Automating Azure Infrastructure with PowerShell & CLI

Hands-On Lab Guide

Ahmed Abdelwahed
ahmed_abdulwahed@outllook.com
www.abdelwahed.me
LinkedIn

1. Prerequisites, Installation & Authentication

1.1 Install and Verify the Azure PowerShell (Az) Module

The Az PowerShell module runs on PowerShell 7.4+ (recommended across Windows, Linux, and macOS) as well as Windows PowerShell 5.1. Ensure any legacy AzureRM module is uninstalled before installing Az to prevent command conflicts.

POWERSHELL (AZ MODULE SETUP)

```
# Check if legacy AzureRM is present and remove if needed
Get-Module -Name AzureRM -ListAvailable

# Install the modern Az PowerShell module for the current user
Install-Module -Name Az -Repository PSGallery -Force -AllowClobber -Scope CurrentUser

# Verify installed Az module version (Az.Compute, Az.Network, Az.Resources, Az.Storage)
Get-InstalledModule -Name Az
```

1.2 Install and Update Azure CLI

Azure CLI (az) is cross-platform and receives monthly updates. Keep your local installation current so newer VM images, SKUs, and security defaults work seamlessly.

AZURE CLI (VERSION CHECK & UPGRADE)

```
# Check installed Azure CLI version and extensions
az version

# Upgrade Azure CLI to the latest release (supported on Windows, Linux, and macOS)
az upgrade --yes
```

1.3 Connect to Azure and Set Active Subscription Context

Always verify which Azure subscription is active after signing in, especially when managing multiple tenants or lab/production environments.

POWERSHELL — SIGN IN & SELECT SUBSCRIPTION

```
# Interactive browser sign-in (supports MFA and Entra ID Conditional Access)
Connect-AzAccount

# Sign in to a specific Microsoft Entra tenant
Connect-AzAccount -Tenant '<Tenant-ID>'

# List available subscriptions and set the active context
Get-AzSubscription
Set-AzContext -Subscription '<Subscription-Name-or-ID>'
```

AZURE CLI — SIGN IN & SELECT SUBSCRIPTION

```
# Interactive browser sign-in (replaces deprecated -u / -p plaintext flags)
az login

# Use device code flow for headless SSH terminals or jump hosts
az login --use-device-code

# List subscriptions in table format and set the active subscription
az account list --output table
az account set --subscription '<Subscription-Name-or-ID>'
```

2. Core Infrastructure: Resource Groups, Storage & Networking

2.1 Create a Resource Group

A Resource Group acts as a logical lifecycle container for your Virtual Machine and its dependent resources (VNets, NICs, Public IPs, Managed Disks, and NSGs).

POWERSHELL & AZURE CLI — CREATE RESOURCE GROUP

```
# PowerShell (Az.Resources)
New-AzResourceGroup -Name 'AZRG01' -Location 'eastus'

# Azure CLI
az group create --name 'AZRG01' --location 'eastus'
```

2.2 Create a Storage Account

Modern Azure VMs use Azure Managed Disks by default and do not require manual storage account creation for OS/data disks. However, Storage Accounts remain essential for boot diagnostics, cloud shell, scripts, and blob/file storage. Storage account names must be 3–24 characters, globally unique, and strictly lowercase alphanumeric.

POWERSHELL & AZURE CLI — CREATE STORAGE ACCOUNT

```
# PowerShell (Az.Storage) — enforces HTTPS and TLS 1.2 by default
New-AzStorageAccount -ResourceGroupName 'AZRG01' `
    -Name 'azbiglab01' `
    -Location 'eastus' `
    -SkuName 'Standard_LRS' `
    -Kind 'StorageV2' `
    -MinimumTlsVersion 'TLS1_2'

# Azure CLI (fixed parameter flags: --location / -l and --sku)
az storage account create `
    --name 'azbiglab02' `
    --resource-group 'AZRG01' `
    --location 'eastus' `
    --sku 'Standard_LRS' `
    --kind 'StorageV2' `
    --min-tls-version 'TLS1_2'
```

2.3 Create a Virtual Network (VNet) and Subnet

Networking Best Practice Fix (RFC 1918 Address Space)
Version 1.0 used 200.200.200.0/24—a routable public IP range—for an internal Virtual Network. Always use RFC 1918 private IP ranges (10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16) to avoid routing conflicts with public internet endpoints.

