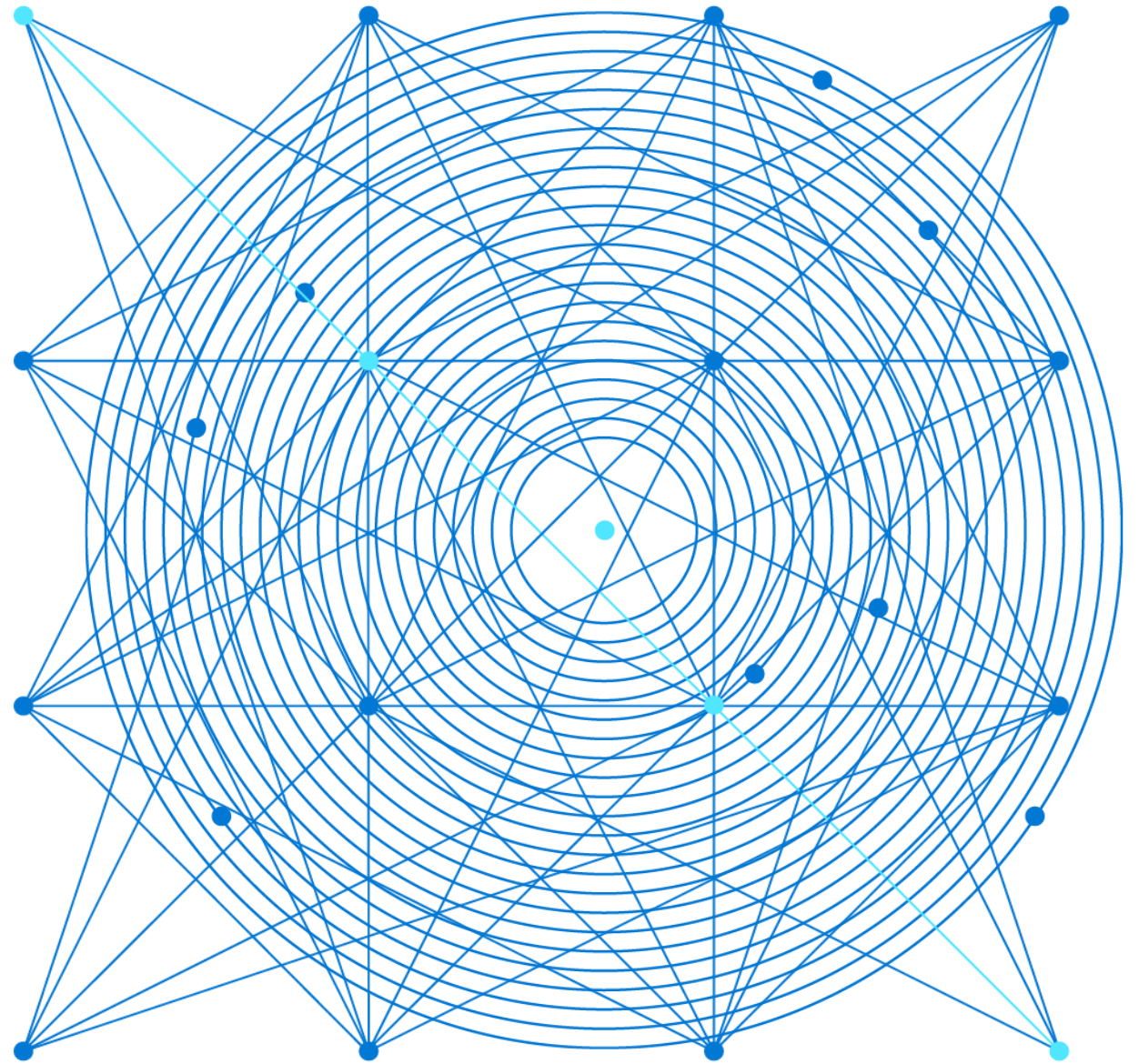


# AZ-303: Microsoft Azure Architect Technologies



# Module 9: Automate Deployment and Configuration of Resources

Azure Resource Manager Templates, Virtual Hard Disk Template, and Automation Runbooks

# Learning Objectives

You will learn the following:

- Azure Resource Manager Templates
- Save a template for a VM
- Evaluate Location for New Resources
- Configure a Virtual Hard Disk Template
- Deploy from a Template
- Create and Execute an Automation Runbook

