Agenda: Managing Azure with Windows PowerShell and Azure CLI.

- Installing the required modules
- Login to Azure Portal
- Managing Resource Group
- Managing App Service Plans and App Service Web Apps
- Create and Configure a Storage Account
- Managing Storage Accounts using PowerShell
- Azure CLI Introduction and Setup
- Creating Resources
- Listing and Formatting output
- Connect a Web App to SQL Database
- Deleting Resources
- Interactive Mode

Installing Required PowerShell Modules

Introduction:

- Azure PowerShell is a set of modules that provide **cmdlets** to manage Azure with **Windows PowerShell**. You can use the cmdlets to create, test, deploy, and manage solutions and services delivered through the Azure platform.
- Typically, as a developer, you might want to automate some management tasks by creating reusable scripts, or combine management of Azure resources with management of other network and infrastructure services.

You can use either of the below to install Azure PowerShell:

1. Note: The Azure PowerShell module has a dependency on the Microsoft .NET Framework 4.5 and **the Web Platform Installer** (Web PI) checks for this during installation.
   You can launch the Web PI from
   - https://azure.microsoft.com/en-us/downloads/
   - Search PowerShell → "Windows Install" → Install

Note: For non-windows platform: https://docs.microsoft.com/en-us/powershell/azure/install-azurermps-maclinux?view=azurermps-5.3.0

2. **PowerShell Gallery**.

In administrator mode, Open Standard Windows PowerShell console,

Install the Azure Resource Manager (ARM) modules from the PowerShell Gallery

```
Set-ExecutionPolicy Remote-Signed
Install-Module AzureRM
```

Import-Module AzureRM

If already existing, use following command to update

Update-Module -Name AzureRM

Note:

- PowerShell Gallery Modules Location: **C:\Program Files\WindowsPowerShell\Modules**.
- Web PI Installed Modules Location: **C:\Program Files (x86)\WindowsPowerShell\Modules**
- If an error occurs during install, you can manually remove the Azure* folders in your **%ProgramFiles%\WindowsPowerShell\Modules** folder, and try the installation again.

To make sure the Azure PowerShell module is available after you install

Get-Module –ListAvailable

Get help for cmdlets

To get detailed help for any cmdlet that you see in this tutorial, use the Get-Help cmdlet.

Get-Help <cmdlet-name> -Detailed

For example, to get help for the Get-AzureRmResource cmdlet, type:

Get-Help Get-AzureRmResource -Detailed

Managing Azure Accounts and Subscriptions

To log in to Azure Resource Manager

Login-AzureRmAccount

You can also use a specific Tenant if you would like a faster log in experience

Login-AzureRmAccount -TenantId 2de8d54d-5576-4bf6-b419-6065cb1e700e

To view all subscriptions for your account

Get-AzureRmSubscription

To select a default subscription for your current session

Get-AzureRmSubscription –Subscription “your sub” | **Select-AzureRmSubscription**

Select-AzureRmSubscription -Subscription "your sub"
Or

```
Set-AzureRmContext -Subscription "Sandeep Soni - Visual Studio Enterprise"

# View your current Azure PowerShell session context
# This session state is only applicable to the current session and will not affect other sessions
Get-AzureRmContext
```

Managing a Resource Group

1. Open the Windows PowerShell ISE using Windows → Search → Window Powershell
2. In the Windows PowerShell ISE, at the command prompt, type the following command, and then press Enter:

   Login-AzureRmAccount
3. In the Windows PowerShell ISE, execute the following lines:

```
$locName = "Central US"
$rgName = "DemoRG"
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

Delete a resource group

```
Remove-AzureRmResourceGroup -Name $rgName
```

Get the list of all resource groups

```
Get-AzureRmResourceGroup
```

Moves a resource to a different resource group or subscription.

