AZ-104

Tag 2

Administer Azure Resources

Guten Morgen!

AZ-104 Agenda

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

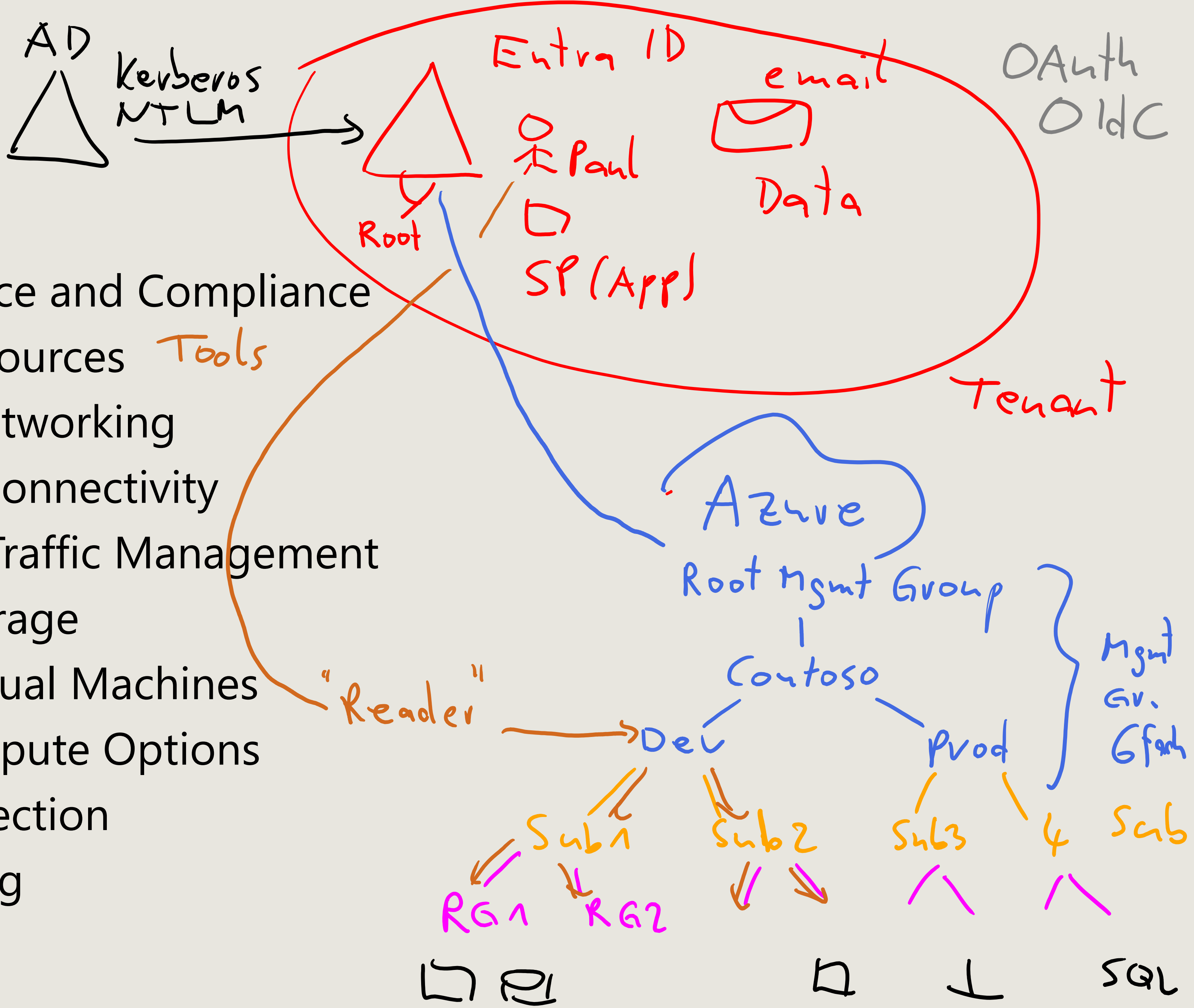
07: Administer Azure Storage

08: Administer Azure Virtual Machines

09: Administer PaaS Compute Options

10: Administer Data Protection

11: Administer Monitoring



Learning Objectives - Azure Resources

- Configure Azure Resources with Tools
  - Manage services with the Azure portal ←
  - Introduction to PowerShell mit Az Module
  - Introduction to Bash
- Deploy Azure resources with templates terraform Bicep
- Lab 03 - Manage Azure resources by Using ARM Templates

