Microsoft

AZ-104

Tag 2

Administer Azure Resources

Guten Morgen!

AZ-104 Course Outline

01: Administer Identity
02: Administer Governance and Compliance
03: Administer Azure Resources
04: Administer Virtual Networking
05: Administer Intersite Connectivity
06: Administer Network Traffic Management
07: Administer Azure Storage
08: Administer Azure Virtual Machines
09: Administer PaaS Compute Options
10: Administer Data Protection
11: Administer Monitoring

Lab4
Lab5
Lab 6

Learning Objectives - Azure Resources

- Configure Azure Resources with Tools

- Configure Resources with ARM Templates

- Lab 03a - Manage Azure resources by Using the Azure Portal
- Lab 03b - Manage Azure resources by Using ARM Templates
- Lab 03c - Manage Azure resources by Using Azure PowerShell (optional)
- Lab 03d - Manage Azure resources by Using Azure CLI (optional)

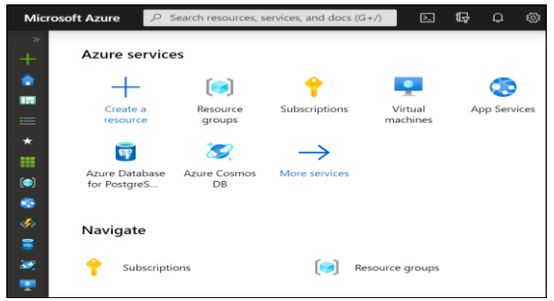
Configure Azure Resources with Tools

Compare Administrator tools

Linux (Ubuntu → Mariner)

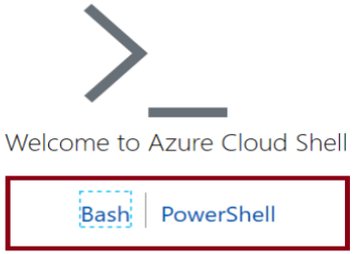
ACI

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
- 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

