



# AZ-800

## Administer Windows Server

## Hybrid Core Infrastructure



# Agenda AZ-800

---

- 1 Deploy and manage identity infrastructure – Windows Server
- 2 Deploy and manage identity infrastructure – Hybrid
  
- 3 Administering Windows Server Hybrid Core Infrastructure – Windows Server
- 4 Administering Windows Server Hybrid Core Infrastructure – Hybrid
  
- 5 Manage virtualization and containers – Windows Server
- 6 Manage virtualization and containers – Hybrid ←
  
- 7 Implement and manage networking infrastructure – Windows Server
- 8 Implement and manage networking Infrastructure – Hybrid
  
- 9 Configure storage and file services – Windows Server
- 10 Configure storage and file services – Hybrid

# Manage virtualization and containers in a hybrid environment *(Deploying and configuring Azure VMs)*

- [Plan and deploy Windows Server IaaS virtual machines](#)
- [Customize Windows Server IaaS virtual machine images](#)
- [Automate the configuration of Windows Server IaaS virtual machines](#)
- [Lab 06 – Deploying and configuring Windows Server on Azure VMs](#)

# Plan and deploy Windows Server IaaS virtual machines

# Learning Objectives – Plan and deploy Windows Server IaaS virtual machines

- Describe Azure compute
- Describe Azure storage
- Deploy Azure VMs
- Demonstration – Create a VM from the Azure portal
- Demonstration – Create a VM from Azure Cloud Shell
- Deploy Azure VMs by using templates
- Learning recap

Size Standard\_DS2\_v3

Status

Stopped (deallocated)

Stopped

HCL Azure AWS ...

Bicep Azure - -

json → ARM

# Describe Azure compute

- Azure compute is an on-demand computing service for running cloud-based applications
- Azure VMs provide a high degree of flexibility and control
- There are two tiers of Azure VMs: **Basic** and **Standard**.
- You can provision Gen 1 and Gen 2 Azure VMs. Gen 2 VMs will allow use UEFI and build VMs up to 12 TB.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes the Microsoft Azure logo, a search bar, and a 'Home > Virtual machines >' breadcrumb trail. The main title is 'Create a virtual machine'. On the left, there are several configuration fields with validation stars: 'Size \*' (dropdown menu open), 'Administrator account' (dropdown menu open), 'Authentication type \*' (dropdown menu open), 'Username \*' (text input field containing 'AzureUser'), 'SSH public key source' (dropdown menu open), and 'Key pair name \*' (text input field containing 'Name the SSH public key'). On the right, a dropdown menu titled 'Your recently used sizes' lists various VM sizes. The 'Standard\_B1s - 1 vcpu, 1 GiB memory (£6.42/month)' option is highlighted with a red box. Other listed options include 'Standard\_B1s - 1 vcpu, 1 GiB memory (£6.42/month)', 'Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (£63.11/month)', 'Standard\_D4s\_v3 - 4 vcpus, 16 GiB memory (£126.23/month)', and 'Standard\_E2s\_v3 - 2 vcpus, 16 GiB memory (£84.88/month)'. A green checkmark is visible next to the 'AzureUser' username entry.

# VM categories (1 of 2)

## VM series identifier

The leading letter and a number designates a collection of VM sizes – referred to as VM series or VM family – that share common configuration characteristics.

Category	Description
<b>General purpose</b>	This category offers a balanced CPU-to-memory ratio, making it most suitable for testing, development, and hosting small-to-medium databases or web servers. This category includes <b>Av2, Dv4, Dv5, DSv4, and DSv5 series</b> VM sizes.
<b>Confidential compute</b>	Azure confidential computing VMs are designed to protect the confidentiality and the integrity of your data and code while it's processed in the cloud. <b>DC series</b> .
<b>Compute optimized</b>	This category offers a high CPU-to-memory ratio, making it most suitable for compute-intensive workloads without extensive memory requirements. Typically, these are medium-size traffic web servers or application servers, network appliances, or servers handling batch processing. This category includes <b>F, Fs, and Fsv2 series</b> VM sizes.

