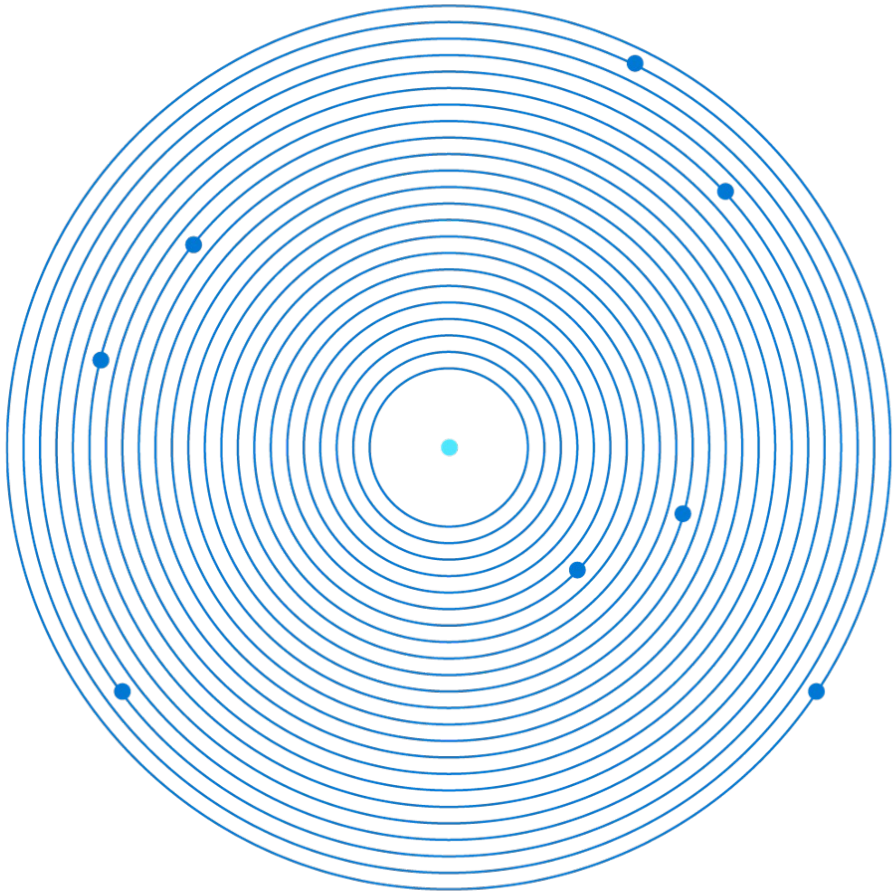
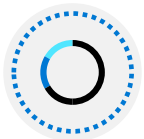
Microsoft Azure

AZ-104T00A Module 03: Azure Administration



Module Overview

Lesson 01: Resource Manager



Lesson 02: Azure Portal and Cloud Shell

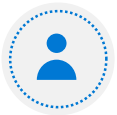
Lesson 03: Azure PowerShell and CLI

Lesson 04: ARM Templates

Lesson 05: Module 03 Lab and Review

Lesson 01: Resource Manager

Resource Manager Overview



Resource Manager

Terminology

Resource Group Deployments

Resource Manager Locks

Moving Resources

Removing Resources and Resource Groups

Resource Limits

Demonstration – Resource Groups

Resource Manager

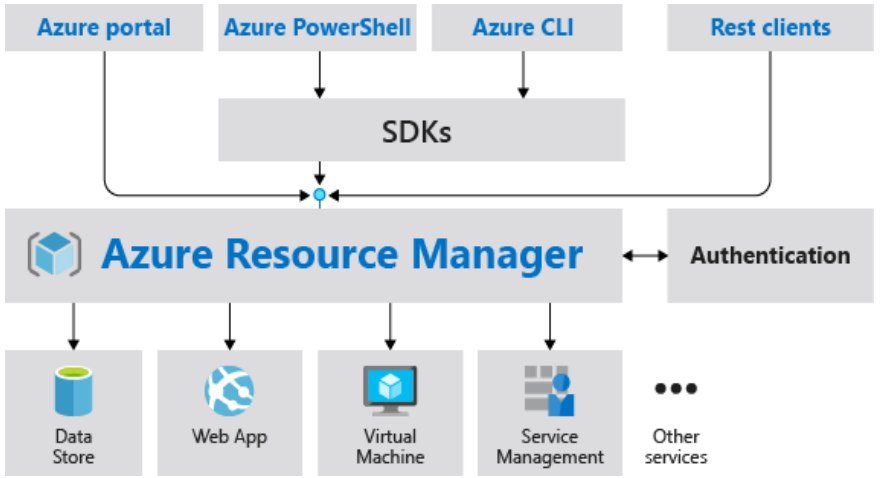
Provides a consistent management layer

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



Terminology

A **resource** is simply a single service instance in Azure

A **resource group** is a logical grouping of resources

An **Azure Resource Manager template** is a JSON file that allows you to declaratively describe a set of resources

