

Cloud
Computing

Caltech

Center for Technology &
Management Education

Post Graduate Program in Cloud Computing

Cloud Computing

Caltech

**Center for Technology &
Management Education**

**PG CC - Microsoft Azure Architect
Technologies: AZ:303**



Automate Deployment and Configuration of Resources

Learning Objectives

By the end of this lesson, you will be able to:

- 👁️ Analyze Azure resource manager and its terminologies
- 👁️ Illustrate Azure resource manager templates
- 👁️ Implement virtual hard disk
- 👁️ Configure templates
- 👁️ Configure automation runbook



A Day in the Life of an Azure Architect

The teams in your organization have adopted agile development methods. These teams iterate quickly and need to repeatedly deploy their solutions to the cloud and know their infrastructure is in a reliable state.

As infrastructure has become part of the iterative process, the division between operations and development has disappeared. Now, these teams need to manage infrastructure and application code through a unified process.

To meet these challenges, you have been asked to suggest an azure automation solution so that anyone from the team can run the code and deploy similar environments leveraging Infrastructure as a Code.

To achieve all of the above along with some additional features, we will be learning a few concepts in this lesson that will help you find a solution for the given scenario.



Azure Resource Manager

Azure Resource Manager

Azure Resource Manager is a service that manages and deploys Azure resources.

Benefits

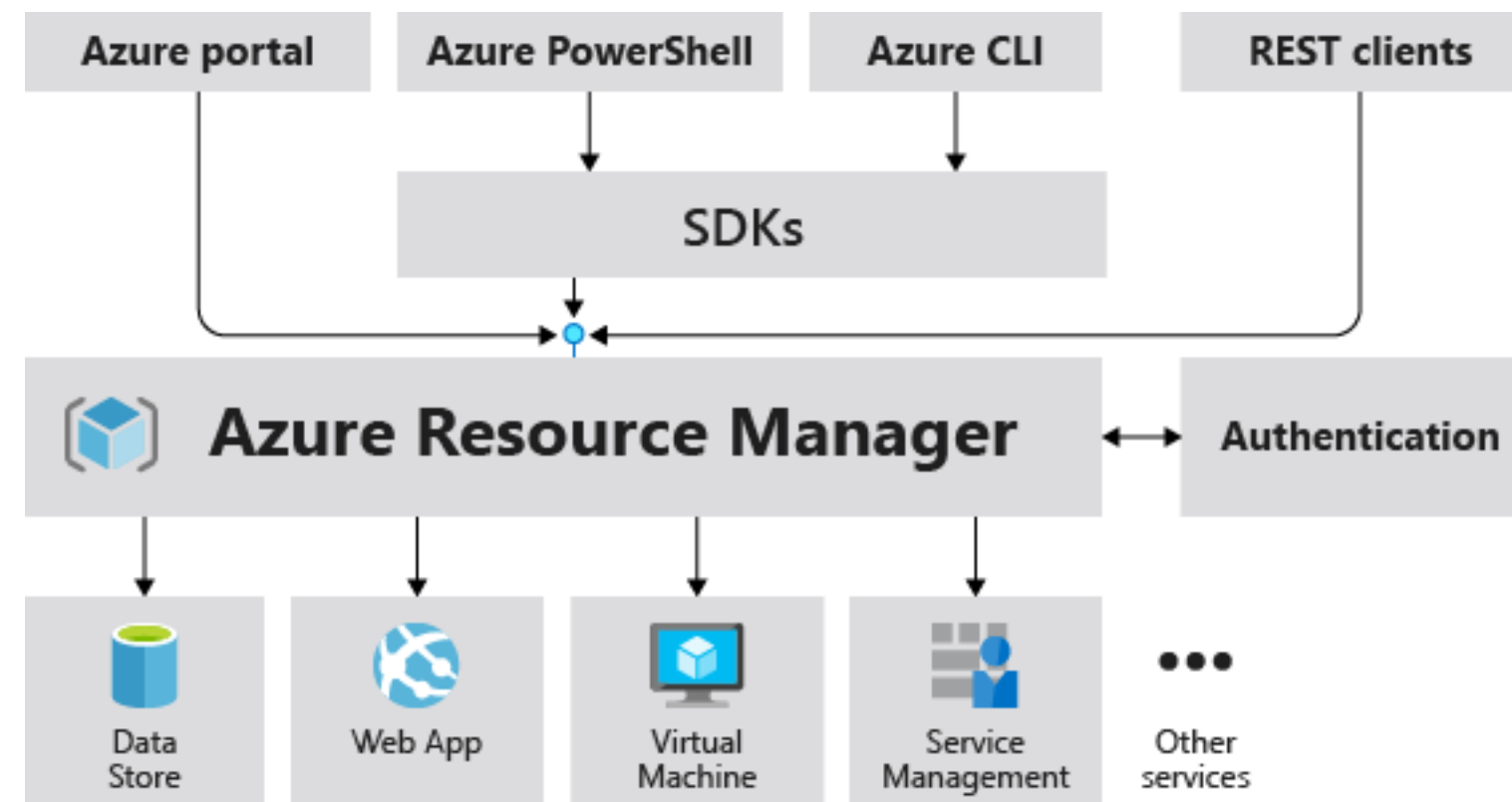
With Resource Manager, the user can:

- Manage the infrastructure
- Deploy, manage, and monitor all the resources
- Redeploy consistently
- Define dependencies between the resources
- Apply access control
- Apply tags

Azure Resource Manager has a management layer that allows the user to create, update, and delete Azure account resources.

Azure Resource Manager

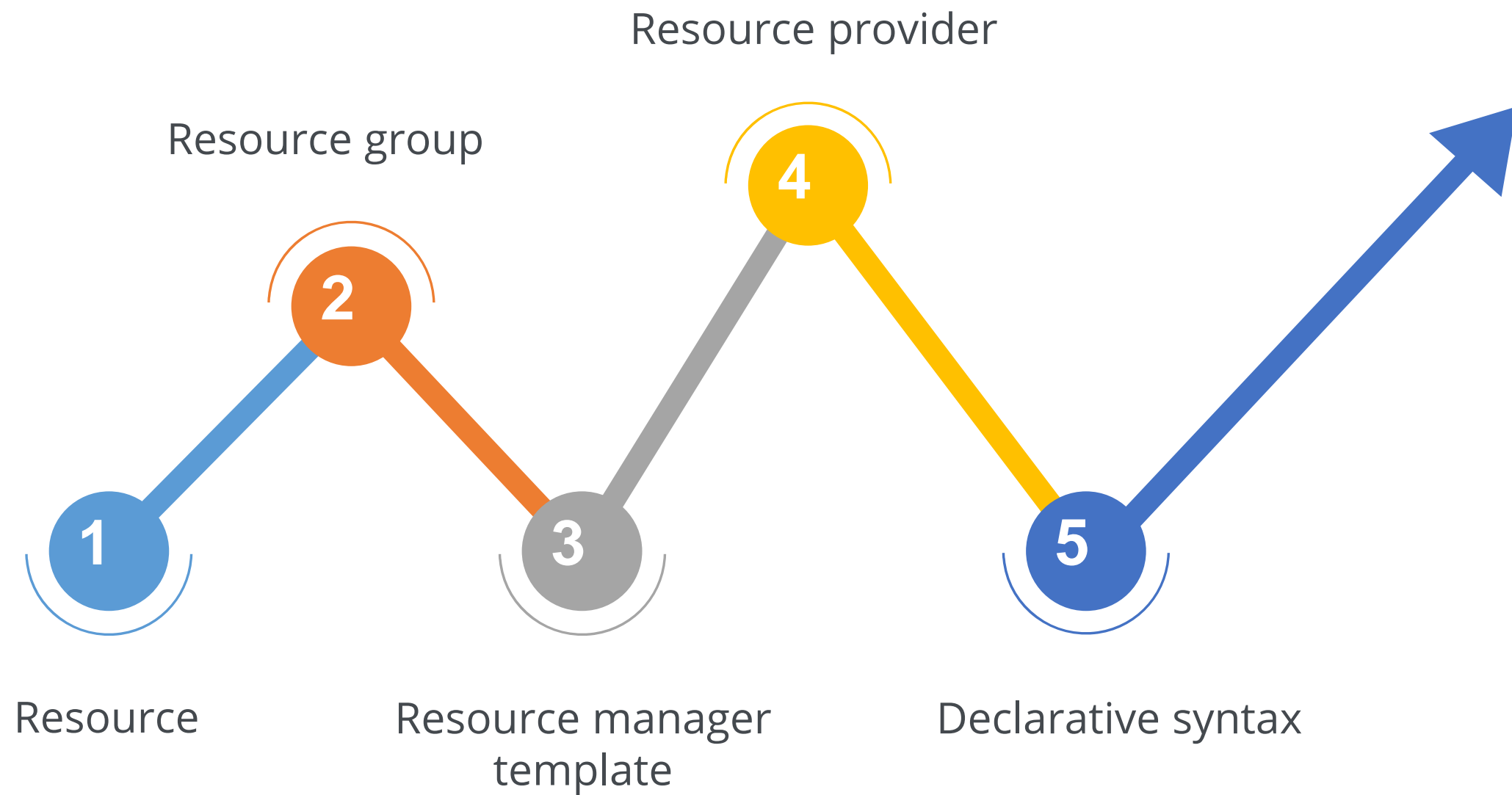
The flowchart given below shows the role of the Azure Resource Manager in handling Azure requests:



After deployment, the user can use administration tools like access control, locks, and tags to secure and arrange their resources.

