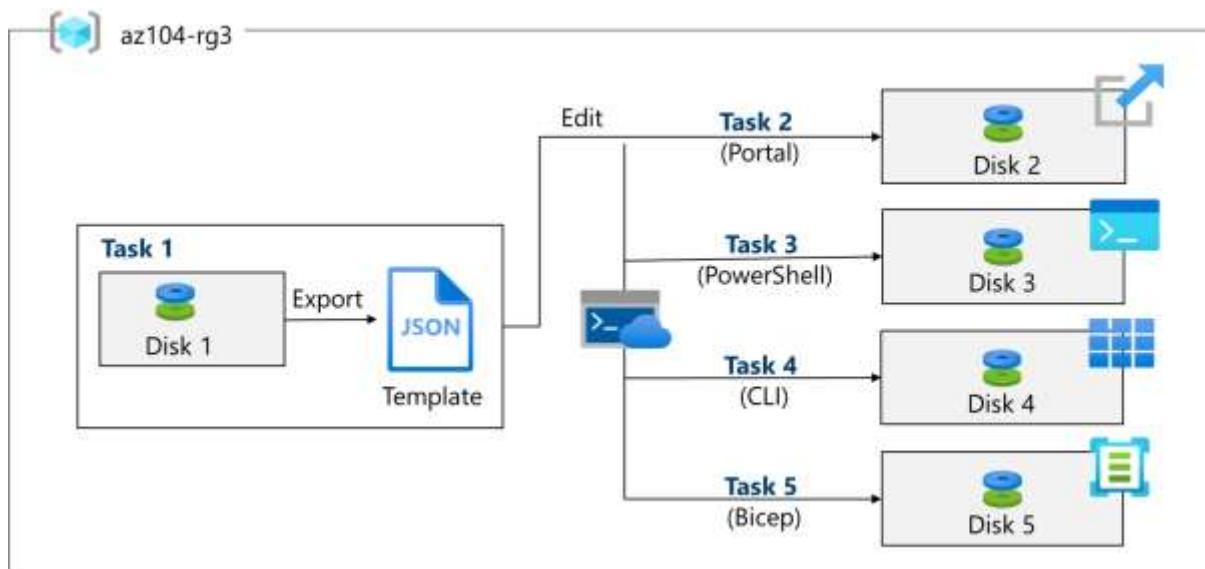


## Manage Azure resources by using Azure Resource Manager Templates



### Objective

Automate resource deployment using Azure Resource Manager (ARM) templates and Bicep. Deploy five managed disks using different methods: Portal, Template Editor, PowerShell, CLI, and Bicep.

### Task 1: Create a Managed Disk and Export ARM Template

#### Steps

1. Sign in to [Azure Portal](#)
2. Search for **Disk** → Click **Create**
3. Configure:
  - Resource Group: az104-rg3
  - Disk Name: az104-disk1
  - Region: East US
  - Performance: Standard HDD
  - Size: 32 GiB
4. Click **Review + Create** → **Create**
5. After deployment, click **Go to resource**
6. In the left pane, select **Automation** → **Export Template**
7. Click **Download** → Extract the ZIP → Locate template.json and parameters.json

#### Screenshots to Take

- ✓ Disk creation form before clicking “Review + Create”

Validation passed

Basics    Encryption    Networking    Advanced    Tags    Review + create

### Basics

Subscription	AZ-104-Recycling-1d49705936
Resource group	(new) az104-rg3
Region	East US
Disk name	az104-disk1
Availability zone	No infrastructure redundancy required
Source type	None
VM architecture	x64

### Size

Size	32 GiB
Storage type	Standard HDD LRS

- ✓ Automation blade showing export template

Home > az104-disk1

## az104-disk1 | Export template

Disk

Download Copy content Deploy Feedback

ARM Template Bicep Terraform

Include parameters ⓘ

Template Parameters

To export all resources in this resource group, navigate to the “Export template” experience under “Automation” on the left menu of the resource

