



AZ-104 LAB REPORT [WEEK #5]

BY

EMMANUEL MUTURIA™

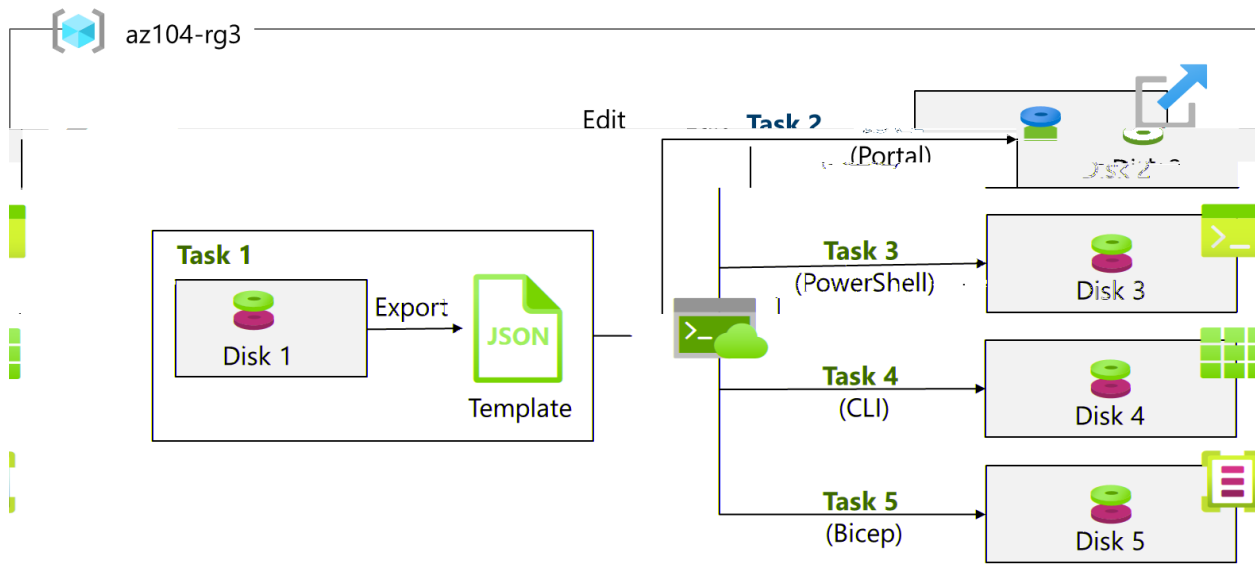
ADMISSION NUMBER
[ADC-SE02-25011]

TABLE OF CONTENTS

INTRODUCTION.....	
TASK #1 [Create an Azure Resource Manager template].....	
TASK #2 [Edit an Azure Resource Manager template and then redeploy the template].....	
TASK #3 [Configure the Cloud Shell and deploy a template with PowerShell].....	
TASK #4 [Deploy a template with the CLI].....	
TASK #5 [Deploy a resource by using Azure Bicep].....	
CONCLUSION.....	
REFERENCES.....	

