Azure Resource Manager templates and Azure SQL(slash)MI

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:40 PM PDT

Contents

- Symptoms:
- Analysis and Mitigation:
- What is ARM template?
- PowerShell cmdlet?
- What if customer has issues like template deployment failu...
- Walk-through of this sample JSON and script execution:
- Connect to tenant and select subscription
- Create parameters
- Create Resource Group
- Deploy template and create resources using ARM template
- Successful creation on Azure portal
- Appendix:
 - Sample JSON:
 - Deployment error due to region not allowed for subscripti...
 - Internal Reference :
 - Classification

Symptoms:

Azure Resource Manager templates enable you to define your infrastructure as code and deploy your solutions to Azure cloud. These can be used to deploy and configure Azure SQL databases.

Customers choose this supporting topic when they see, deployment failures with timeout error or need help with ARM template deployment.

Analysis and Mitigation:

What is ARM template?

Azure Resource Manager (ARM) templates provide the ability to deploy resources on Azure platform through the resource manager in JSON format. This format enables service providers or administrators to deploy Azure resources including but not limited to Azure SQL like Azure SQL Database, Managed Instance, PostgreSQL or MySQL.

Create and deploy your first Azure Resource Manager template

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-create-first-template

PowerShell cmdlet?

The primary PowerShell cmdlet used for the resource manager deployments through ARM template is New-AzureRmResourceGroupDeployment. And the parameter -TemplateFile points to the JSON based ARM template that can be built to suit the requirement.

New-AzureRmResourceGroupDeployment

https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/new-<u>azurermresourcegroupdeployment</u>

What if customer has issues like template deployment failures?

Most of the scenarios in this category relate to the scripts that customers may have developed. However, simple test can be done to confirm whether the issue lies with PaaS or the script customer uses. And this can be done via a test like below.

In following sample PowerShell script uses JSON that is inspired from this blog:

https://blogs.msdn.microsoft.com/azuresqldbsupport/2018/06/26/updated-arm-template-to-deploy-serverwith-auditing-and--detection-turned-on/

The JSON can be modified for better fitting scenarios to test functionality of PaaS for customer as appropriate. The PowerShell script below though should largely remain same or similar in most cases since it needs a few basic information to provision the Azure SQL server and DB property values.

```
PowerShell:
<# Connect to your tenant:#>
Connect-AzureRmAccount
<# Choose the right subscription, either pull subscription name or ID from portal or use "Get-</p>
AzureRmSubscription" to see this list in PowerShell. #>
Select-AzureRmSubscription -Subscription YOURSUBSCRIPTIONID
<# Create parameters list - this is based on the list of parameters the JSON is expecting - in</p>
this section:
{ "$schema": "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
"contentVersion": "1.0.0.0",
"parameters": {}
Choices may need to change depending on a different scenario, for given sample JSON the
current list with given parameters is sufficient.
$parameters = @{"databaseserver"="YOURSERVERNAME";
         "databaselist" = @(@{databaseName = "YOURDATABASE1NAME";
                      collation = "SQL_Latin1_General_CP1_CI_AS";
                      databaseEdition = "Standard";
                      maxSizeBytes= "268435456000";},
                     @{databaseName = "YOURDATABASE2NAME";
                      collation = "SQL_Latin1_General_CP1_CI_AS";
                      databaseEdition = "Standard";
                      maxSizeBytes= "268435456000";})
         "firewallruleList" = @(@{name = "allowAll";
                        startlpAddress = "0.0.0.0";
                        endlpAddress = "255.255.255.255";})
         "sgladminLogin" = "YOURSQLADMIN";
```

```
"sgladminpassword" = "YOURPASSWORD!";
"emailaddresses" = "YOUREMAILID"}
```

<#Run the actual deployment through New-AzureRmResourceGroupDeployment, be sure to</p> provide your choice of RG name that needs to be present already. If wanting to create RG from within PowerShell, use New-AzureRmResourceGroup with -Name and -Location options.#>