Az
- Az.Compute
- Az.Storage
- ⋮

	PowerShell Module Az	Azure CLI az (Binary)
PowerShell	✓	✓
Bash	✗	✓
	Get-Az Vm	az vm list

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

Learning Objectives – Configure Azure Resources with tools

- Compare Administration tools
- Demonstration – Azure Portal
- Demonstration – Azure Cloud Shell
- Learning Recap

Review Resource Manager Benefits

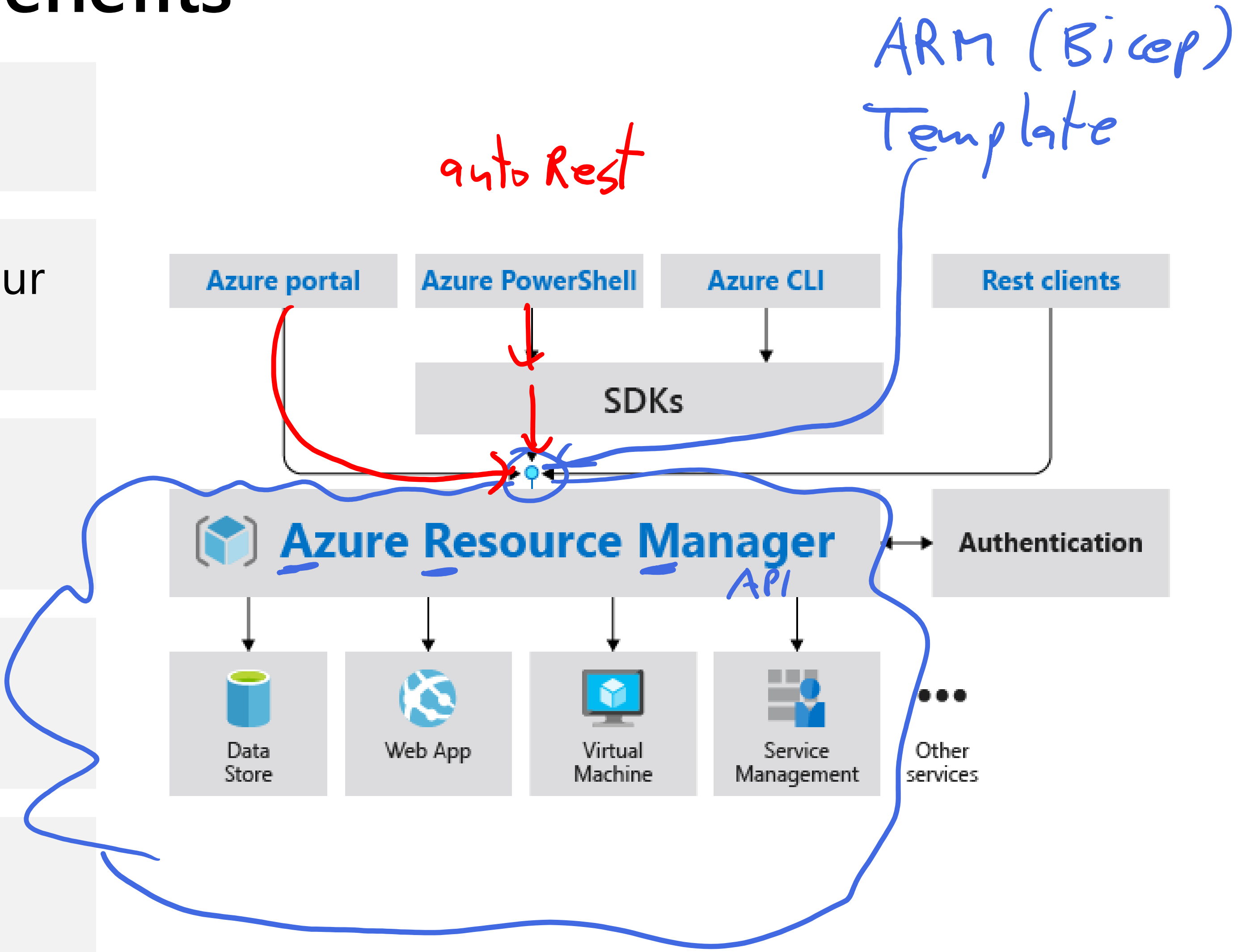
Provides a consistent management layer

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



Review Azure Resource Terminology

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

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

Bicepr

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

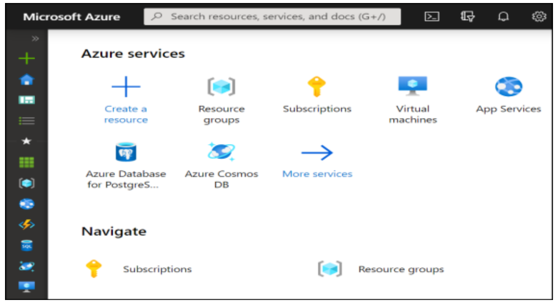
A **declarative syntax** is what a template uses to state what you intend to create

Semper idem!

