Managing Azure with 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 **PowerShell**. You can use the cmdlets to create, test, deploy, and manage solutions and services delivered through the Azure platform.
- Since PowerShell 6.x can run on **Linux**, **macOS**, and **Windows**, Azure PowerShell is now available for all platforms.
- 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.
- 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 uses the **.NET Standard library**, which means it runs on PowerShell 5.x and PowerShell 6.x.

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

In administrator mode, Open Standard Windows PowerShell console,

```
Set-ExecutionPolicy Remote-Signed
Install-Module -Name Az -AllowClobber -Force
Import-Module Az
```

If already existing, use following command to update

```
Update-Module -Name Az
```

Note:

- PowerShell Gallery Modules Location: **C:\Program Files\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-AzResource -Detailed
```

Formatting Output:

https://docs.microsoft.com/en-us/powershell/scripting/samples/using-format-commands-to-change-output-view

Managing Azure Accounts and Subscriptions

```
# To log in to Azure Resource Manager
Login-AzAccount


# To view all subscriptions for your account
Get-AzSubscription


# To select a default subscription for your current session
Get-AzSubscription –Subscription “your sub” | Select-AzSubscription
```

```
Select-AzSubscription -Subscription "your sub"
Or
Set-AzContext -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-AzContext
```

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-AzAccount
   ```

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

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

Delete a resource group

```
Remove-AzResourceGroup -Name $rgName
```

Get the list of all resource groups

```
Get-AzResourceGroup
```

Moves a resource to a different resource group or subscription.

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

Searches for resources based on specified parameters:

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

Managing an App Service Plans

List all Existing

```
Get-AzAppServicePlan -ResourceGroupname DemoRG
```

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

Create

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

To change the Tier

```
Set-AzAppServicePlan -Name DemoPlan -ResourceGroupName DemoRG -Tier Basic
```

Delete

```
Remove-AzAppServicePlan -Name DemoPlan -ResourceGroupName DemoRG
```

Manage an Azure Web Apps

List existing Web Apps

```
Get-AzWebApp -ResourceGroupname DemoRG
```

Create:

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

Delete:

```
Remove-AzWebApp -Name ContosoWebApp -ResourceGroupName ContosoAzureResourceGroup
```

Configure an existing Web App

```
$appsettings = @{Key1 = "Key1value"; Key2 = "Key2value"}
Set-AzWebApp -Name DssDemoWebApp -ResourceGroupName DemoRG -AppSettings $appsettings
```

Start / Stop / Restart:

```
Start-AzWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG
Stop-AzWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG
```

Restart-AzWebapp -Name DssDemoWebApp -ResourceGroupName DemoRG

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 x`
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.

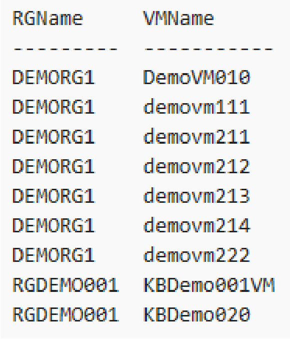
Example:

- 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
```