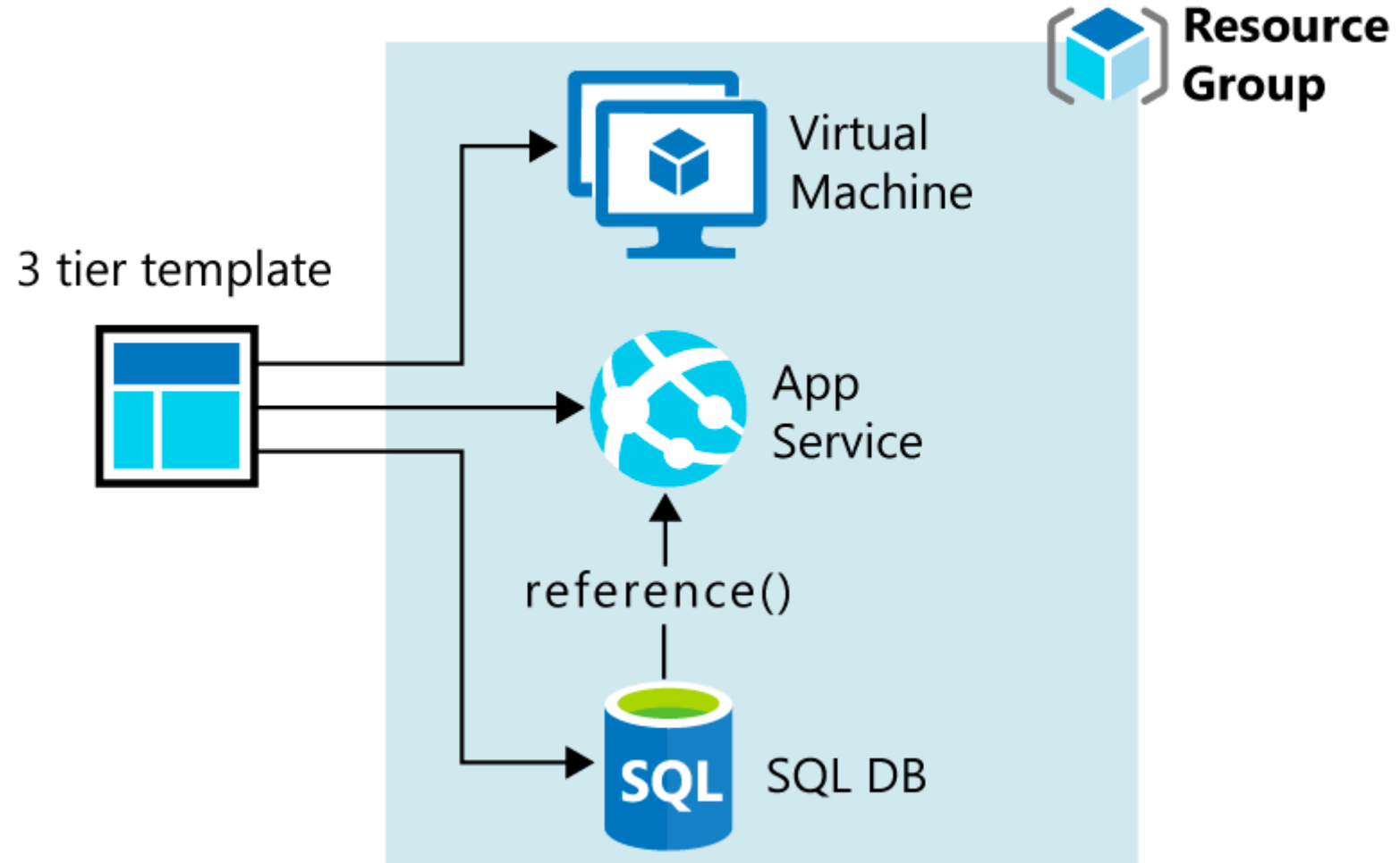


# Azure Resource Manager Templates

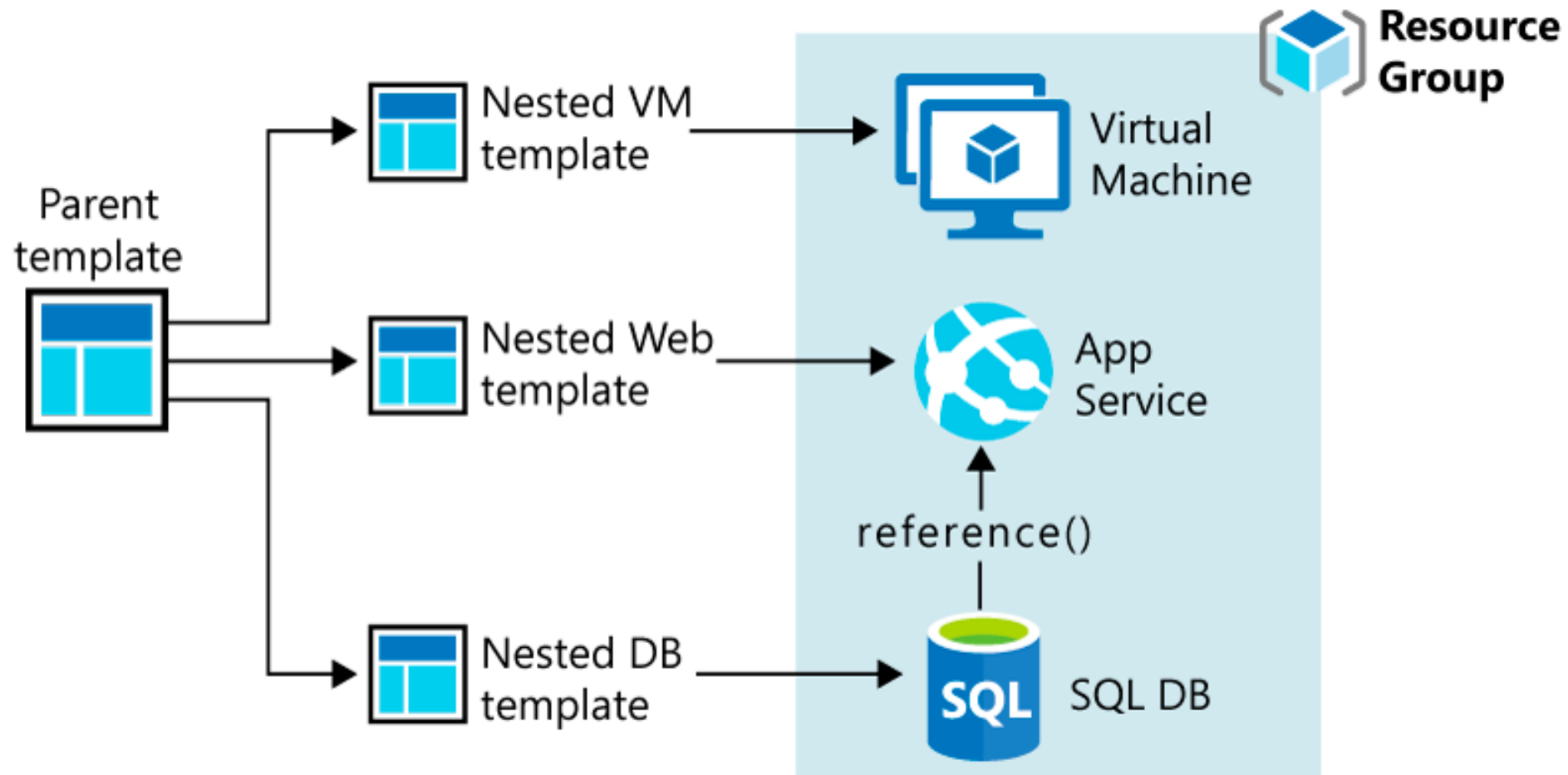


