Microsoft

AZ-104 Administer Azure Resources

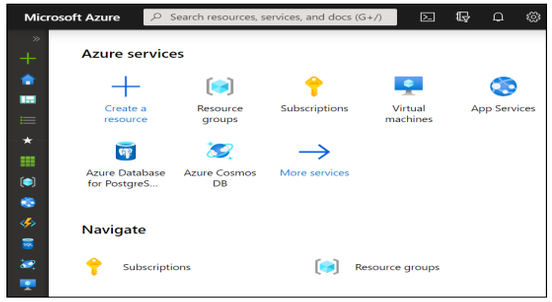
Learning Objectives - Azure Resources

- Configure Azure Resources with Tools
  - Manage services with the Azure portal
  - Introduction to PowerShell
  - Introduction to Bash
- Deploy Azure resources with templates
- Lab 03b - Manage Azure resources by Using ARM Templates

Configure Azure Resources with the Azure Portal, PowerShell, and the CLI

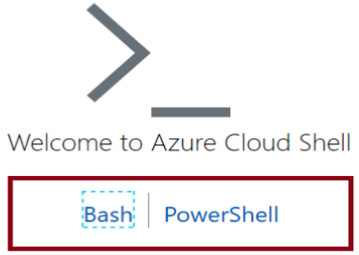
Compare Administrator tools

Azure Portal



- View and manage resources
- Visual interface
- Unified hub – training and documentation
- Personalize your experience
- Mobile app
- Access the Cloud Shell
- One-off creation scenarios

Azure Cloud Shell



- Interactive and browser-accessible with file storage
- Offers Bash or PowerShell
- Authenticates automatically
- Provided on a per-session and per-user basis
- Temporary - times out after 20 minutes

Azure PowerShell and CLI

```
az vm restart -g
MyResourceGroup -n MyVm
```

- Command line programs
- Interactive and scripting modes
- Cross-platform
- Good for repeatable deployments
- Familiar coding experience

Deploy Azure resources with templates

Learning Objectives – Deploy Azure resources using templates

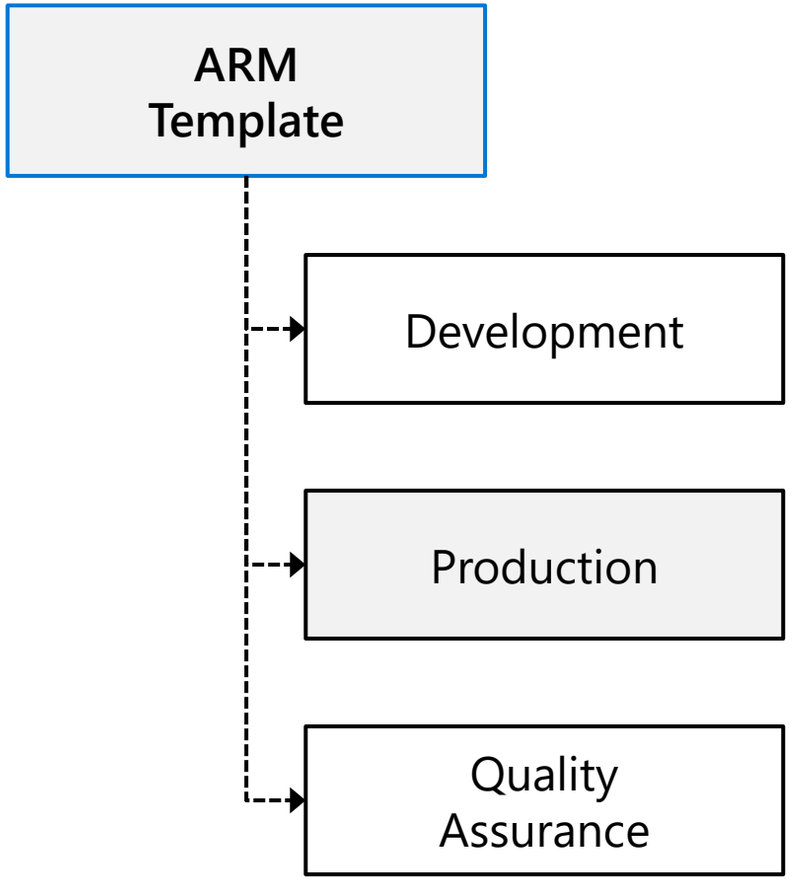
- Explore the JSON Template Schema
- Explore the JSON Template Parameters
- Consider Azure Bicep Files
- Demonstration – QuickStart Templates
- Learning Recap

Deploy and manage Azure compute resources (20–25%): Automate deployment of resources by using ARM templates or Bicep files

- Interpret an ARM template or a Bicep file
- Modify an existing ARM template
- Modify an existing Bicep file
- Deploy resources by using an ARM template or a Bicep file
- Export a deployment as an ARM template or convert an ARM template to a Bicep file

Review ARM Template Advantages

- Improves consistency and promotes reuse
- Reduce manual, error prone, and repetitive tasks
- Express complex deployments
- Express requirements through code
- Provides validation tasks
- Modular and can be linked
- Simplifies orchestration



Explore the JSON Template Schema

Defines all the Resource manager resources in a deployment

Written in JSON

A collection of key-value pairs

Each key is a string

Each value can be a string, number, Boolean expression, list of values, object

```
{
    "$schema":
      "http://schema.management.
      azure.com/schemas/2019-04-
      01/deploymentTemplate.json#",
    "contentVersion": "",
    "parameters": {},
    "variables": {},
    "functions": [],
    "resources": [],
    "outputs": {}
}
```

