

Lab Deploy a storage account and container using an ARM template

Methods: Deployed a storage account and blob container using an ARM template.

Purpose: Learn how to automate Azure resource creation through Infrastructure as Code.

Tools: Used Azure Cloud Shell with PowerShell to deploy and verify resources.

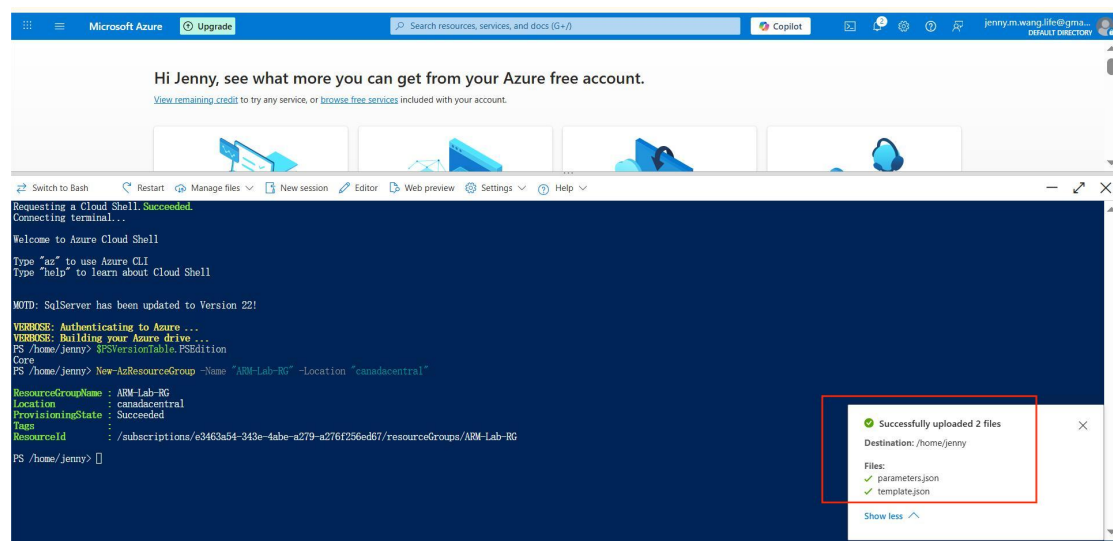
Skills Covered: Created, parameterized, and redeployed ARM templates for secure storage setup.

Step 1 – Preparing files

Create template.json and parameters.json under the lab file.

Step 2 – Deploy using CloudShell

Create a resource group, Upload both template.json and parameters.json to Cloud Shell:



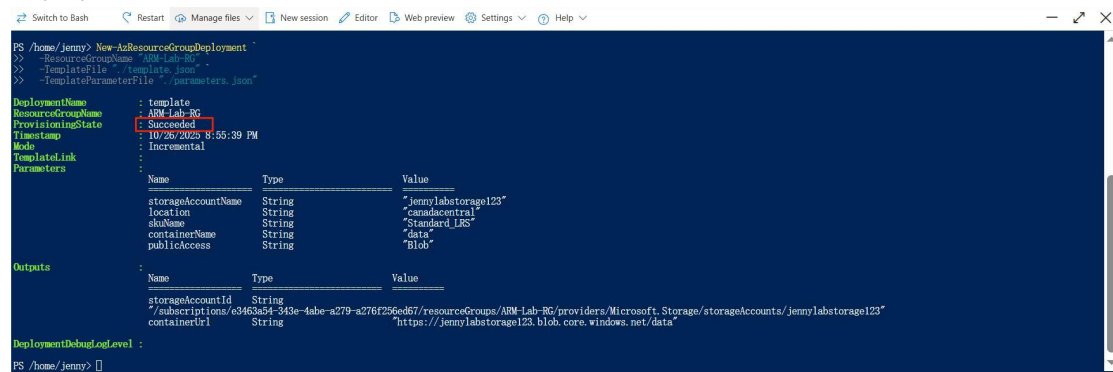
The screenshot shows the Azure Cloud Shell interface. At the top, there's a banner for Jenny's Azure free account. Below it, a terminal window shows the command `New-AzResourceGroup` being executed to create a resource group named 'ARM-Lab-RG' in the 'canadacentral' location. The command succeeds. A red box highlights a notification that says 'Successfully uploaded 2 files' to the destination '/home/jenny'. The files listed are 'parameters.json' and 'template.json'.

```
PS /home/jenny> New-AzResourceGroup -Name "ARM-Lab-RG" -Location "canadacentral"

ResourceGroupName : ARM-Lab-RG
Location           : canadacentral
ProvisioningState   : Succeeded
Tags               :
ResourceId          : /subscriptions/e3463a54-343e-4abe-a279-a276f256ed67/resourceGroups/ARM-Lab-RG

PS /home/jenny>
```