# VM categories (2 of 2)

Category	Description
<b>Memory optimized</b>	This category offers a high memory-to-CPU ratio, making it most suitable for memory-intensive workloads without extensive compute requirements. Typically, these are workloads that keep the bulk of their operational content in memory, such as database or caching servers. This category includes <b>Ev5, Esv5, and Mv2 series</b> VM sizes.
<b>Storage optimized</b>	This category offers high-performance disk I/O, most suitable for big data processing with both SQL and NoSQL database management systems. This category consists of the <b>Lsv3</b> VM sizes.
<b>GPU</b>	This category offers graphics processing unit (GPU) support, with thousands of CPU cores. This typically is ideal for implementing workloads such as graphic rendering, video editing, crash simulations, or deep learning. This category includes <b>NC, NCv2, NCv3, NVv4, and ND series</b> VM sizes.
<b>High performance compute</b>	This category offers VMs with the fastest CPUs and optional high-throughput Remote Direct Memory Access (RDMA) network interfaces. This category includes <b>H series</b> VM sizes.

# Describe Virtual Machine storage (1/2)

All Azure VM sizes support **Standard** Storage

A large number of Azure VM sizes also support **Premium** Storage

**Premium** Storage vs. **Standard** Storage:

- Premium Storage offers superior performance, equivalent to what SSD technology provides.
- Standard Storage provides performance similar to commodity magnetic disks, referred to typically as hard disk drives (HDD).

Type of disk	Description
Operating system disks	Each Azure VM has one operating system disk that contains the operating system drive.
Data disks	The maximum number of data disks you can attach to the Azure VM is dependent on the Azure VM size.
Temporary disks	You can have one temporary disk per Azure VM.



Network Security Group NSG

# Deploy Azure virtual machines

Method	Explanation
The Azure portal	This method is most straightforward because it provides a wizard-driven interface with default settings that represent the most common configuration options.
Azure PowerShell <i>New-AzVM</i>	This method offers automation and full flexibility, including support for multiple NICs and custom images.
Azure CLI	This method is equivalent to using Azure PowerShell in terms of flexibility and automation capabilities. The difference is merely the preference of the person performing the deployment.
Azure Resource Manager templates	This method provides full flexibility and <u>the best</u> performance for large Azure VM deployments. <i>Infra as Code</i>
Windows Admin Center	This method enables you to create an Azure VM with custom settings by using Storage Migration Service or by using Storage Replica.

Azure CLI deployment example:

```
az vm create --resource-group ContosoResourceGroup --name ContosoVM4  
--image win2019datacenter --admin-username azureuser
```

# Demonstration – Create a virtual machine using the portal

Create a VM

Connect to a VM

Install a web server

Access the Microsoft  
Internet Information  
Services (IIS)  
welcome page

# Demonstration – Create a virtual machine using Azure CLI

From Cloud  
Shell create a  
resource group

Create a Windows  
Server VM

Enable a port  
for web traffic

Install a web server

# Deploy Azure virtual machines using templates (1/3)

## Benefits of templates:

- Deployment templates are for provision multiple solutions with the same general design.
- Templates are idempotent
- Templates support VM extensions such as DSC, Chef, or Puppet.
- Allowed define dependencies between resources to control the resource-provisioning sequence

## Considerations for using templates:

- When creating and working with resource templates, you should consider:
- Which resources you are going to deploy
  - Where your resources will be located
  - Which version of the resource provider application programming interface (API) you will use
  - Whether there are dependencies between resources
  - When you will specify values of resource properties.

# ARM template schema (2/3)

- **\$schema** – The predefined URL identifying the location of the JSON schema file
- **Contentversion** – Custom value to track changes
- **Parameters** – Parameters to provide during deployment or from a parameter file
- **Variables** – Values referenced during template deployment
- **Functions** – Custom functions calling during template deployment
- **Resources** – Resources to be provisioned or modified
- **Outputs** – Values returned by the deployment

```
{  
  "$schema":  
    "http://schema.management.azure.com/...#",  
  "contentVersion": "",  
  "parameters": { },  
  "variables": { },  
  "functions": [ ],  
  "resources": [ ],  
  "outputs": {}  
}
```

# Deploy Azure virtual machines using Bicep

(3/3)

## What is Bicep?

- New language that uses declarative syntax to deploy Azure resources
- Easier and more concise syntax than JSON
- Converts Bicep files into ARM template JSON
- Infrastructure-as-code solution in Azure