POWERSHELL & AZURE CLI — CREATE VIRTUAL NETWORK & SUBNET

```
# PowerShell (Az.Network) — define subnet first, then attach to Virtual Network
$subnet = New-AzVirtualNetworkSubnetConfig -Name 'default' -AddressPrefix '10.0.1.0/24'
New-AzVirtualNetwork -ResourceGroupName 'AZRG01' `
    -Name 'AZVN01' `
    -Location 'eastus' `
    -AddressPrefix '10.0.0.0/16' `
    -Subnet $subnet

# Azure CLI — creates VNet and default subnet in a single command
az network vnet create `
    --name 'AZVN02' `
    --resource-group 'AZRG01' `
    --location 'eastus' `
    --address-prefixes '10.1.0.0/16' `
    --subnet-name 'default' `
    --subnet-prefixes '10.1.1.0/24'
```

3. Deploying Virtual Machines (Linux & Windows)

3.1 Discover Supported VM Sizes and Marketplace Images

Before deploying a VM, inspect available hardware SKUs in your target region and query current Marketplace image aliases (such as Ubuntu2404, RHEL, or Win2022Datacenter).

POWERSHELL & AZURE CLI — LIST VM SIZES & IMAGES

```
# PowerShell — list available VM sizes in East US
Get-AzVMSize -Location 'eastus' | Select-Object Name, NumberOfCores, MemoryInMB | Format-Table

# PowerShell — browse Ubuntu Server image SKUs
Get-AzVMImageSku -Location 'eastus' -PublisherName 'Canonical' -Offer 'ubuntu-24_04-lts'

# Azure CLI — list VM sizes in table format
az vm list-sizes --location 'eastus' --output table

# Azure CLI — list popular built-in image aliases and query Red Hat / Ubuntu offers
az vm image list --output table
az vm image list --publisher 'RedHat' --offer 'RHEL' --all --output table
```

3.2 Create a Linux VM (Ubuntu 24.04 LTS) using Azure CLI

CentOS Linux reached End of Life and is deprecated in Azure. Modern Linux deployments use Ubuntu LTS or Red Hat Enterprise Linux with SSH key authentication and Standard SKU Public IPs (Static allocation by default).

AZURE CLI — DEPLOY UBUNTU 24.04 LTS VIRTUAL MACHINE

```
az vm create `
    --resource-group 'AZRG01' `
    --name 'azbiglabvm04' `
    --location 'eastus' `
    --image 'Ubuntu2404' `
    --size 'Standard_D2s_v5' `
    --admin-username 'azureuser' `
    --generate-ssh-keys `
    --public-ip-address 'AZVMIP01' `
    --public-ip-sku 'Standard'
```

3.3 Create a Windows Server VM using Az PowerShell

When running New-AzVM with minimal parameters, Azure automatically creates a companion Resource Group, Virtual Network, Subnet, Network Security Group, Public IP, and NIC using the VM name if they do not already exist. For production and structured labs, explicitly specify your existing VNet, Subnet, and Standard SKU Public IP.

POWERSHELL — DEPLOY WINDOWS SERVER 2022 VIRTUAL MACHINE

```
# Prompt securely for local administrator credentials
$cred = Get-Credential

# Deploy Windows Server 2022 VM into existing VNet with Standard Static Public IP
New-AzVM -ResourceGroupName 'AZRG01' `
    -Location 'eastus' `
    -Name 'AZVM002' `
    -Image 'Win2022Datacenter' `
    -Size 'Standard_D2s_v5' `
    -VirtualNetworkName 'AZVN01' `
    -SubnetName 'default' `
    -PublicIpAddressName 'AZVMIP02' `
    -AllocationMethod 'Static' `
    -PublicIpSku 'Standard' `
    -OpenPorts 3389 `
    -Credential $cred
```

3.4 Open Inbound Ports on a VM (NSG Rule Shortcut)

If you need to allow inbound SSH (port 22), RDP (port 3389), or HTTP/HTTPS (ports 80/443) after VM creation, use az vm open-port or update the attached Network Security Group.

AZURE CLI & POWERSHELL — OPEN INBOUND PORT

```
# Azure CLI — open SSH (port 22) with priority 1000
az vm open-port --resource-group 'AZRG01' --name 'azbiglabvm04' --port 22 --priority 1000

# PowerShell — add an inbound HTTPS rule to an existing NSG
$nsg = Get-AzNetworkSecurityGroup -ResourceGroupName 'AZRG01' -Name 'AZVM002-nsg'
$nsg | Add-AzNetworkSecurityRuleConfig -Name 'Allow-HTTPS' -Access 'Allow' `
    -Protocol 'Tcp' -Direction 'Inbound' -Priority 1010 `
    -SourceAddressPrefix '*' -SourcePortRange '*' `
    -DestinationAddressPrefix '*' -DestinationPortRange 443 | Set-AzNetworkSecurityGroup
```