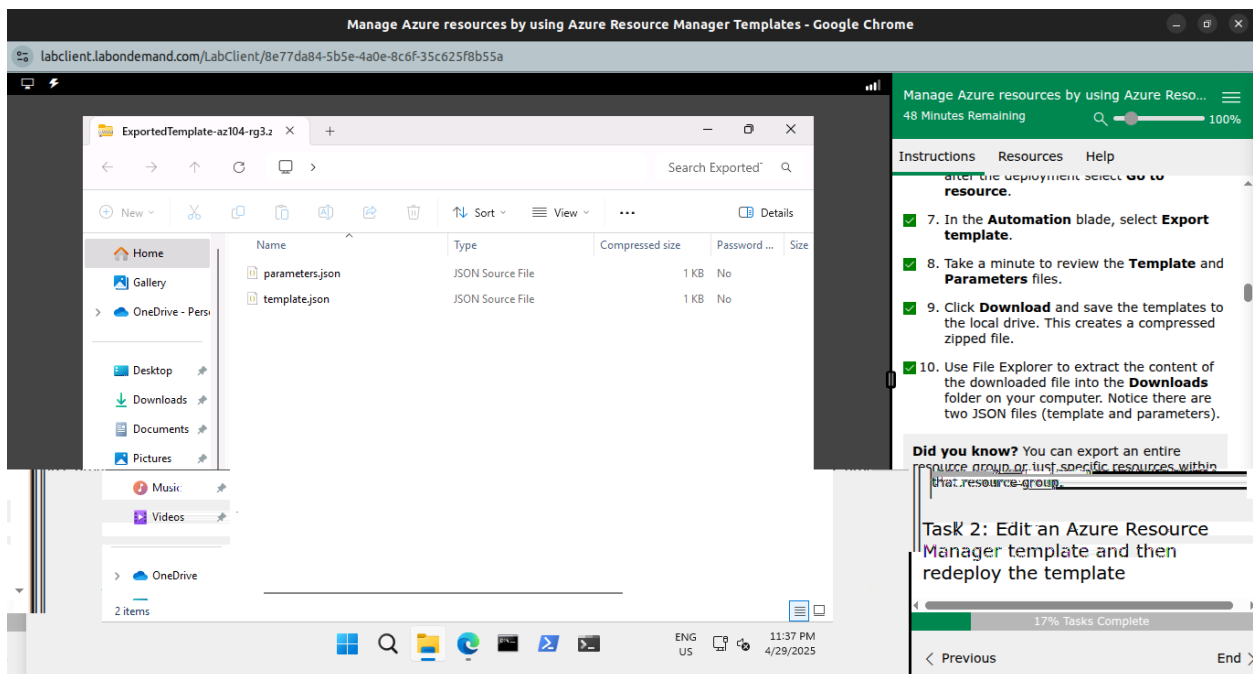
INTRODUCTION

This report documents my completion of the **Manage Azure resources by using Azure Resource Manager Templates Lab**. In this Lab, I learnt how to automate resource deployments. I learnt about Azure Resource Manager templates and Bicep templates, and the different ways of deploying the templates.



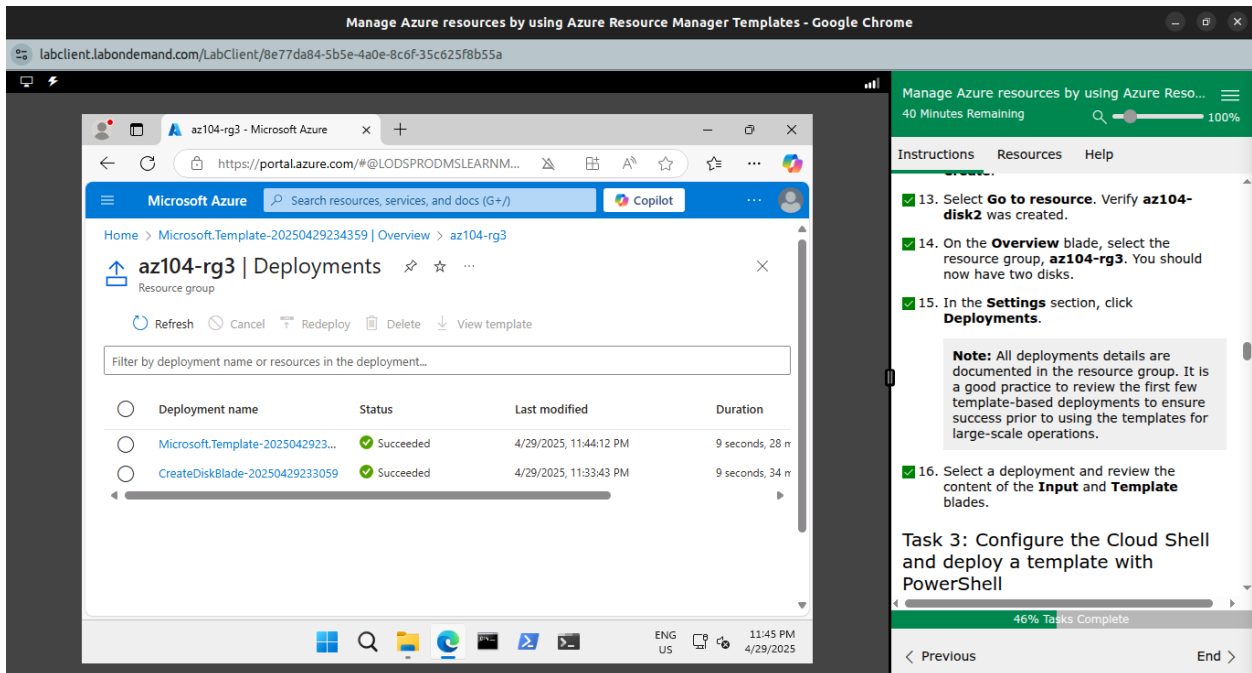
TASK #1 [Create an Azure Resource Manager template]