Deploy:



The screenshot shows the deployment of an ARM template using `New-AzResourceGroupDeployment`. The command is executed with the template file 'template.json' and parameter file 'parameters.json'. The deployment succeeds. The output shows the resource group name 'ARM-Lab-RG' and the provisioning state 'Succeeded'. Below the deployment details, there's a table of parameters and a table of outputs.

```
PS /home/jenny> New-AzResourceGroupDeployment -ResourceGroupName "ARM-Lab-RG" -TemplateFile ".\template.json" -TemplateParameterFile ".\parameters.json"

DeploymentName      : template
ResourceGroupName  : ARM-Lab-RG
ProvisioningState   : Succeeded
Timestamp          : 10/26/2025 8:55:39 PM
Mode               : Incremental
TemplateLink        :
Parameters          :
+-----+-----+-----+
| Name | Type | Value |
+-----+-----+-----+
| storageAccountName | String | "jennylabstorage123" |
| location | String | "canadacentral" |
| skuName | String | "Standard_LRS" |
| containerName | String | "data" |
| publicAccess | String | "Blob" |
+-----+-----+-----+

Outputs
+-----+-----+-----+
| Name | Type | Value |
+-----+-----+-----+
| storageAccountId | String | "/subscriptions/e3463a54-343e-4abe-a279-a276f256ed67/resourceGroups/ARM-Lab-RG/providers/Microsoft.Storage/storageAccounts/jennylabstorage123" |
| containerUrl | String | "https://jennylabstorage123.blob.core.windows.net/data" |
+-----+-----+-----+

DeploymentDebugLogLevel :
PS /home/jenny>
```

Step 3 – Verify deployment

List resources in the group, and get the output values from the deployment:

```

Parameters
  Name      Type      Value
  -----
  storageAccountName String  "jennylabstorage123"
  location   String  "CanadaCentral"
  skuName    String  "Standard_LRS"
  containerName String  "data"
  publicAccess String  "Blob"

Outputs
  Name      Type      Value
  -----
  storageAccountId String  "/subscriptions/e3463a54-343e-4abe-a279-a276f256ed67/resourceGroups/ARM-Lab-RG/providers/Microsoft.Storage/storageAccounts/jennylabstorage123"
  containerUrl String  "https://jennylabstorage123.blob.core.windows.net/data"

DeploymentDebug> Get-AzResource -ResourceGroupName "ARM-Lab-RG"
Name : jennylabstorage123
ResourceGroupName : ARM-Lab-RG
ResourceType : Microsoft.Storage/storageAccounts
Location : CanadaCentral
ResourceId : /subscriptions/e3463a54-343e-4abe-a279-a276f256ed67/resourceGroups/ARM-Lab-RG/providers/Microsoft.Storage/storageAccounts/jennylabstorage123
Tags : (Get-AzResourceGroupDeployment -ResourceGroupName "ARM-Lab-RG") [-1].Outputs

Key      Value
-----
storageAccountId ...
containerUrl ... (Get-AzResourceGroupDeployment -ResourceGroupName "ARM-Lab-RG") [-1].Outputs.containerUrl.Value
https://jennylabstorage123.blob.core.windows.net/data
PS /home/jenny> Get-AzResourceGroupDeployment -ResourceGroupName "ARM-Lab-RG" [-1].Outputs

```

Click the link:

The browser window shows the URL `jennylabstorage123.blob.core.windows.net/data`. Below the address bar, a message states: "This XML file does not appear to have any style information associated with it. The document tree is shown below." The XML content is as follows:

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Error>
  <Code>ResourceNotFound</Code>
  <Message>The specified resource does not exist. RequestId:41638116-201e-005a-04bb-469e07000000 Time:2025-10-26T21:03:07.4978398Z</Message>
</Error>

```

It means the storage account exists but the container is empty.
 Azure Blob Storage returns that XML when there are no blobs to show.
 To confirm the container really exists:

```

storageAccountId ...
containerUrl ... https://jennylabstorage123.blob.core.windows.net/data
Storage Account Name: jennylabstorage123
Name      PublicAccess  LastModified  IsDeleted  VersionId
----
Blob      Blob            10/26/2025 8:55:39 PM +00:00

```

My ARM template declared "allowBlobPublicAccess: true" and "publicAccess: "Blob"" sets the container so its blobs can be read without authentication. That's why the container is publicly readable, and the browser can directly open the URL.
 The container data exists with public access: Blob -- deployment is successful.
 To make the container private, I need to disable public access in both the container and storage account by editing the two parts in my template.json: "allowBlobPublicAccess": false and "publicAccess": "None":