<>

Parameters (1)

Variables (0)

Resources (1)

[parameters('disks\_az104\_disk1\_name') (Microsoft.Compute/disks)]

```

1  {
2      "$schema": "https://schema.management.azure.com/schemas/2019-
3      deploymentTemplate.json#",
4      "contentVersion": "1.0.0.0",
5      "parameters": {
6          "disks_az104_disk1_name": {
7              "defaultValue": "az104-disk1",
8              "type": "String"
9          }
10     }
11 }
```

---

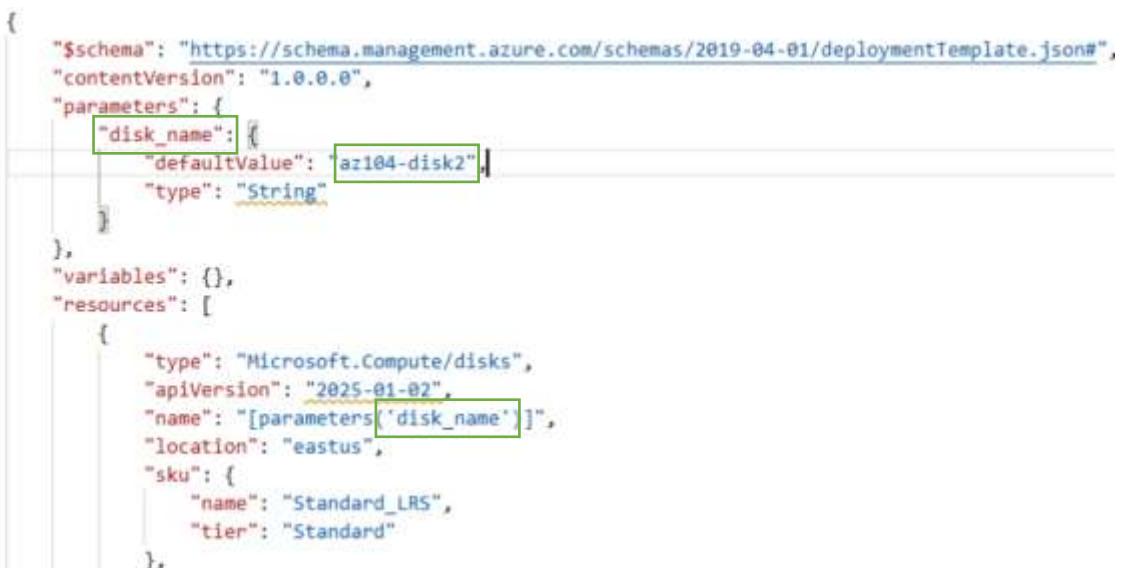
## Task 2: Edit and Redeploy ARM Template via Portal

### Steps

1. In Azure Portal, search **Deploy a custom template**
2. Click **Build your own template in the editor**
3. Load template.json → Edit:
  - o Replace disks\_az104\_disk1\_name with disk\_name (2 places)
  - o Change disk name to az104-disk2
4. Load parameters.json → Edit:
  - o Replace disks\_az104\_disk1\_name with disk\_name
5. Set deployment values:
  - o Resource Group: az104-rg3
  - o Disk Name: az104-disk2
6. Click **Review + Create** → **Create**
7. Go to resource → Confirm disk created
8. Navigate to az104-rg3 → Click **Deployments** → Review Input and Template blades

### Screenshots to Take

-  Template editor with changes highlighted



```
1  {
2    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5      "disk_name": {
6        "defaultValue": "az104-disk2",
7        "type": "String"
8      }
9    },
10   "variables": {},
11   "resources": [
12     {
13       "type": "Microsoft.Compute/disks",
14       "apiVersion": "2025-01-02",
15       "name": "[parameters('disk_name')]",
16       "location": "eastus",
17       "sku": {
18         "name": "Standard_LRS",
19         "tier": "Standard"
20       }
21     }
22   ]
23 }
```

