Microsoft Azure

# AZ-104

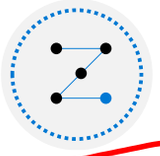
Tag 2

# Administer Azure Resources

Guten Morgen!

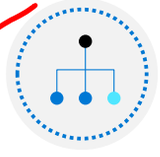
# About this course: Course Outline

01: Administer Identity



02: Administer Governance and Compliance

03: Administer Azure Resources



04: Administer Virtual Networking

05: Administer Intersite Connectivity

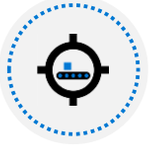
06: Administer Network Traffic Management

07: Administer Azure Storage

08: Administer Azure Virtual Machines

09: Administer PaaS Compute Options

10: Administer Data Protection



11:  Administer Monitoring

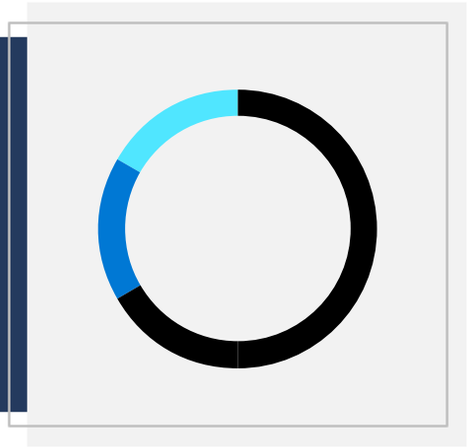
# Administer Azure Resources Introduction

Configure Azure Resources with Tools

Configure Resources with ARM Templates

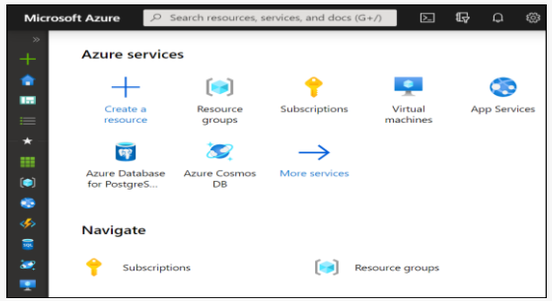
Lab 03a – Manage Azure resources with the Azure portal
Lab 03b – Manage Azure resources by using ARM Templates
Lab 03c – Manage Azure resources by using Azure PowerShell
Lab 03d – Manage Azure resources by using Azure CLI

# Configure Azure Resources with Tools



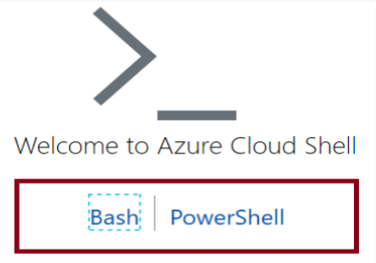
# Compare Administrator tools

## 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
- 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 PowerShell Cmdlets and Modules