Review Resource Manager Benefits

ARM Azure Resource Manager 2013

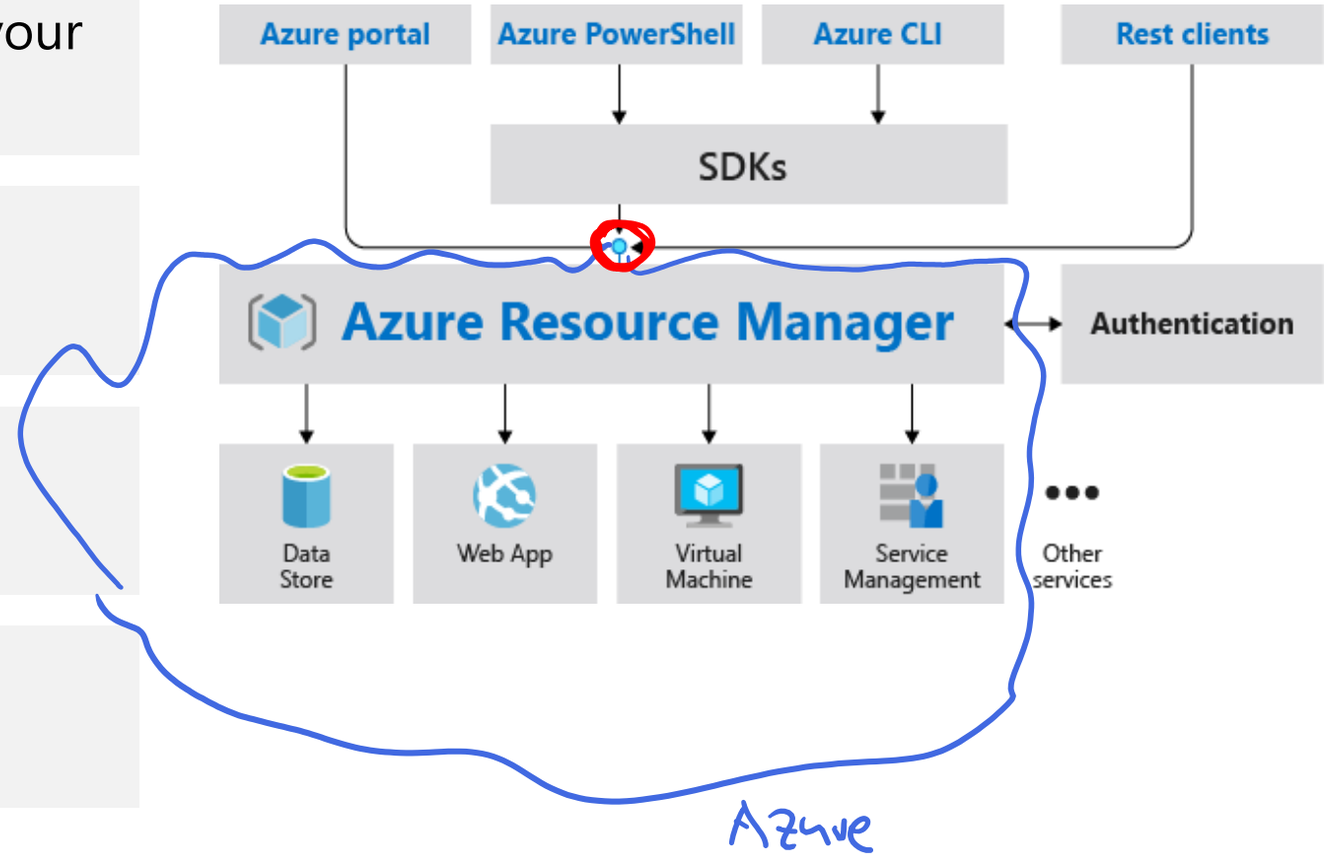
Provides a consistent management layer

Enables you to work with the resources in your solution as a group

Deploy, update, or delete in a single, coordinated operation

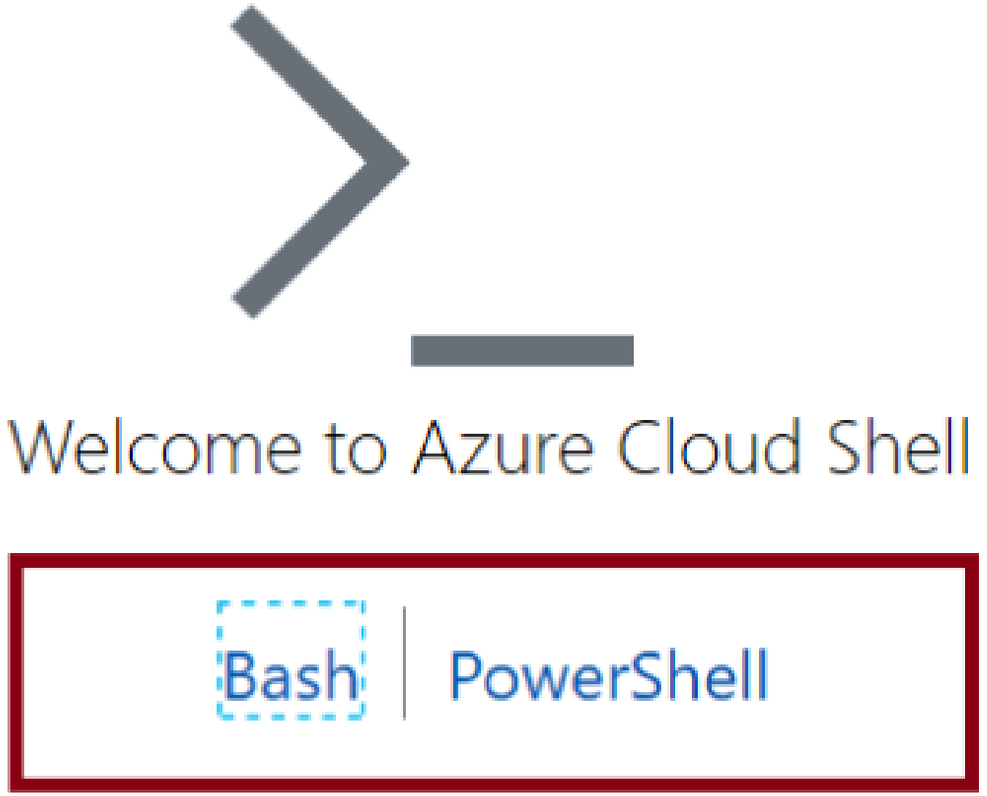
Provides security, auditing, and tagging features