A **declarative syntax** is what a template uses to state what you intend to create

A **resource provider** is service that supplies the resources you can deploy and manage through Resource Manager

Resource Groups

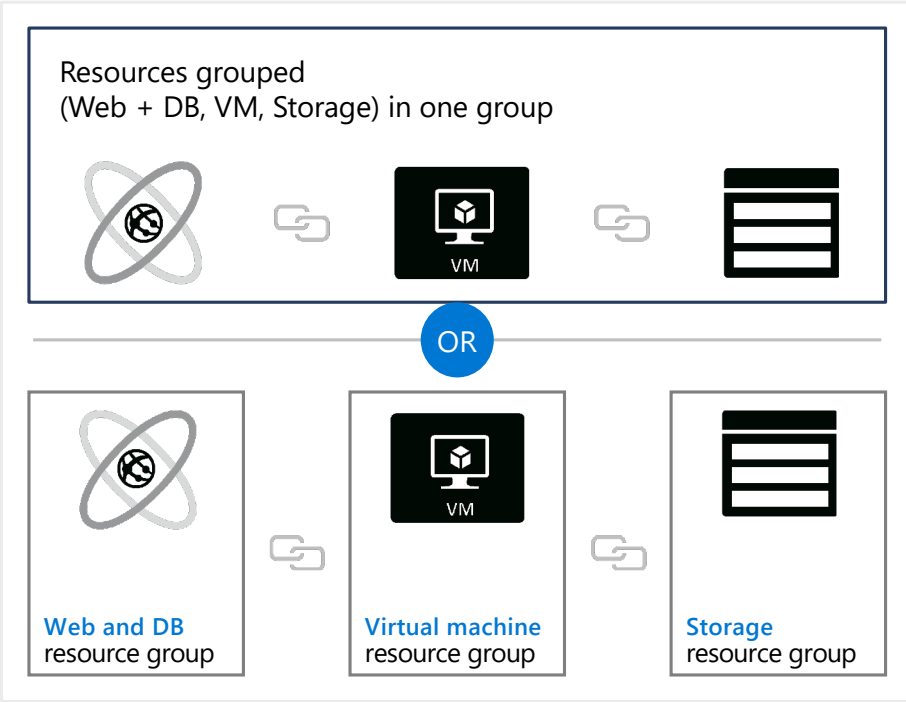
Resources can only exist in one resource group

Groups cannot be renamed

Groups can have resources of many different types (services)