```
$resource = Get-AzureRmResource -ResourceType "Microsoft.ClassicCompute/storageAccounts" -ResourceName "DssStorageAccountName"
Move-AzureRmResource -ResourceId $resource.ResourceId -DestinationResourceGroupName "NewRG"
```

Searches for resources based on specified parameters:

```
$multipleresources = Find-AzureRmResource -ResourceType "microsoft.web/sites" -ResourceGroupNameContains "RG"
$multipleresources = Find-AzureRmResource -ResourceType "microsoft.web/sites" -ResourceNameContains "test"
```

Managing an App Service Plans

List all Existing

```
Get-AzureRmAppServicePlan -ResourceGroupname DemoRG
```

Note: If ResourceGroupName is not provided all the Service Plans will be listed.

Create

New-AzureRmAppServicePlan -Name DemoPlan -Location "Central US" -ResourceGroupName DemoRG -Tier Standard -WorkerSize Medium -NumberofWorkers 2

To change the Tier

Set-AzureRmAppServicePlan -Name DemoPlan -ResourceGroupName DemoRG -Tier **Basic**

Delete

Remove-AzureRmAppServicePlan -Name DemoPlan -ResourceGroupName DemoRG

Manage an Azure Web Apps

List existing Web Apps

Get-AzureRmWebApp -ResourceGroupname DemoRG

Create:

New-AzureRmWebApp -Name DssDemoWebApp -AppServicePlan DemoPlan -ResourceGroupName DemoRG -Location "Central US"

Delete:

Remove-AzureRmWebApp -Name ContosoWebApp -ResourceGroupName ContosoAzureResourceGroup

Configure an existing Web App

$appsettings = @{Key1 = "Key1value"; Key2 = "Key2value"}

Set-AzureRmWebApp -Name DssDemoWebApp -ResourceGroupName DemoRG -**AppSettings** $appsettings

Start / Stop / Restart:

Start-AzureRmWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG

Stop-AzureRmWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG

Restart-AzureRmWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG

Introducing the NEW Azure PowerShell Az module

- Starting in December 2018, the **Azure PowerShell Az** module is in general release and now the intended PowerShell module for interacting with Azure.
- Az offers **shorter commands, improved stability, and cross-platform support**.
- Az also offers feature parity and an easy migration path from AzureRM.

- Az uses the **.NET Standard library**, which means it runs on PowerShell 5.x and PowerShell 6.x.
- Since PowerShell 6.x can run on **Linux**, **macOS**, and **Windows**, Azure PowerShell is now available for all platforms.
- Az is a new module, so the version has been reset to **1.0.0**.

Install Az Module

```
Install-Module -Name Az
```

OR

Recommended

Step1: Uninstall AzureRM Modules

```
Uninstall-AzureRm
```

Step2: Install Az Module

1. OPTIONAL: Check version of Windows PowerShell and it should be 5.x or 6.x

   ```
   $PSVersionTable.PSVersion
   ```

   Install the latest version of powershell. Download and run the MSI from https://github.com/PowerShell/PowerShell/releases
2. Install Azure Az PowerShell Modules

   ```
   Install-Module -Name Az
   ```

Step3: To enable the compatibility mode for the Az module. PowerShell Copy Try It