Terminology

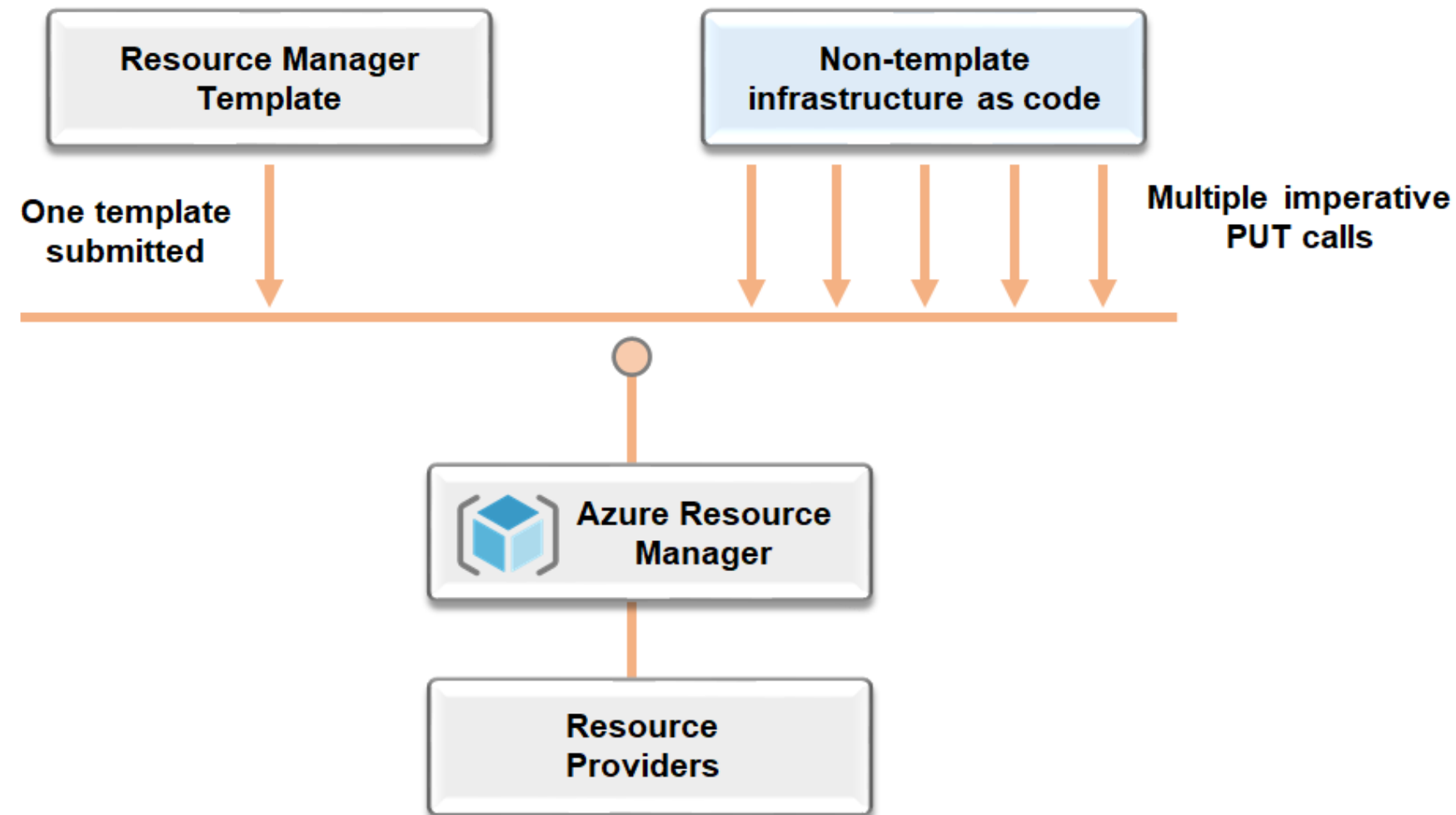
If you're new to Azure Resource Manager, there are a few terms you may not be acquainted with:



Azure Resource Manager Template

Azure Resource Manager Templates

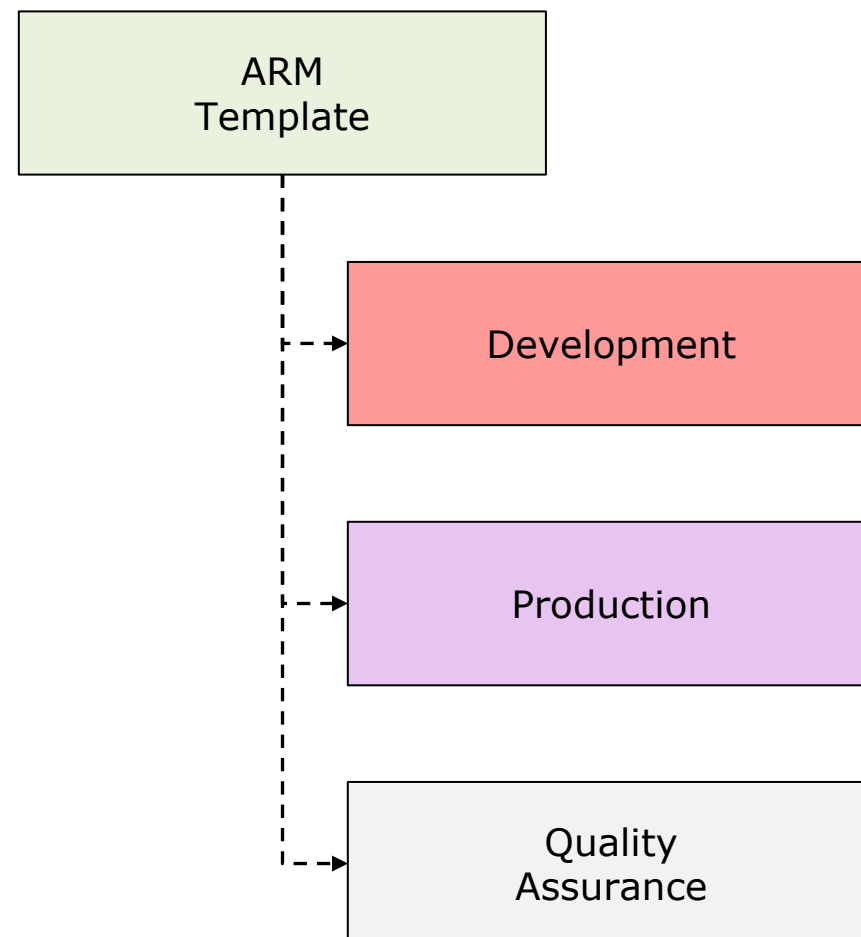
Azure Resource Manager templates are used to implement infrastructure as code for the Azure solutions.



It is a JavaScript Object Notation (JSON) file that defines all the resource manager resources in a deployment.

Template Advantages

Resource Manager templates will make the deployments quicker and more repeatable.

