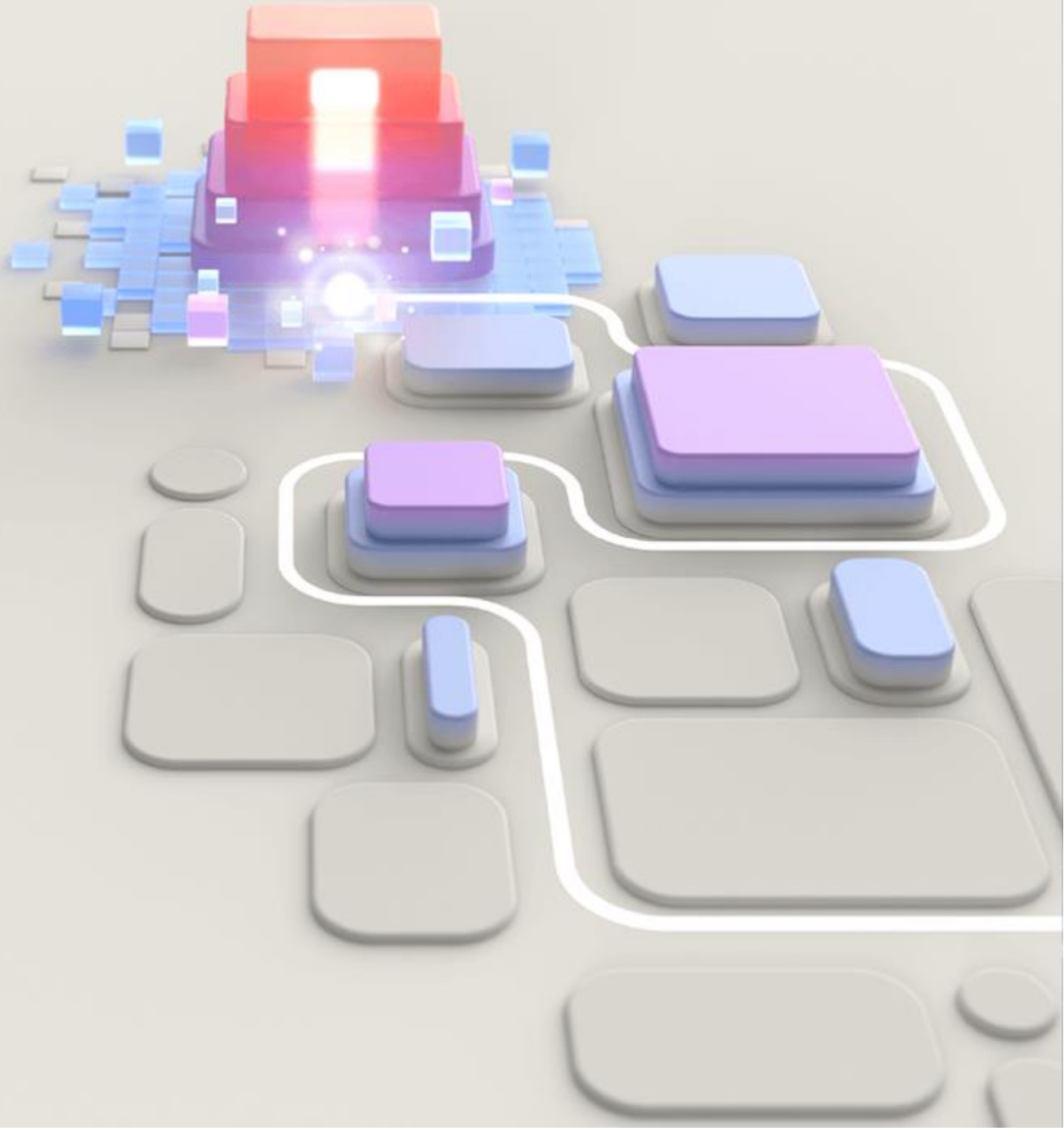
Microsoft

AZ-104

Administer Azure Resources



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

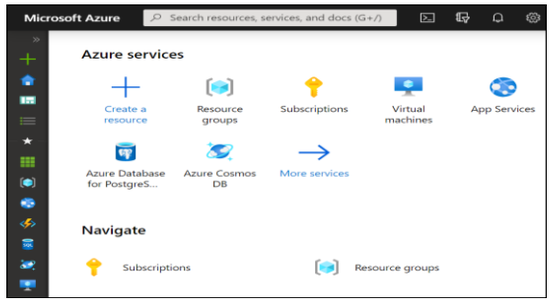
Learning Objectives - Azure Resources

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

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

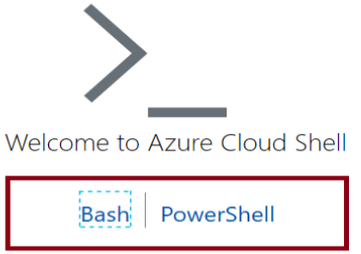
Compare Administrator tools