Explore the JSON Template Parameters

- Specifies which values are configurable when the template runs
- This example has two parameters: one for a VM's username (adminUsername), and one for its password (adminPassword)

```
"parameters": {
  "adminUsername": {
    "type": "string",
    "metadata": {
      "description": "Username for the VM."
    }
  },
  "adminPassword": {
    "type": "securestring",
    "metadata": {
      "description": "Password for the VM."
    }
  }
}
```

Consider Azure Bicep Files

Simpler syntax for writing templates

Smaller module files you can reference from a main template

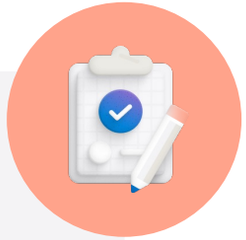
Automatically detect dependencies between your resources

Visual Studio Code extension with validation and IntelliSense

Bicep file

```
resource storageAccount
'Microsoft.Storage/storageAccounts@
2021-01-01' = {
  name: storageAccountName
  location: location
  tags: {
    displayName: storageAccountName
  }
  kind: 'StorageV2'
  sku: {
    name: 'Standard_LRS'
  } }
```

Learning Recap – Deploy Azure resources with templates



Reference modules

- Create Azure resources using Azure Resource Manager templates
- Introduction to infrastructure as code using Bicep

Check your knowledge questions and additional study

Lab - Manage Azure resources by Using ARM Templates

Lab 03 – Manage Azure resources with templates

In this lab, you learn how to define your resource infrastructure using Azure Resource Manager templates.

Templates ensure consistency and let you create multiple resources at one time.

You learn to deploy templates with the Azure portal, Azure PowerShell, or the CLI.

Job Skills

Task 1: Create an Azure Resource Manager template.

Task 2: Edit an Azure Resource Manager template and redeploy the template.

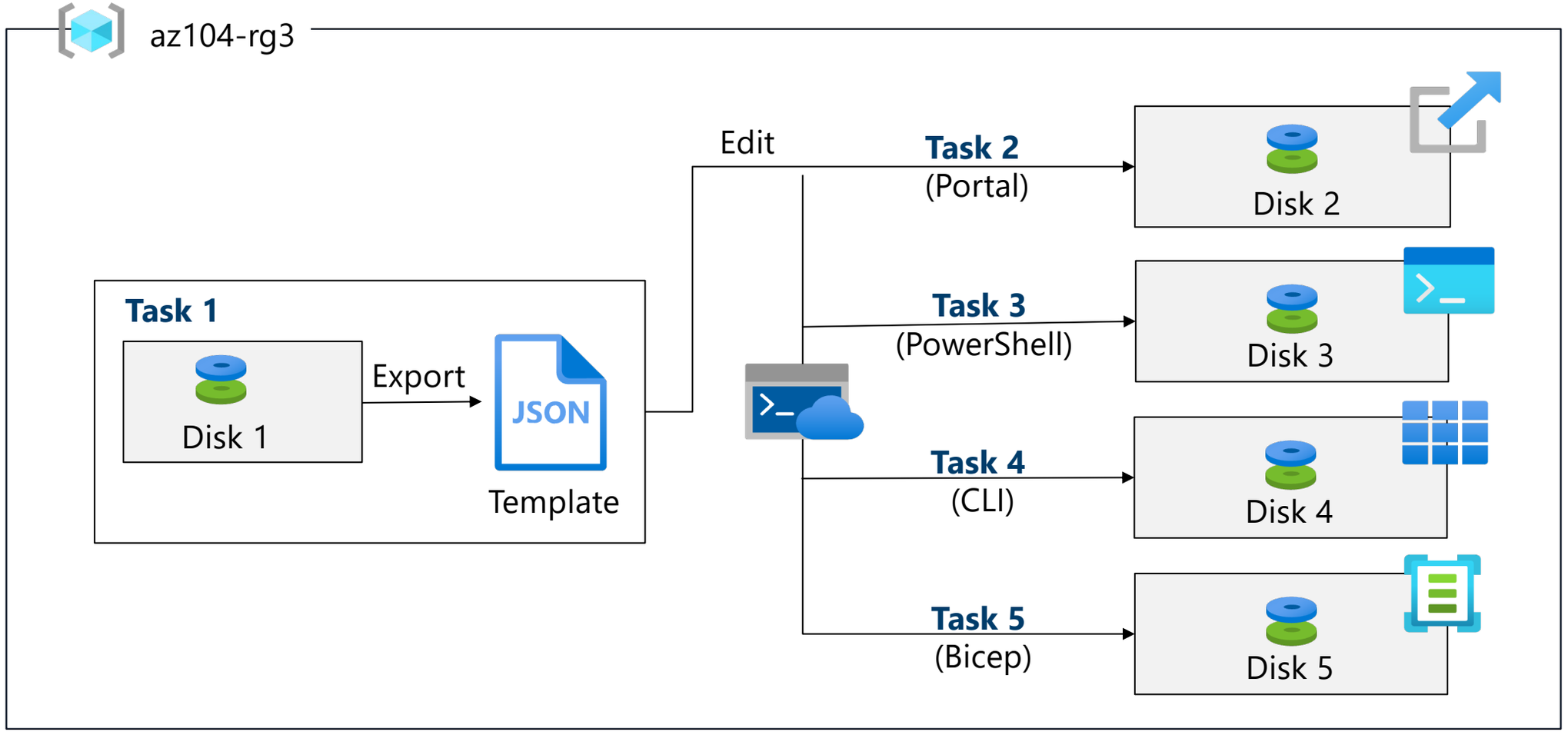
Task 3: Configure the Cloud Shell and deploy a template with Azure PowerShell.

Task 4: Deploy a template with the CLI.

Task 5: Deploy a managed disk by using Azure Bicep.

Next slide for an architecture diagram

Lab 03 – Architecture diagram



End of presentation