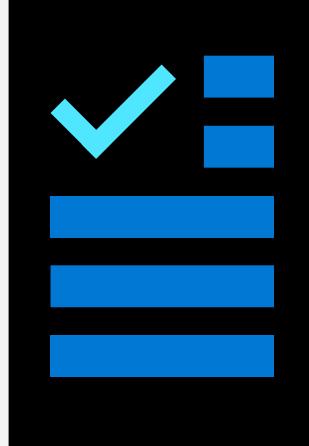
```
targetScope = '<scope>'  
@<decorator>(<argument>) param <parameter-name>  
<parameter-data-type> = <default-value>  
  
var <variable-name> = <variable-value> resource  
  
<resource-symbolic-name>'<resource-type>@<api-version>'  
= { <resource-properties> }  
  
module <module-symbolic-name> '<path-to-file>' = { name:  
'<linked-deployment-name>' params: { <parameter-names-  
and-values> } } output <output-name>  
  
<output-data-type> = <output-value>
```

# Methods for deploying VMs from an ARM template

Method	Explanation
Azure PowerShell	You can initiate deployment by running the <b>New-AzResourceGroupDeployment</b> cmdlet.
Azure CLI	You can run the <b>az group deployment create</b> Azure CLI command with either the --template-file or --template-uri * parameter and the --resource_group parameters to initiate deployment.
The Azure portal	The Custom deployment pane in the Azure portal provides a convenient way to deploy Azure Resource Manager template-based resources.
Azure QuickStart Templates GitHub repository	Every QuickStart template published on GitHub has a corresponding <b>Deploy to Azure</b> link. <a href="https://github.com/Azure/azure-quickstart-templates">Azure/azure-quickstart-templates: Azure Quickstart Templates (github.com)</a>
Microsoft Visual Studio or Visual Studio Code	You can author and deploy templates by using <b>Visual Studio</b> and <b>Visual Studio Code</b> . <a href="https://docs.microsoft.com/en-us/learn/modules/plan-deploy-windows-server-iaas-virtual-machines/7-deploy-azure-virtual-machines-using-templates">https://docs.microsoft.com/en-us/learn/modules/plan-deploy-windows-server-iaas-virtual-machines/7-deploy-azure-virtual-machines-using-templates</a>