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

Review Resource Manager Benefits

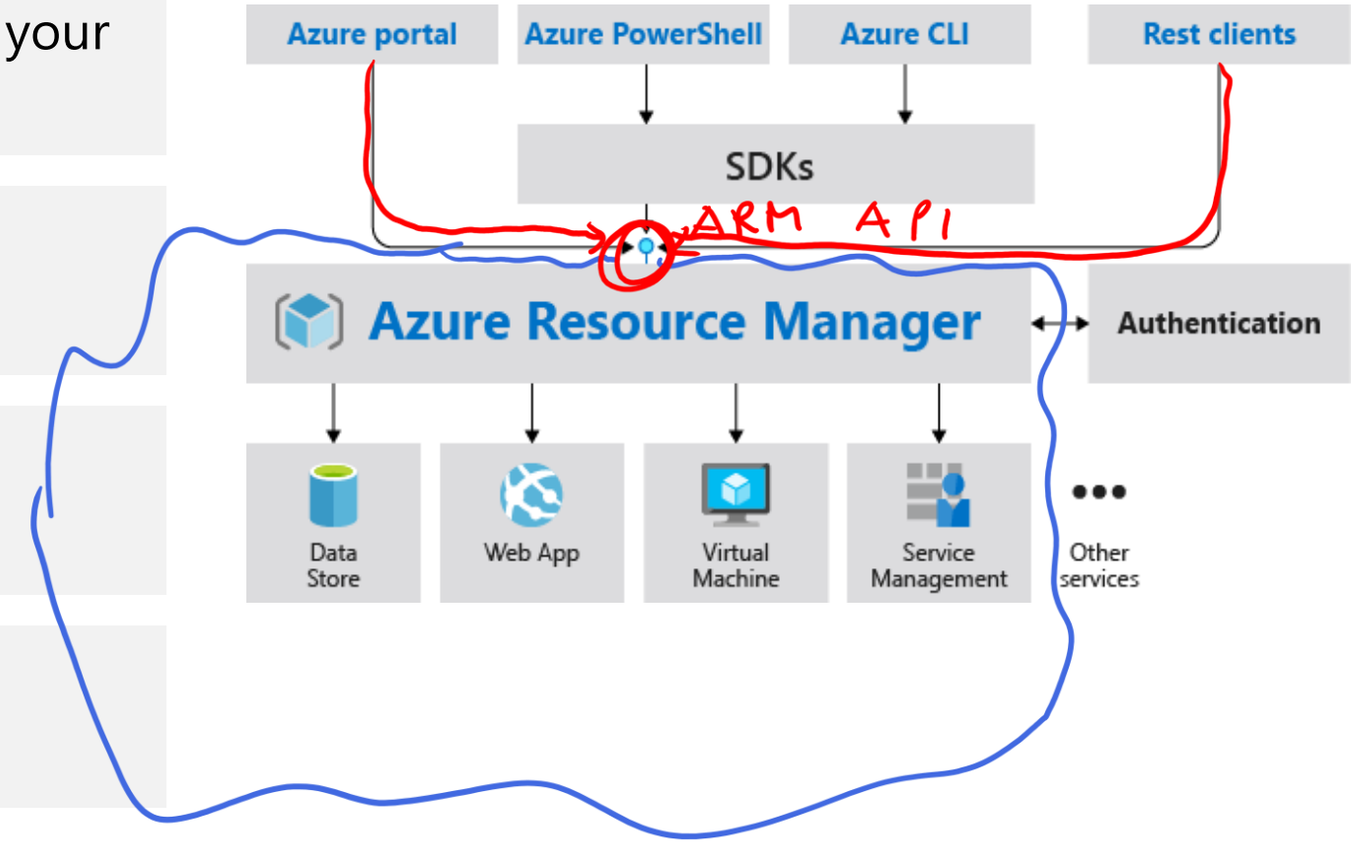
Provides a consistent management layer

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

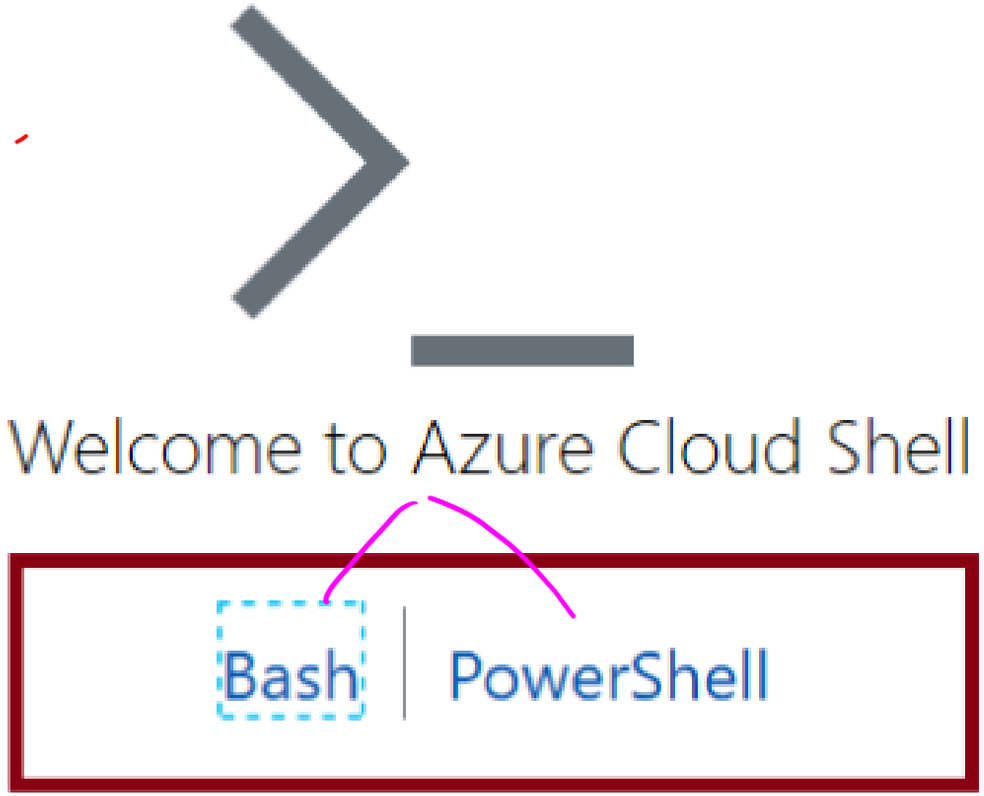


Use Azure Cloud Shell

v2

Editor code (vs code) → v1

- 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" `
...
```

$RgName

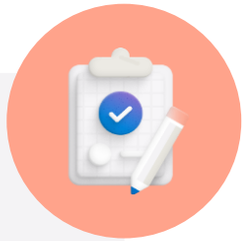