4. Inspecting, Monitoring & Managing Virtual Machines

4.1 Retrieve VM Details, Power State, Public IPs & Storage Accounts

POWERSHELL — QUERY VMS, POWER STATES, PUBLIC IPS & STORAGE

```
# List all VMs in the subscription with current power state (Running / Deallocated)
Get-AzVM -Status | Select-Object Name, ResourceGroupName, Location, PowerState | Format-Table

# Retrieve assigned Public IP address details
Get-AzPublicIpAddress -ResourceGroupName 'AZRG01' -Name 'AZVMIP02' | `
    Select-Object Name, IpAddress, PublicIpAllocationMethod, @{N='Sku';E={$_.Sku.Name}} | Format-
Table

# List Storage Accounts and their locations/SKUs
Get-AzStorageAccount | Select-Object StorageAccountName, ResourceGroupName, Location,
@{N='Sku';E={$_.Sku.Name}} | Format-Table
```

AZURE CLI — QUERY VMS, PUBLIC IPS & STORAGE

```
# List all VMs with detailed networking and power state info
az vm list --show-details --output table

# List Public and Private IP addresses for a specific VM
az vm list-ip-addresses --resource-group 'AZRG01' --name 'azbiglabvm04' --output table

# List Storage Accounts in table format
az storage account list --output table
```

4.2 VM Power Lifecycle: Start, Stop (Deallocate) & Restart

Critical Cost-Optimization Note: Stopped (Deallocated) vs. Stopped (Allocated)

Running Stop-AzVM (without -StayProvisioned) or az vm deallocate shuts down the OS and releases underlying Azure compute hardware, stopping hourly VM compute charges. Running az vm stop or Stop-AzVM -StayProvisioned halts the OS while keeping hardware reserved—you continue paying full compute charges!

POWERSHELL — START, DEALLOCATE, STOP & RESTART VMS

```
# Start a stopped/deallocated VM
Start-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002'

# Stop AND deallocate VM (stops compute billing — recommended)
Stop-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002' -Force

# Stop OS only while keeping compute hardware allocated (billing continues)
Stop-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002' -StayProvisioned -Force

# Restart a running VM
Restart-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002'
```

AZURE CLI — START, DEALLOCATE, STOP & RESTART VMS

```
# Start a VM
az vm start --resource-group 'AZRG01' --name 'azbiglabvm04'

# Stop AND deallocate VM (stops compute billing — recommended)
az vm deallocate --resource-group 'AZRG01' --name 'azbiglabvm04'

# Stop OS only while keeping hardware allocated (billing continues)
az vm stop --resource-group 'AZRG01' --name 'azbiglabvm04'

# Restart a VM
az vm restart --resource-group 'AZRG01' --name 'azbiglabvm04'
```

4.3 Resize a Virtual Machine

Resizing scales vCPUs and RAM up or down. If the target VM size is not available in the current hardware cluster, deallocate the VM first before resizing.

POWERSHELL & AZURE CLI — RESIZE A VIRTUAL MACHINE

```
# PowerShell — resize an existing VM to Standard_B2s
$vm = Get-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002'
$vm.HardwareProfile.VmSize = 'Standard_B2s'
Update-AzVM -ResourceGroupName 'AZRG01' -VM $vm

# Azure CLI — resize an existing VM to Standard_B2s
az vm resize --resource-group 'AZRG01' --name 'azbiglabvm04' --size 'Standard_B2s'
```

4.4 Execute Scripts Inside a VM Without SSH/RDP (Run Command)

Azure VM Run Command executes PowerShell or Bash scripts directly inside the guest OS through the Azure VM Agent—ideal for troubleshooting broken firewall rules or locked-out SSH/RDP sessions.

POWERSHELL & AZURE CLI — REMOTE RUN COMMAND

```
# PowerShell — run a PowerShell command inside a Windows VM
Invoke-AzVMRunCommand -ResourceGroupName 'AZRG01' -VMName 'AZVM002' `
    -CommandId 'RunPowerShellScript' -ScriptString 'Get-Service -Name TermService'

# Azure CLI — run a shell command inside a Linux VM
az vm run-command invoke --resource-group 'AZRG01' --name 'azbiglabvm04' `
    --command-id 'RunShellScript' --scripts 'uptime && df -h'
```

5. Identity & Access Management (Azure RBAC)

Grant administrators or lab users scoped permissions to manage VMs at the Resource Group or individual VM level using Azure Role-Based Access Control (RBAC).

POWERSHELL & AZURE CLI — ASSIGN, LIST & REMOVE RBAC ROLES

```
# PowerShell — assign Contributor role on a Resource Group (fixed -RoleDefinitionName spelling)
New-AzRoleAssignment -SignInName 'ahmed@abdelwahed.me' `
    -RoleDefinitionName 'Contributor' `
    -ResourceGroupName 'AZRG01'

# PowerShell — verify role assignments on the Resource Group
Get-AzRoleAssignment -ResourceGroupName 'AZRG01' | Select-Object DisplayName, RoleDefinitionName,
Scope

# Azure CLI — assign Virtual Machine Contributor role using explicit resource group scope
$rgId = az group show --name 'AZRG01' --query id --output tsv
az role assignment create --assignee 'ahmed@abdelwahed.me' `
    --role 'Virtual Machine Contributor' `
    --scope $rgId
```

6. Cleanup, Decommissioning & Session Sign-Out

6.1 Delete Individual Virtual Machines and Storage Accounts

Orphaned Resources Warning
Deleting an individual VM via Remove-AzVM or az vm delete removes the VM compute object, but its OS disk, Network Interface (NIC), and Public IP address remain in the Resource Group and continue incurring storage/IP charges unless deleted separately or via Resource Group deletion.

