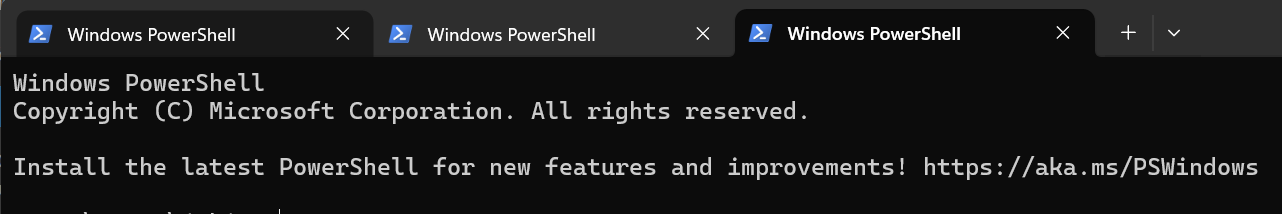
Class1- Labsetup

Software Needed: Azure CLI, AWS CLI, Visual studio and GIT, Terminal [Super putty for windows]



Method of Payment- Free Trial / Pay as you go

Terms: Putty, Superputty, user, administrator, package manager

**Introduction to Azure CLI**

Azure CLI is a cross-platform command-line tool that enables users to connect to Azure and execute administrative commands on Azure resources. It is designed for efficiency and automation, allowing for scripting and integration into various development workflows.

**Install Azure CLI:**

* **Windows**: Download and run the installer from the official Azure documentation.
* **macOS**: Use Homebrew:
* brew update && brew install azure-cli
* **Linux**: Use the package manager specific to your distribution. For example, on Ubuntu:
* curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

**Verify Installation:**

az --version

This command displays the installed version of Azure CLI and confirms a successful installation.

**Sign In to Azure:**

az login

This command opens a browser window for authentication. After logging in, the CLI is ready to interact with Azure resources.

**List Available Subscriptions:**

az account list --output table

Displays all subscriptions associated with the account in a table format.

**Set Active Subscription:**

az account set --subscription "SUBSCRIPTION\_NAME"

Sets the specified subscription as the active context for subsequent commands.

**Create a Resource Group:**

az group create --name RESOURCE\_GROUP\_NAME --location LOCATION

Creates a new resource group in the specified location.

**Create a Virtual Network (VNet):**

az network vnet create --resource-group RESOURCE\_GROUP\_NAME --name VNET\_NAME --address-prefixes ADDRESS\_PREFIX

Creates a virtual network with the specified address range.

**Create a Subnet within a VNet:**

az network vnet subnet create --resource-group RESOURCE\_GROUP\_NAME --vnet-name VNET\_NAME --name SUBNET\_NAME --address-prefix ADDRESS\_PREFIX

Adds a subnet to the existing virtual network.

**Create a Virtual Machine (VM):**

az vm create --resource-group RESOURCE\_GROUP\_NAME --name VM\_NAME --image IMAGE --admin-username ADMIN\_USERNAME --admin-password ADMIN\_PASSWORD

Deploys a virtual machine using the specified image and credentials.

**Start a Virtual Machine:**

az vm start --resource-group RESOURCE\_GROUP\_NAME --name VM\_NAME

Initiates the specified virtual machine.

**Stop a Virtual Machine:**

az vm stop --resource-group RESOURCE\_GROUP\_NAME --name VM\_NAME

Stops the specified virtual machine.

**Delete a Resource Group:**

az group delete --name RESOURCE\_GROUP\_NAME

Removes the specified resource group and all associated resources.

**Automating Tasks with Azure CLI**

Azure CLI supports automation through scripting. For example, to automate the shutdown of all virtual machines with a specific tag at a certain time, you can use a script that:

* Identifies all VMs with the tag Env=Dev.
* Initiates a shutdown command for each identified VM.

This approach is particularly useful for cost management by ensuring that development environments are not running during off-hours.

**Role-Based Access Control (RBAC) via Azure CLI**

Managing access to Azure resources is streamlined with Azure CLI. To create a custom role with specific permissions:

**Define the Role in a JSON File:**

{

"Name": "CustomRoleName",

"Description": "Description of the custom role.",

"Actions": [

"Microsoft.Compute/\*/read",

"Microsoft.Network/\*/read",

"Microsoft.Storage/\*/read"

],

"AssignableScopes": [

"/subscriptions/SUBSCRIPTION\_ID"

]

}

This JSON defines a role with read permissions on compute, network, and storage resources.

**Create the Custom Role:**

az role definition create --role-definition @roleDefinition.json

This command creates the role based on the JSON definition.

**Assign the Role to a User or Group:**

az role assignment create --assignee USER\_OR\_GROUP\_ID --role "CustomRoleName" --scope "/subscriptions/SUBSCRIPTION\_ID/resourceGroups/RESOURCE\_GROUP\_NAME"

Assigns the custom role to the specified user or group within the given scope.

**Best Practices**

* **Use Resource Tags:** Tagging resources facilitates organization, cost tracking, and automation. For example, tagging resources with Env=Dev or Env=Prod helps in identifying the environment of each resource.
* **Implement Automation for Routine Tasks:** Automate common tasks such as starting and stopping VMs during specific hours to optimize costs and resource utilization.
* **Regularly Review and Update Permissions:** Periodically review role assignments and permissions to ensure they align with current organizational policies and least privilege principles.

This guide provides a structured overview of Azure CLI installation, common commands, automation, and best practices to efficiently manage Azure resources.