# Overview of Resource Manager Templates (1 of 3)

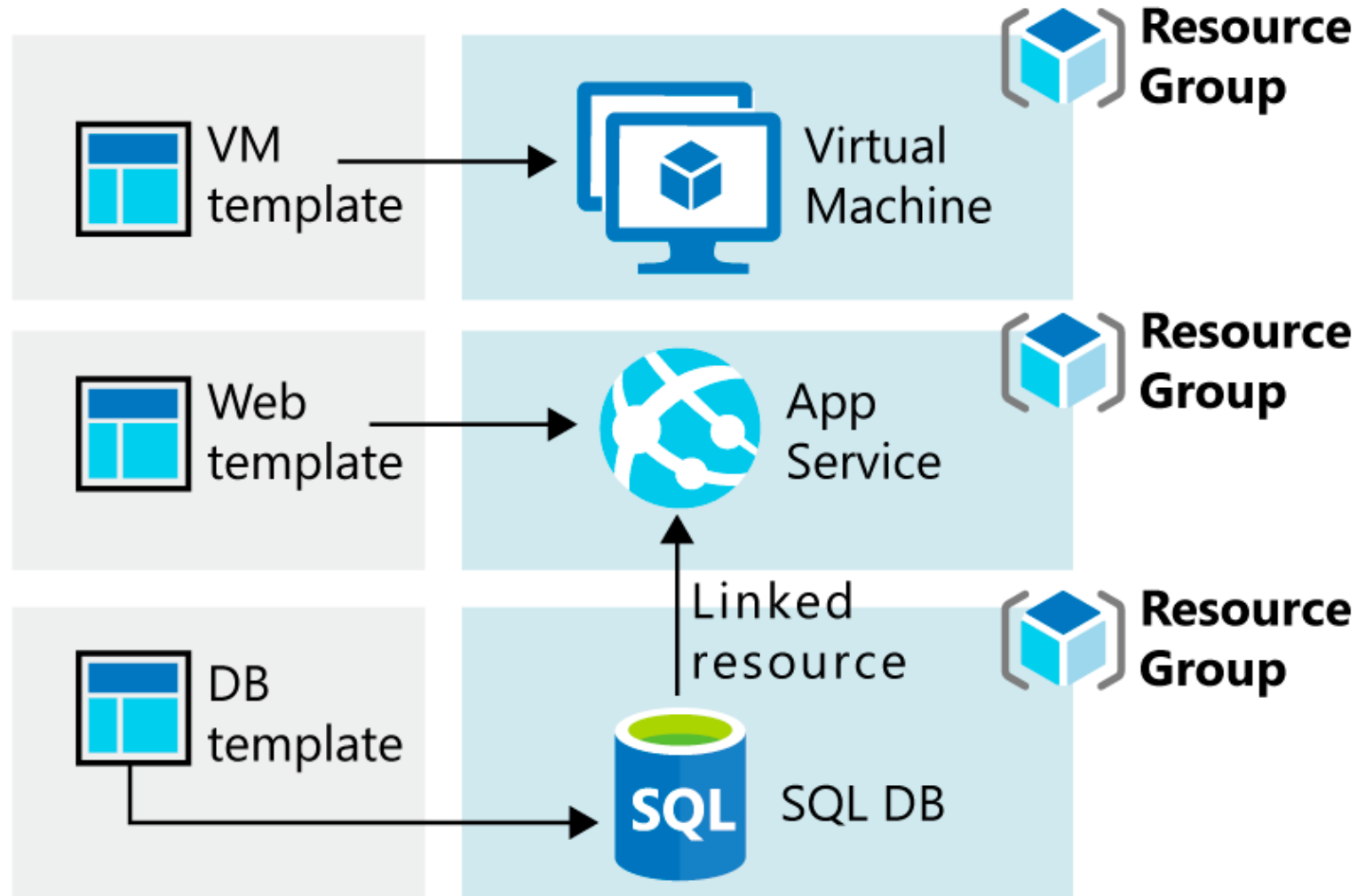
- What are Resource Manager templates?
- Template design