-  Parameters editor with changes

Home >

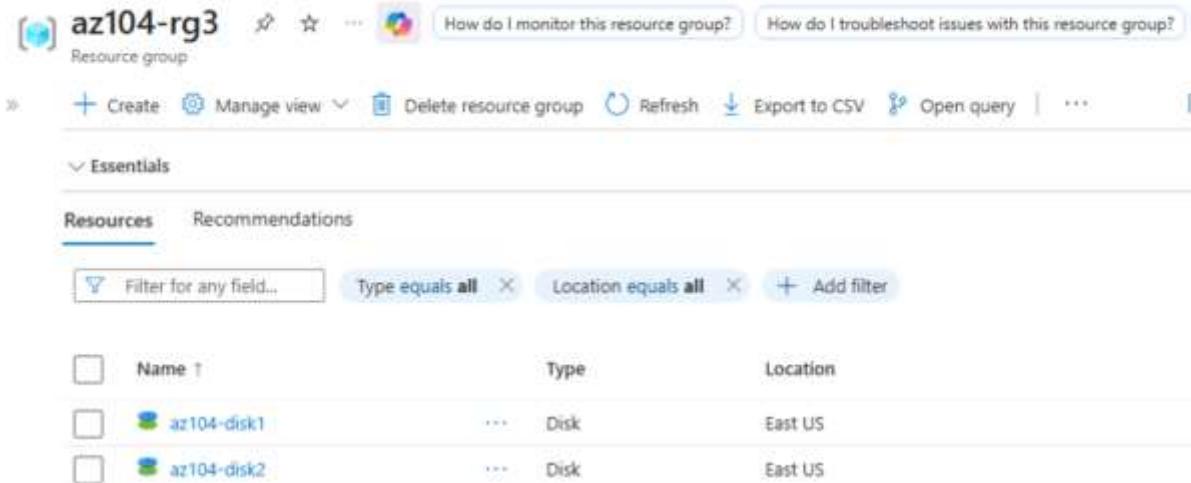
## Edit parameters

 Load file  Download

```
1 {  
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",  
3   "contentVersion": "1.0.0.0",  
4   "parameters": {  
5     "disk_name": {  
6       "value": null  
7     }  
8   }  
9 }
```

-  Resource group showing both disks

Home > Microsoft.Template-20250924053818 | Overview >



The screenshot shows the Azure portal's resource group overview for 'az104-rg3'. At the top, there are links for monitoring and troubleshooting. Below that is a toolbar with 'Create', 'Manage view', 'Delete resource group', 'Refresh', 'Export to CSV', 'Open query', and more. A filter bar allows filtering by field, type, location, and adding filters. The main table lists resources under the 'Essentials' section, with 'Resources' selected. The table has columns for Name, Type, and Location. Two entries are listed:

Name	Type	Location
az104-disk1	Disk	East US
az104-disk2	Disk	East US

-  Deployment blade showing input/template

## Microsoft.Template-20250924053818 | Inputs

Deployment

Search

Overview

Inputs

Outputs

Template

disk\_name

az104-disk2



## Microsoft.Template-20250924053818 | Template

Deployment

Search

Download

Copy content

Deploy

Feedback

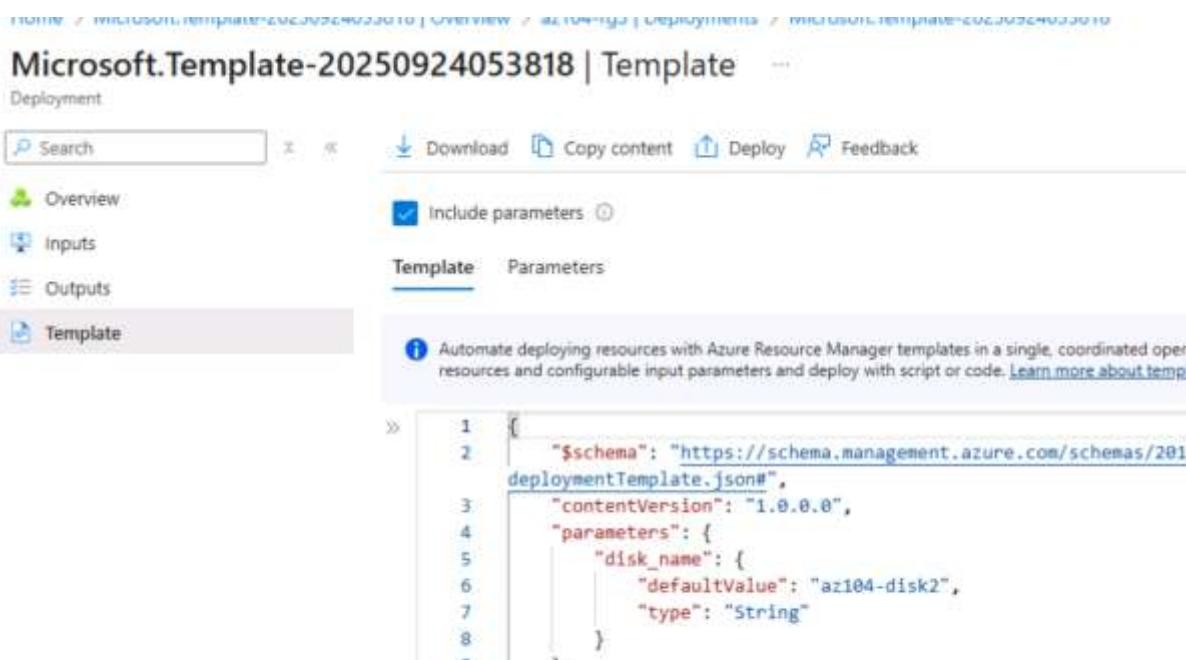
Include parameters

Template

Parameters

Automate deploying resources with Azure Resource Manager templates in a single, coordinated open resources and configurable input parameters and deploy with script or code. [Learn more about templates](#)

```
1  [
2   "$schema": "https://schema.management.azure.com/schemas/201
deploymentTemplate.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "disk_name": {
6       "defaultValue": "az104-disk2",
7       "type": "String"
8     }
9   },
10 }
```



### ✓ Task 3: Deploy Template via Cloud Shell (PowerShell)

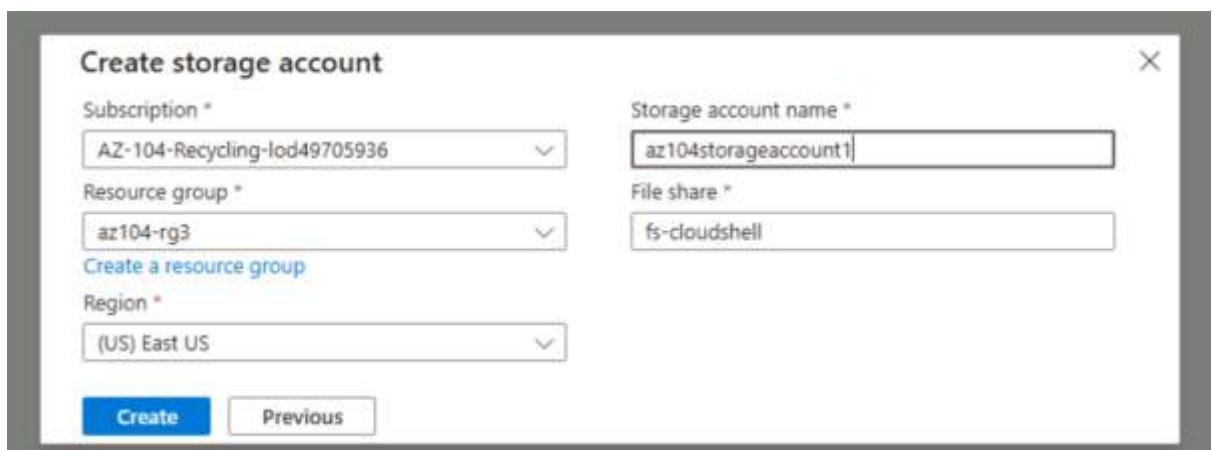
#### Steps

1. Open [Cloud Shell](#) → Select **PowerShell**
2. Mount storage → Create:
  - Storage Account: globally unique name
  - File Share: fs-cloudshell
3. Upload template.json and parameters.json
4. Open Editor → Change disk name to az104-disk3 → Save