# Learning recap – Plan and deploy Windows Server IaaS virtual machines

Module  
assessment



**Microsoft Learn Modules ([docs.microsoft.com/Learn](https://docs.microsoft.com/Learn))**

Plan and deploy Windows Server IaaS Virtual Machines

# Customize Windows Server IaaS virtual machine images

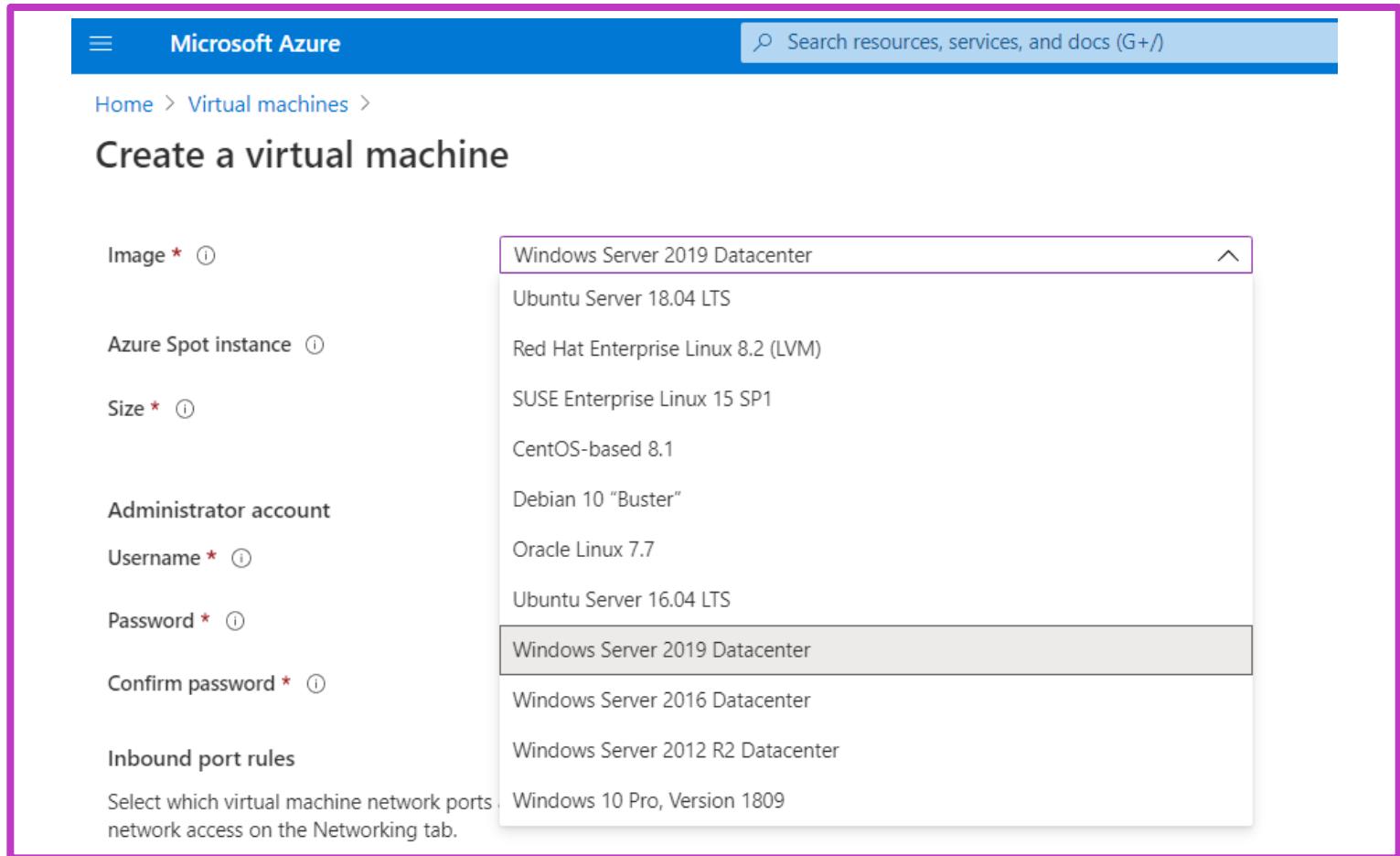
# Learning Objectives – Customize Windows Server IaaS virtual machine images

- Create a generalized image
- Create a new VM from a generalized image
- Create a managed image of a generalized VM in Azure
- Demonstration – Create a VM from a managed image
- Describe Azure Image Builder
- Use Azure Image Builder to create a Windows image
- Learning recap

# Available VM images

## What are VM images?

- A VM image contains a generalized operating system and optionally, other preconfigured software.
- A VM image is a template from which you can create the VHDs to run a VM. The VHDs for a typical VM image contain a preconfigured version of an operating system.
- By Browsing in Azure Marketplace, you can find images and build your own on top.
- In Marketplace you can find Linux, Windows Server, MS SQL Server, MySQL, MongoDB, Cassandra, or other already configured images.



# Create a generalized image

## What is a generalized image?

Following items must be reset to their default settings before you capture an image. When you reset these items in a VM, you generalize the VM.

- The host name of your VM.
- The administrator username and credentials.
- Log files.
- Security identifiers for various operating system services.

## Generalize a VM

Use the **Sysprep.exe** tool to generalize a Windows VM. Follow these steps:

- Sign in to the Windows VM.
- Open an elevated **Command Prompt**.
- Navigate to the **C:\windows\system32\sysprep** folder.
- Run **sysprep.exe**.
- In the **System Preparation Tool** dialog box, select the settings, and then select OK.