## Overview of Resource Manager Templates (2 of 3)



# Overview of Resource Manager Templates (3 of 3)



# What's in a Resource Manager Template? (1 of 6)

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "",
  "parameters": {  },
  "variables": {  },
  "functions": [  ],
  "resources": [  ],
  "outputs": {  }
}
```



## What's in a Resource Manager Template? (2 of 6)

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

## What's in a Resource Manager Template? (3 of 6)

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

# What's in a Resource Manager Template? (4 of 6)

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

## What's in a Resource Manager Template? (5 of 6)

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

# What's in a Resource Manager Template? (6 of 6)

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


# Azure Quickstart Templates (1 of 5)

Browse to Quickstart template gallery

## Most popular


### Create a Standard Storage Account

This template creates a Standard Storage Account

 by [Kay Singh](#),  
Last updated: 12/4/2018


### Create a Virtual Network with two Subnets

This template allows you to create a Virtual Network with two subnets.

 by [Telmo Sampaio](#),  
Last updated: 10/12/2018


### Create an Azure VM with a new AD Forest

This template creates a new Azure VM, it configures the VM to be an AD DC for a new Forest

 by [Simon Davies](#),  
Last updated: 7/4/2018

### Join a VM to an existing domain

This template demonstrates domain join to a private AD domain up in cloud.


 by [Kay Singh](#),  
Last updated: 5/25/2018

# Azure Quickstart Templates (2 of 5)

Select a template

[Templates](#) / Deploy a simple Windows VM

## Deploy a simple Windows VM

 by [Brian Moore](#)

[Deploy to Azure](#) [Browse on GitHub](#)

This template allows you to deploy a simple Windows VM using a few different options for the Windows version, using the latest patched version. This will deploy a A2 size VM in the resource group location and return the FQDN of the VM.

# Azure Quickstart Templates (3 of 5)

View the template's source code on GitHub

## Very simple deployment of a Windows VM



Deploy to Azure



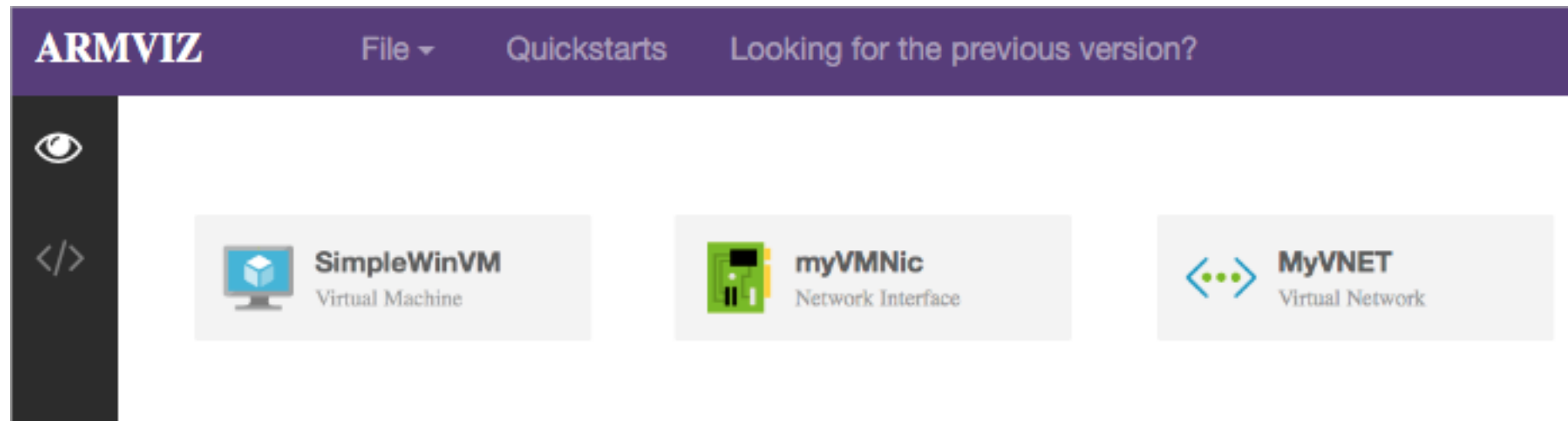
Visualize

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



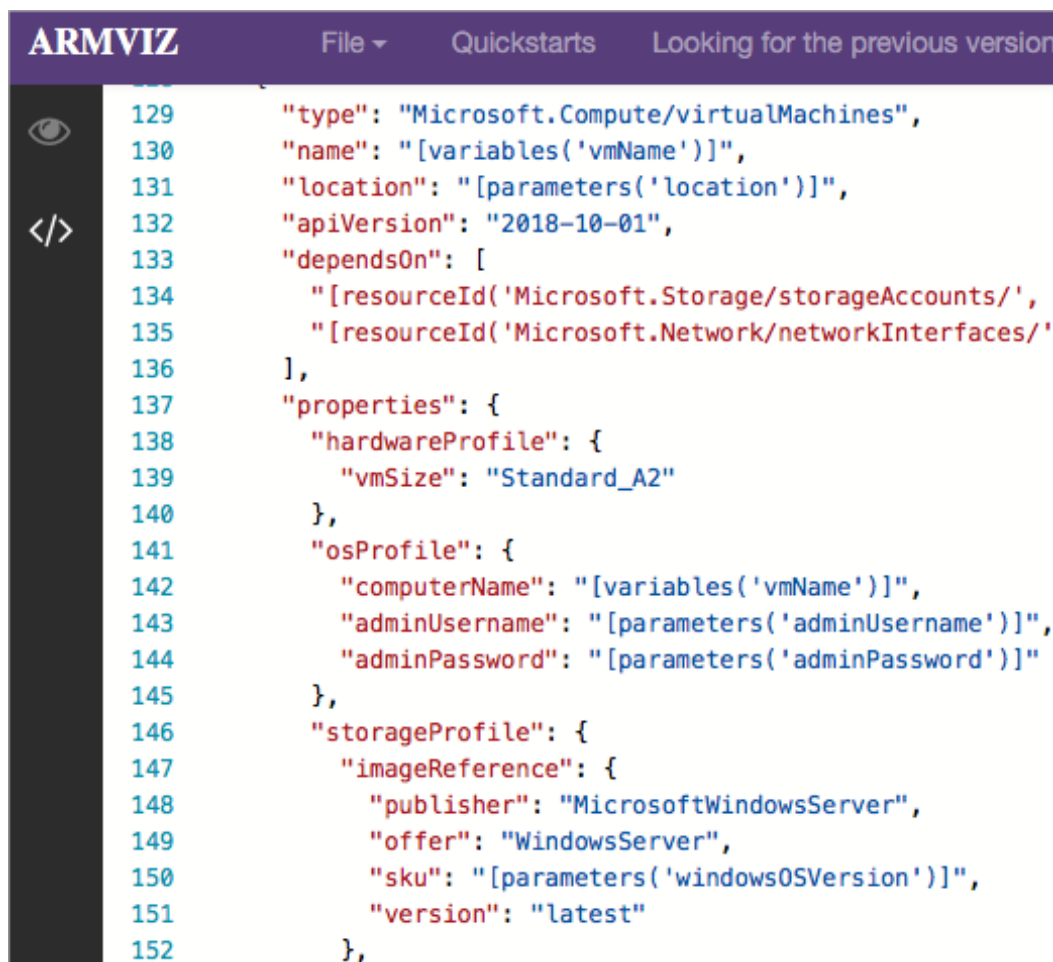
# Azure Quickstart templates (4 of 5)

Visualize the template



# Azure Quickstart templates (5 of 5)

Review JSON that defines the Azure resource



The screenshot shows the ARMVIZ application interface. The top bar is purple with the text 'ARMVIZ' on the left, and 'File', 'Quickstarts', and 'Looking for the previous version' on the right. On the left side of the editor, there is a dark vertical bar with an eye icon and a code icon. The main area displays a JSON snippet for an Azure resource, with line numbers 129 through 152 on the left. The JSON defines a virtual machine resource with various properties like name, location, apiVersion, dependsOn, hardwareProfile, osProfile, and storageProfile.