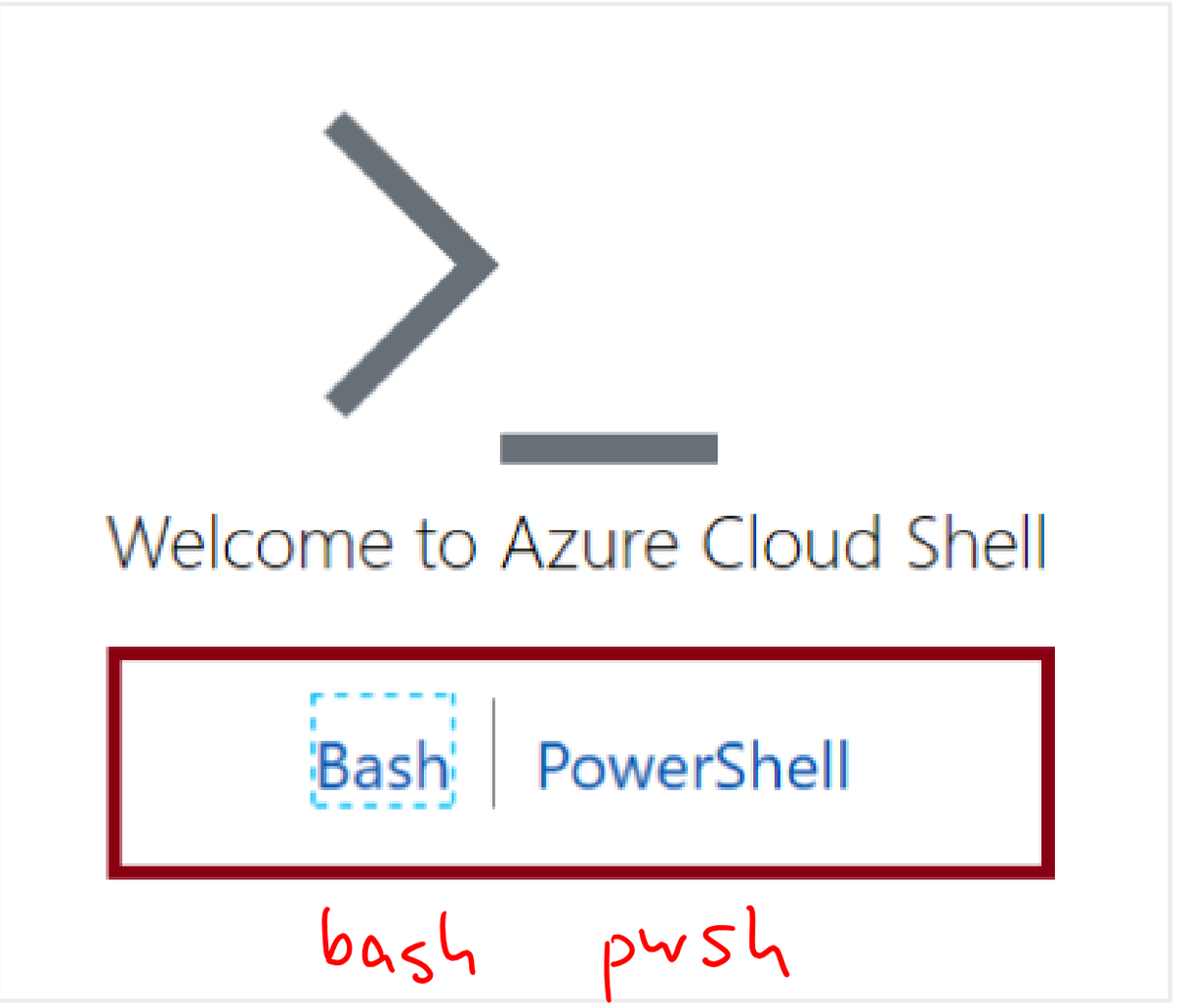
```
Get-Module
# Output
ModuleType Version    Name
---------- -------    ----
Manifest   3.1.0.0    Microsoft.PowerShell.Management
Manifest   3.1.0.0    Microsoft.PowerShell.Utility
Binary     1.0.0.1     PackageManagement
Script     1.0.0.1     PowerShellGet
Script     2.0.0      PSReadline
```

AZ

- Cmdlets follow a verb-noun naming convention; shipped in modules
- Modules are a DLL file with the code to process each cmdlet
- Load cmdlets by loading the module containing them
- Use **Get-Module** to see a list of loaded modules