Choose the tools and APIs that work best for you



Use Azure Cloud Shell

- Interactive, browser-accessible shell
- Offers either Bash or PowerShell
- Is temporary and provided on a per-session, per-user basis
- Requires a resource group, storage account, and Azure File share
- Authenticates automatically
- Integrated graphical text editor
- Is assigned one machine per user account
- Times out after 20 minutes



Use Azure PowerShell

Verb-Noun

```
New-AzVm `
  -ResourceGroupName "CrmTestingResourceGroup" `
  -Name "CrmUnitTests" `
  -Image "UbuntuLTS" `
...
```

$9 = "b"

" $9 " b

' $9 ' $9

- Connect to your Azure subscription and manage resources
- Adds the Azure-specific commands
- Available inside a browser via the Azure Cloud Shell
- Available as a local installation on Linux, macOS, or Windows
- Has an interactive and a scripting mode

Use Azure CLI

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

--name MyVm

- Cross-platform command-line program
- Runs on Linux, macOS, and Windows
- Can be used interactively or through scripts
- Commands are structured in _*groups*_ and _*subgroups*_
- Use *find* to locate commands
- Use *--help* for more detailed information

- declarative
- idempotent

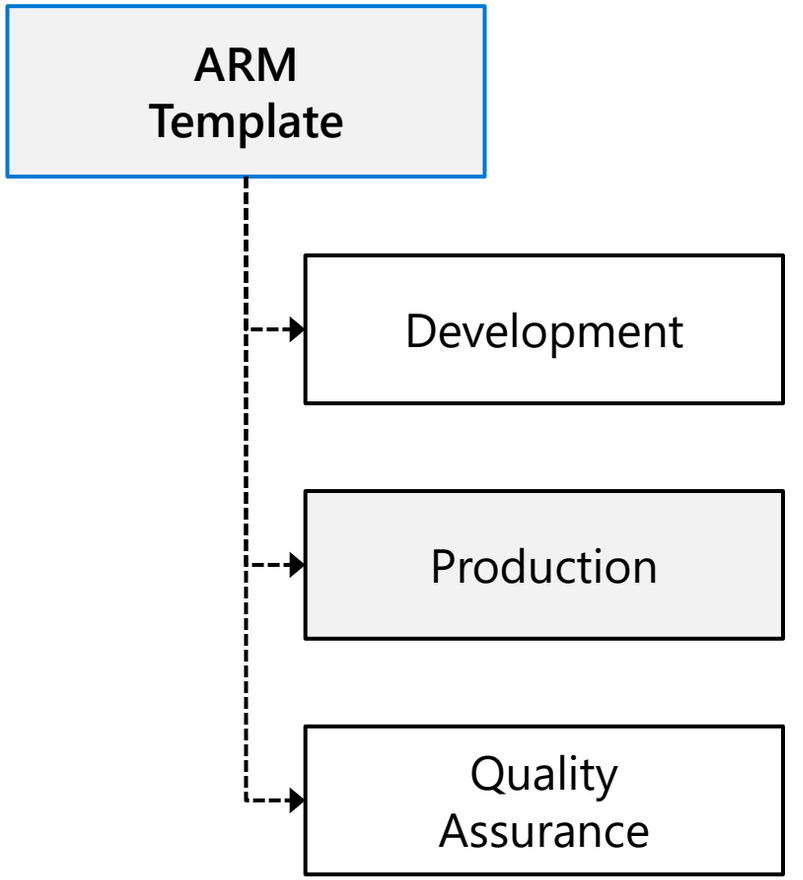
Microsoft: Bicep → json → ARM API

Terraform: HCL → ARM API

Configure Resources with ARM Templates

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

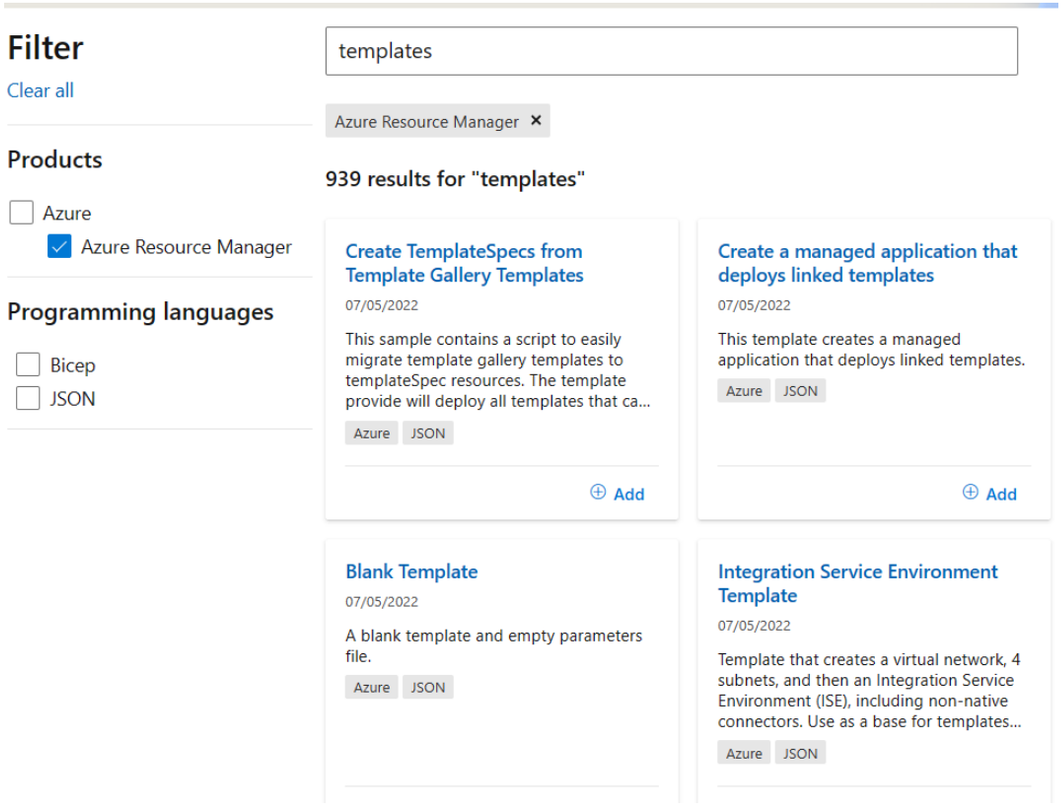
Visual Studio Code extension with validation and IntelliSense

Bicep file

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

Quickstart templates

- Locate the Azure Quickstart template gallery
- Deploy a JSON or Bicep template



Lab 03a - Manage Azure resources by Using the Azure Portal
Lab 03b - Manage Azure resources by Using ARM Templates
Lab 03c - Manage Azure resources by Using Azure PowerShell
Lab 03d - Manage Azure resources by Using Azure CLI

Lab 03b – Manage Azure resources with templates

Now that you explored the basic Azure administration capabilities associated with provisioning resources and organizing them based on resource groups, you need to carry out the equivalent task by using Azure Resource Manager templates

Objectives

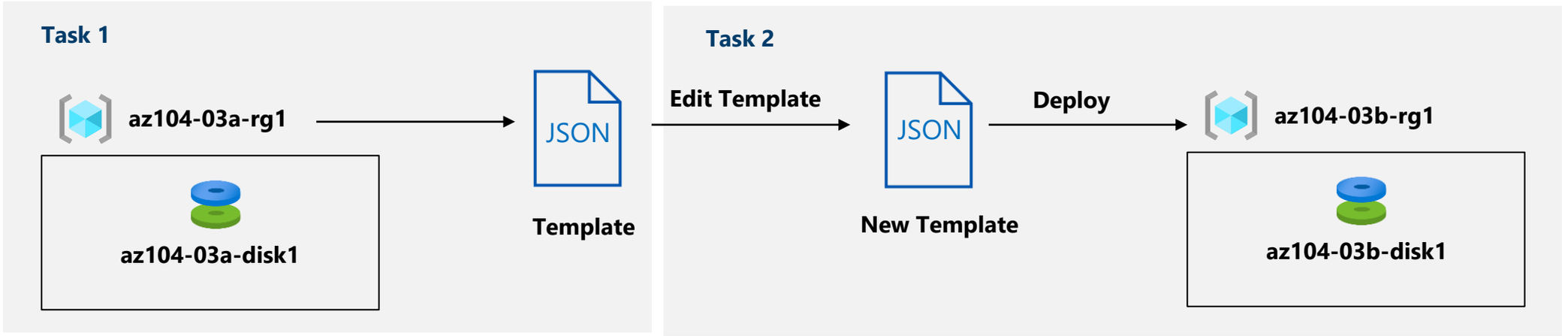
Task 1: Review an ARM template for deployment of an Azure managed disk