```
129     "type": "Microsoft.Compute/virtualMachines",
130     "name": "[variables('vmName')]",
131     "location": "[parameters('location')]",
132     "apiVersion": "2018-10-01",
133     "dependsOn": [
134       "[resourceId('Microsoft.Storage/storageAccounts/',",
135       "[resourceId('Microsoft.Network/networkInterfaces/',",
136     ],
137     "properties": {
138       "hardwareProfile": {
139         "vmSize": "Standard_A2"
140       },
141       "osProfile": {
142         "computerName": "[variables('vmName')]",
143         "adminUsername": "[parameters('adminUsername')]",
144         "adminPassword": "[parameters('adminPassword')]"
145       },
146       "storageProfile": {
147         "imageReference": {
148           "publisher": "MicrosoftWindowsServer",
149           "offer": "WindowsServer",
150           "sku": "[parameters('windowsOSVersion')]",
151           "version": "latest"
152         },
```

# Save a Template for a VM



# Download the Template for a VM

Export and download the template using the Azure portal

The screenshot shows the Azure portal interface for exporting a virtual network template. The breadcrumb navigation at the top indicates the path: Home > Virtual networks > myVnet | Export template. The page title is 'myVnet | Export template' with a sub-label 'Virtual network'. On the right, there are links for 'Documentation' and a close icon. Below the title bar, there is a search bar and three action buttons: 'Download', 'Add to library (preview)', and 'Deploy'. A left-hand navigation pane lists various network-related features, with 'Export template' highlighted. The main content area includes an information message about selecting resources for export. Below this, there is a section for 'Include parameters' with tabs for 'Template', 'Parameters', and 'Scripts'. The 'Template' tab is active, showing a tree view of resources: 'Parameters (1)', 'Variables (0)', and 'Resources (2)'. The resources listed are '[parameters('virtualNetworks\_myVnet\_name')]' and '[concat(parameters('virtualNetworks\_myVnet\_name'), '/mySubnet')]' (Microsoft.Network/virtualNetworks/subnets). To the right, a code editor displays the JSON deployment template, which includes schema information, content version, parameters (with a default value for 'virtualNetworks\_myVnet\_name'), and resource definitions for the virtual network and its subnets.

Home > Virtual networks > myVnet | Export template

Virtual network

Search (Ctrl+/)

Download Add to library (preview) Deploy

To export related resources, select the resources from the Resource Group view then select the "Export template" option from the tool bar.

Include parameters

Template Parameters Scripts

Parameters (1)

Variables (0)

Resources (2)

- [parameters('virtualNetworks\_myVnet\_name')] (Microsoft.Network/virtualNetworks)
- [concat(parameters('virtualNetworks\_myVnet\_name'), '/mySubnet')] (Microsoft.Network/virtualNetworks/subnets)

```
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       "type": "Microsoft.Network/
30       virtualNetworks/subnets",
31       "apiVersion": "2020-04-01",
32       "name": "[concat(parameters
33       ('virtualNetworks_myVnet_name'), '/mySubnet')]",
34       "location": "eastus",
35       "properties": {
36         "addressPrefix": "192.168.0.0/24",
37         "enableDdosProtection": false,
38         "enableForwarding": false
39       }
40     }
41   ]
42 }
```

# Download the Template using PowerShell

An example of downloading an ARM template using PowerShell:

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

# Create a Virtual Hard Disk Template



# Disk Images for Azure VMs

- What is an Azure virtual hard disk?
- What is a virtual machine image?
- What is a generalized image?
- What is a specialized virtual image?

# 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 (...)
```