Groups can have resources from many different regions

Deployments are incremental



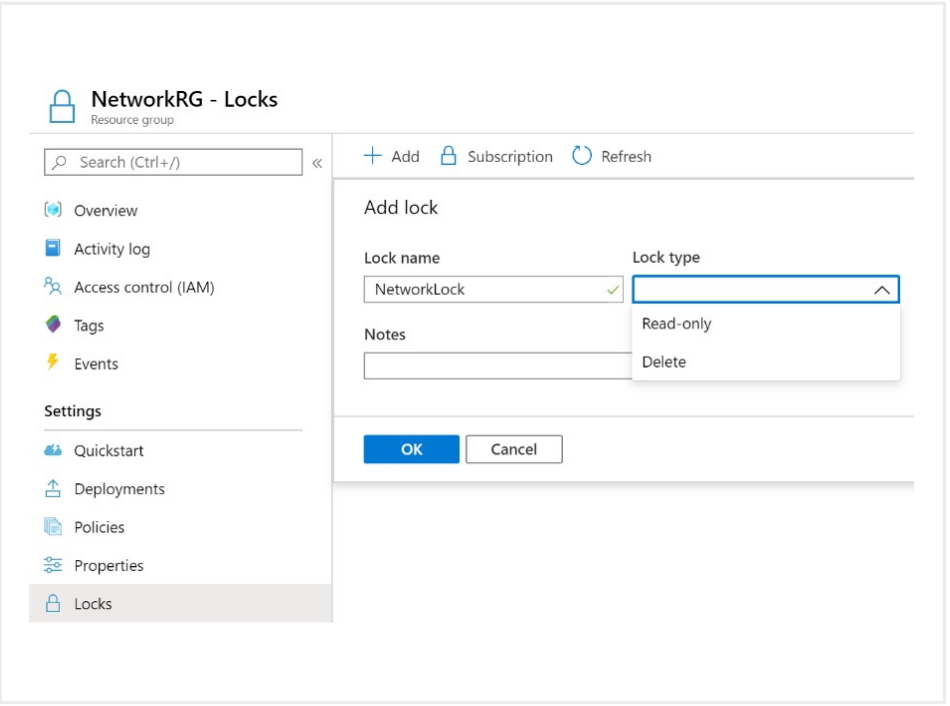
Resource Manager Locks

Associate the lock with a subscription, resource group, or resource

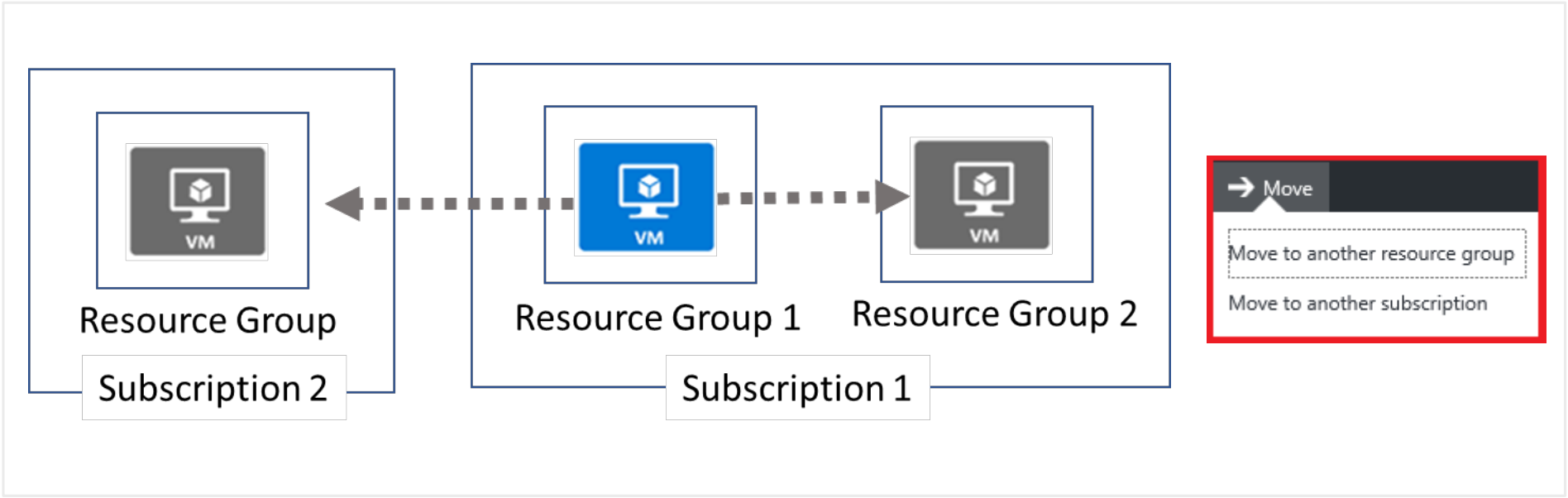
Locks are inherited by child resources

Read-Only locks prevent any changes to the resource

Delete locks prevent deletion



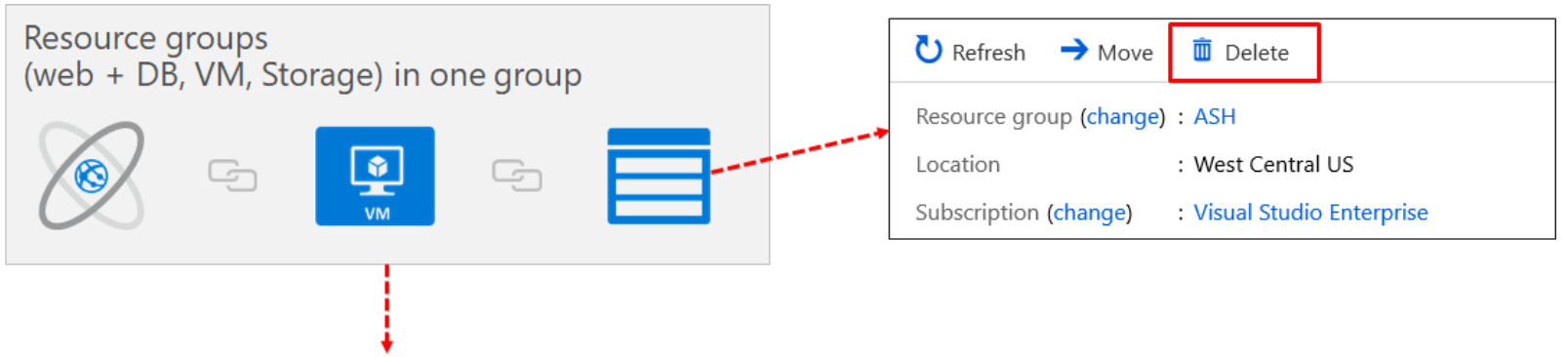
Moving Resources



When moving resources, both the source group and the target group are locked during the operation