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

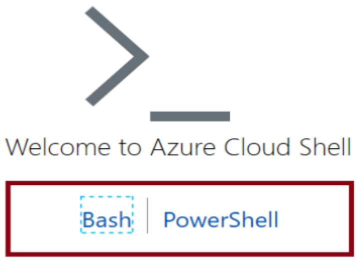
Compare Administrator tools

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

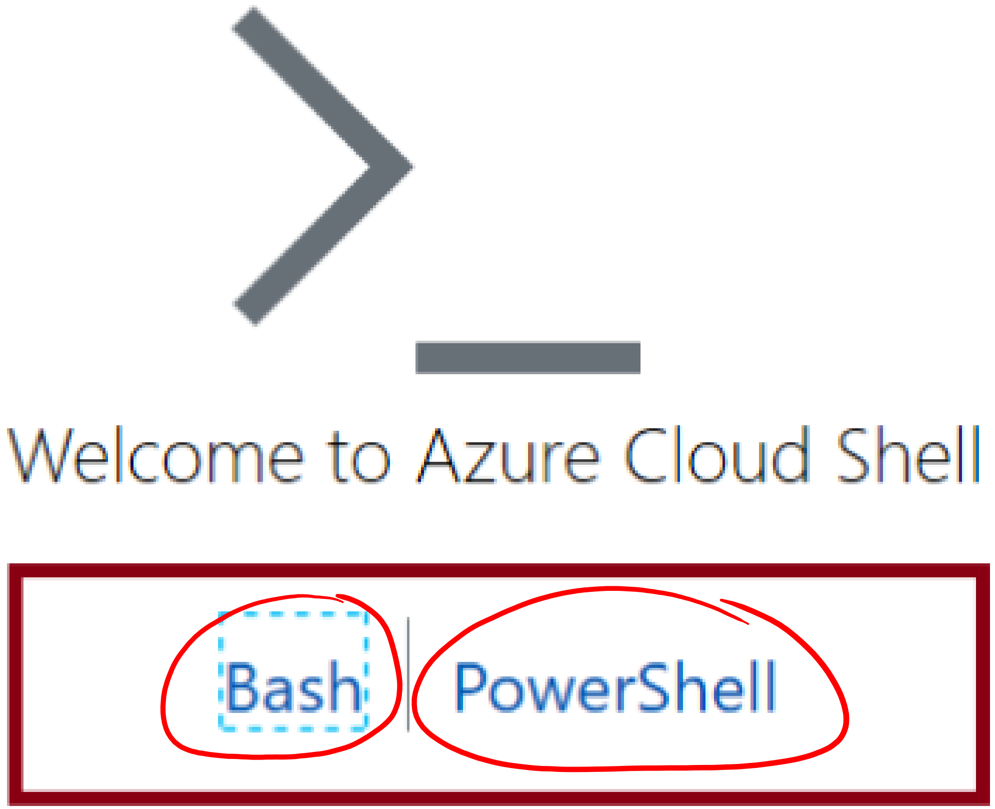
```
az vm restart -g
MyResourceGroup -n MyVm
```

