title

Subtitle



Author

Azure SQL TOC

Azure sql Consistent Copy 1

Requirements 1

Automation account 2

# Azure sql Consistent Copy

To get started right away, just tap any placeholder text (such as this) and start typing.

## Requirements

**Create Azure SQL Database:**

You require a source active Azure database. You can create it using the Azure portal, Azure CLI, PowerShell

**Create Azure Automation Account:**

Automation provides runbooks to specify your scripts in PowerShell, Python. These runbooks define scripts with the workflow to deploy your resources.

“Quote”

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.

## Azure Automation

**Define Mandatory Variables:**

At the beginning of the script, we define a mandatory parameter and ask the user to provide a new Azure SQL DB name. It stores the user input in the $NewDBName parameter.

|  |
| --- |
| PARAM  (      [Parameter(Mandatory=$true)]      [string]$NewDbName  )  Read-Host $NewDbName = "Please Enter New Azure SQL Database Name" |

**Connection Details**

In this section, we use [Get-AzAutomationConnection](https://docs.microsoft.com/en-us/powershell/module/az.automation/get-azautomationconnection?view=azps-5.3.0) retrieves metadata for AzureRunAsConnection for automation.

|  |
| --- |
| $connectionName = "AzureRunAsConnection"  # Get the connection "AzureRunAsConnection "  $servicePrincipalConnection=Get-AutomationConnection -Name $connectionName  Connect-AzAccount `          -ServicePrincipal `          -TenantId $servicePrincipalConnection.TenantId `          -ApplicationId $servicePrincipalConnection.ApplicationId `          -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint |

**Connect to Account**

Now, we use PowerShell cmdlet [Connect-AzAccount](https://docs.microsoft.com/en-us/powershell/module/az.accounts/connect-azaccount?view=azps-5.3.0) for connecting to Azure with the authenticated account. Here, we connect using the certificate-based service principal authentication.

|  |  |
| --- | --- |
| |  | | --- | | Connect-AzAccount `      -ServicePrincipal `     -TenantId $servicePrincipalConnection.TenantId `     -ApplicationId $servicePrincipalConnection.ApplicationId `     -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint | |

**Create Variables:**

Earlier, we created variables for storing the Azure server and database names. Now, we fetch the values of the variables and store them into the variables using the [Get-AutomationVariable](https://docs.microsoft.com/en-us/powershell/module/servicemanagement/azure.service/get-azureautomationvariable?view=azuresmps-4.0.0) cmdlet as below:

$SQLServerName: In this variable, we store the Azure server name

$database: It stores the source Azure SQL DB name

We use the Write-Output cmdlet to print this information on the PowerShell console.

|  |
| --- |
| #Enter the name for your server variable  $SQLServerName = Get-AutomationVariable -Name "SqlServer"  #Enter the name for your database variable  $database = Get-AutomationVariable -Name "Database"    Write-Output "Azure SQL Database server"    Write-Output $SQLServerName    Write-Output "Azure SQL Database name"  Write-Output $database |

**Copy Azure SQL Database:**

In this section, we implement the following tasks for our azure automation runbook.

Initially, it uses [Get-AzSqlDatabase](https://docs.microsoft.com/en-us/powershell/module/az.sql/get-azsqldatabase?view=azps-5.3.0) to check the target ( copy) azure SQL DB existence and stores information in the $replicadb variable

Note here that we specify the parameter value $NewDbName for checking the azure database

Our intention here is to check whether the required database ( copied database) already exists in the resource group

If the target database already exists, the script terminates and prints the message “DB name already exists”

If the target database does not exist, the script enters in the else condition

It prints a message Creating DB before starting database copy

It uses the PowerShell cmdlet [New-AzSqlDatabase](https://docs.microsoft.com/en-us/powershell/module/az.sql/new-azsqldatabase?view=azps-5.3.0) to copy the database with the specified source and target resource group, azure server and database information

Source: We specify source configuration using the -ResourceGroupName, -ServerName, -DatabaseName arguments

Target: It uses arguments -CopyResourceGroupName, -CopyServerName and -CopyDatabaseName in the script

|  |
| --- |
| $replicaDb = (Get-AzSqlDatabase -ResourceGroupName $resourceGroupName -DatabaseName $NewDbName -ServerName $SQLServerName -ErrorAction SilentlyContinue)  write-Output $replicaDb    if($replicaDb)  {      write-Output " Specified target Azure SQL Database already exists"  }  else  {      write-Output " Creating a database copy using Azure Automation runbook”      New-AzSqlDatabaseCopy -ResourceGroupName $resourceGroupName -ServerName $SQLServerName -DatabaseName $database -CopyResourceGroupName $resourceGroupName -CopyServerName $SQLServerName -CopyDatabaseName $NewDbName    } |

**Create Azure SQL Database:**

You require a source active Azure database. You can create it using the Azure portal, Azure CLI, PowerShell

|  |
| --- |
| PARAM  (      [Parameter(Mandatory=$true)]      [string]$NewDbName  )  Read-Host $NewDbName = "Please Enter New Azure SQL Database Name" |