New-AzureRmResourceGroupDeployment -ResourceGroupName YOURRGNAME -TemplateFile "YOURJSONFILELOCATION\armtemplates2.json" -TemplateParameterObject \$parameters

Walk-through of this sample JSON and script execution:

Connect to tenant and select subscription

<Connect to your tenant:#>

Connect-AzureRmAccount

<Choose the right subscription, either pull subscription name or ID from portal or use "Get-AzureRmSubscription" to see this list in PowerShell. #>

Select-AzureRmSubscription -Subscription YOURSUBSCRIPTIONID

```
Microsoft ASEA PowerShell Cmdlets, version 2.1.0.5, have loaded successfully.
PS C:\Users\tapanm> Connect-AzureRmAccount
Account
SubscriptionName : Microsoft Azure Internal Consumption
SubscriptionId :
                : 72f988bf-86f1-41af-91ab-2d7cd011db47
TenantId
                : AzureCloud
Environment
PS C:\Users\tapanm> Select-AzureRmSubscription -Subscription "Microsoft Azure Internal Consumption"
                : Microsoft Azure Internal Consumption - 95d5acb3-e6fa-47ab-a8a2-9ed64cba9ac3
Name
                : tapanm@microsoft.com
Account
SubscriptionName : Microsoft Azure Internal Consumption
TenantId
               : 72f988bf-86f1-41af-91ab-2d7cd011db47
Environment
                 : AzureCloud
```

Create parameters

<Create parameters list - this is based on the list of parameters the JSON is expecting - in this section:</p>

{ "\$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

```
"parameters": {}
```

Choices may need to change depending on a different scenario, for given sample JSON the current list with given parameters is sufficient.

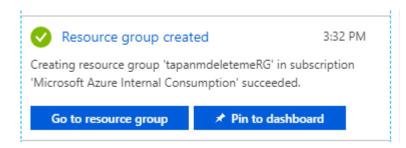
Parameters in PowerShell	JSON portion of parameters list
<pre>\$parameters = @{"databaseserver"="YOURSERVERNAME"</pre>	{ "\$schema": "https://schema.management.azure.com/schemas/2015 01-01/deploymentTemplate.json#", "contentVersion": "1.0.0.0",
"SQL_Latin1_General_CP1_CI_AS"; databaseEdition ="Standard";	"parameters": { "databaseserver": { "type": "string"
maxSizeBytes= "268435456000";}, @{databaseName = "YOURDATABASE2NAME";	}, "databaselist": {
collation = "SQL_Latin1_General_CP1_CI_AS"; databaseEdition	"type": "array", "metadata": { }
="Standard"; maxSizeBytes= "268435456000";}) "firewallruleList" = @(@{name =	<pre>}, "firewallruleList": { "type": "array",</pre>
"allowAll"; startIpAddress = "0.0.0.0"; endIpAddress =	"metadata": { } },
"255.255.255.255";}) "sqladminLogin" = "YOURSQLADMIN";	"sqladminLogin": { "type": "string" },
"sqladminpassword" = "YOURPASSWORD!"; "emailaddresses" = "YOUREMAILID"}	"sqladminpassword": { "type": "securestring" }, "emailaddresses": {

```
"type": "string"
},
```

```
databaseEdition = "Standard";
maxSizeBytes= "268435456000";},
@{databaseName = "tapanmtestdb2";
                                   collation = "SQL_Latin1_General_CP1_CI_AS";
                                  databaseEdition ="Standard";
maxSizeBytes= "268435456000";})
               "sqladminLogin" = "tapanm";
               "sqladminpassword" = "AzureSQLDB!";
               "emailaddresses" = "tapanm@microsoft.com"}
PS C:\Users\tapanm> $parameters
                             Value
Name
sqladminpassword
                             AzureSQLDB!
                             {System.Collections.Hashtable, System.Collections.Hashtable}
databaselist
firewallruleList
                             {System.Collections.Hashtable}
                             tapanm@microsoft.com
emailaddresses
sqladminLogin
                             tapanm
                             tapanmtestsqlserver
databaseserver
PS C:\Users\tapanm>
```

