Lab: Automating Azure SQL Database

Lab Overview

In this lab, you will be introduced to using the Azure PowerShell cmdlets to automate tasks with Azure

Exercise 1: Create an Azure SQL Server and Database

In this exercise you will create an Azure SQL Server and Database instance using the Azure PowerShell cmdlets.

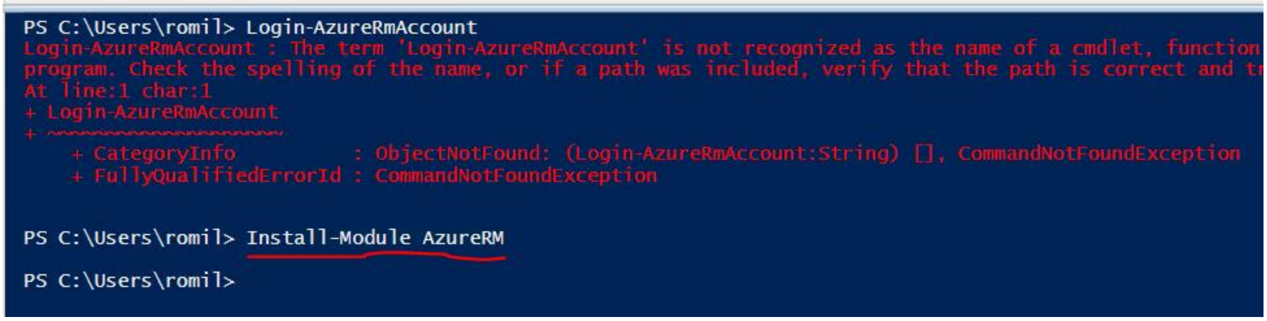
1. Launch PowerShell_ISE by clicking the PowerShell_ISE icon.

2. In the PowerShell ISE console execute the Login-AzureRmAccount cmdlet to login to Azure.

```
Login-AzureRmAccount
```

Note:If Login-AzureRmAccount command is not working please install Azure Resource Manager module

```
Install-Module AzureRM
```



```
PS C:\Users\romil> Login-AzureRmAccount
Login-AzureRmAccount : The term 'Login-AzureRmAccount' is not recognized as the name of a cmdlet, function
program. Check the spelling of the name, or if a path was included, verify that the path is correct and tr
At line:1 char:1
+ Login-AzureRmAccount
+ ~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Login-AzureRmAccount:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\romil> Install-Module AzureRM

PS C:\Users\romil>
```

3. To view the available cmdlets for managing an Azure SQL Database server execute the following command:

```
Get-Command | where name -Like "*azurerm*sql*"
```

4. Execute the following commands to create variables for the lab.
   a. $rgName should be set to the value SQLPSDEMO-RG.
   b. $sqlServerName should be a unique name for your SQL Server instance.
   c. $location should point to the region you wish to deploy to.

```
$rgName        = "SQLPSDEMO-RG"
$sqlServerName = "[unique name for the SQL server instance]"
$location      = "[location value]"
```

5. Execute the Get-Credential cmdlet and store the result in a variable named $credentials.

```
$credentials   = Get-Credential
```

6. Create the new resource group by using the New-AzureRmResourceGroup cmdlet.

```
New-AzureRmResourceGroup -Name $rgName -Location $location
```

7. Use the New-AzureRmSQLServer cmdlet to create the SQL Database Server instance.

```
<# Note: you can specify all of the parameters on the same line, just
remember to remove the line continuation character `.#>

New-AzureRmSqlServer -ServerName $sqlServerName `
                     -SqlAdministratorCredentials $credentials `
                     -Location $location `
                     -ServerVersion "12.0" `
                     -ResourceGroupName $rgName
```

8. Declare two variables that will contain the start IP and the end IP for a firewall rule to allow connectivity to the virtual machine.

For normal operations this would either be a range of known addresses, or a specific address such as the public IP address of your web server. In this case we're opening it up completely for simplicity.

```
$startIP = "0.0.0.0"
$endIP   = "255.255.255.255"
```

9. Execute the following code to create the new rule.

```
New-AzureRmSqlServerFirewallRule -ResourceGroupName $rgName `
                                -ServerName $sqlServerName `
                                 -FirewallRuleName "TestRule" `
                                 -StartIpAddress $startIP `
                                 -EndIpAddress $endIP
```

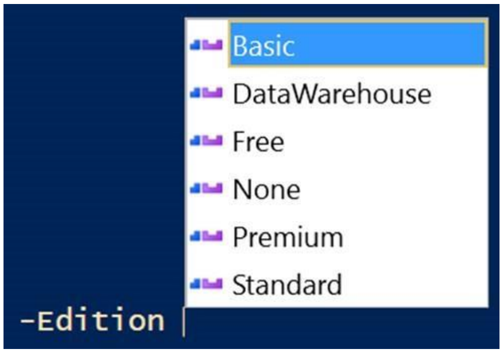
10. Declare a variable named $dbName and assign it the value PSDB, and create another variable called service tier to hold the value S1.

```
$dbName           = "SQLPSDB"
$performanceLevel = "S1"
```