Services that cannot be moved: Azure AD Domain Services, ExpressRoute, and Site Recovery. Other restrictions apply

Removing Resources and Resource Groups



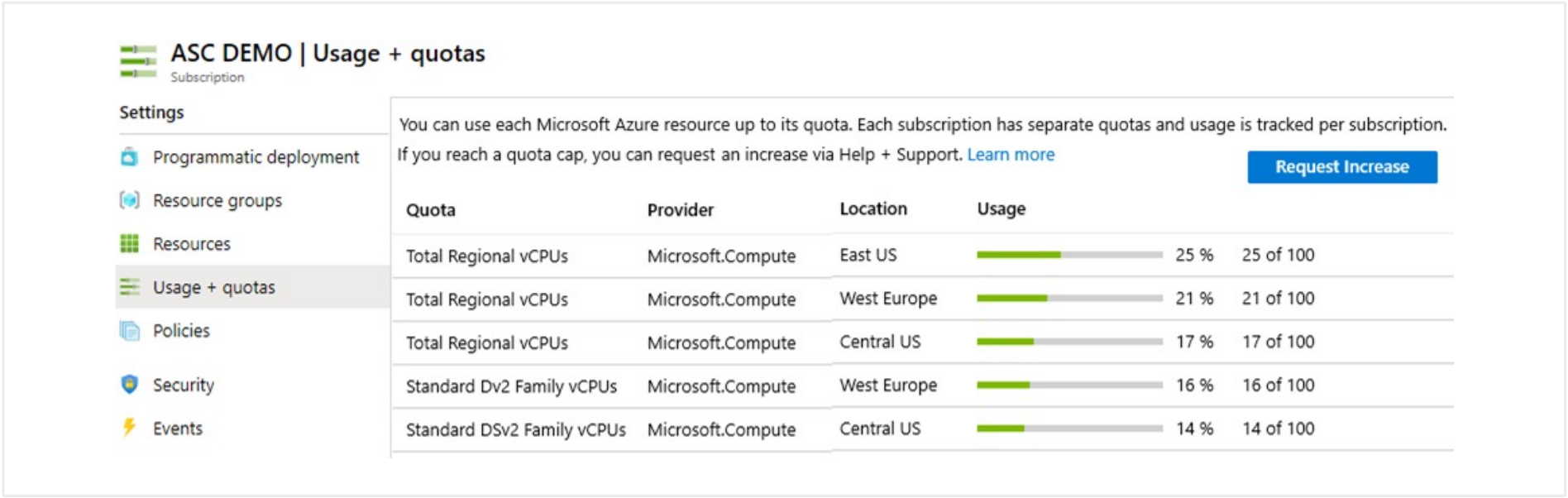

```
Get-AzResourceGroup -Name 'az104-03*' | Remove-AzResourceGroup -Force -AsJob
```

Remove Azure resources that you no longer use

Ensures you will not see unexpected charges

Remove individual resources or remove the resource group

Resource Limits



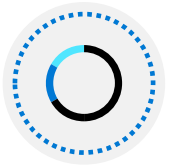
Resources have a default limit also known as quota