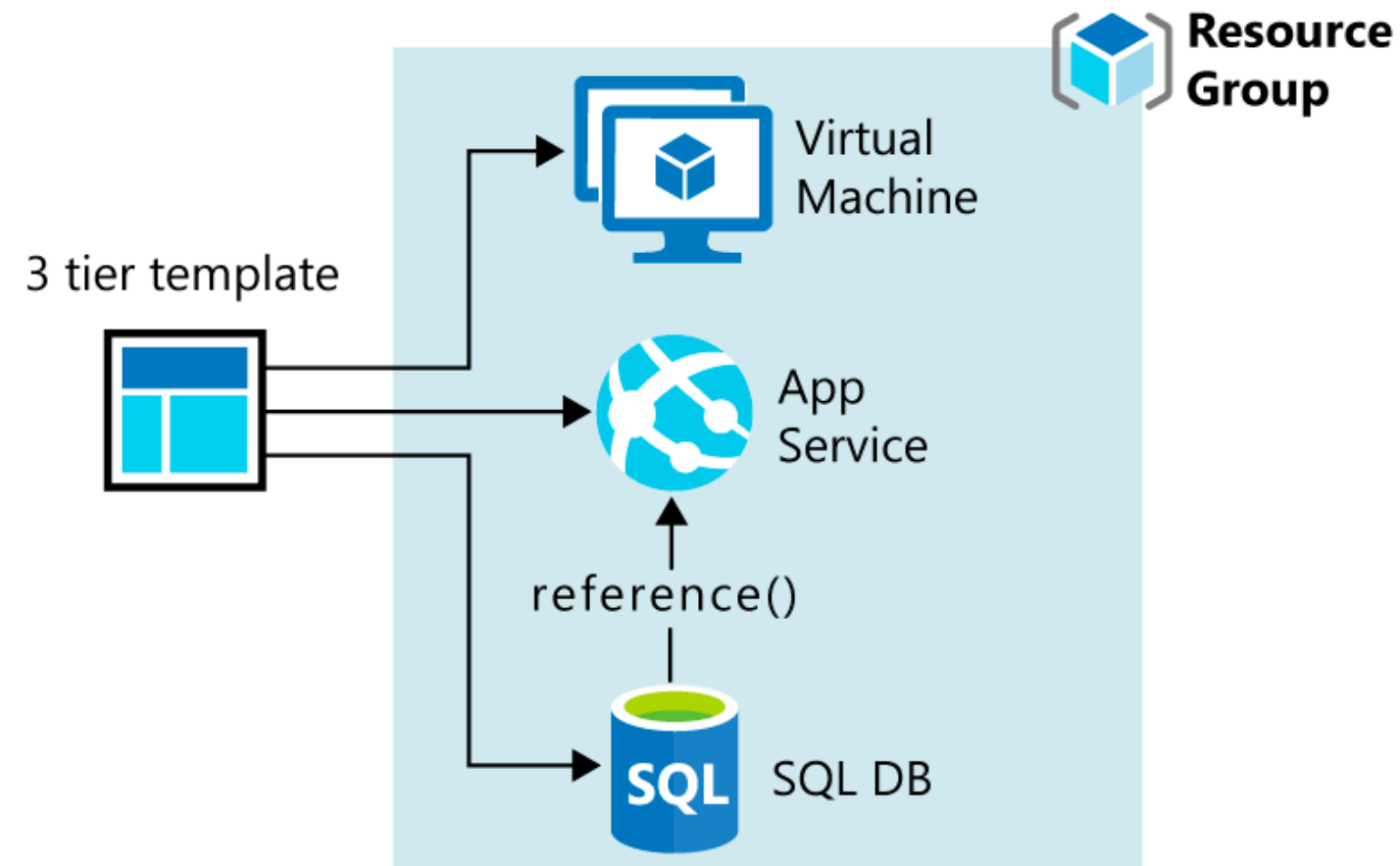


Advantages

- Improves consistency
- Express complex deployments
- Reduce manual and error prone tasks
- Express requirements through code
- Promotes reuse
- Simplifies orchestration

Template Design

The users can define templates and resource groups depending on how they want to manage the solution.



Example: The users can deploy their three-tier application through a single template to a single resource group.

image source: <https://docs.microsoft.com/en-in/>

Template Design

The users can divide their deployment requirements into a set of targeted, purpose-specific templates.

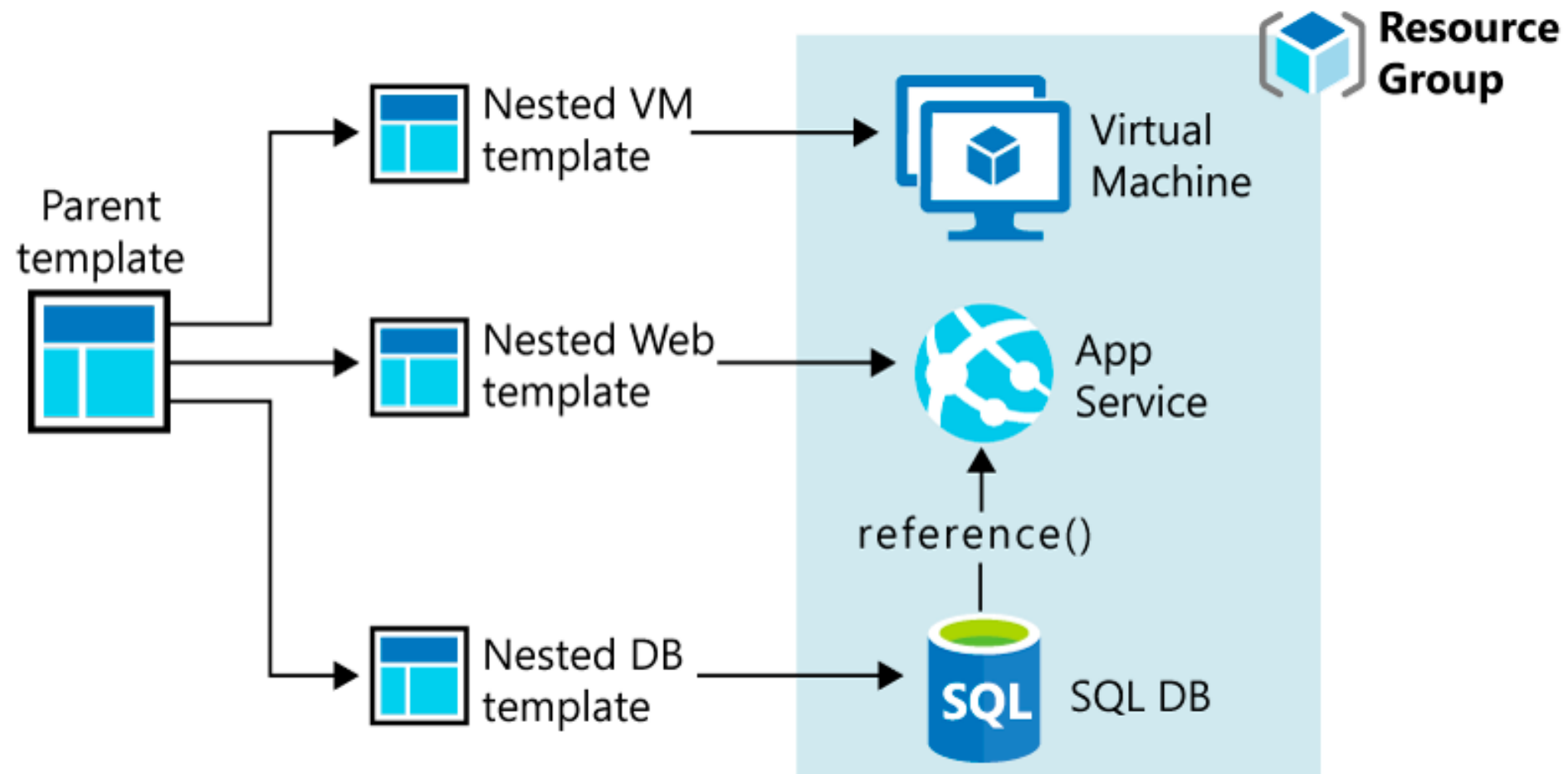


image source: <https://docs.microsoft.com/en-in/>

Template Design

If the user tiers have separate lifecycles, the user can deploy his three tiers to separate resource.