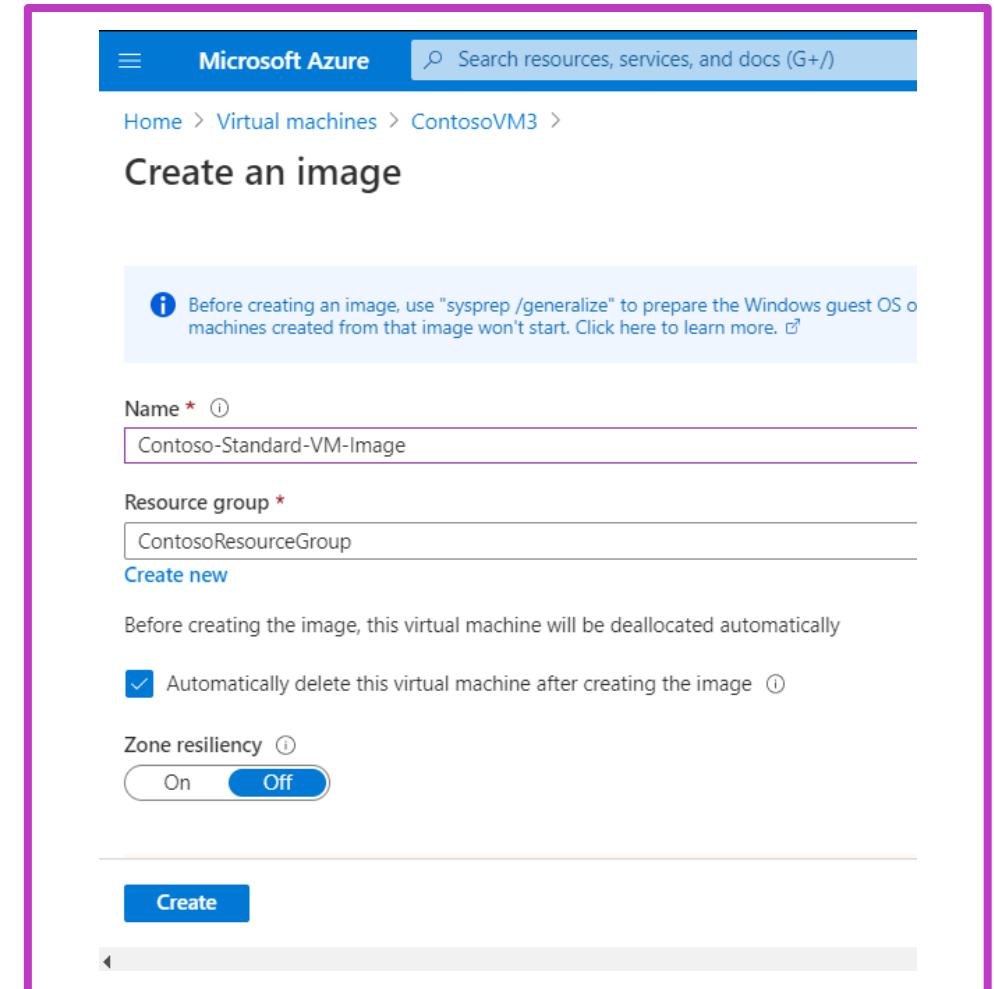
# Create a managed image from a generalized VM

To create an image in the Azure portal, use the following procedure:

1. Select the appropriate VM.
2. On the toolbar, select **Capture**.
3. On the **Create an image** blade, enter a **Name** and then select a **Resource group**.
4. If you want to remove the source VM, select the **Automatically delete this virtual machine after creating the image** check box.
5. Select **Create**.