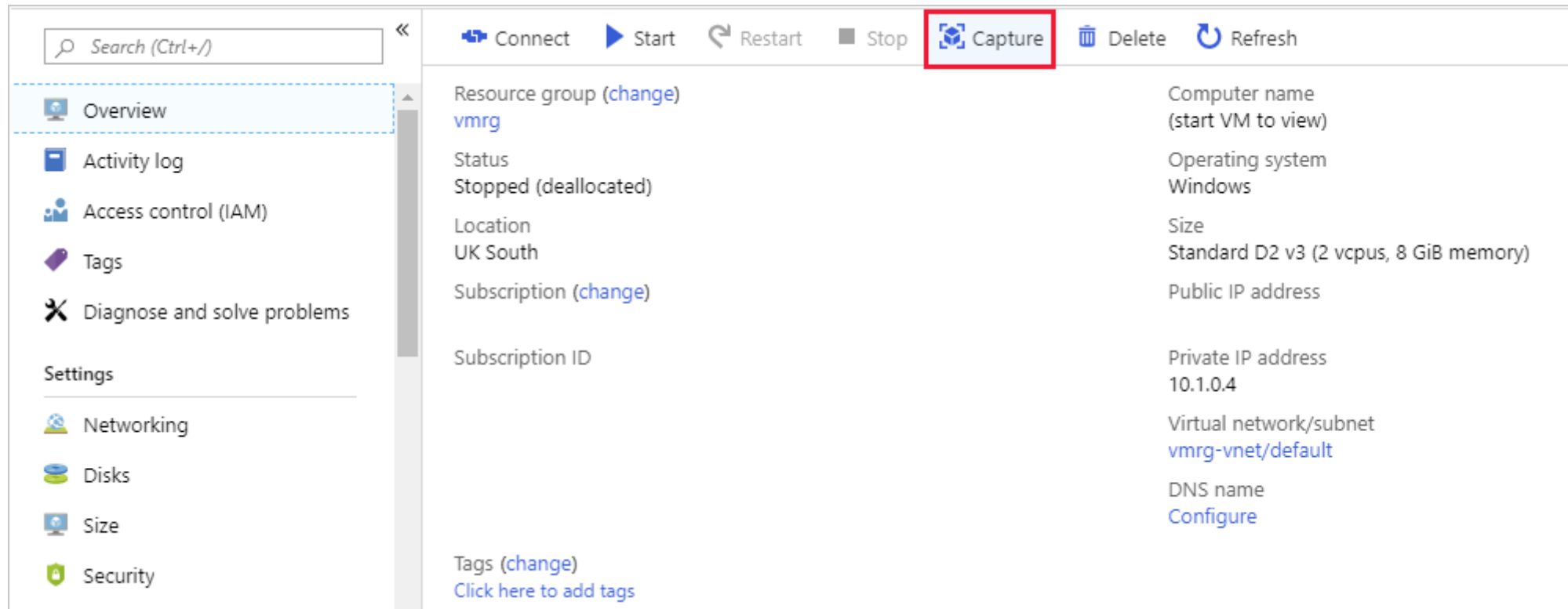
# Virtual hard Disk Deployment Template

```
"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 (1 of 3)

To create an image in the Azure portal:

- Generalize the OS of an Azure VM and stop/deallocate it
- Go to the blade of the virtual machine and select **Capture**




# Create a VM from a VHD (2 of 3)

Provide **Create image** settings


Alternatively, you 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

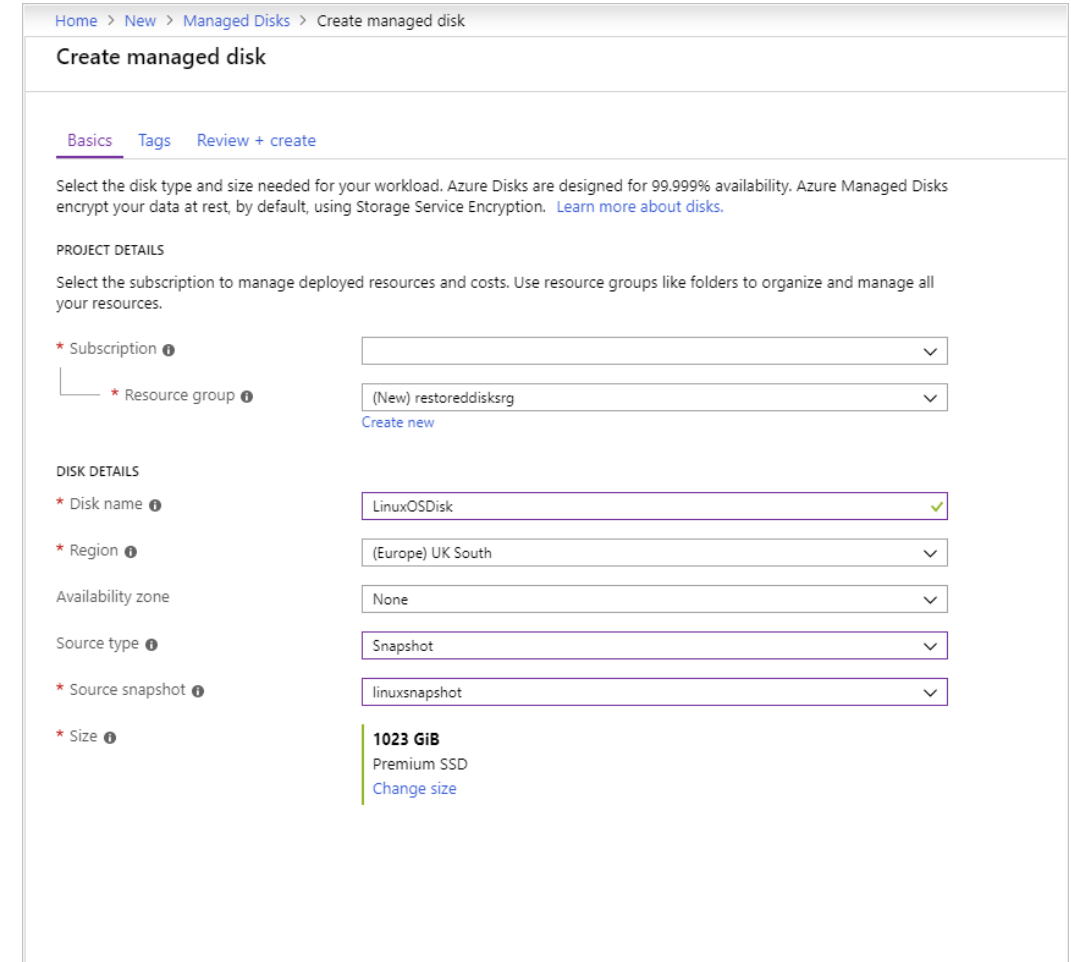
Off

Create

# Create a VM from a VHD (3 of 3)

You can also create an Azure VM from a snapshot of its VHD files:

- Create a snapshot of each VHD
- For each snapshot, create a managed disk
- Create an Azure VM using the managed disks



The screenshot shows the 'Create managed disk' page in the Azure portal. The breadcrumb navigation at the top reads 'Home > New > Managed Disks > Create managed disk'. The page title is 'Create managed disk'. Below the title are tabs for 'Basics' (selected), 'Tags', and 'Review + create'. A descriptive paragraph states: 'Select the disk type and size needed for your workload. Azure Disks are designed for 99.999% availability. Azure Managed Disks encrypt your data at rest, by default, using Storage Service Encryption. [Learn more about disks.](#)'

