**Step-by-Step PowerShell Commands**

**Prerequisites:**

Azure PowerShell module installed (Az)

Logged in via **Connect-AzAccount**

1. **Set Variables**

**# Variables**

**$resourceGroupName = "MyResourceGroup"**

**$location = "EastUS"**

**$appServicePlanName = "MyAppServicePlan"**

**$webAppName = "MyUniqueWebAppName1234" # Must be globally unique**

**$osType = "Linux" # Options: "Linux" or "Windows"**

**$runtime = "DOTNET|6.0" # Format: '<RUNTIME>|<VERSION>'**

**$tier = "Basic" # Options: ‘**Free, Shared, Basic, Standard, PremiumV2, PremiumV3, Isolated, IsolatedV2**’**

1. **Create Resource Group**

**New-AzResourceGroup -Name $resourceGroupName -Location $location**

1. **Create App Service Plan**

**# For Linux**

New-AzAppServicePlan -Name $appServicePlanName `

-Location $location `

-ResourceGroupName $resourceGroupName `

-Tier $**tier** `

-NumberofWorkers 1 `

-WorkerSize "Small" `

-Linux

**# For Windows**

New-AzAppServicePlan -Name $appServicePlanName `

-Location $location `

-ResourceGroupName $resourceGroupName `

-Tier $**tier** `

-NumberofWorkers 1 `

-WorkerSize "Small"

1. **Create Web App**

**# For Linux**

New-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-Location $location `

-AppServicePlan $appServicePlanName `

-Runtime $runtime

**# For Windows**

New-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-Location $location `

-AppServicePlan $appServicePlanName

1. **Delete a web app**

Remove-AzWebApp -ResourceGroupName $resourceGroupName -Name $webAppName

1. **Delete App Service Plan or Resource Group**

Remove-AzAppServicePlan -Name $appServicePlanName -ResourceGroupName $resourceGroupName

# OR

Remove-AzResourceGroup -Name $resourceGroupName

1. **Enable HTTPS-only access**

Set-AzWebApp -ResourceGroupName $resourceGroupName -Name $webAppName -HttpsOnly $true

1. **Bind a custom domain and SSL certificate**

New-AzWebAppSSLBinding -ResourceGroupName $resourceGroupName `

-WebAppName $webAppName `

-Name "yourdomain.com" `

-SslState SniEnabled `

-Thumbprint "CERT\_THUMBPRINT"

1. **Manually scale out**

Set-AzAppServicePlan -Name $appServicePlanName `

-ResourceGroupName $resourceGroupName `

-NumberofWorkers 3

1. **Configure deployment source (e.g. GitHub)**

Set-AzWebAppSourceControl -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-RepoUrl "https://github.com/username/repo" `

-Branch "main" `

-IsManualIntegration $false

1. **ZIP Deploy**

Publish-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-ArchivePath "C:\Path\To\app.zip"

1. **Enable diagnostic logs**

Set-AzWebAppDiagnosticLog -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-ApplicationLoggingFileSystem $true `

-WebServerLoggingFileSystem $true `

-DetailedErrorMessages $true `

-FailedRequestTracing $true

1. **Stream logs in real-time**

Get-AzWebAppLog -ResourceGroupName $resourceGroupName -Name $webAppName

1. **Swap slots**

Get-AzWebAppLog -ResourceGroupName $resourceGroupName -Name $webAppName

**Full PowerShell Script – With Optional Features**

# -------------------------------------

# VARIABLES - Customize as needed

# -------------------------------------

$resourceGroupName = "MyResourceGroup"

$location = "EastUS"

$appServicePlanName = "MyAppServicePlan"

$webAppName = "MyUniqueWebAppName1234" # Must be globally unique

$osType = "Linux" # "Linux" or "Windows"

$runtime = "NODE|18-lts" # Only applies to Linux

$slotName = "staging"

$enableDeploymentSlot = $true

$setEnvironmentVariables = $true

# -------------------------------------

# LOGIN AND RESOURCE GROUP

# -------------------------------------

Connect-AzAccount

New-AzResourceGroup -Name $resourceGroupName -Location $location -Force

# -------------------------------------

# APP SERVICE PLAN

# -------------------------------------

if ($osType -eq "Linux") {

Write-Host "Creating Linux App Service Plan..."

New-AzAppServicePlan -Name $appServicePlanName `

-Location $location `

-ResourceGroupName $resourceGroupName `

-Tier "Basic" `

-NumberofWorkers 1 `

-WorkerSize "Small" `

-Linux

}

elseif ($osType -eq "Windows") {

Write-Host "Creating Windows App Service Plan..."

New-AzAppServicePlan -Name $appServicePlanName `

-Location $location `

-ResourceGroupName $resourceGroupName `

-Tier "Basic" `

-NumberofWorkers 1 `

-WorkerSize "Small"

}

# -------------------------------------

# CREATE WEB APP

# -------------------------------------

if ($osType -eq "Linux") {

Write-Host "Creating Linux Web App with runtime: $runtime"

New-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-Location $location `

-AppServicePlan $appServicePlanName `

-Runtime $runtime

}

elseif ($osType -eq "Windows") {

Write-Host "Creating Windows Web App..."

New-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-Location $location `

-AppServicePlan $appServicePlanName

}

# -------------------------------------

# DEPLOYMENT SLOT (OPTIONAL)

# -------------------------------------

if ($enableDeploymentSlot) {

Write-Host "Creating deployment slot: $slotName"

New-AzWebAppSlot -Name $webAppName `

-ResourceGroupName $resourceGroupName `

-Slot $slotName `

-AppServicePlan $appServicePlanName

}

# -------------------------------------

# APP SETTINGS / ENVIRONMENT VARIABLES (OPTIONAL)

# -------------------------------------

if ($setEnvironmentVariables) {

Write-Host "Setting application environment variables..."

$appSettings = @{}

$appSettings["ENVIRONMENT"] = "Production"

$appSettings["MySecret"] = "SuperSecret123"

Set-AzWebApp -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-AppSettings $appSettings

if ($enableDeploymentSlot) {

Set-AzWebAppSlot -ResourceGroupName $resourceGroupName `

-Name $webAppName `

-Slot $slotName `

-AppSettings @{ "ENVIRONMENT" = "Staging"; "MySecret" = "SlotSecret456" }

}

}