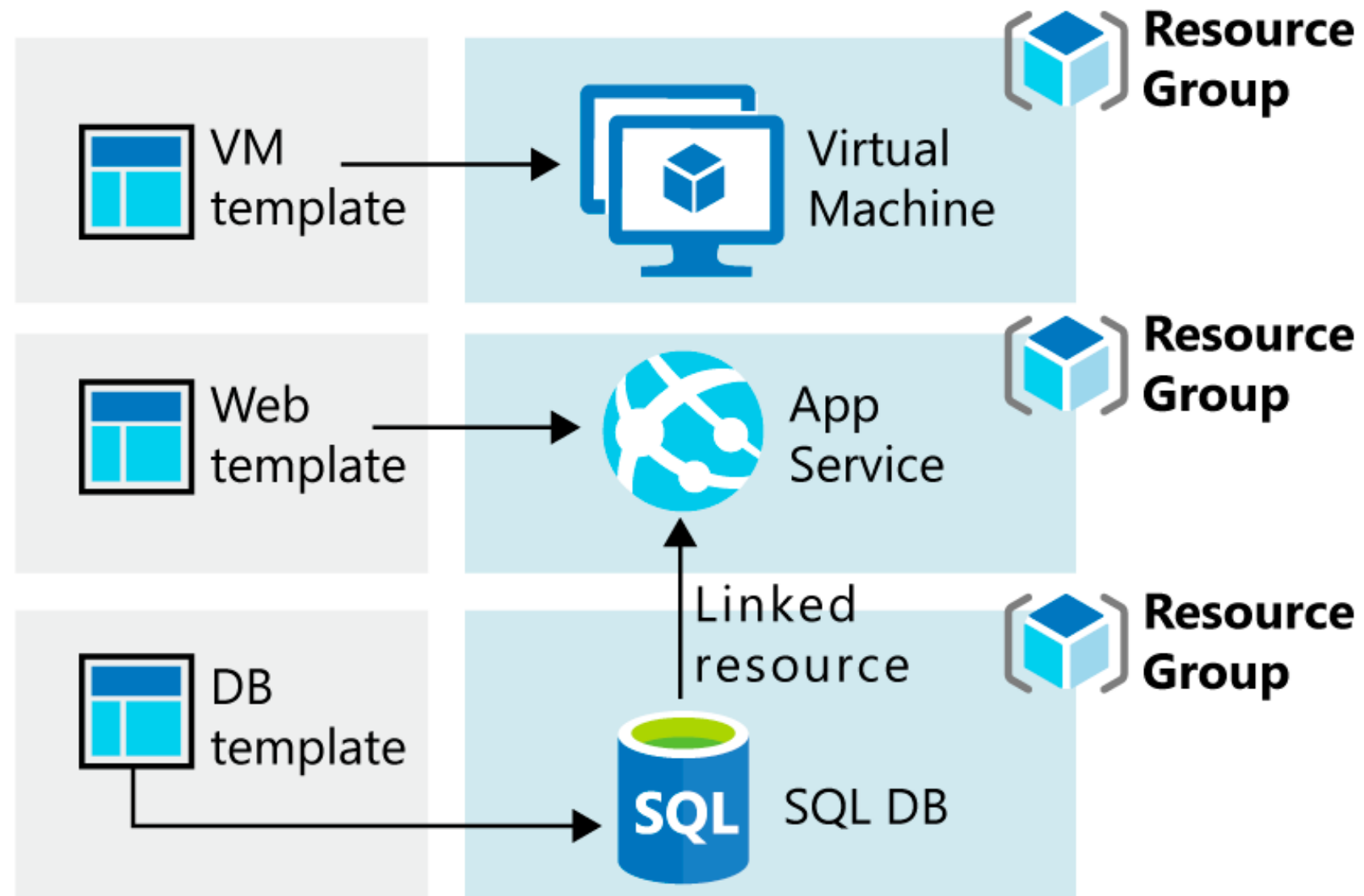


image source: <https://docs.microsoft.com/en-in/>

Template Schema

Template Schema defines all the resource manager resources in a deployment.

Features

- Is written in JSON
- Is a collection of key-value pairs
- Each key is a string
- Each values can be a string, number, Boolean expression, list of values, or object

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "",
  "parameters": { },
  "variables": { },
  "functions": [ ],
  "resources": [ ],
  "outputs": { }
}
```

Resource Manager Template Sections

Element name	Required	Description
\$schema	Yes	Location of the JSON schema file that describes the version of the template language. Use the URL shown in the preceding example.
contentVersion	Yes	Version of the template (such as 1.0.0.0). The users can provide any value for this element. Use this value to document significant changes in their template. When deploying resources using the template, this value can be used to make sure that the right template is being used.
parameters	No	Values that are provided when deployment is executed to customize resource deployment.
variables	No	Values that are used as JSON fragments in the template to simplify template language expressions.
functions	No	User-defined functions that are available within the template.
resources	Yes	Resource types that are deployed or updated in a resource group.
outputs	No	Values that are returned after deployment.

Resource Manager Template

Parameters

It specifies the values that are configurable when the template runs.

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

Resource Manager Template

Variables

- Define values that are used throughout the template
- Makes the templates easier to maintain

E.g: This example provides variables that describe networking features for a virtual machine

```
"variables": {  
  "nicName": "myVMNic",  
  "addressPrefix": "10.0.0.0/16",  
  "subnetName": "Subnet",  
  "subnetPrefix": "10.0.0.0/24",  
  "publicIPAddressName": "myPublicIP",  
  "virtualNetworkName": "MyVNET"  
}
```

Resource Manager Template

Functions

It defines procedures which the users don't want to repeat throughout the template.

```
"functions": [  
  {  
    "namespace": "contoso",  
    "members": {  
      "uniqueName": {  
        "parameters": [  
          {  
            "name": "namePrefix",  
            "type": "string"  
          }  
        ],  
        "output": {  
          "type": "string",  
          "value": "[concat(toLower(parameters('namePrefix')),  
uniqueString(resourceGroup().id))]"  
        }  
      }  
    }  
  }  
],
```


Resource Manager Template

Resources

This is where a user defines the Azure resources that make up the deployment.

```
"resources": [  
  {  
    "type": "Microsoft.Network/publicIPAddresses",  
    "name": "[variables('publicIPAddressName')]",  
    "location": "[parameters('location')]",  
    "apiVersion": "2018-08-01",  
    "properties": {  
      "publicIPAllocationMethod": "Dynamic",  
      "dnsSettings": {  
        "domainNameLabel": "[parameters('dnsLabelPrefix')]"  
      }  
    }  
  }  
],
```

Resource Manager Template

Outputs

Outputs define any information a user would like to receive when the template runs.

```
"outputs": {  
  "hostname": {  
    "type": "string",  
    "value":  
    "[reference(variables('publicIPAddressName')).dnsSettings.fqdn]"  
  }  
}
```

Azure Quickstart Templates

Azure Quickstart templates are resource manager templates that are provided by the Azure community.

Quickstart templates are available on GitHub, where the user can:

- Select a template
- View the template's source code on GitHub
- Visualize the template
- Review JSON that defines the Azure resource

Very simple deployment of a Windows VM

This template allows you to deploy a simple Windows VM using a few different options. This will deploy a A2 size VM in the resource group location name of the VM.

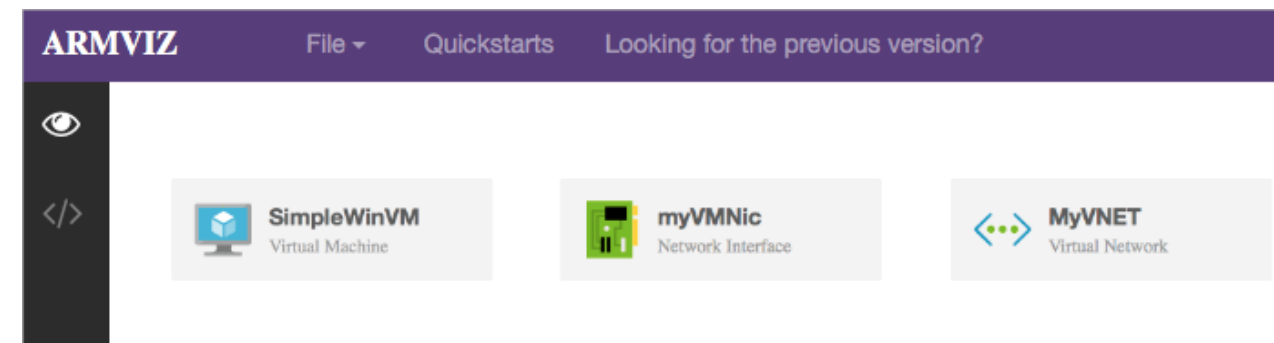


image source: <https://docs.microsoft.com/en-in/>

Downloading a Template

Download the Template for a VM

A user can export and download the template using the Azure portal.

The screenshot shows the Azure portal interface for a virtual network named 'myVnet'. The left sidebar contains navigation options: Connected devices, Subnets, DDoS protection, Firewall, Security, DNS servers, Peerings, Service endpoints, Private endpoints, Properties, Locks, Export template (highlighted), Monitoring (Diagnostic settings, Logs, Connection monitor, Diagram), and Support + troubleshooting (Connection troubleshoot, New support request). The main area is titled 'myVnet | Export template' and includes a toolbar with 'Download', 'Add to library (preview)', and 'Deploy' buttons. A message states: 'To export related resources, select the resources from the Resource Group view then select the "Export template" option from the tool bar.' Below this, there are tabs for 'Template', 'Parameters', and 'Scripts'. The 'Template' tab is active, showing a list of resources: Parameters (1), Variables (0), and Resources (2). The resources listed are: [parameters('virtualNetworks_myVnet_name')] (Microsoft.Network/virtualNetworks) and [concat(parameters('virtualNetworks_myVnet_name'), '/mySubnet')] (Microsoft.Network/virtualNetworks/subnets). The right pane displays the JSON template for the virtual network, which includes schema information, content version, parameters, variables, and resource definitions.

