -n ScriptRG -l eastus
az vm create -g ScriptRG -n ScriptVM \
  --image UbuntuLTS --admin-username azureuser --generate-ssh-keys
```

PowerShell Example

```
az group create -n ScriptRG -l eastus
az vm create -g ScriptRG -n ScriptVM --image UbuntuLTS --admin-username azureuser --generate-ssh-keys
```

🔹 9. Where Can You Use Azure CLI?

1. **Local Machine** → Installed on Windows/Linux/Mac.

2. **Azure Cloud Shell** → In-browser shell with pre-installed tools.

3. **CI/CD pipelines** → Jenkins, GitHub Actions, Azure DevOps.

🔹 10. Azure Portal vs Azure CLI

Azure Portal (GUI)	Azure CLI (az)
Web interface (point & click)	Text-based commands
Easy for beginners	Faster for pros
Manual, not repeatable	Scriptable, repeatable
Visual feedback	Great for automation

📝 **Summary Notes**

- Azure CLI (az) = Fast, scriptable, cross-platform way to manage Azure.
- Install via **MSI**, **winget**, **brew**, or use **Cloud Shell**.
- Login with **az login** (interactive), **service principal**, or **managed identity**.
- Common resources: az group, az vm, az storage, az network, az webapp.
- Output formats: json, table, yaml, tsv.
- Best for **automation, DevOps, and repeatable infrastructure**.