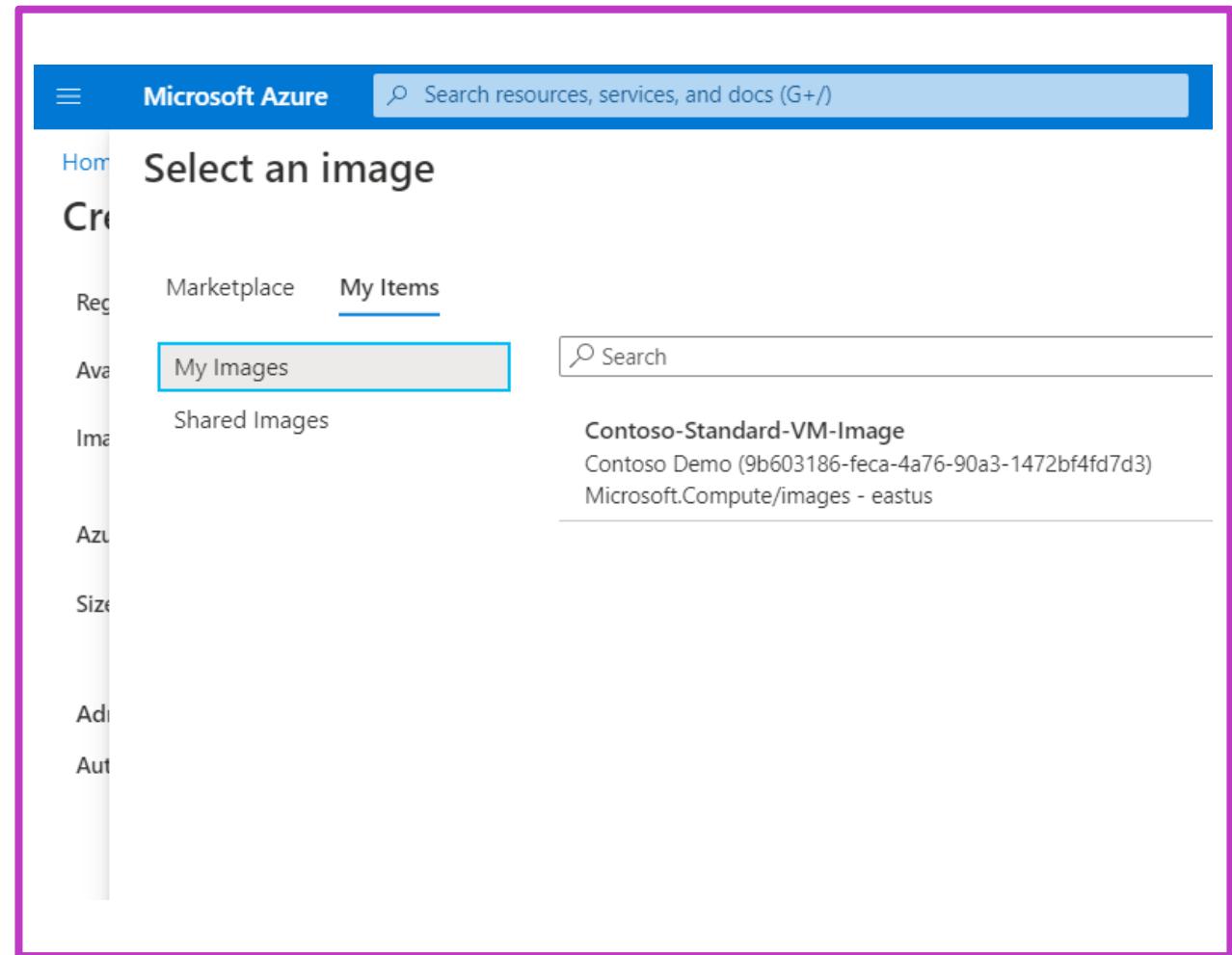
Azure CLI command:

```
az image create \
  --name <image name> \
  --resource-group <resource group> \
  --source <generalized virtual machine>
```



# Create a new VM from a managed image

1. In Virtual machines select **+ Add**.
2. Provide Resource group, VM name, and Region.
3. Then select **Browse all public and private images**.
4. On the **Select an image** blade, select the **My Items** tab.
5. Your newly created image should be present in the list. Select it.
6. Configure Size, Username, Password, Inbound port rules, and License type.
7. When you're done, select **Review + create**.
8. When prompted, select **Create**.



# Demonstration – Create a managed image of generalized VM

Generalize the Windows VM using **Sysprep.exe**.

Create a managed image in the Azure portal

Test that the new image is available to use.

Delete the VM to not use it again.