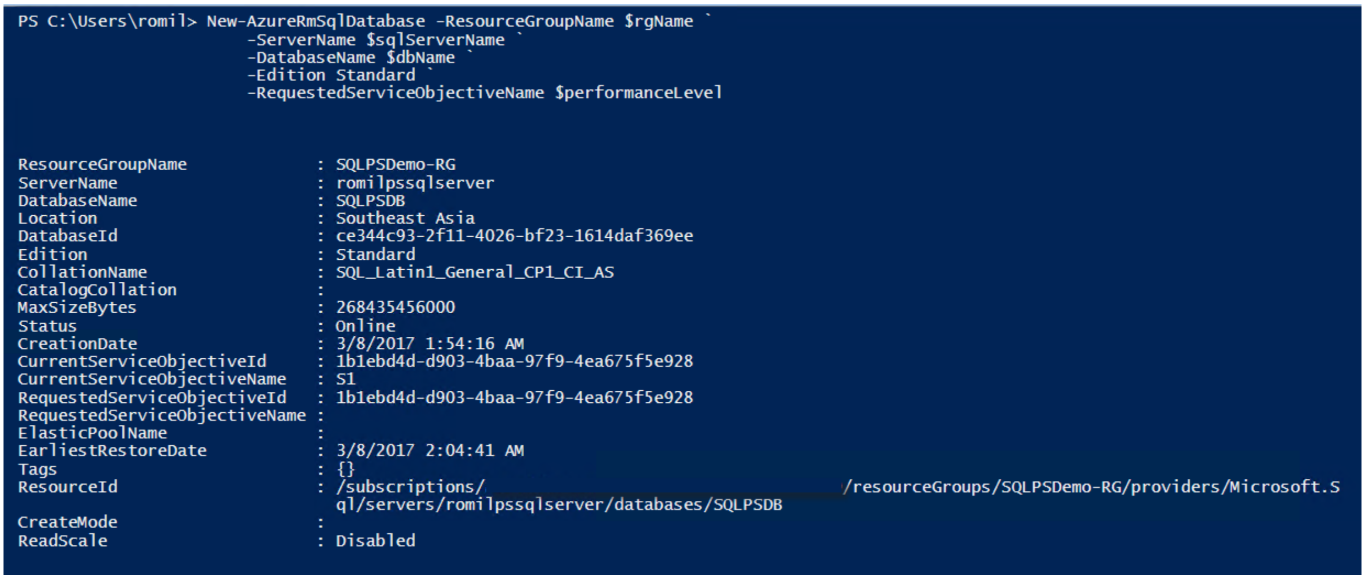
11. Use the New-AzureRmSqlDatabase cmdlet to create the SQL database instance.

```
New-AzureRmSqlDatabase -ResourceGroupName $rgName `
                       -ServerName $sqlServerName `
                       -DatabaseName $dbName `
                       -Edition Standard `
                       -RequestedServiceObjectiveName $performanceLevel
```

Note: The –Edition parameter allows you to select the Pricing Tier.



```
-Edition
```



```
PS C:\Users\romil> New-AzureRmSqlDatabase -ResourceGroupName $rgName `
                       -ServerName $sqlServerName `
                       -DatabaseName $dbName `
                       -Edition Standard `
                       -RequestedServiceObjectiveName $performanceLevel



ResourceGroupName             : SQLPSDemo-RG
ServerName                    : romilpssqlserver
DatabaseName                  : SQLPSDB
Location                      : Southeast Asia
DatabaseId                    : ce344c93-2f11-4026-bf23-1614daf369ee
Edition                       : Standard
CollationName                 : SQL_Latin1_General_CP1_CI_AS
CatalogCollation              :
MaxSizeBytes                  : 268435456000
Status                        : Online
CreationDate                  : 3/8/2017 1:54:16 AM
CurrentServiceObjectiveId     : 1b1ebd4d-d903-4baa-97f9-4ea675f5e928
CurrentServiceObjectiveName   : S1
RequestedServiceObjectiveId   : 1b1ebd4d-d903-4baa-97f9-4ea675f5e928
RequestedServiceObjectiveName :
ElasticPoolName               :
EarliestRestoreDate           : 3/8/2017 2:04:41 AM
Tags                          : {}
ResourceId                    : /subscriptions/[redacted]/resourceGroups/SQLPSDemo-RG/providers/Microsoft.S
                                ql/servers/romilpssqlserver/databases/SQLPSDB
CreateMode                    :
ReadScale                     : Disabled
```

Exercise 2: Manage your SQL Database Instance

In this exercise, you will view the deployed status of your database as well as perform some common management tasks.

1. To view the current configuration of all databases (should be only one) in your resource group execute the following code:

```
Get-AzureRmSqlDatabaseExpanded -ResourceGroupName $rgName `
                               -ServerName $sqlServerName
```

2. Execute the following code to change the performance level to S3.

```
Set-AzureRmSqlDatabase -ResourceGroupName $rgName `
                       -ServerName $sqlServerName `
                       -DatabaseName $dbName `
                       -Edition Standard `
                       -RequestedServiceObjectiveName "S3"
```

3. Next, you will create a point-in-time recovery of the database you deployed. The following code will create a copy of the database from a point in time 5 minutes before.

```
$pitDate = (Get-Date).AddMinutes(-5).ToUniversalTime()
```

```
$Database = Get-AzureRmSqlDatabase -ResourceGroupName $rgName `
                                   -ServerName $sqlServerName `
                                   -DatabaseName $dbName
```