```
1 {
2   "$schema": "https://
3     schema.management.azure.com/schemas/2015-01-01/
4     deploymentTemplate.json#",
5   "contentVersion": "1.0.0.0",
6   "parameters": {
7     "virtualNetworks_myVnet_name": {
8       "defaultValue": "myVnet",
9       "type": "String"
10    },
11   "variables": {},
12   "resources": [
13     {
14       "type": "Microsoft.Network/
15       virtualNetworks",
16       "apiVersion": "2020-04-01",
17       "name": "[parameters
18         ('virtualNetworks_myVnet_name')]",
19       "location": "eastus",
20       "properties": {
21         "addressSpace": {
22           "addressPrefixes": [
23             "192.168.0.0/16"
24           ]
25         }
26       }
27     }
28   ]
29 }
```

Download the Template for a VM Using PowerShell

A user can download ARM template using PowerShell.

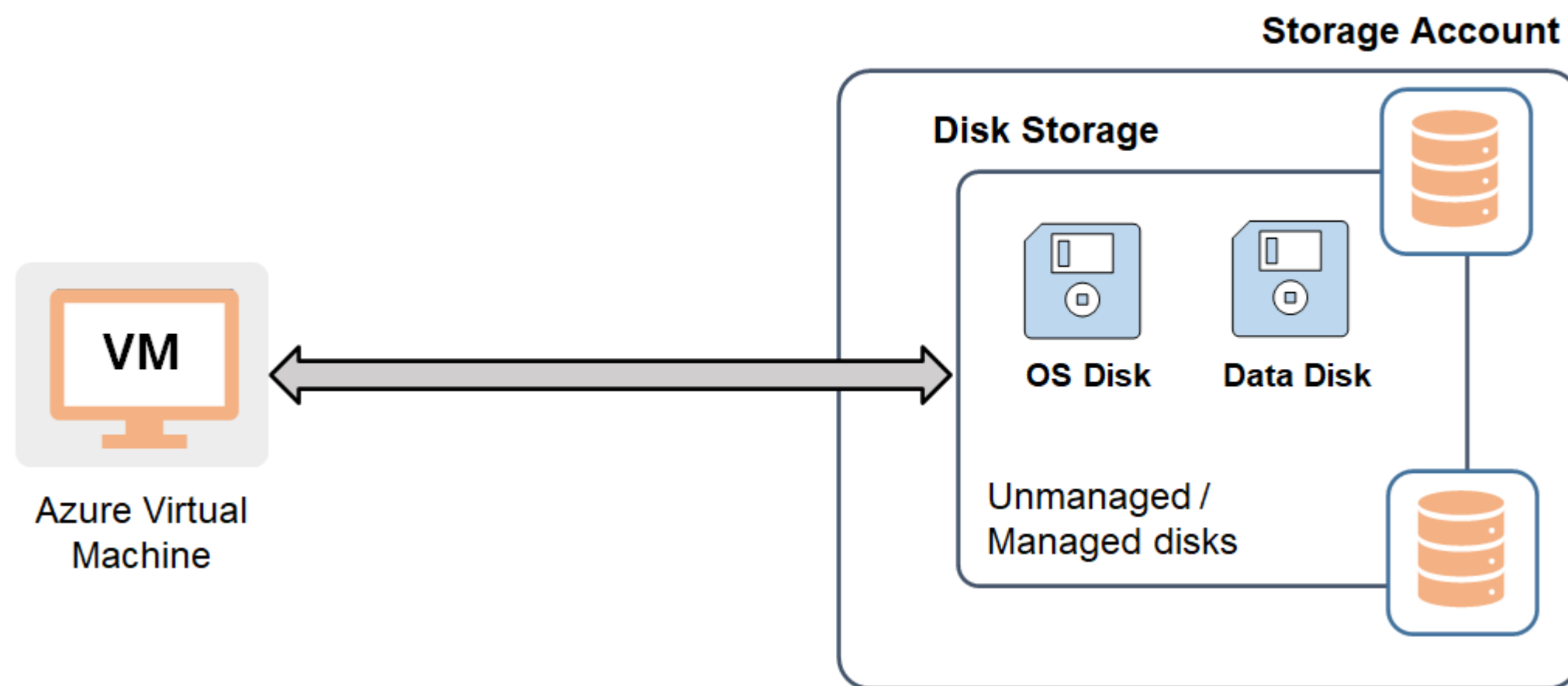
Example

```
Export-AzResourceGroup `
  -ResourceGroupName "myResourceGroup" `
  -Path "C:\users\public\downloads"
```


Virtual Hard Disk Template

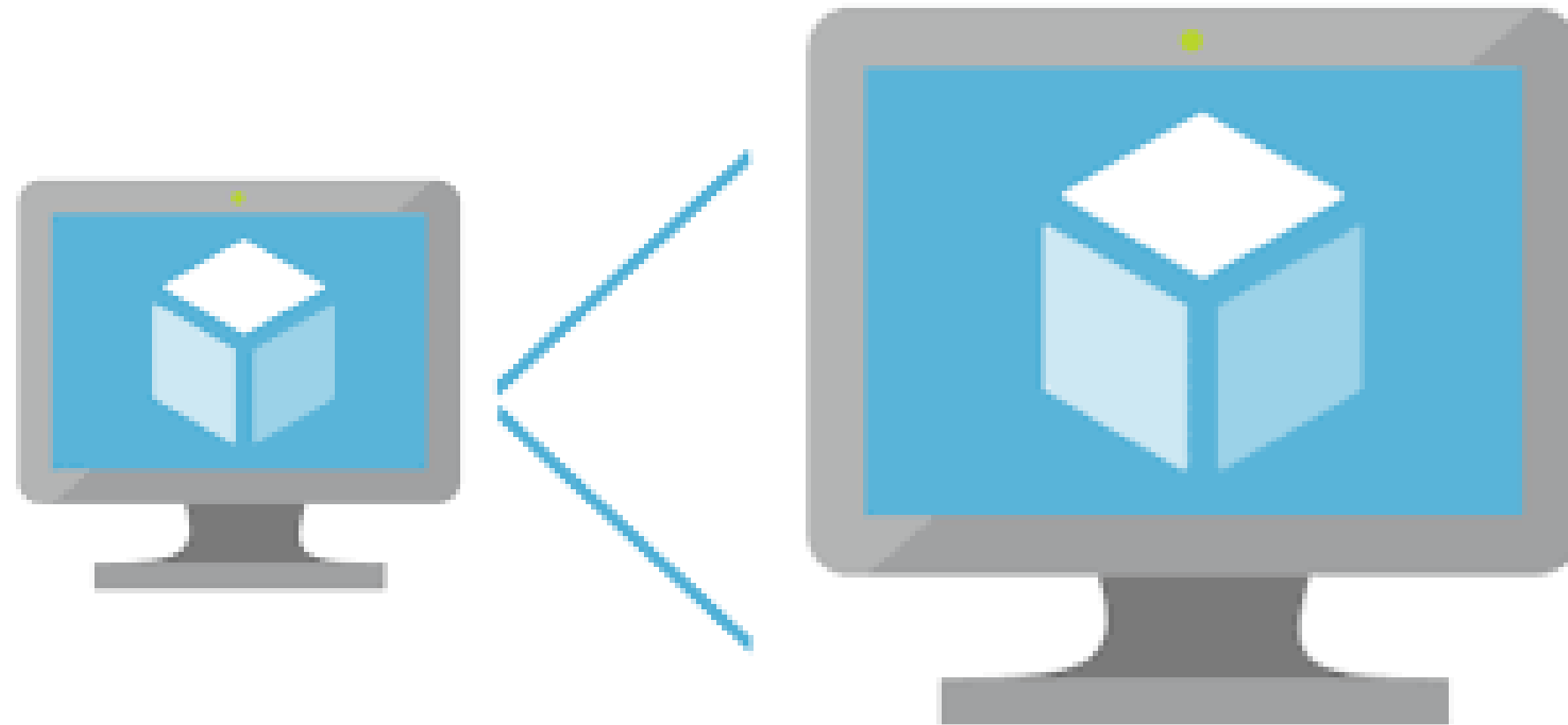
Azure Virtual Hard Disk

A virtual machine can have multiple VHDs. It has an operating system VHD that has an OS installed. VM also has data VHDs, which contains applications and user specific data.



Virtual Machine Image

Virtual machine image is a template from which a user can create the VHDs to run a virtual machine.



Deploy an Azure VM from a VHD

Use a deployment template:

Include the `vhdUrl` parameter (URL of the virtual hard disk)

Run a PowerShell script:

```
# specify storage account of an existing generalized VHD
$storageaccount = (...)
set generalized VHD URL
$vhdUrl =
"https://$storageaccount.blob.core.windows.net/vhds/(...) .
vhd"
# deploy a VM using the existing VHD
New-AzResourceGroupDeployment -vhdUrl $vhdUrl (...)
```

Virtual Machine Image

Virtual Hard Disk Deployment Template:


```
"storageProfile": {
  "osDisk": {
    "name": "[concat(parameters('vmName'), '-osDisk')]",
    "osType": "[parameters('osType')]",
    "caching": "ReadWrite",
    "image": {
      "uri": "[parameters('vhdUrl')]"
    },
    "vhd": {
      "uri": "[variables('osDiskVhdName')]"
    },
    "createOption": "FromImage"
  }
},
```

Create a VM from a VHD


To create an image settings the user can use:

- Azure PowerShell
- Azure CLI


Create image




Before creating an image, use "sysprep /generalize" to prepare the Windows guest OS on the virtual machine. If you create an image from a virtual machine that hasn't been generalized, any virtual machines created from that image won't start. Click here to learn more.




Capturing a virtual machine image will make the virtual machine unusable. This action cannot be undone.

* Name 


windows-image-20190708130059 


* Resource group

(New) myimagesrg 

[Create new](#)

Before creating the image, this virtual machine will be deallocated automatically


☒ Automatically delete this virtual machine after creating the image 


Zone resiliency 

On

Off

Create

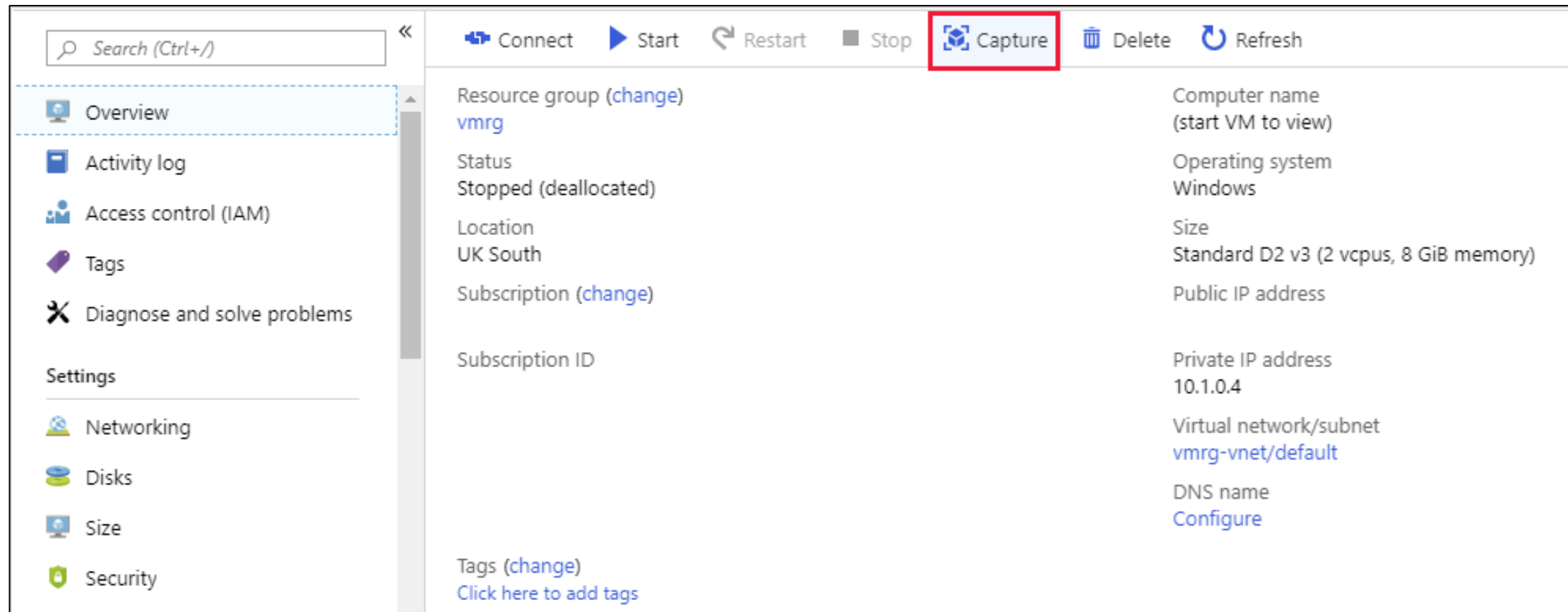
Powered by 

 Center for Technology & Management Education

Create a VM from a VHD

To create an image in the Azure portal:

- Generalize the OS of an Azure VM, and stop or deallocate it.
- Go to the blade of the virtual machine, and select Capture.



Deploying a Template

Template Deployment Options

The templates can be deployed by these options:

Using Azure Portal

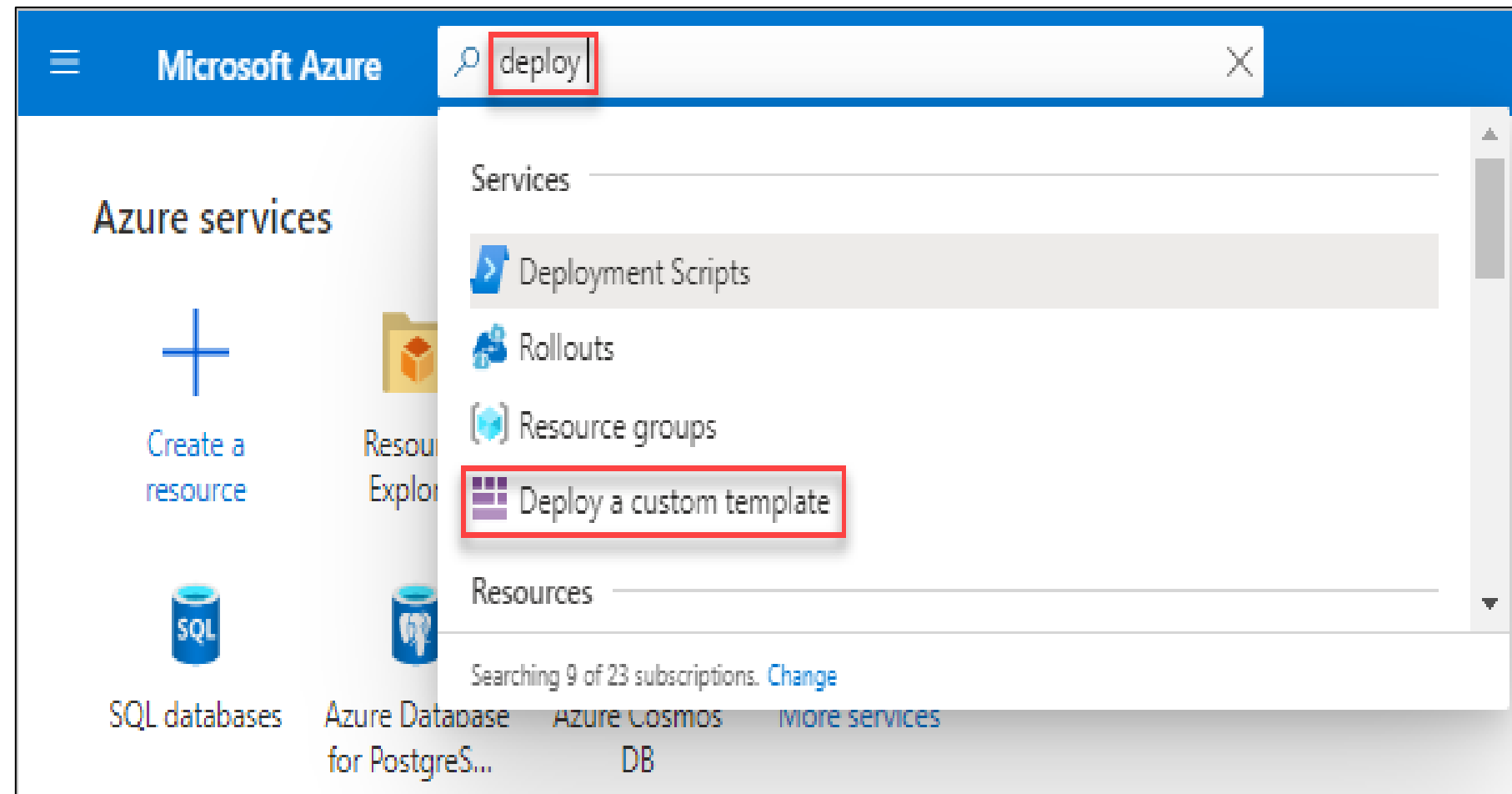
Using CLI

Creating ARM template
using VS code

Using Powershell

Template Deployment Options

The user can deploy the custom template from the Azure portal.



Template Deployment Options

The user can also deploy the template using PowerShell.

Example