5. Run:
6. New-AzResourceGroupDeployment -ResourceGroupName az104-rg3 -TemplateFile template.json -TemplateParameterFile parameters.json
7. Confirm success → Run:
8. Get-AzDisk | ft

#### Screenshots to Take

- Cloud Shell setup screen with storage account creation



- Editor showing updated disk name

```

1  {
2    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deployment-schema.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5      "disks_az104_disk1_name": {
6        "defaultValue": "az104-disk3",
7        "type": "String"
8      }
9    },
10   "variables": {},
11   "resources": [
12     {
13       "type": "Microsoft.Compute/disks",
14       "apiVersion": "2025-01-02",
15       "name": "[parameters('disks_az104_disk1_name')]",
16       "location": "eastus",

```

- PowerShell command output with ProvisioningState: Succeeded

```

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/labuser-54947493> New-AzResourceGroupDeployment -ResourceGroupName az104-rg3 -TemplateFile template.json --parameterFile parameters.json

DeploymentName      : template
ResourceGroupName  : az104-rg3
ProvisioningState  : Succeeded
Timestamp          : 9/24/2025 12:52:41 PM
Mode               : Incremental
TemplateLink       :
Parameters         :
    Name          Type          Value
    =============
    disks_az104_disk1_name String        "az104-disk3"

Outputs           :
DeploymentLogLevel :

PS /home/labuser-54947493> []

```

- Get-AzDisk output showing az104-disk3

```

PS /home/labuser-54947493> Get-AzDisk | ft

ResourceGroupName ManagedBy ManagedByExtended Sku
-----          ---          ---          ---
AZ104-RG3          {}          Microsoft.Azure.Management.Compute.Models.DiskSku 9/24/2025
AZ104-RG3          {}          Microsoft.Azure.Management.Compute.Models.DiskSku 9/24/2025
AZ104-RG3          {}          Microsoft.Azure.Management.Compute.Models.DiskSku 9/24/2025

PS /home/labuser-54947493>

```

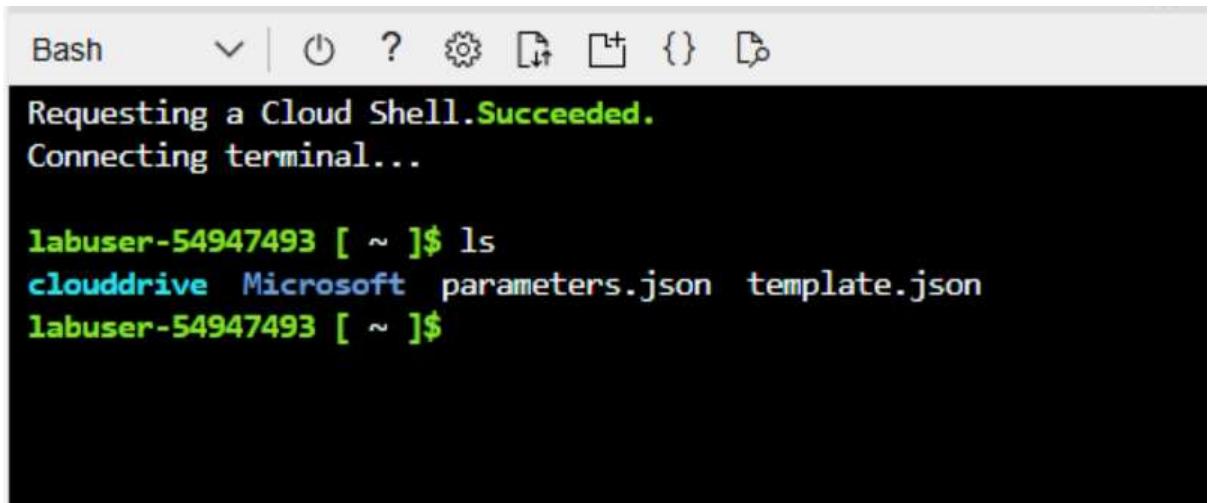
#### Task 4: Deploy Template via CLI (Bash)