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

Container

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



Welcome to Azure Cloud Shell

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
```

- 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 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 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

Learning Recap – Azure Resources with Tools

Check your knowledge questions and additional study

Reference modules

- Describe features and tools for managing and deploying Azure resources
- Introduction to Azure Cloud Shell
- Introduction to PowerShell

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

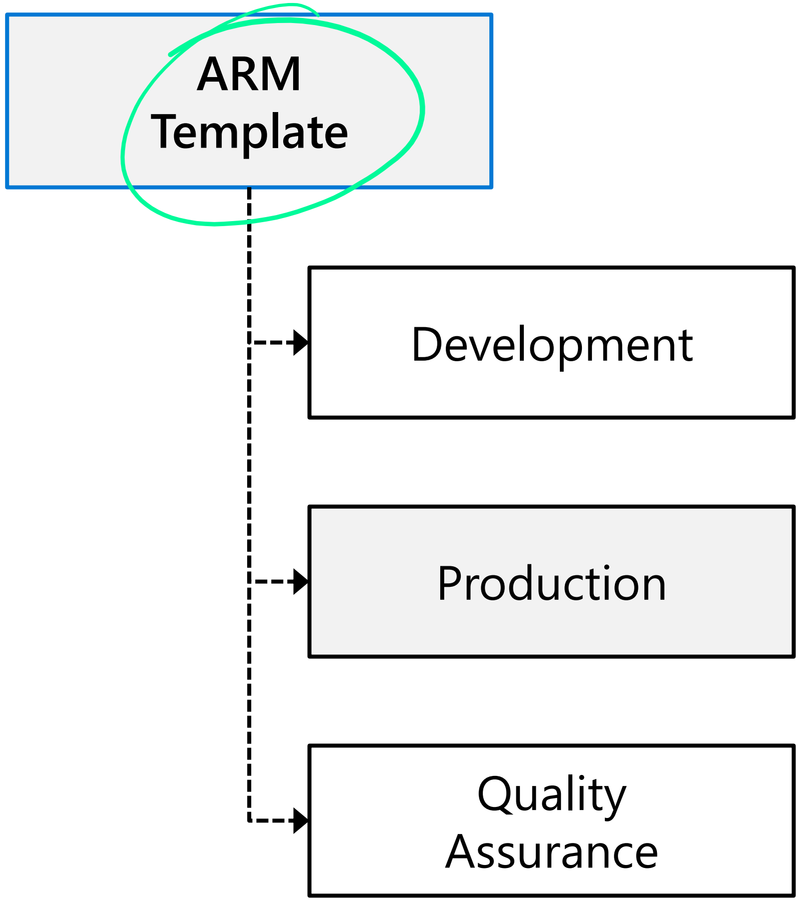
Deploy and manage Azure compute resources (20–25%): Automate deployment of resources by using ARM templates or Bicep files

- 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

IaC

- 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

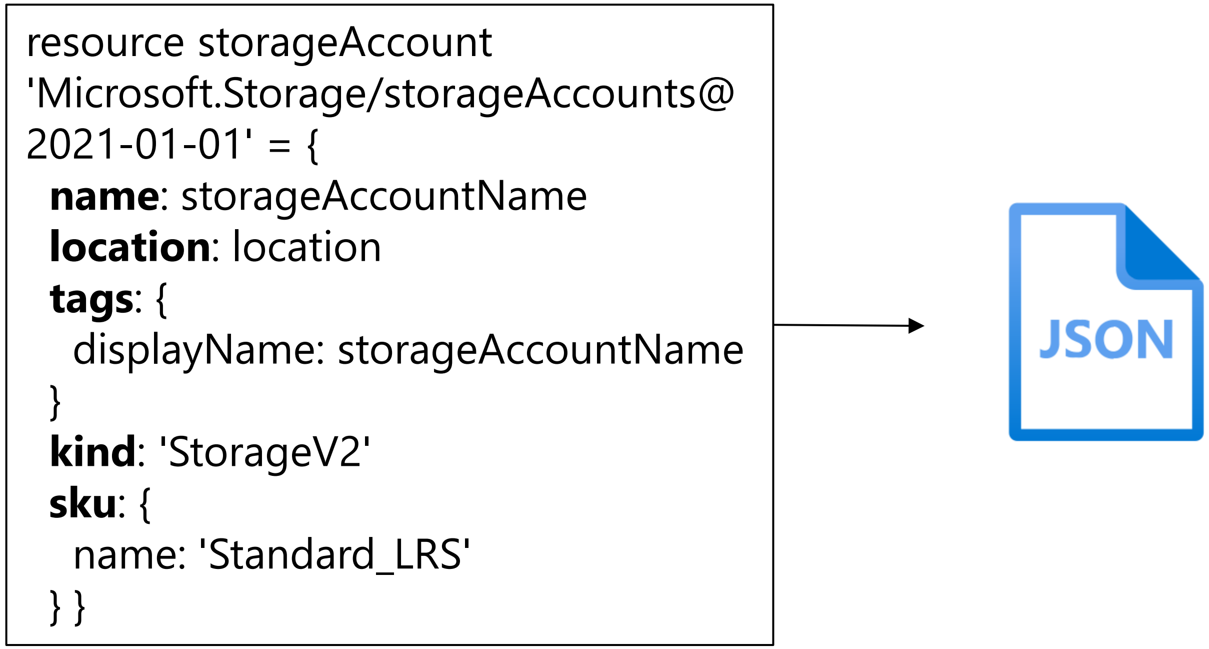
- Specifies which values are configurable when the template runs
- This example has two parameters: one for a VM’s username (adminUsername), and one for its password (adminPassword)