Helpful to track current usage, and plan for future use

You can open a free support case to increase limits to published maximums

Lesson 02: Azure Portal and Cloud Shell

Azure Portal and Cloud Shell overview



Azure Portal

Demonstration – Azure Portal

Azure Cloud Shell

Demonstration – Cloud Shell

Azure Portal

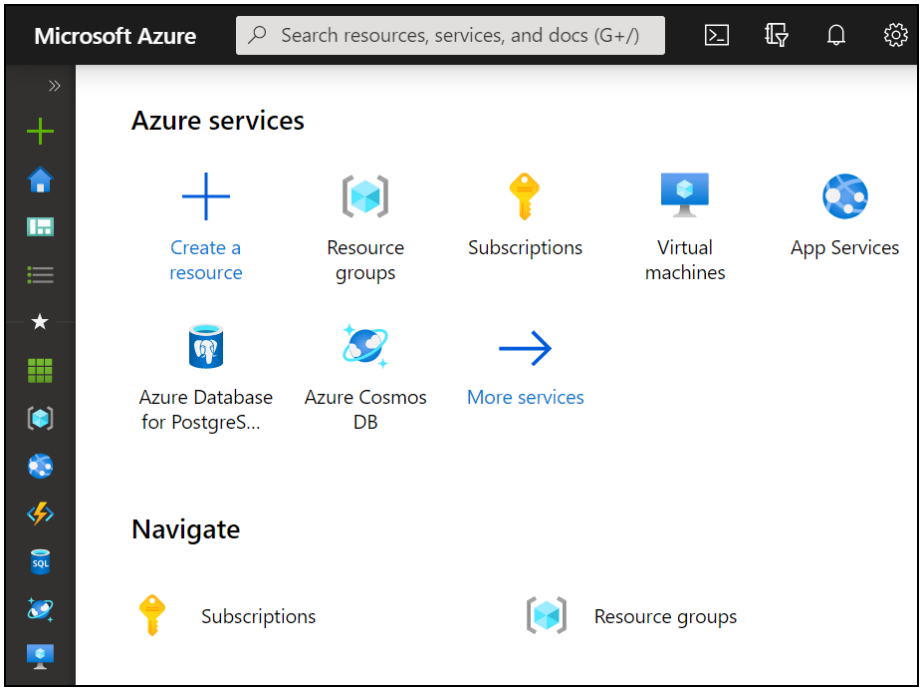
Search resources, services, and docs

Manage resources

Create customized dashboards and favorites

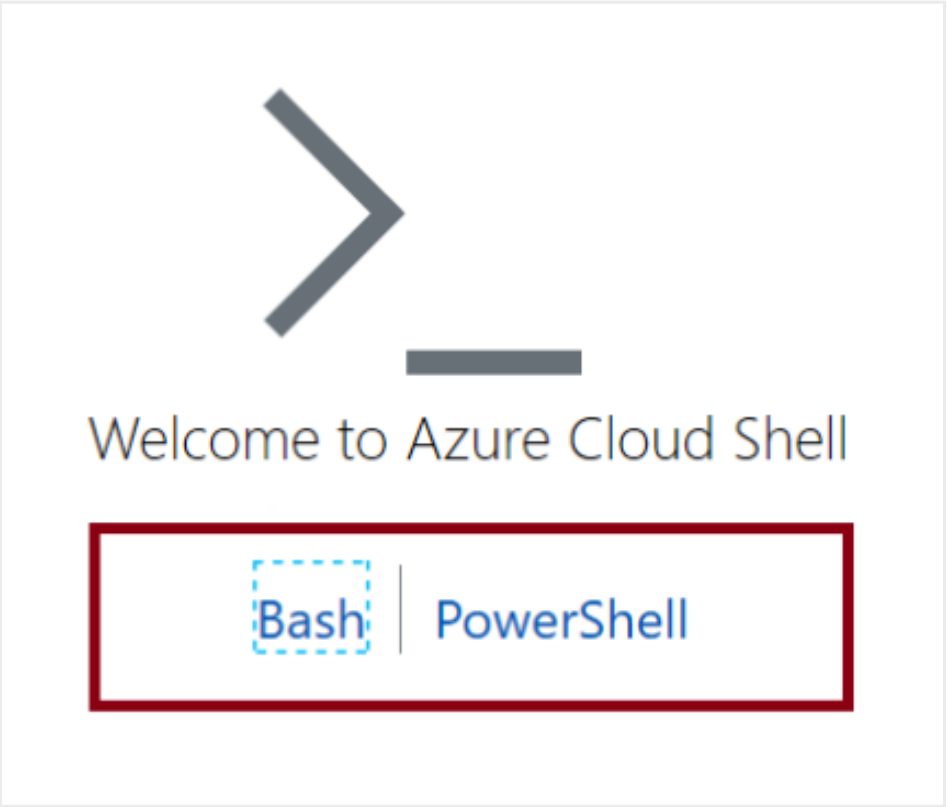
Access the Cloud Shell

Receive notifications

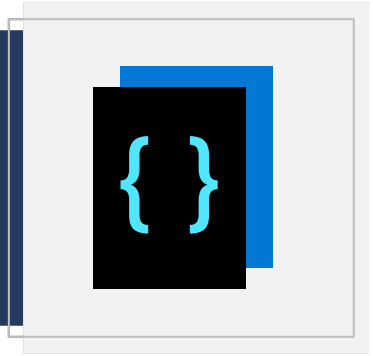


Azure Cloud Shell

- 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

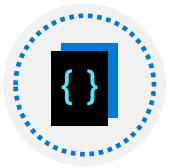


Lesson 03: Azure PowerShell and CLI



Azure PowerShell and CLI Overview

Azure PowerShell



PowerShell Cmdlets and Modules

Demonstration – Working with PowerShell locally

Azure CLI

Demonstration – Working with Azure CLI locally

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