```
$templateFile = "{provide-the-path-to-the-template-file}"  
New-Az Resource Group Deployment `   
  -Name blank template `   
  -ResourceGroupName my Resource Group `   
  -Template File $template File
```

Deploy an ARM Template Using CLI

The user can deploy the ARM template using CLI.

Example

```
"resources": [  
  {  
    "type": "Microsoft.Storage/storageAccounts",  
    "apiVersion": "2019-04-01",  
    "name": "[variables('storageAccountName')]",  
    "location": "[parameters('location')]",  
    "sku": {  
      "name": "[parameters('storage SKU')]"  
    },  
    "kind": "Storage V2",  
    "properties": {  
      "supportsHttpsTrafficOnly": true  
    }  
  },  
]
```

Assisted Practice

ARM Template

Duration: 10 Min

Problem Statement:

You are given a project to download ARM Template to declare the objects you want, as well as their types, names, and properties, in a JSON file that can be checked into source control and maintained like any other code file.

Assisted Practice: Guidelines

Steps to create an Azure SQL DB are:

1. Login to your Azure portal.
2. Click on Create a resource
3. Select Storage and click on Storage Account
4. Download the ARM Template



Automation Runbook

Runbook in Azure Automation

The following figure illustrates the lifecycle of a runbook job for different types of runbooks:

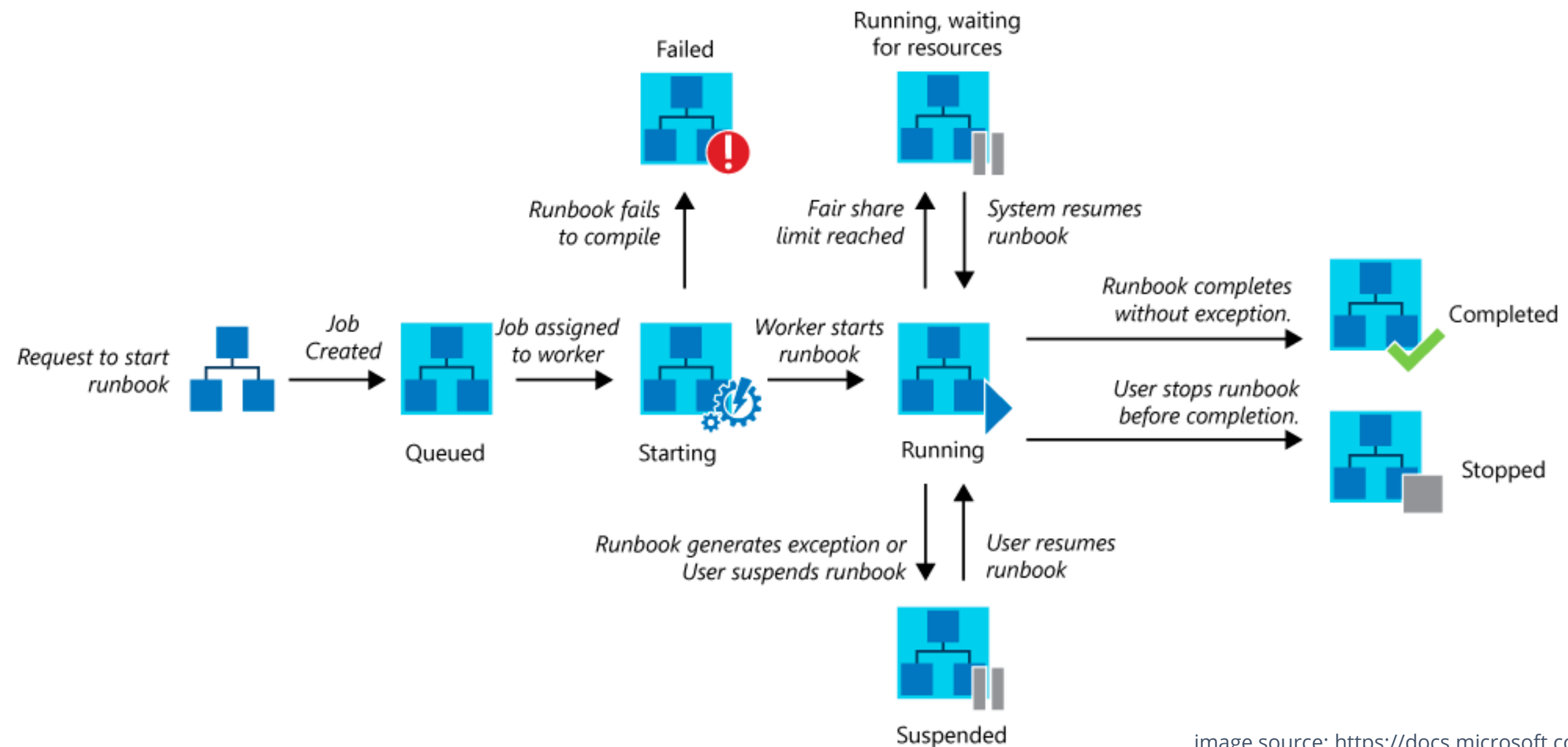
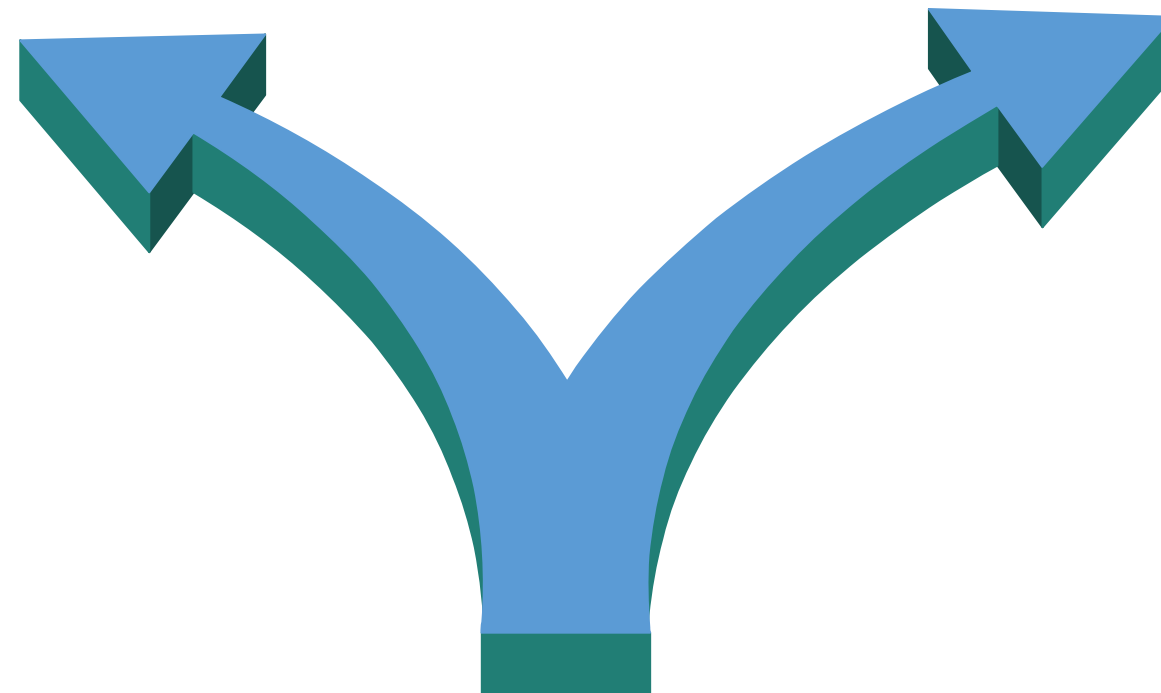


image source: <https://docs.microsoft.com/en-in/>

Runbooks in Azure Automation

An Azure sandbox:
A shared environment,
against Azure resources.



A hybrid runbook worker:
In any environment, directly
on the computer that hosts
the worker role and against
local resources in the
environment.

**Azure Automation
Runbooks can run in**

Importing PowerShell Runbook

Import a PowerShell Runbook from the Runbook Gallery

In the Azure portal:

- Select Runbooks gallery under **Process Automation**
- Select Source: **PowerShell Gallery**
- Locate and select the gallery item you want to import

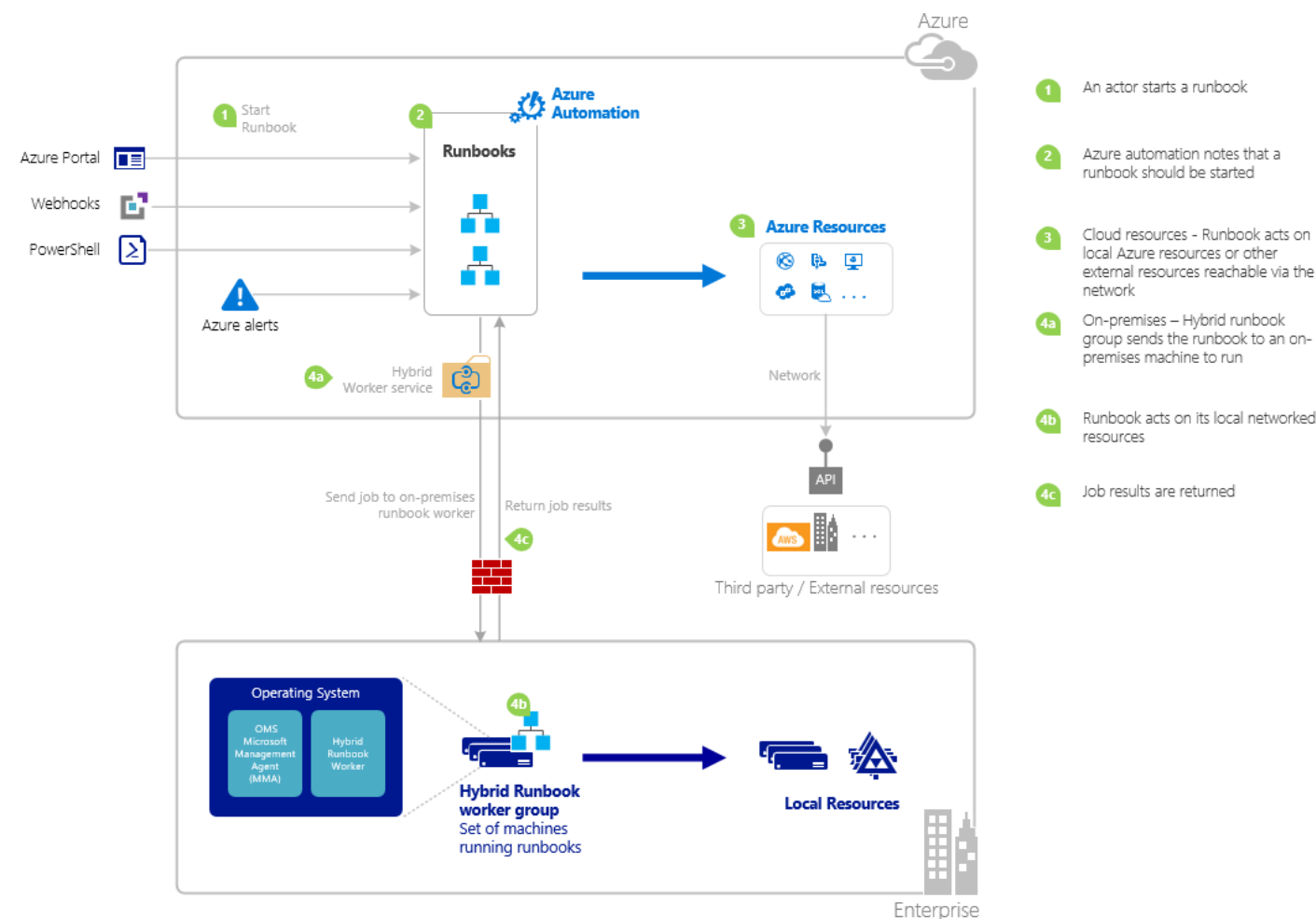
Other options are:

- Adding a PowerShell Runbook to the gallery
- Importing a module from the module gallery



Start a Runbook in Azure Automation

The following diagram shows the life cycle of a Runbook:



- 1 An actor starts a Runbook
- 2 Azure automation
- 3 Cloud Resources
- 4a on-premise
- 4b Hybrid Runbook worker group
- 4c Return job results

image source: <https://docs.microsoft.com/en-in/>

Start a Runbook in Azure Automation

A runbook can be started using:

1

Azure portal

2

PowerShell