# Demonstration – Create a VM from managed image

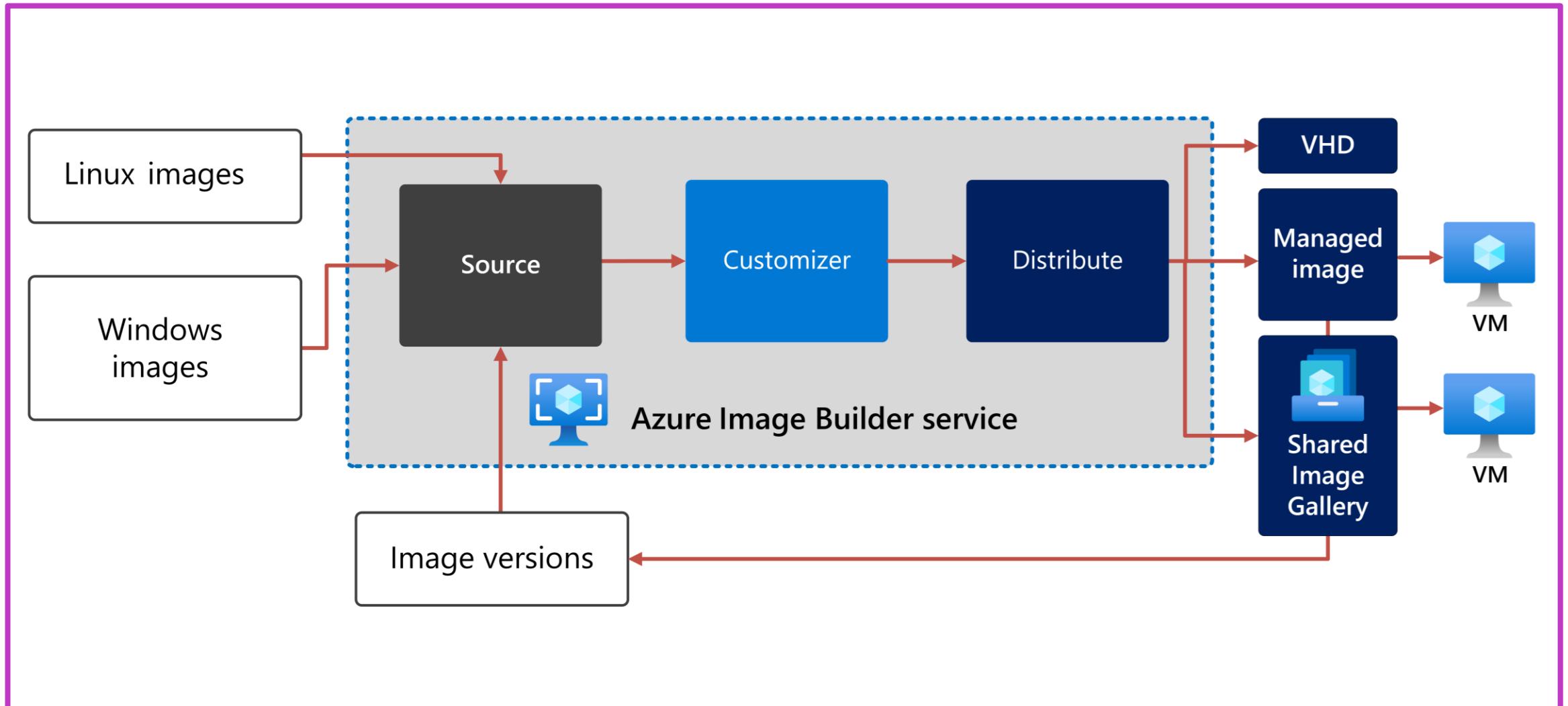
From Azure portal create a resource group