```

{
  "kind": "StorageV2",
  "sku": { "name": "[parameters('skuName')]" },
  "properties": {
    "minimumTlsVersion": "TLS1_2",
    "allowBlobPublicAccess": false
  },
},
{
  "type": "Microsoft.Storage/storageAccounts/blobServices/containers",
  "apiVersion": "2023-01-01",
  "name": "[format('{0}/default/{1}', parameters('storageAccountName'), parameters('containerName'))]",
  "dependsOn": [
    "[resourceId('Microsoft.Storage/storageAccounts', parameters('storageAccountName'))]"
  ],
  "properties": {
    "publicAccess": "None"
  }
},

```

then redeploy:

```
PS /home/jenny> New-AzResourceGroupDeployment `
>> -ResourceGroupName "ARM-Lab-RG" `
>> -TemplateFile "/template.json" `
>> -TemplateParameterFile "/parameters.json"
DeploymentName      : template
ResourceGroupName  : ARM-Lab-RG
ProvisioningState   : Succeeded
Timestamp          : 10/26/2025 9:24:45 PM
Mode               : Incremental
TemplateLink       :
Parameters
  Name      Type      Value
  ----      -
  storageAccountName String  "jennylabstorage123"
  location   String  "CanadaCentral"
  skuName    String  "Standard_LRS"
  containerName String  "data"
  publicAccess String  "Blob"

Outputs
  Name      Type      Value
  ----      -
  storageAccountId String  "/subscriptions/e3463a54-343e-44be-a279-a276f256ed67/resourceGroups/ARM-Lab-RG/providers/Microsoft.Storage/storageAccounts/jennylabstorage123"
  containerUrl String   "https://jennylabstorage123.blob.core.windows.net/data"

DeploymentDebugLogLevel :
PS /home/jenny>
```

The redeployment didn't overwrite the container's access setting. Fix it manually:

```
Microsoft Azure | Upgrade | Search resources, services, and docs (G+/) | Copilot | jenny.m.wang.life@gmail.com | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

PS /home/jenny> Get-AzStorageAccount -ResourceGroupName "ARM-Lab-RG"
StorageAccountName ResourceGroupName PrimaryLocation SkuName Kind AccessTier CreationTime ProvisioningState EnableHttpsTrafficOnly LargeFileShares
-----
jennylabstorage123 ARM-Lab-RG CanadaCentral Standard_LRS StorageV2 Hot 10/26/2025 8:55:17 PM Succeeded True

PS /home/jenny> $ctx = New-AzStorageContext -StorageAccountName "jennylabstorage123" -UseConnectedAccount
PS /home/jenny> Get-AzStorageContainer -Context $ctx | Format-List

CloudBlobContainer : Microsoft.Azure.Storage.Blob.CloudBlobContainer
Permission          : Microsoft.Azure.Storage.Blob.BlobContainerPermissions
AccessPolicy        : Azure.Storage.Blobs.Models.BlobContainerAccessPolicy
PublicAccess        : Blob
LastModified        : 10/26/2025 9:24:44 PM +00:00
ContinuationToken   :
IsDeleted           :
VersionId           :
BlobContainerClient : Azure.Storage.Blobs.BlobContainerClient
BlobContainerProperties Azure.Storage.Blobs.Models.BlobContainerProperties
Context             : Microsoft.WindowsAzure.Commands.Storage.Common.AzureStorageContext
Name                : data

PS /home/jenny> $ctx = New-AzStorageContext -StorageAccountName "jennylabstorage123" -UseConnectedAccount
PS /home/jenny> Set-AzStorageContainerAcl -Name "data" -PublicAccess Off -Context $ctx
PS /home/jenny> Get-AzStorageContainer -Context $ctx | Format-List

CloudBlobContainer : Microsoft.Azure.Storage.Blob.CloudBlobContainer
Permission          : Microsoft.Azure.Storage.Blob.BlobContainerPermissions
AccessPolicy        : Azure.Storage.Blobs.Models.BlobContainerAccessPolicy
PublicAccess        : Off
LastModified        : 10/26/2025 9:38:39 PM +00:00
ContinuationToken   :
IsDeleted           :
VersionId           :
BlobContainerClient : Azure.Storage.Blobs.BlobContainerClient
BlobContainerProperties Azure.Storage.Blobs.Models.BlobContainerProperties
Context             : Microsoft.WindowsAzure.Commands.Storage.Common.AzureStorageContext
Name                : data

PS /home/jenny>
```

My container is now private.