```
"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

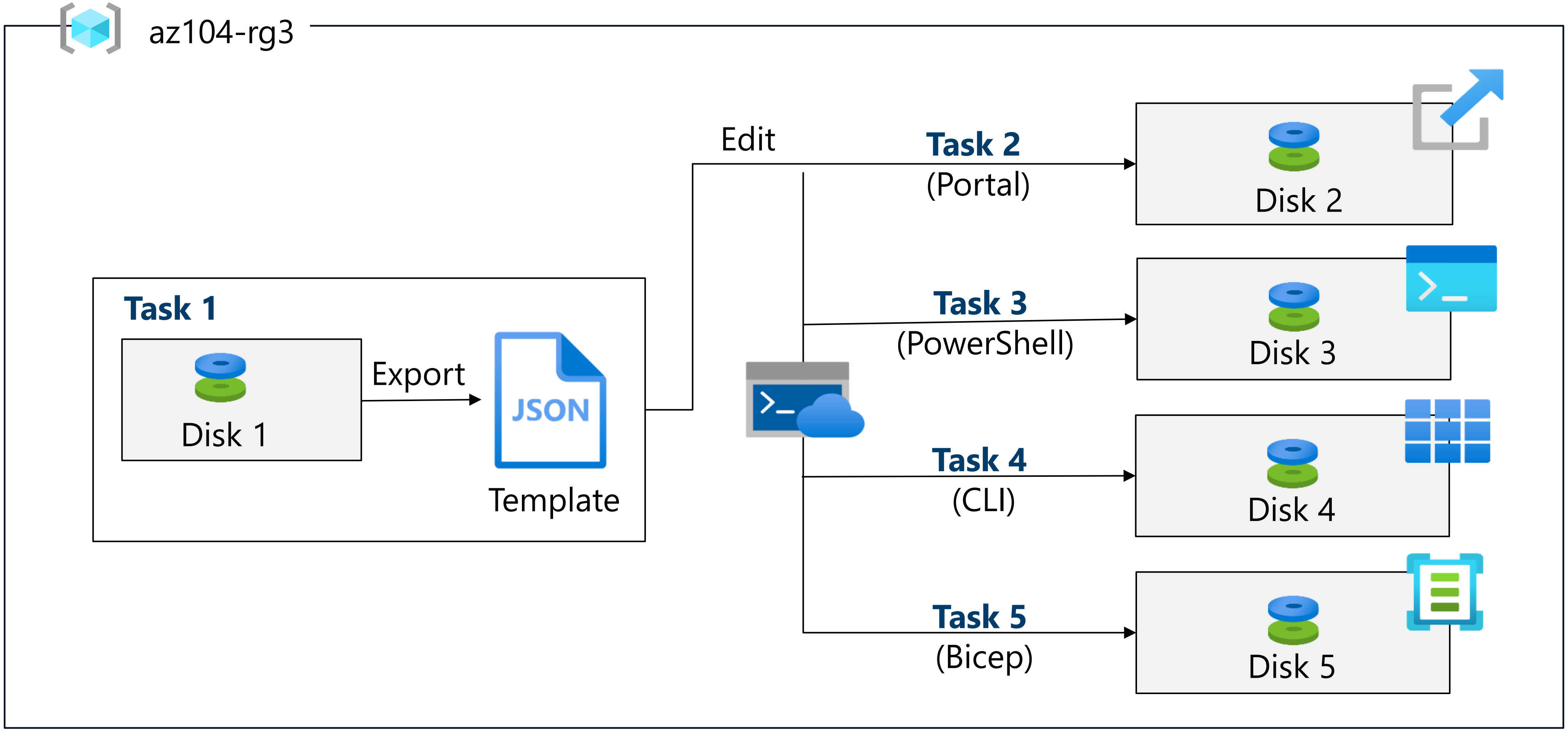
Check your knowledge questions and additional study

Reference modules

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

Lab - Manage Azure resources by Using ARM Templates

Lab 03 – Architecture diagram



End of presentation