The 'PROJECT DETAILS' section includes instructions: 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' It contains two dropdown menus: 'Subscription' and 'Resource group'. The 'Resource group' dropdown is currently set to '(New) restoreddisksrg' with a 'Create new' link below it.

The 'DISK DETAILS' section contains several fields: 'Disk name' (set to 'LinuxOSDisk' with a green checkmark), 'Region' (set to '(Europe) UK South'), 'Availability zone' (set to 'None'), 'Source type' (set to 'Snapshot'), and 'Source snapshot' (set to 'linuxsnapshot'). The 'Size' field is set to '1023 GiB' with 'Premium SSD' and a 'Change size' link below it.

# Deploy from a Template



# Deploy an ARM template using CLI

Sample deployment of an Azure storage account

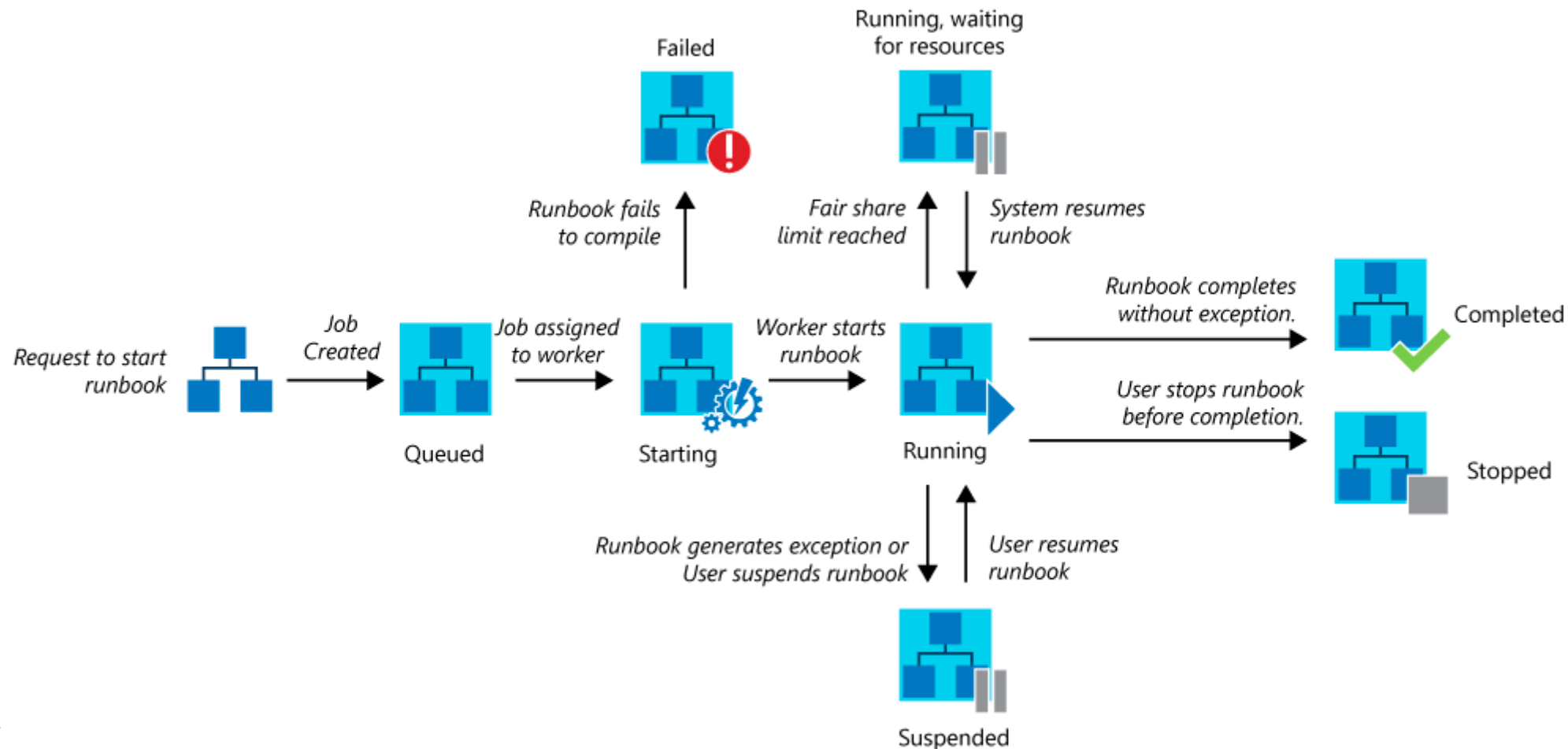
```
"resources": [  
  {  
    "type": "Microsoft.Storage/storageAccounts",  
    "apiVersion": "2019-04-01",  
    "name": "[variables('storageAccountName')]",  
    "location": "[parameters('location')]",  
    "sku": {  
      "name": "[parameters('storageSKU')]"  