- Connect to your Azure subscription and manage resources
- Adds the Azure-specific commands
- Available inside a browser via the Azure Cloud Shell
- Available as a local installation on Linux, macOS, or Windows
- Has an interactive and a scripting mode

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

- Manage services with the Azure portal
- Introduction to PowerShell
- Introduction to Bash
- Use Azure Resource Manager

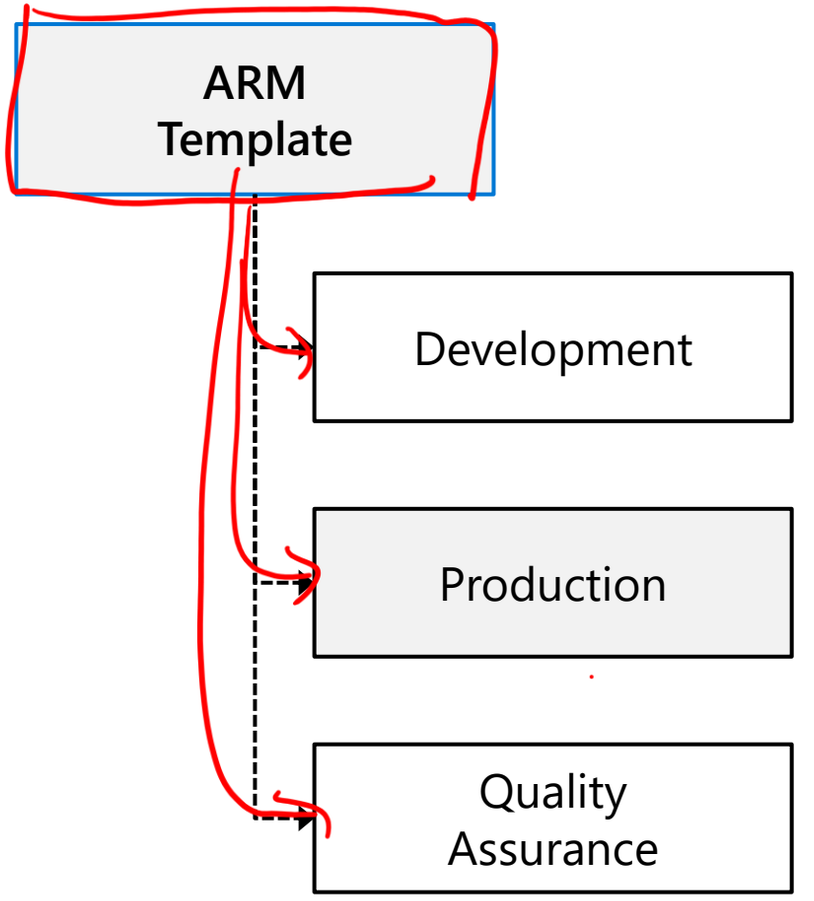
Configure Resources with ARM Templates

Review ARM Template Advantages

Infra as Code

- 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

Bicep



Alternative: Terraform (IBM)

Explore the JSON Template Schema

Defines all the Resource manager resources in a deployment

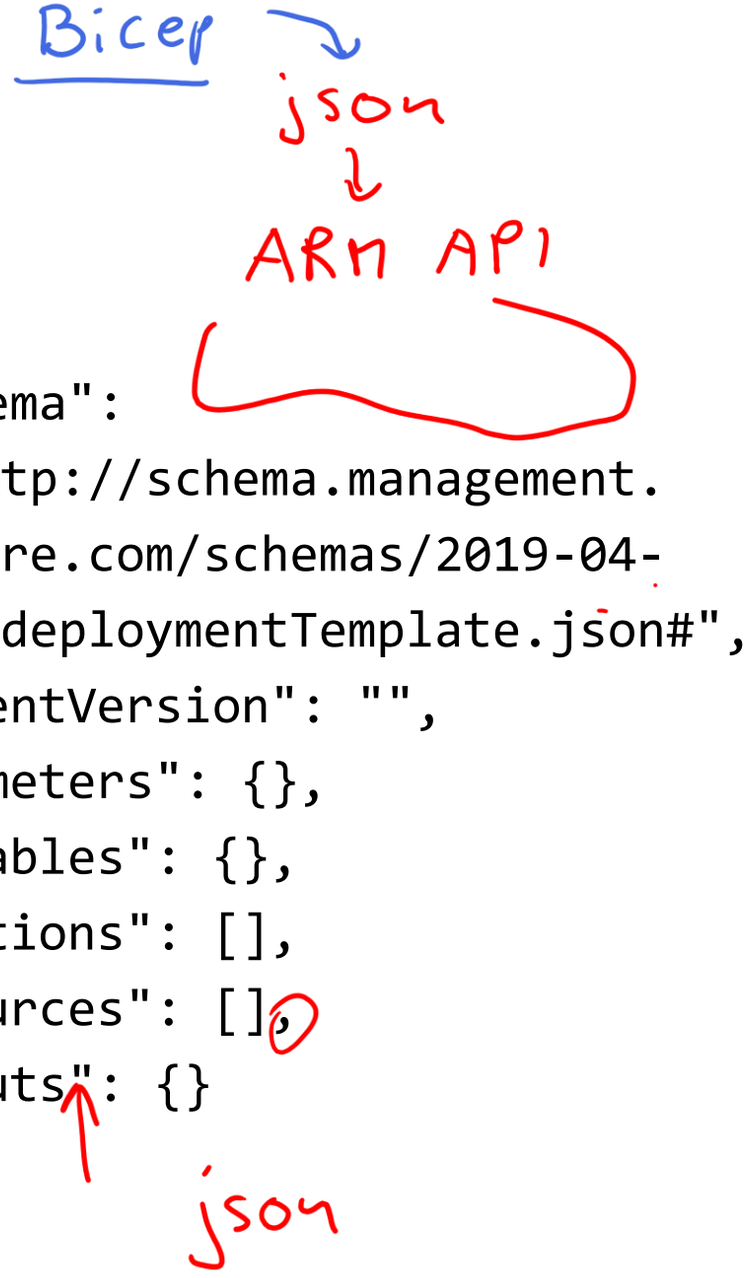
Written in JSON

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."
    }
  }
}
```