##### Steps

1. Switch Cloud Shell to **Bash**
2. Confirm files exist → ls
3. Open Editor → Change disk name to az104-disk4 → Save
4. Run:
5. az deployment group create --resource-group az104-rg3 --template-file template.json --parameters parameters.json
6. Confirm success → Run:
7. az disk list --output table

##### Screenshots to Take

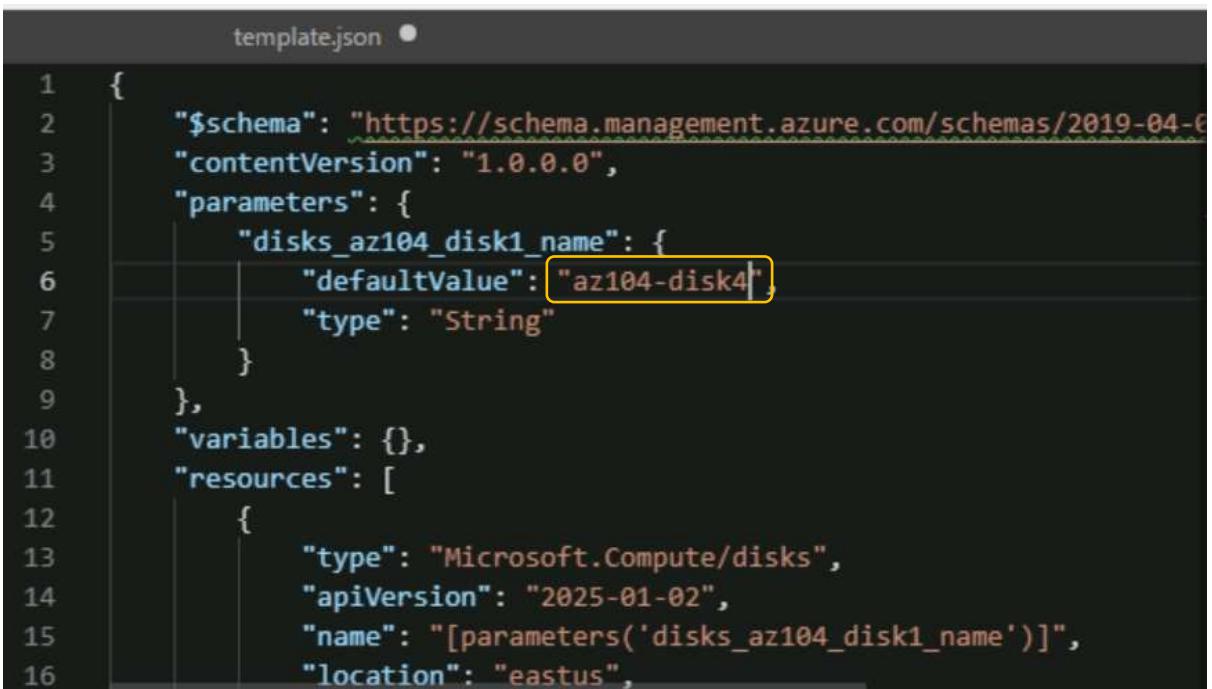
- Bash shell showing ls output



```
Bash      ▾ | ⌁ ? ⚙ ⌂ ⌂ {} ⌂
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

labuser-54947493 [ ~ ]$ ls
clouddrive Microsoft parameters.json template.json
labuser-54947493 [ ~ ]$
```