Create Resource Group

```
PS C:\Users\tapanm> New-AzureRmResourceGroupDeployment -ResourceGroupName tapanmdeletemeRG -TemplateFile "c:\data\CaseData\ARMTemplate-CreateSQLServerDB-WithAuditing.jsc
PS C:\Users\tapanm> |
```

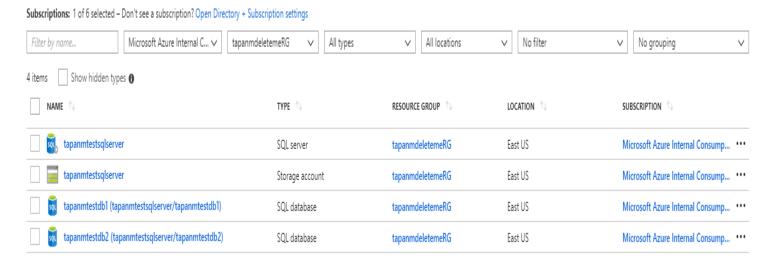


Deploy template and create resources using ARM template

New-AzureRmResourceGroupDeployment -ResourceGroupName YOURRGNAME -TemplateFile "YOURJSONFILELOCATION\armtemplates2.json" -TemplateParameterObject \$parameters

```
PS C:\Users\tapann> New-AzureRmResourceGroupDeployment -ResourceGroupDeployment -ResourceGroupDe
                                                                       : ARMTemplate-CreateSQLServerDB-WithAuditing
DeploymentName
                                                                       : tapanmdeletemeRG
ResourceGroupName
ProvisioningState
                                                                       : Succeeded
                                                                       : 7/17/2018 9:05:57 PM
Timestamp
Mode
                                                                       : Incremental
TemplateLink
Parameters
                                                                              Name
                                                                                                                                Type
                                                                                                                                                                                                                  Value
                                                                              databaseserver String
                                                                                                                                                                                                                 tapanmtestsqlserver
                                                                              databaselist
                                                                                                                              Array
                                                                                         "databaseName": "tapanmtestdb1",
"collation": "SQL_Latin1_General_CP1_CI_AS",
"databaseEdition": "Standard",
                                                                                           "maxSizeBytes": "268435456000
                                                                                            "databaseName": "tapanmtestdb2",
                                                                                           "collation": "SQL_Latin1_General_CP1_CI_AS",
                                                                                            "databaseEdition": "Standard",
                                                                                           "maxSizeBytes": "268435456000"
                                                                              firewallruleList Array
                                                                                            "startIpAddress": "0.0.0.0",
                                                                                           "name": "allowAll",
"endIpAddress": "255.255.255.255"
                                                                              sqladminLogin String
                                                                              sqladminpassword SecureString
                                                                              emailaddresses String
                                                                                                                                                                                                                  tapanm@microsoft.com
Outputs
DeploymentDebugLogLevel:
PS C:\Users\tapanm>
```

Successful creation on Azure portal



Appendix:

Sample JSON:

ARMTemplate-CreateSQLServerDB-WithAuditing.json

<< ARMTemplate-CreateSQLServerDB-WithAuditing.json>>

Deployment error due to region not allowed for subscription

If the execution fails with this error, be sure to change the region since the region in JSON provided may not be allowed to create resources in your subscription:

```
PS C:\Users\tapanmo New-AzureRmResourceGroupDeployment -ResourceGroupName tapanmdeletemeRG -TemplateFile "c:\data\CaseData\ARMTemplate-CreateSQLServerDB-WithAuditing.json" -TemplateParameterObject $parameters
```

Change this portion to the region as required in JSON file:

"databaseServerLocation": "East US",

"defaultSecondaryLocation": "West US",

Internal Reference:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-create-first-template

Classification

Root cause path - CRUD/User issue/error/Uncoded

How good have you found this content?