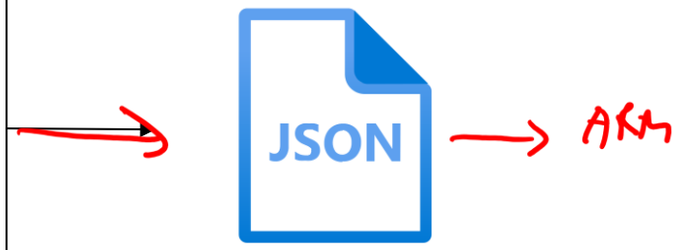
Consider Azure Bicep Files

- 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

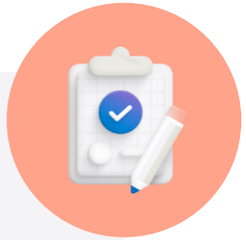
param location = 'westeurope'

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 – Azure Resource Manager templates

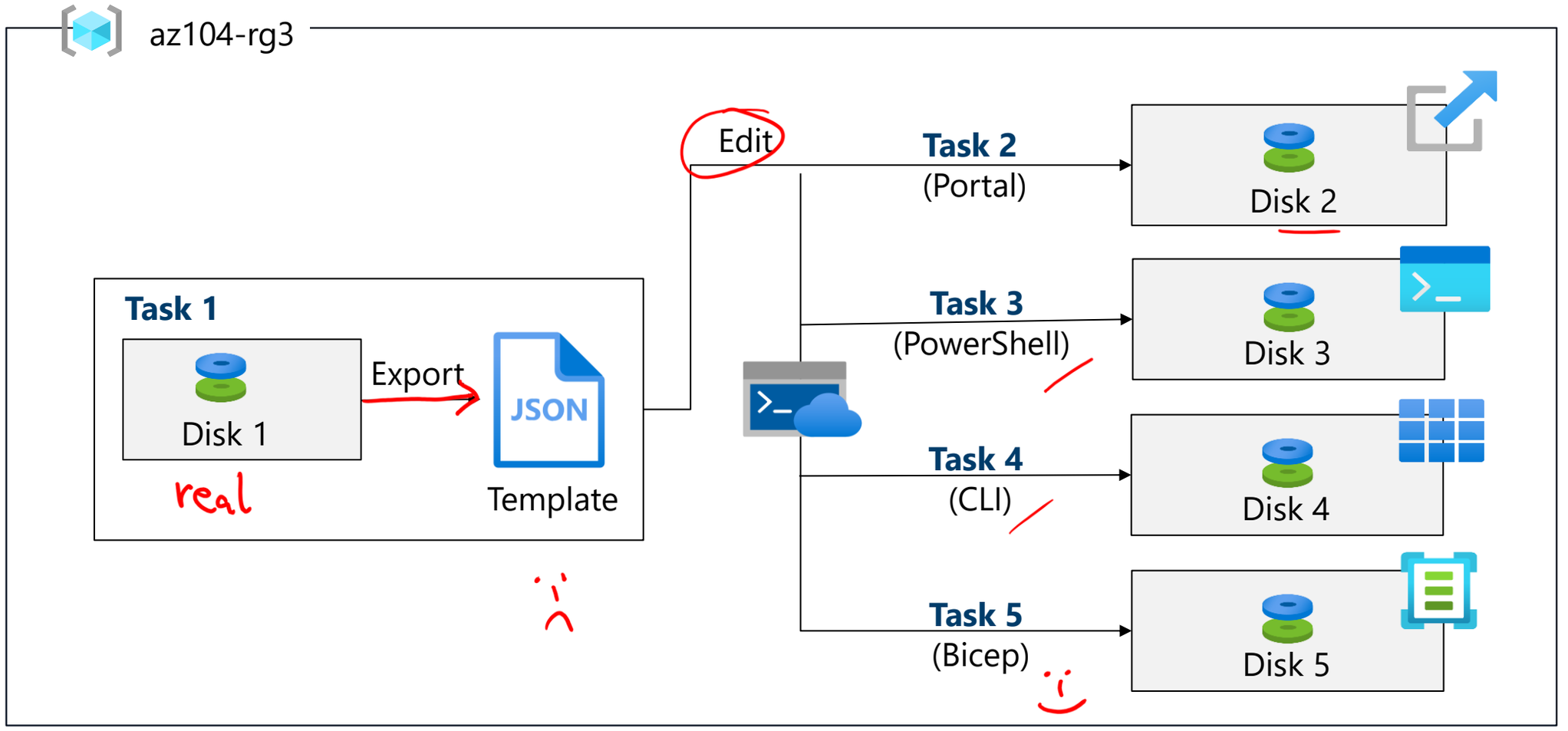


Check your knowledge questions and additional study

- Create Azure resources using Azure Resource Manager templates
- Deploy Azure infrastructure by using JSON ARM templates
- Introduction to infrastructure as code using Bicep
- Build your first Bicep template

Lab - Manage Azure resources by Using ARM Templates

Lab 03 – Architecture diagram



End of presentation