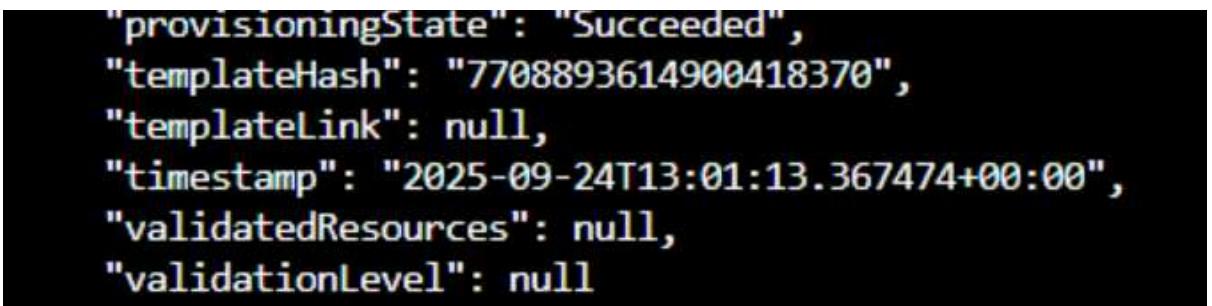
- Editor with updated disk name



```
template.json •

1  {
2      "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
3      "contentVersion": "1.0.0.0",
4      "parameters": {
5          "disks_az104_disk1_name": {
6              "defaultValue": "az104-disk4",
7              "type": "String"
8          }
9      },
10     "variables": {},
11     "resources": [
12         {
13             "type": "Microsoft.Compute/disks",
14             "apiVersion": "2025-01-02",
15             "name": "[parameters('disks_az104_disk1_name')]",
16             "location": "eastus",
17         }
18     ]
19 }
```

- CLI deployment command with success output



```
"provisioningState": "Succeeded",
"templateHash": "7708893614900418370",
"templateLink": null,
"timestamp": "2025-09-24T13:01:13.367474+00:00",
"validatedResources": null,
"validationLevel": null
```

- Disk list showing az104-disk4

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

## Task 5: Deploy Managed Disk via Bicep

### Steps

1. Upload azuredeploydisk.bicep to Cloud Shell
2. Open Editor → Make changes:
  - Line 2: az104-disk5
  - Line 7: 32
  - Line 26: StandardSSD\_LRS
3. Save → Run:
4. az deployment group create --resource-group az104-rg3 --template-file azuredeploydisk.bicep
5. Confirm success → Run:
6. az disk list --output table

### Screenshots to Take

-  Bicep file in editor with changes

```
1  @description('Name of the managed disk to be copied')
2  param managedDiskName string = 'az104-disk5'
3
4  @description('Disk size in GiB')
5  @minValue(4)
6  @maxValue(65536)
7  param diskSizeinGiB int = 8
8
```

```
resource managedDisk 'Microsoft.Compute/disks@2020-09-30' = {
  name: managedDiskName
  location: location
  sku: {
    name: 'StandardSSD_LRS'
  }
}
```

```
@description('Disk size in GiB')
@minValue(4)
@maxValue(65536)
param diskSizeinGiB int = 32
```

- ✓ CLI deployment command with success output

```
}
],
"provisioningState": "Succeeded",
"templateHash": "15558544284772728026",
"templateLink": null,
"timestamp": "2025-09-24T13:13:57.266150+00:00",
"validatedResources": null,
"validationLevel": null
},
"resourceGroup": "az104-rg3",
"tags": null,
"type": "Microsoft.Resources/deployments"
}
labuser-54947493 [ ~ ]$
```

- ✓ Disk list showing az104-disk5

```
labuser-54947493 [ ~ ]$ az disk list --output table
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
az104-disk5  AZ104-RG3  eastus   StandardSSD_LRS  32  Succeeded
labuser-54947493 [ ~ ]$
```