```
Enable-AzureRmAlias -Scope CurrentUser
```

In general, the module names have been changed so that **AzureRM** and Azure become **Az**, and the same for cmdlets.

For example:

- AzureRM.Compute module has been renamed to Az.Compute.
- New-AzureRMVM has become New-AzVM,
- Get-AzureStorageBlob is now Get-AzStorageBlob.

To Uninstall Az Module:

https://docs.microsoft.com/en-us/powershell/azure/uninstall-az-ps?view=azps-1.1.0

Azure CLI

- With Azure CLI, you can create, manage, and delete services on the command line via cmd.exe, bash or {your shell} on the operating system of your choice.

- You can use it in your browser with Azure Cloud Shell, or you can install it on macOS, Linux, and Windows and run it from the command line.
- Azure CLI 2.0 is optimized for managing and administering Azure resources from the command line, and for building automation scripts that work against the Azure Resource Manager.

Install locally Azure CLI 2.0:

Download the MSI installer(https://docs.microsoft.com/en-us/cli/azure/install-azure-cli) and then run it to install or update.

Note: Same MSI can be used for **uninstalling** the same.

Az login

To sign in, use a web browser to open the page https://aka.ms/devicelogin and enter the code XXXXXXX to authenticate.

#To search for commands, use az find.

az find -q secret

#Use the --help argument to get a complete list of commands and subgroups of a group.

az network nsg --help

Creating Resources

#Create a Resource Group:

az group create -n DemoRG -l southindia

 # Create an Azure AppService that we can host any number of web apps within

az appservice plan create -n MyAppServicePlan -g DemoRG

 # Create Two Web Apps within the AppService (note: name param must be a unique DNS entry)

az webapp create -n DssMyWebApp1 -g DemoRG --plan MyAppServicePlan

az webapp create -n DssMyWebApp2 -g DemoRG --plan MyAppServicePlan

Following are some popular Azure resource types and the corresponding Azure CLI create commands to create them:

Resource Type	Azure CLI create command
Resource Group	az group create
Virtual Machine	az vm create

Virtual Network	az network vnet create
Load Balancer	az network lb create
Managed Disk	az disk create
Storage account	az storage account create
Virtual Machine Scale Set	az vmss create
Azure Container	az acs create
Web App	az webapp create
SQL Database Server	az sql server create
Document DB	az documentdb create

Note:

If you do not need to wait on creation of a resource before continuing, you can use the **no-wait** option to start a create action in the background.

az webapp create -n DssMyWebApp2 -g DemoRG --plan MyAppServicePlan **--no-wait**

Listing resources and formatting output

To list all the items for a particular category:

az webapp **list**

Output Formats:

--output	**Description**
json	json string. json is the default.
jsonc	colored json string.
table	table with column headings.
tsv	tab-separated values.

Example:

- az webapp list -g DemoRG --**output json**
- az webapp list -g DemoRG --**output table**
- az webapp list -g DemoRG --**output tsv**

Query:

az vm list **--query [*].[name,resourceGroup]** --out table

az vm list **--query "[].{RGName:resourceGroup, VMName:name}" --out** table

```
RGName      VMName
----------  -----------
DEMORG1     DemoVM010
DEMORG1     demovm111
DEMORG1     demovm211
DEMORG1     demovm212
DEMORG1     demovm213
DEMORG1     demovm214
DEMORG1     demovm222
RGDEMO001   KBDemo001VM
RGDEMO001   KBDemo020
```

Interactive Mode

You can use Azure CLI 2.0 in interactive mode by running the `az **interactive**` command.

az **interactive**

Note: Interactive mode optionally displays command descriptions, parameter descriptions, and command examples. You can turn descriptions and examples on or off using `F1`.

You can turn the display of parameter defaults on or off using `F2`.

Connect a WebApp to SQL database

 # Create a SQL Server

az sql server create --name **DssDemoSqlServer** --resource-group DemoRG --location southindia --admin-user "dssadmin" --admin-password "Password@123"

 # Configure Firewall for Azure Access

az sql server firewall-rule create --resource-group DemoRG --server dssdemosqlserver --name AllowYourIp --start-ip-address "0.0.0.0" --end-ip-address "0.0.0.0"

 # Create Database on Server

az sql db create --resource-group DemoRG **–**server **dssdemosqlserver** --name MySampleDatabase --service-objective S0

 # Assign the connection string to an App Setting in the Web App

az webapp config appsettings set --settings "SQLSRV_CONNSTR=Server=tcp:**dssdemosqlserver**.database.windows.net;Database=**dssdemosqlserver**;User ID=dssadmin@dssdmosqlserver;Password=Password@123;Trusted_Connection=False;Encrypt=True;" --name **DssMyWebApp** --resource-group DemoRG

 # Deleting Resource (WebApp)

```
az webapp delete -n DssMyWebApp -g DemoRG
az group delete -g DemoRG
```