```

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

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

Azure CLI

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

Cross-platform command-line program

Runs on Linux, macOS, and Windows

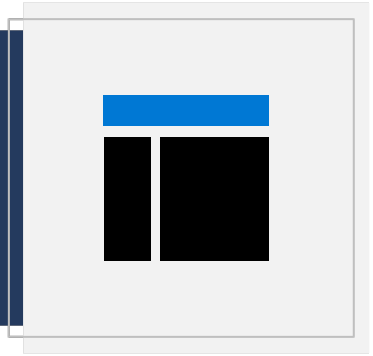
Can be used interactively or through scripts

Commands are structured in *_groups_* and *_subgroups_*

Use *find* to locate commands

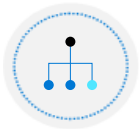
Use *--help* for more detailed information

Lesson 04: ARM templates



ARM Templates Overview

Template Advantages



Template Schema

Template Parameters

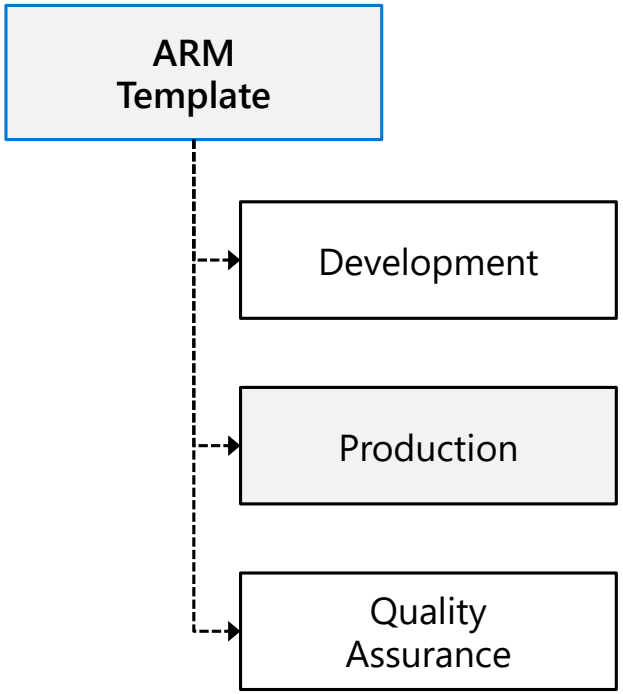
QuickStart Templates

Demonstration – QuickStart Templates

Demonstration – Run Templates with PowerShell

Template Advantages

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



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 values 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": {}
}
```