In this task, I created a managed disk in the Azure portal. Managed disks are storage designed to be used with virtual machines. Once the disk was deployed, I exported a template that I used in other deployments.



TASK #2 [Edit an Azure Resource Manager template and then redeploy the template]

In this task, I used the downloaded template to deploy a new managed disk. This task outlined how to quickly and easily repeat deployments.



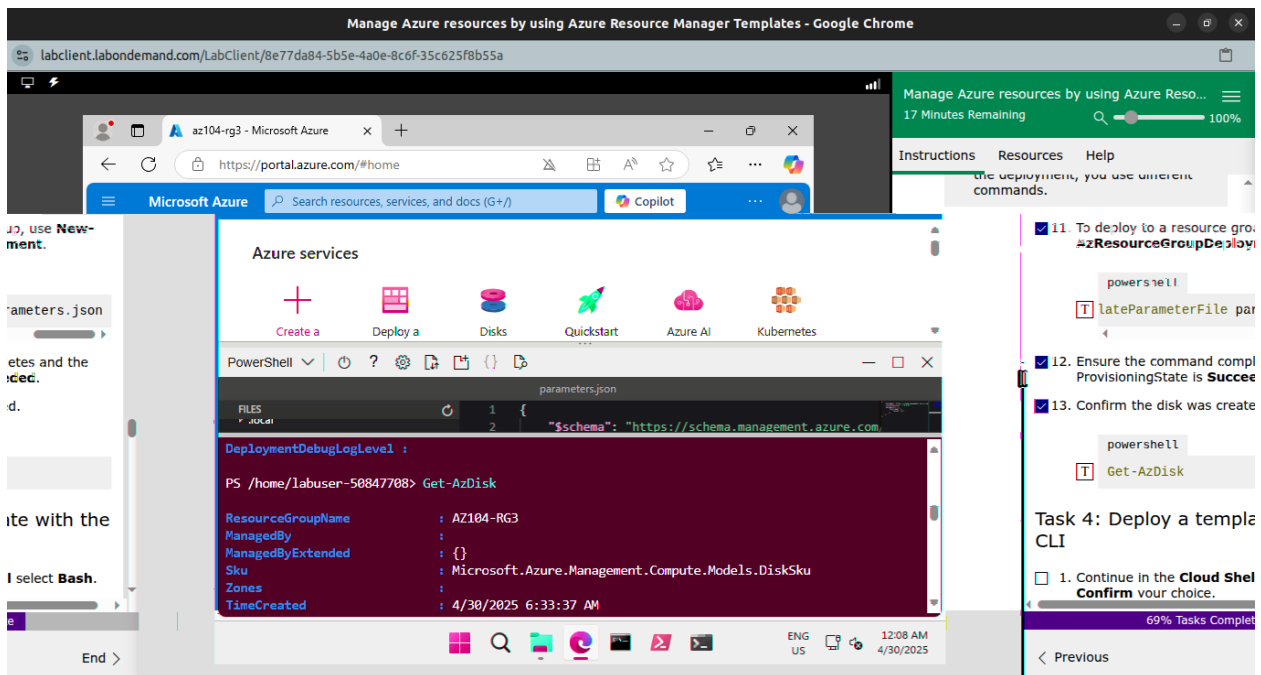
The screenshot displays the Azure portal interface for managing resources using Azure Resource Manager (ARM) templates. The main window shows the 'Deployments' page for the resource group 'az104-rg3'. It lists two successful deployments:

Deployment name	Status	Last modified	Duration
Microsoft.Template-20250429234359	Succeeded	4/29/2025, 11:44:12 PM	9 seconds, 28 n
CreateDiskBlade-20250429233059	Succeeded	4/29/2025, 11:33:43 PM	9 seconds, 34 n

To the right of the main window, a task list is visible, showing steps 13 through 16. Step 16 is the current task: 'Select a deployment and review the content of the Input and Template blades.' A progress bar at the bottom indicates '46% Tasks Complete'.

TASK #3 [Configure the Cloud Shell and deploy a template with PowerShell]

In this task, I worked with the Azure Cloud Shell and Azure PowerShell. Azure Cloud Shell is an interactive, authenticated, browser-accessible terminal for managing Azure resources. It provides the flexibility of choosing the shell experience that best suits the way you work, either Bash or PowerShell. In this task, I used PowerShell to deploy a template.



The screenshot displays the Azure Cloud Shell interface within a Google Chrome browser. The main window shows the PowerShell terminal with the following content:

```

PS /home/labuser-50847708> Get-AzDisk

ResourceGroupName : AZ104-RG3
ManagedBy         :
ManagedByExtended : {}
Sku                : Microsoft.Azure.Management.Compute.Models.DiskSku
Zones              :
TimeCreated        : 4/30/2025 6:33:37 AM
  
```

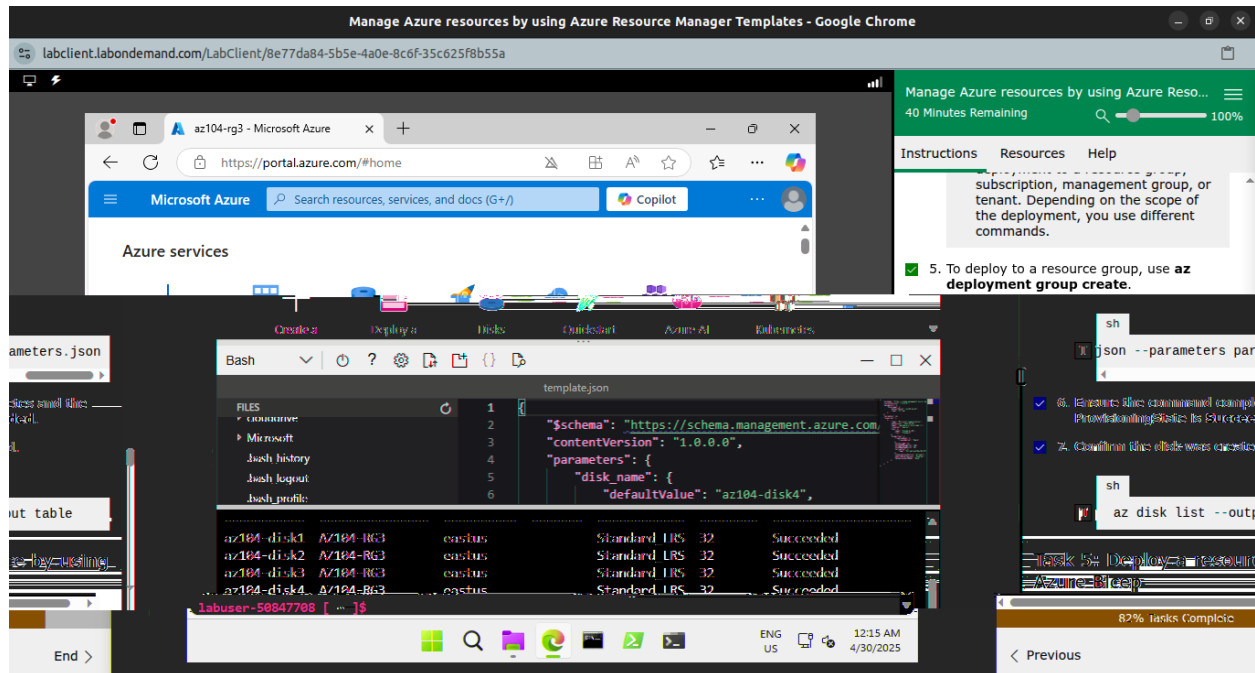
On the right side, a task list is visible, including:

- 11. To deploy to a resource group, use `azResourceGroupDeployment`.
- 12. Ensure the command completes with a `ProvisioningState` of **Succeeded**.
- 13. Confirm the disk was created.

Below the task list, the command `Get-AzDisk` is shown in a PowerShell context.

TASK #4 [Deploy a template with the CLI]

In this task, I deployed an Azure Template using Bash.



The screenshot shows a terminal window with the following content:

```

template.json
1
2
3
4
5
6
"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
"contentVersion": "1.0.0.0",
"parameters": {
  "disk_name": {
    "defaultValue": "az104-disk4",
    "type": "String"
  }
},
"resources": [
  {
    "name": "[parameters('disk_name')]",
    "type": "Microsoft.Storage/storageAccounts/disk",
    "apiVersion": "2019-07-01",
    "location": "East US",
    "sku": "Standard_LRS",
    "size": 32,
    "createOptions": "Empty"
  }
]

```

The terminal also shows the output of the `az deployment group create` command, indicating successful deployment of four disks:

```

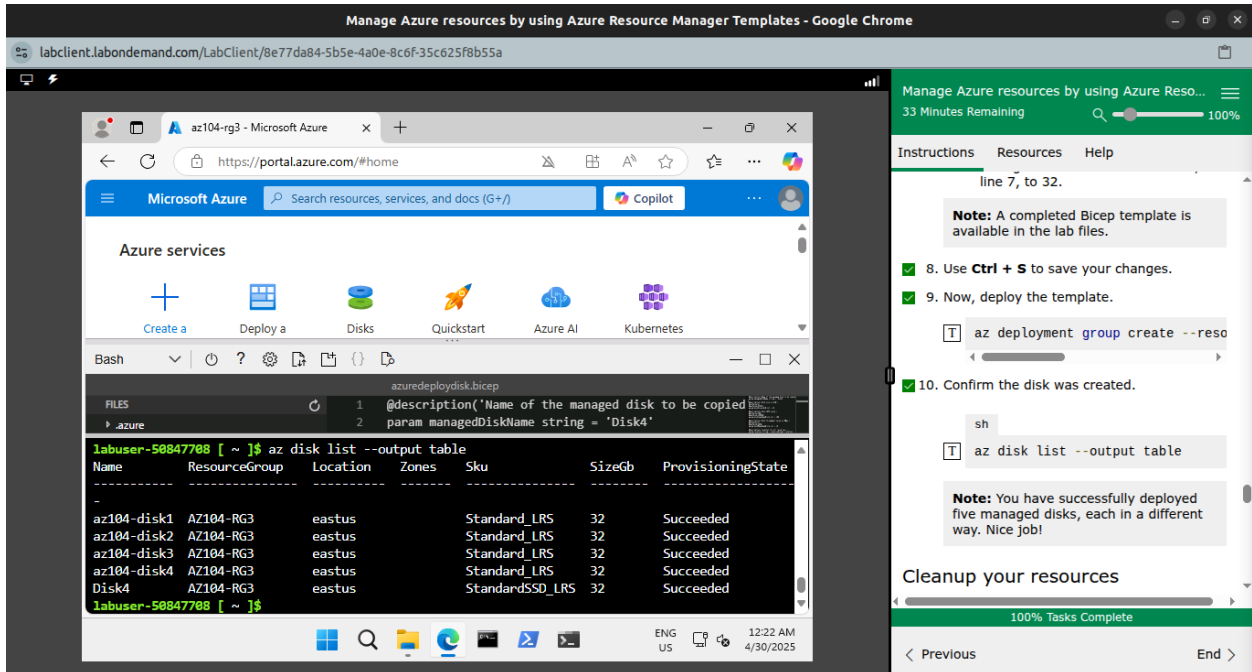
az104-disk1  AZ104-RG3  eastus  Standard_LRS  32  Succeeded
az104-disk2  AZ104-RG3  eastus  Standard_LRS  32  Succeeded
az104-disk3  AZ104-RG3  eastus  Standard_LRS  32  Succeeded
az104-disk4  AZ104-RG3  eastus  Standard_LRS  32  Succeeded