Task 2: Create an Azure managed disk by using an ARM template

Task 3: Review the ARM template-based deployment of the managed disk

Next slide for an architecture diagram

Lab 03b – Architecture diagram



Lab 03c – Manage Azure resources with PowerShell (optional)

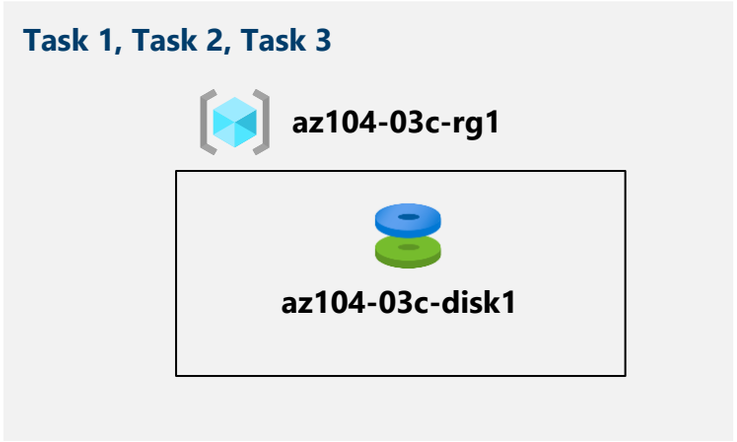
Now that you explored the basic Azure administration capabilities associated with provisioning resources and organizing them based on resource groups by using the Azure portal and Azure Resource Manager templates, you want the equivalent tasks with Azure PowerShell. To avoid installing Azure PowerShell modules, you will leverage the Azure Cloud Shell.

Objectives

Task 1: Start a PowerShell session in Azure Cloud Shell
Task 2: Create a resource group and an Azure managed disk with Azure PowerShell
Task 3: Configure the managed disk by using Azure PowerShell

Next slide for an architecture diagram

Lab 03c – Architecture diagram



Lab 03d – Manage Azure resources with the Azure CLI (optional)

Now that you explored the basic Azure administration capabilities associated with provisioning resources and organizing them based on resource groups by using the Azure portal, Azure Resource Manager templates, and Azure PowerShell, you need to carry out the equivalent task by using Azure CLI. To avoid installing Azure CLI, you will leverage Bash environment available in Azure Cloud Shell.

Objectives

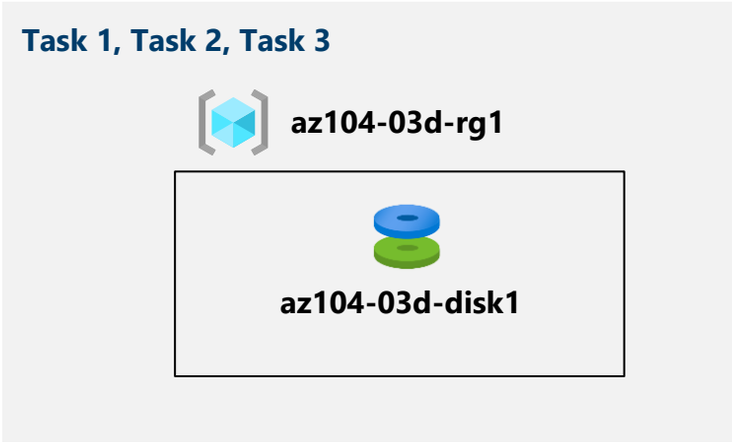
Task 1: Start a Bash session in Azure Cloud Shell

Task 2: Create a resource group and a managed disk by using Azure CLI

Task 3: Configure the managed disk by using Azure CLI

Next slide for an architecture diagram

Lab 03d – Architecture diagram



End of presentation