POWERSHELL & AZURE CLI — DELETE VMS & STORAGE ACCOUNTS

```
# PowerShell — remove a VM without confirmation prompt
Remove-AzVM -ResourceGroupName 'AZRG01' -Name 'AZVM002' -Force

# Azure CLI — delete a VM without confirmation prompt
az vm delete --resource-group 'AZRG01' --name 'azbiglabvm04' --yes

# PowerShell & Azure CLI — delete a Storage Account
Remove-AzStorageAccount -ResourceGroupName 'AZRG01' -Name 'azbiglab01' -Force
az storage account delete --resource-group 'AZRG01' --name 'azbiglab02' --yes
```

6.2 Delete an Entire Resource Group (Complete Lab Teardown)

Deleting a Resource Group permanently removes the group and every resource inside it (VMs, managed disks, VNets, NICs, NSGs, Public IPs, and storage accounts).

POWERSHELL & AZURE CLI — DELETE RESOURCE GROUP

```
# PowerShell — remove Resource Group and all nested resources (-AsJob runs asynchronously)
Remove-AzResourceGroup -Name 'AZRG01' -Force -AsJob

# Azure CLI — delete Resource Group asynchronously without waiting
az group delete --name 'AZRG01' --yes --no-wait
```

6.3 Disconnect Active Sessions

POWERSHELL & AZURE CLI — SIGN OUT

```
# PowerShell — clear cached Azure credentials from current session
Disconnect-AzAccount

# Azure CLI — clear cached Azure CLI credentials
az logout
```

Appendix: Legacy v1.0 to Modern v2.0 Command Crosswalk

Use this reference matrix when upgrading older scripts written for the retired AzureRM or classic Azure Service Management (ASM) modules to modern Az PowerShell and Azure CLI.

Task	Legacy v1.0 (Deprecated)	Modern Az PowerShell (v2.0)	Modern Azure CLI (v2.0)
Install Module	`Install-Module AzureRM`	`Install-Module -Name Az -Scope CurrentUser`	`az upgrade --yes`
Sign In	`Connect-AzureRmAccount` `az login -u user -p pass`	`Connect-AzAccount` `Set-AzContext -Subscription <id>`	`az login` `az account set --subscription <id>`
Create Resource Group	`New-AzureRmResourceGroup`	`New-AzResourceGroup`	`az group create`
Create Storage Account	`New-AzureRmStorageAccount`	`New-AzStorageAccount`	`az storage account create`
Create VNet	`New-AzureRmVirtualNetwork (200.200.200.0/24)`	`New-AzVirtualNetwork (10.0.0.0/16 RFC 1918)`	`az network vnet create (10.1.0.0/16 RFC 1918)`
Create VM	`New-AzureRmVM --image centos --size Standard_A1`	`New-AzVM -Image Win2022Datacenter -Size Standard_D2s_v5`	`az vm create --image Ubuntu2404 --size Standard_D2s_v5`
Inspect VM & Status	`Get-AzureRmVM`	`Get-AzVM -Status`	`az vm list --show-details -o table`
Get Public IP	`Get-AzureRmPublicIpAddress`	`Get-AzPublicIpAddress`	`az vm list-ip-addresses -o table`
Get Storage Account	`Get-AzureRMStorageAccount`	`Get-AzStorageAccount`	`az storage account list -o table`
Start / Stop / Restart VM	`Start-AzureVM / Stop-AzureVM (Classic ASM cmdlets)`	`Start-AzVM / Stop-AzVM` `Restart-AzVM`	`az vm start / az vm deallocate` `az vm restart`
Resize VM	`az vm resize --size Standard_A1`	`Update-AzVM ($vm.HardwareProfile.VmSize)`	`az vm resize --size Standard_B2s`
Assign RBAC Role	`New-AzureRmRoleAssignment -RoleDefnitionName (typo)`	`New-AzRoleAssignment -RoleDefinitionName`	`az role assignment create --assignee --role --scope`
Delete VM & RG	`Remove-AzureVM` `Remove-AzureRmResourceGroup`	`Remove-AzVM -Force` `Remove-AzResourceGroup -Force`	`az vm delete --yes` `az group delete --yes --no-wait`
Sign Out	`Disconnect-AzureRmAccount`	`Disconnect-AzAccount`	`az logout`