# 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

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

- 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

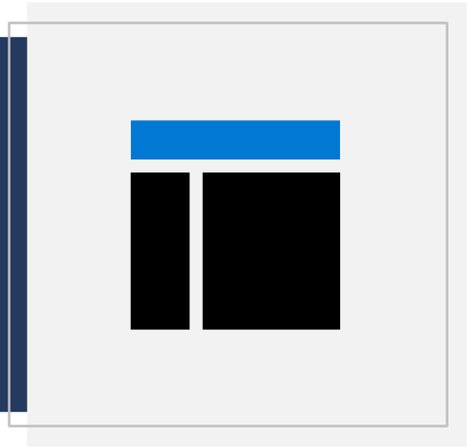
az login

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

Binary

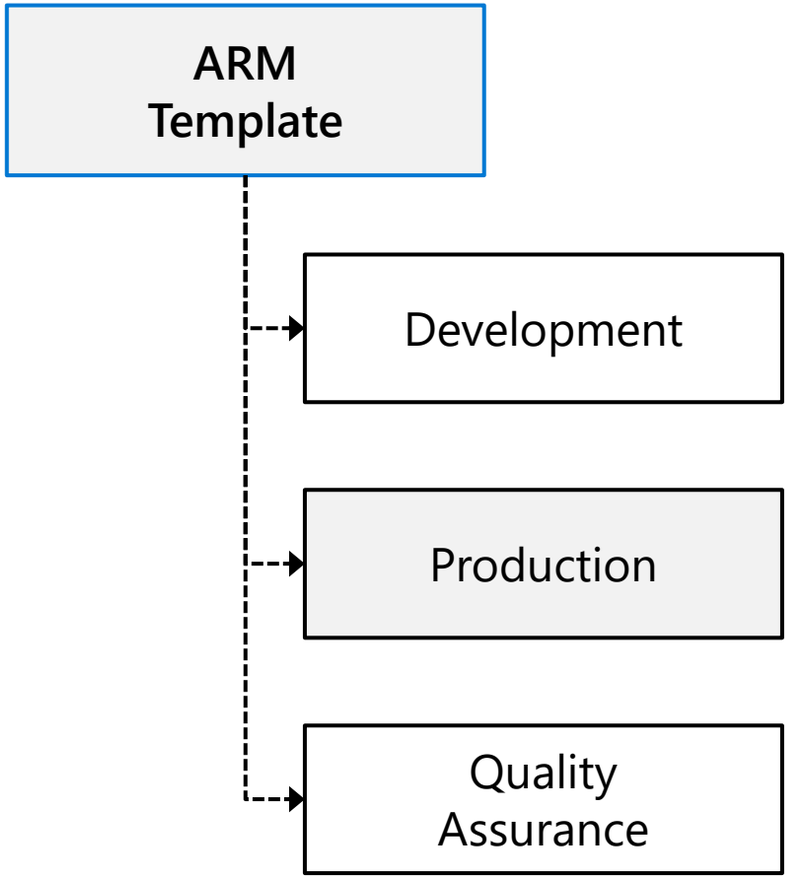
- Cross-platform command-line program
- Runs on Linux, macOS, and Windows
- Can be used interactively or through scripts
- Commands are structured in _*groups*_ and _*subgroups*_
- Use *find* to locate commands
- Use *--help* for more detailed information

# 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

Specify 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)

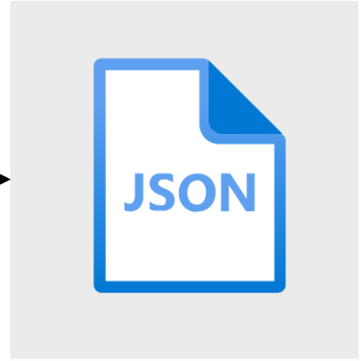
```
"parameters": {
  "adminUsername": {
    "type": "string",
    "metadata": {
      "description": "Username for the VM."
    }
  },
  "adminPassword": {
    "type": "securestring",
    "metadata": {
      "description": "Password for the VM."
    }
  }
}
```

# Consider Azure Bicep Files

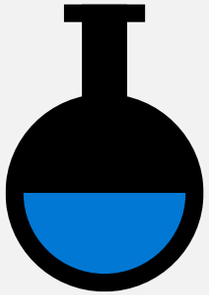
- Simpler syntax for writing templates
- Smaller module files you can reference from a main template
- Automatically detect dependencies between your resources
- Visual Studio Code extension with validation and IntelliSense

Bicep file

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



# Lab 03a - Manage Azure resources with 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 03a – Manage Azure resources with the Azure portal

## Lab scenario

You need to explore the basic Azure administration capabilities associated with provisioning resources and organizing them based on resource groups, including moving resources between resource groups. You also want to explore options for protecting disk resources from being accidentally deleted, while still allowing for modifying their performance characteristics and size

## Objectives

### Task 1:

Create resource groups and deploy resources to resource groups

### Task 2:

Move resources between resource groups

### Task 3:

Implement and test resource locks

Next slide for an architecture diagram

# Lab 03b – Manage Azure resources with templates

## Lab scenario

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, 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:

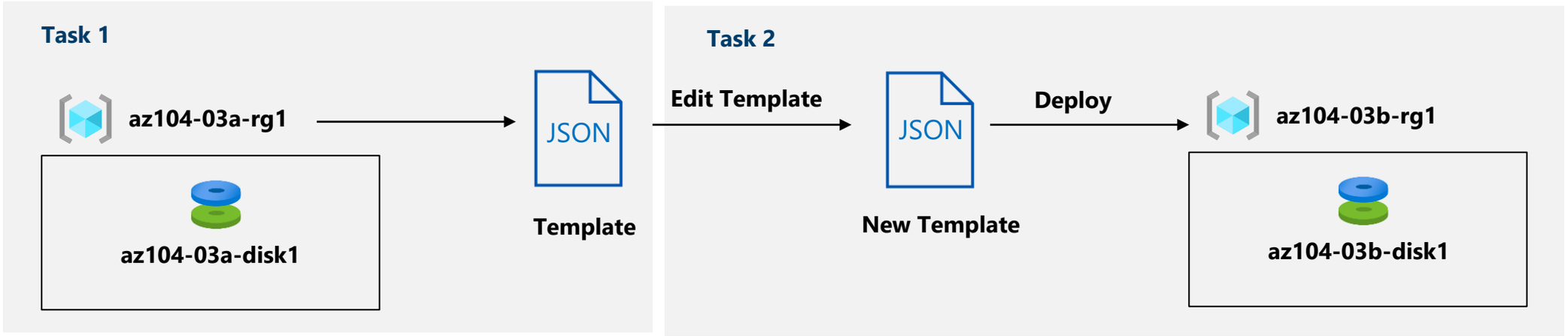
Create an Azure managed disk by using an ARM template

### 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

## Lab scenario

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:

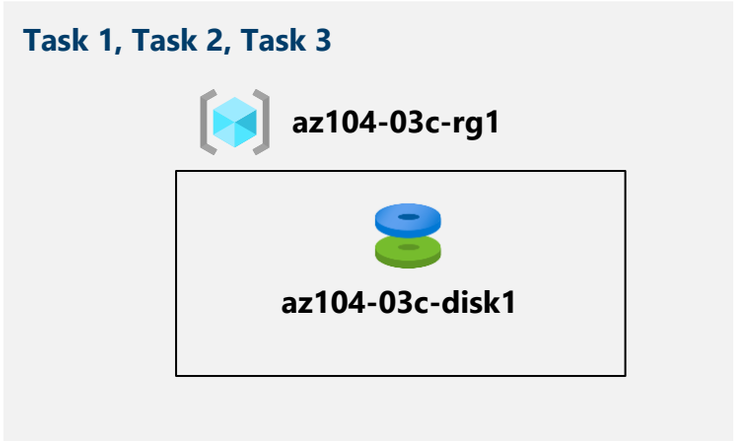
Create a resource group and an Azure managed disk with Azure PowerShell

### 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

## Lab scenario

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:

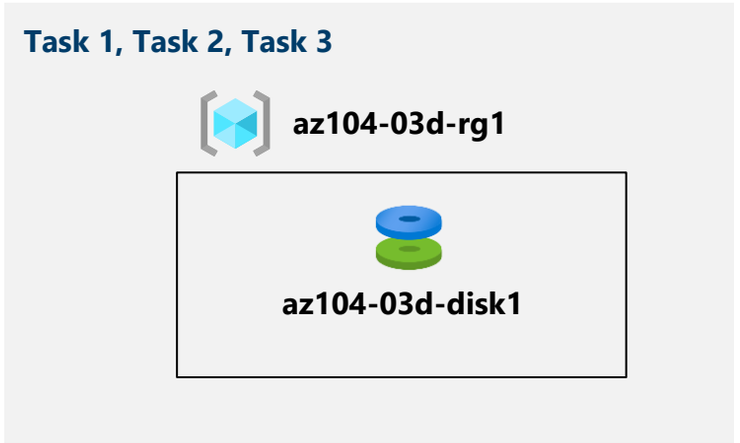
Create a resource group and a managed disk by using Azure CLI

### Task 3:

Configure the managed disk by using Azure CLI

Next slide for an architecture diagram

# Lab 03d – Architecture diagram



# End of presentation