Template Parameters

Specify which values are configurable when the template runs

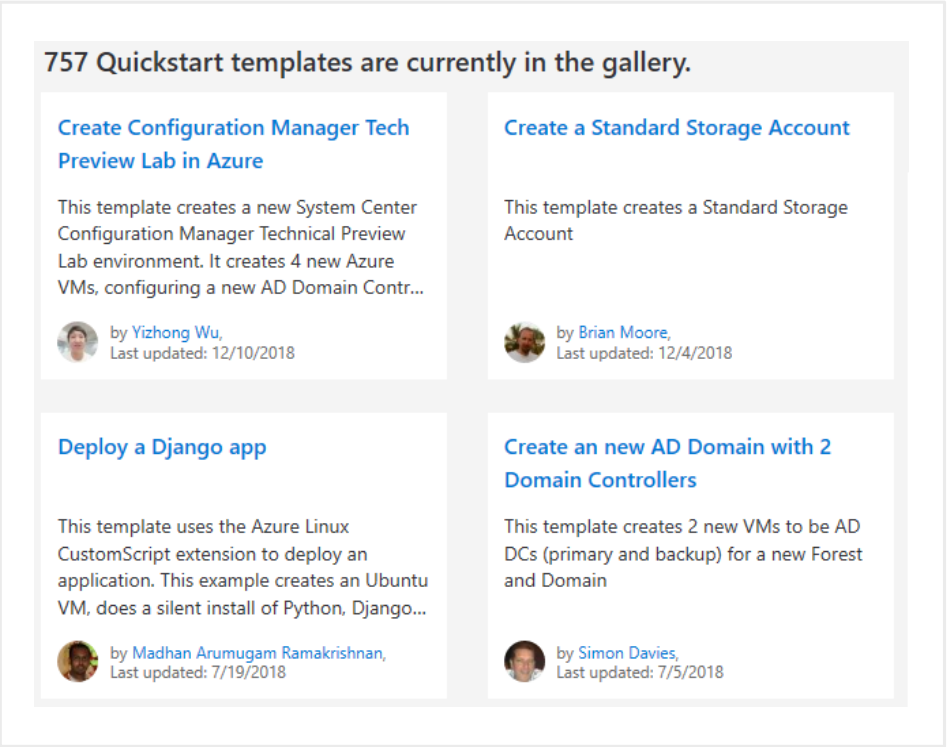
This example has two parameters: one for a VM's username (adminUsername), and one for its password (adminPassword)

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

QuickStart Templates

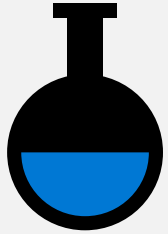
Resource Manager templates provided by the Azure community

Provides everything you need to deploy your solution or serves as a starting point for your template



https://azure.microsoft.com/en-us/resources/templates/

Lesson 05: Module 03 Lab and Review



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

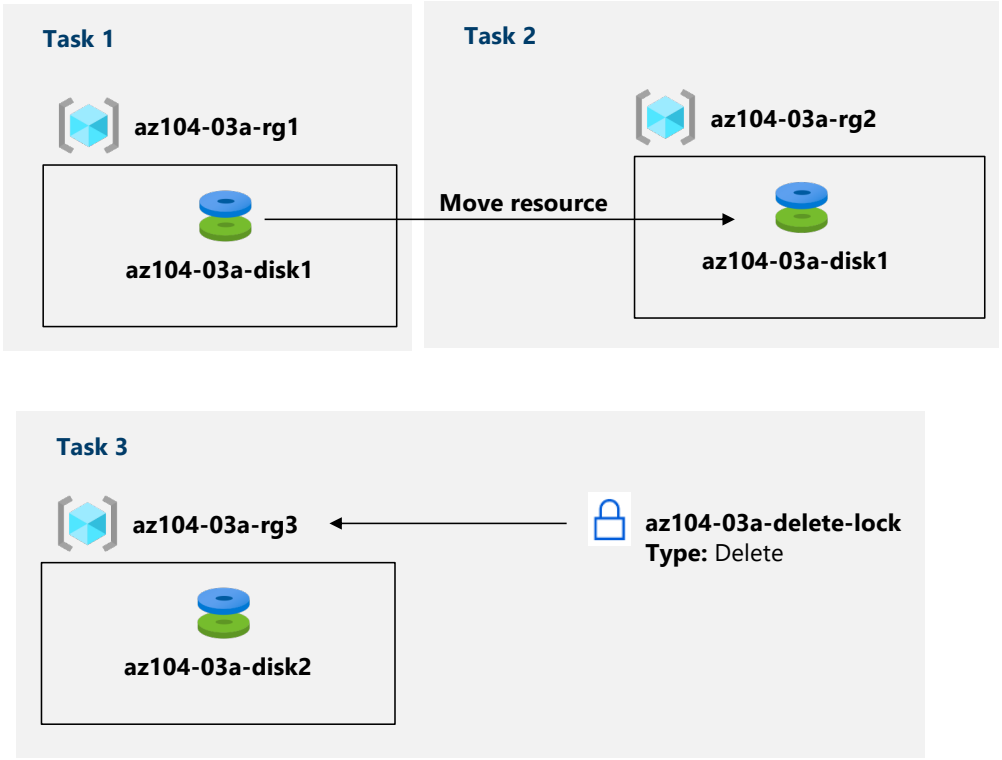
Task 1:
Create resource groups and deploy resources to resource groups