```

The terminal prompt is `labuser-50847708 [~]$`.

TASK #5 [Deploy a resource by using Azure Bicep]

In this task, I used a Bicep file to deploy a managed disk. Bicep is a declarative automation tool that is built on ARM templates.



The screenshot shows a lab environment with a terminal window and a sidebar. The terminal window displays the output of the command `az disk list --output table`, showing five disks successfully deployed. The sidebar on the right shows task instructions and progress.

Name	ResourceGroup	Location	Zones	Sku	SizeGb	ProvisioningState
az104-disk1	AZ104-RG3	eastus		Standard_LRS	32	Succeeded
az104-disk2	AZ104-RG3	eastus		Standard_LRS	32	Succeeded
az104-disk3	AZ104-RG3	eastus		Standard_LRS	32	Succeeded
az104-disk4	AZ104-RG3	eastus		Standard_LRS	32	Succeeded
Disk4	AZ104-RG3	eastus		StandardSSD_LRS	32	Succeeded

The sidebar on the right shows the following instructions:

- line 7, to 32.
- Note:** A completed Bicep template is available in the lab files.
- 8. Use **Ctrl + S** to save your changes.
- 9. Now, deploy the template.
- 10. Confirm the disk was created.
- Note:** You have successfully deployed five managed disks, each in a different way. Nice job!

The progress bar indicates 100% Tasks Complete.

CONCLUSION

This lab helped solidify the management of Azure resources using Azure Resource Manager Templates. I learnt that Azure Resource Manager templates let you deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually. An Azure Resource Manager template is a JavaScript Object Notation [JSON] file that lets you manage your infrastructure declaratively rather than with scripts. Rather than passing parameters as inline values in your template, you can use a separate JSON file that contains the parameter values. Azure Resource Manager templates can be deployed in a variety of ways, including the Azure portal, Azure PowerShell, and CLI. Interestingly, I learnt that Bicep is an alternative to Azure Resource Manager templates. Bicep uses a declarative syntax to deploy Azure resources. Bicep provides concise syntax, reliable type safety, and support for code reuse. Bicep offers a first-class authoring experience for your infrastructure-as-code solutions in Azure.

REFERENCES

Login - *Skillable* *TMS.* (2024). Learnondemand.net.
[https://msle.learnondemand.net/Lab/64587?
instructionSetLang=en&classId=676661](https://msle.learnondemand.net/Lab/64587?instructionSetLang=en&classId=676661)