```
Start-AzAutomationRunbook
  -AutomationAccountName
  "MyAutomationAccount" `
  -Name "Test-Runbook" `
  -ResourceGroupName "ResourceGroup01"
```

Key Takeaways

- Azure Resource Manager Templates are used to implement infrastructure as code for the Azure solutions.
- Azure Quickstart templates are Resource Manager templates that are provided by the Azure community.
- Virtual machine image is a template from which the user can create the Virtual Hard Disks to run a virtual machine.
- Azure Automation Runbooks can run in an Azure Sandbox and Hybrid Runbook Worker.



Automate the deployment of storage accounts

Duration: 10 Min.

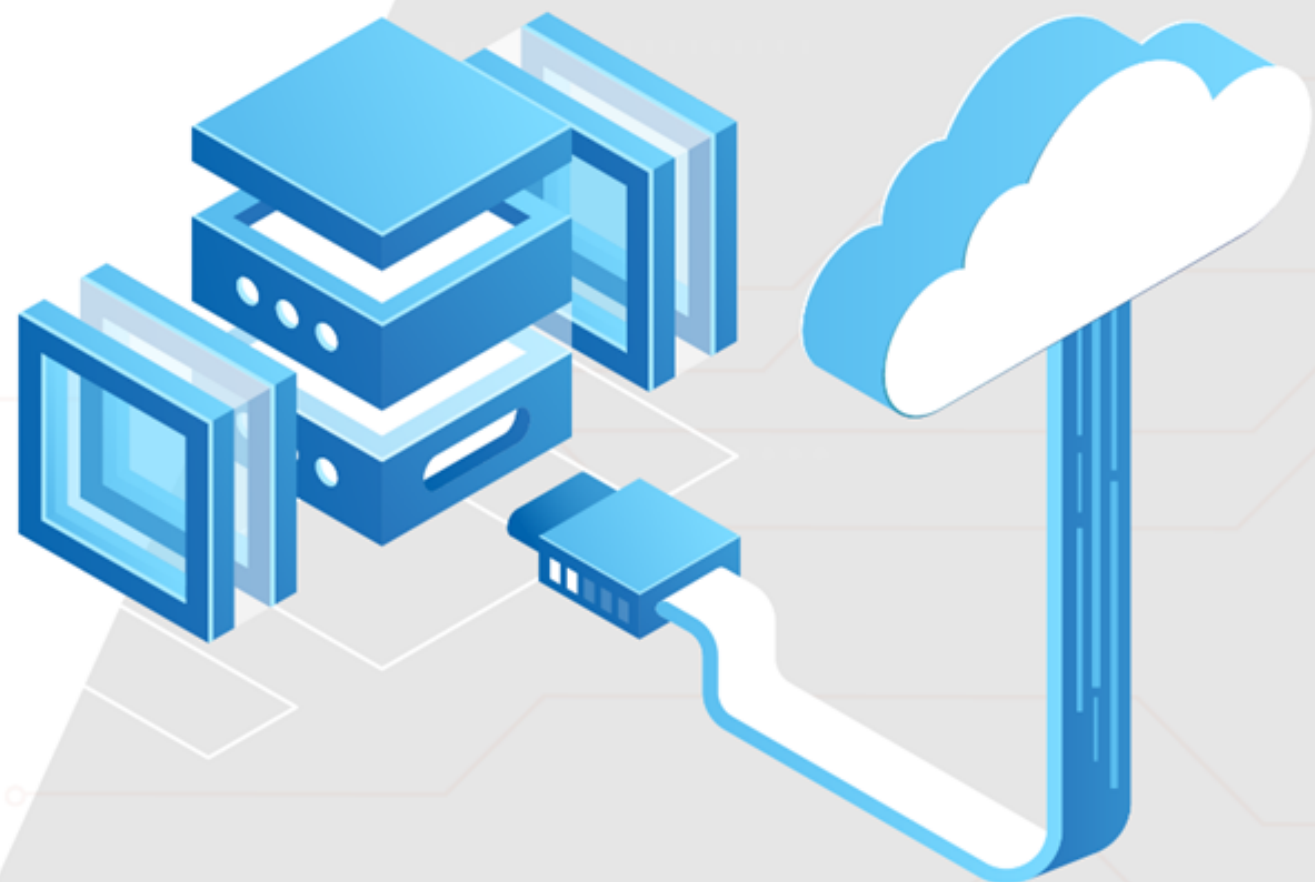
Project agenda: To automate the deployment of storage accounts

Description: You have been given a project to automate the deployment of storage accounts using Infrastructure as a Code. You need to create an ARM template to deploy a storage account which can be used repeatedly for deploying storage accounts.

Perform the following:

Create an ARM Template to deploy a storage account.





Thank you