Task 2:
Move resources between resource groups

Task 3:
Implement and test resource locks

Next slide for an architecture diagram

Lab 03a – 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

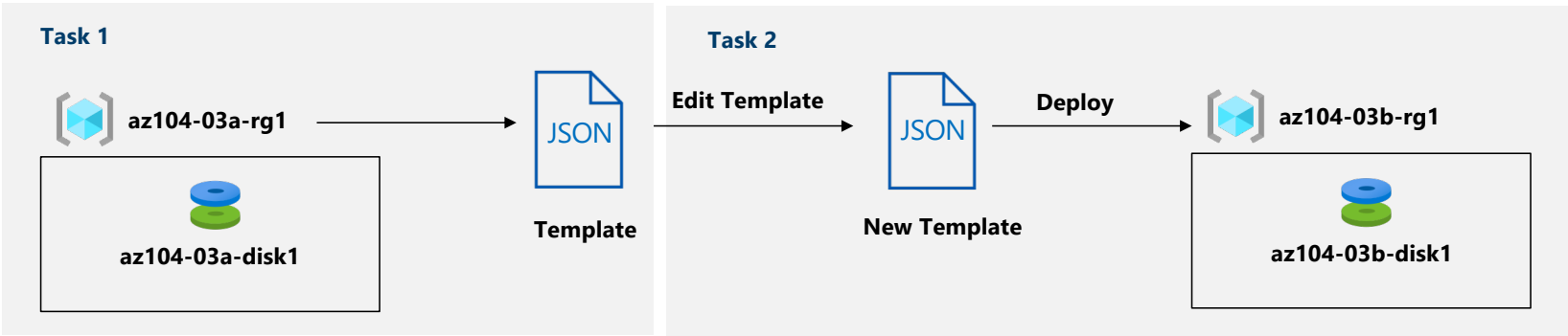
Task 1:
Review an ARM template for deployment of an Azure managed disk

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 (optional)

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:

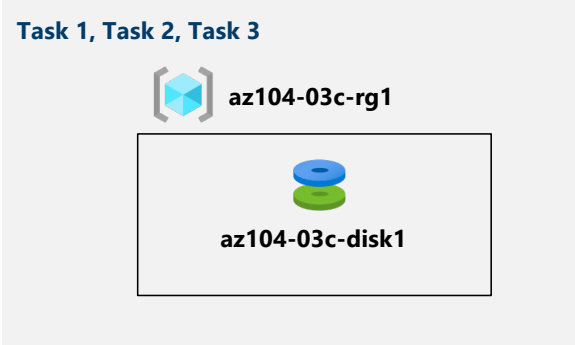
Create a resource group and an Azure managed disk with Azure PowerShell

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 (optional)

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

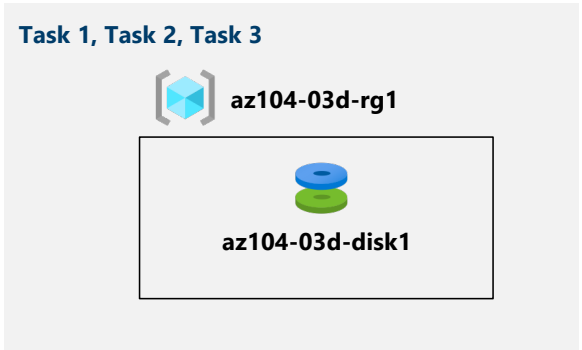
Task 1:
Start a Bash session in Azure Cloud Shell

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



Module Review

Module Review Questions

Microsoft Learn Modules (docs.microsoft.com/Learn)

Core Cloud Services – Manage services with the Azure portal

Control and organize Azure resources with Azure Resource Manager

Build Azure Resource Manager templates

Automate Azure tasks using scripts with PowerShell

Manage virtual machines with the Azure CLI

End of presentation