    },  
    "kind": "StorageV2",  
    "properties": {  
      "supportsHttpsTrafficOnly": true  
    }  
  },  
]
```

# Create and Execute an Automation Runbook



# Runbooks in Azure Automation (1 of 2)

The lifecycle of a runbook job for PowerShell runbooks, PowerShell Workflow runbooks, and graphical runbooks





# Runbooks in Azure Automation (2 of 2)

Azure Automation Runbooks can run in:

- **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



# 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

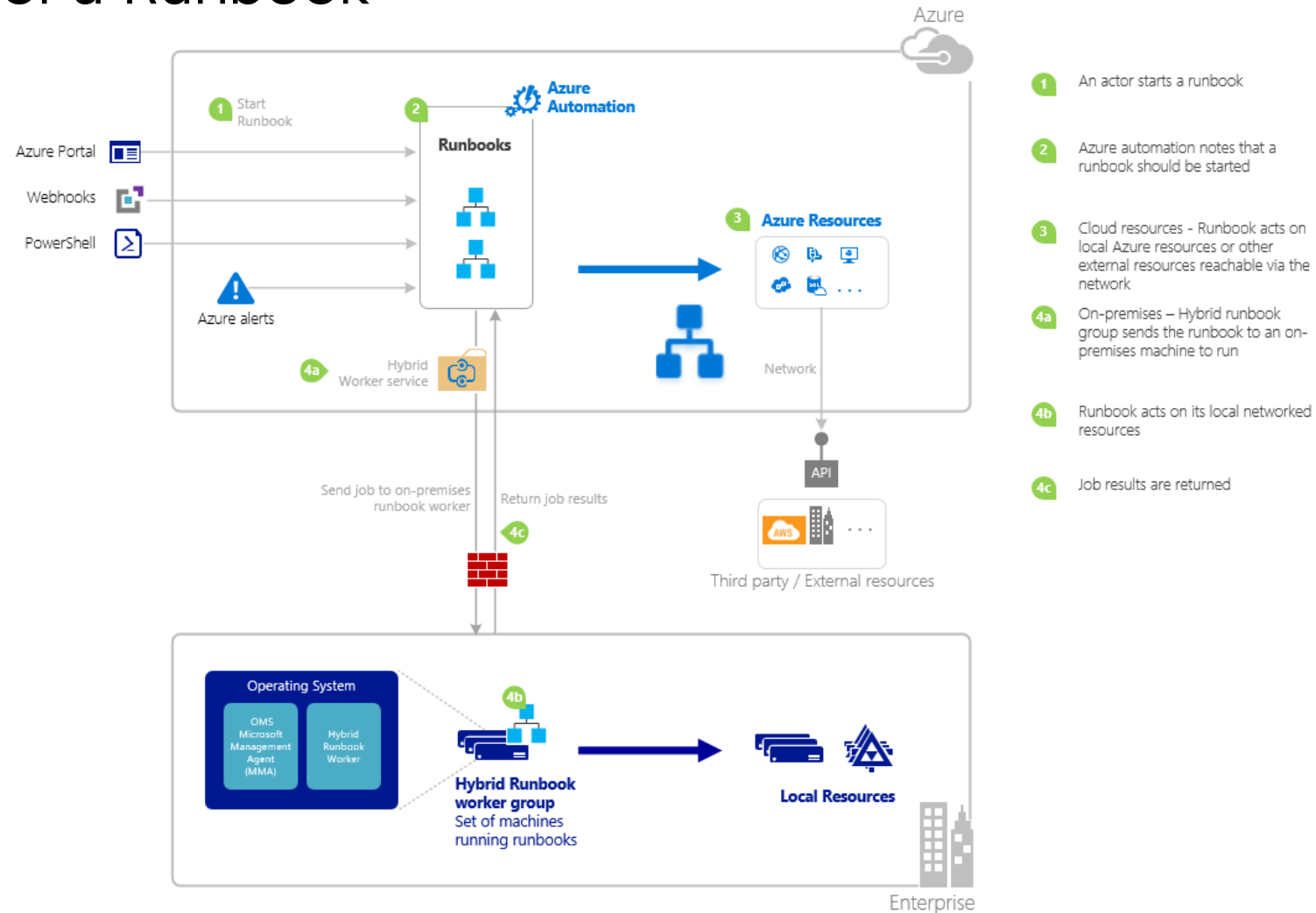
You also have the option of:

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



# Start a Runbook in Azure Automation (1 of 2)

## The life cycle of a Runbook



# Start a Runbook in Azure Automation (2 of 2)

You can start a runbook by using:

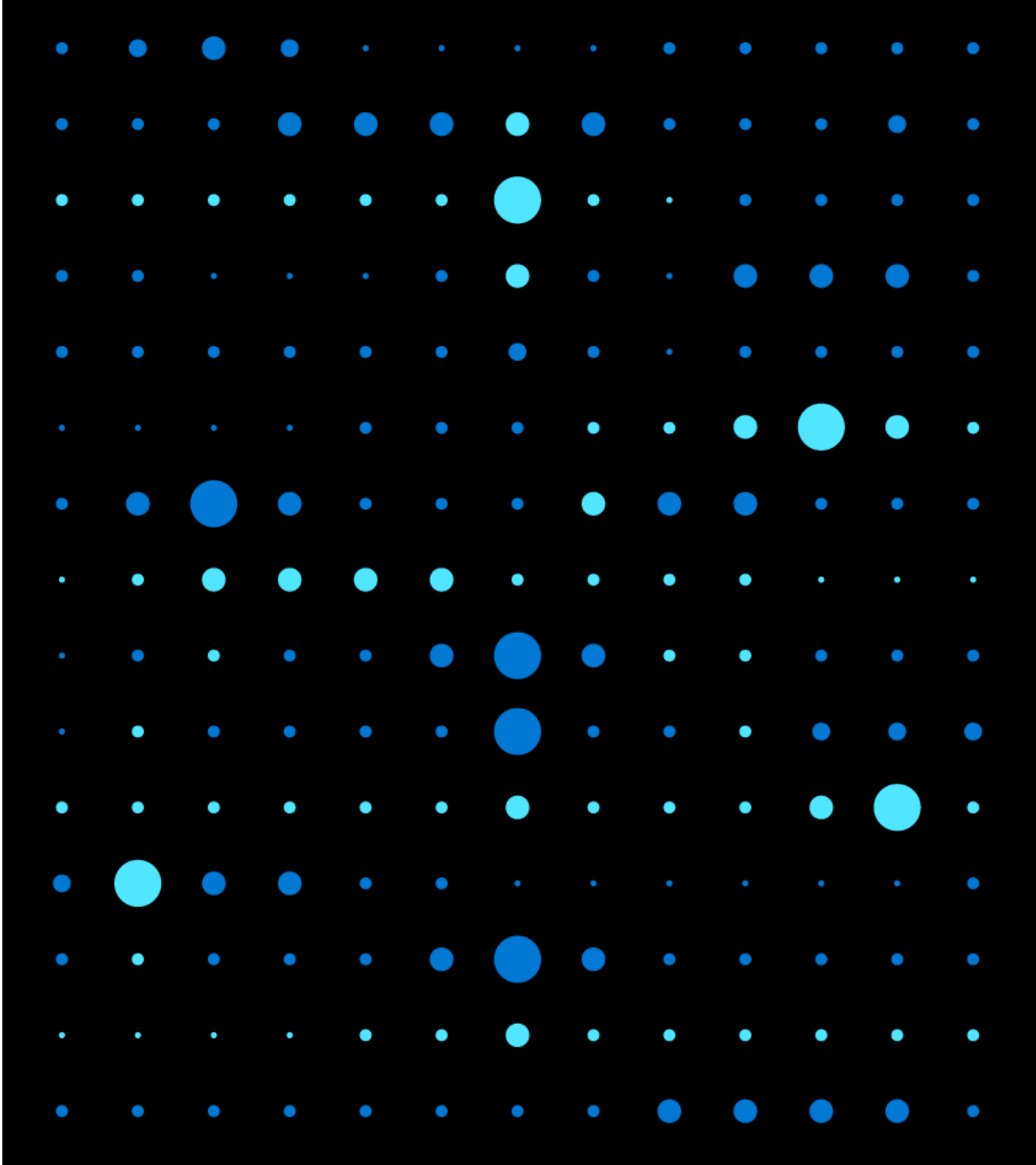
- the Azure portal
- PowerShell

`Start-AzAutomationRunbook`

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



# Demonstration: Create and Run a Workflow Runbook



# Module Review Questions



# Online Role-based training resources:

Microsoft Learn

<https://docs.microsoft.com/en-us/learn/>

Thank you.