Start creating the Azure VM

Select managed image prepared before

Complete VM creation and check connection

# How Azure Image Builder works?

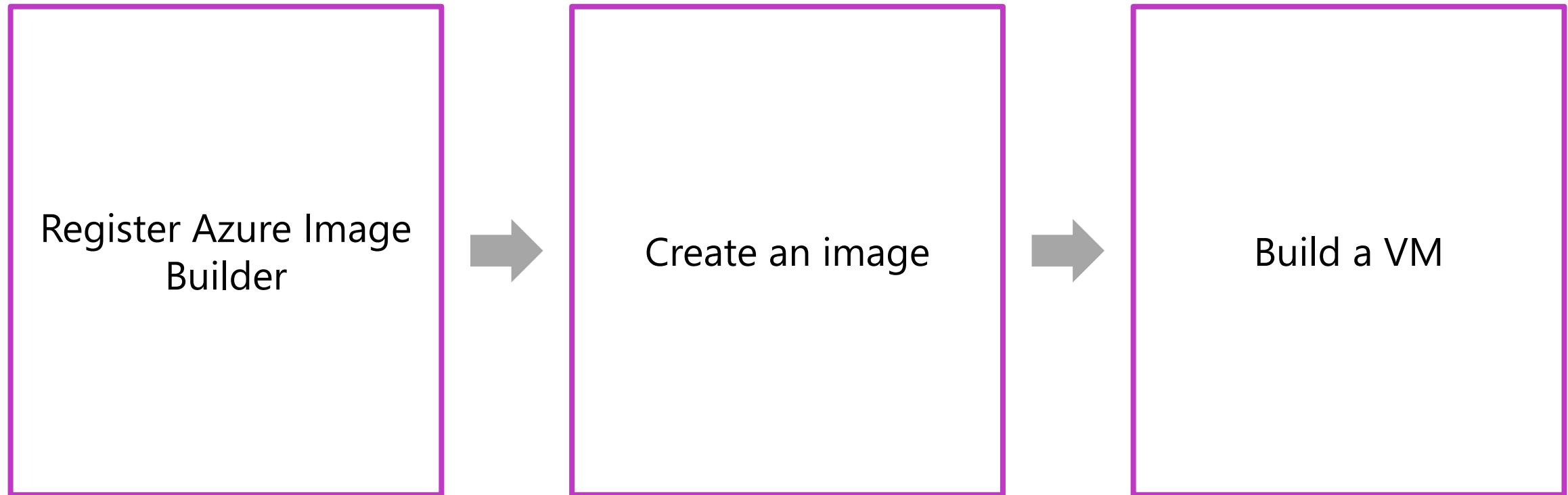


# What is Azure Image Builder?

Azure Image Builder enables you to add your own customizations to existing images

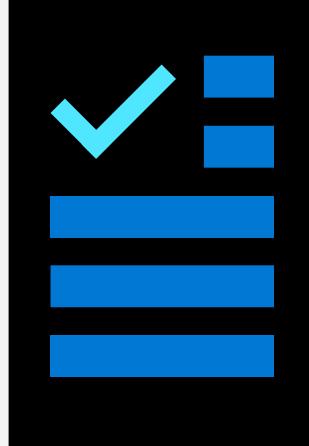
Task	Description
<b>Create golden baseline images</b>	Includes your minimum security and organizational configurations, and enables departments to further customize them for their needs.
<b>Patch existing images</b>	Enables you to continually patch existing custom images.
<b>Connect to existing virtual networks</b>	Connects with existing VNets, enabling you to connect to existing configuration servers, file shares, or any other routable servers/services.
<b>Integrate with Azure Shared Image Gallery</b>	Enables you to manage versions, and distribute and scale images globally. Also provides you with an image management system.
<b>Integrate image build pipelines from your own organization</b>	Enables integration with your existing image build procedures (pipelines).
<b>Migrate an existing image customization pipeline to Azure</b>	Enables you to use your existing scripts, commands, and processes to customize images.
<b>Create images in VHD format</b>	Enables you to support Azure Stack.

# Create a Windows virtual machine using Azure Image Builder template ( Azure CLI or Azure PowerShell )



# Learning recap – Customize Windows Server IaaS virtual machine images

Module  
assessment



**Microsoft Learn Modules ([docs.microsoft.com/Learn](https://docs.microsoft.com/Learn))**

Customize Windows Server IaaS Virtual Machine images

# Automate the configuration of Windows Server IaaS virtual machines

# Learning Objectives – Automate the configuration of Windows Server IaaS virtual machines

- Describe Azure automation
- Implement Azure automation with DSC
- Remediate noncompliant servers
- Describe custom script extension
- Configure a VM by using DSC extensions
- Demonstration – Configure a VM by using DSC
- Learning recap

Automation Account  
= Runtime PS7  
+ DSC Pull "Server"

# What is Azure Automation?

- Azure Automation is a cloud service that provides the ability to automate administrative tasks.
- It integrates with Log Analytics, enabling administrators to manage OS updates, simplify server inventory, and track changes to server configuration.
- Azure Automation can execute long-running tasks on Azure and on-premises servers.
- The core components are credentials, modules, schedules, connections and runbooks.
- Azure Automation offers several ready-to-use runbooks for Update Management, Inventory and Change Tracking.

The screenshot shows the 'Add Automation Account' page in the Microsoft Azure portal. The top navigation bar includes the Microsoft Azure logo, a search bar, and a 'Home > Automation Accounts >' breadcrumb trail. The main title is 'Add Automation Account'. The form contains the following fields:

- Name \***: ContosoAutomation
- Subscription \***: Contoso Demo
- Resource group \***: DefaultResourceGroup-EUS (with a 'Create new' link)
- Location \***: East US
- Create Azure Run As account \***: Yes (selected)

A large blue 'Create' button is at the bottom right of the form area.

# Describe Azure Automation

## Overview of Azure Automation runbooks

- Assets are the building blocks of Automation runbooks.
- A runbook represents a collection of tasks that perform a custom process.
- Runbooks implemented on PowerShell scripts, Python 2 scripts, or Windows PowerShell workflows.
- There are two types of runbooks – Graphical and Textual

## Managing Azure Automation runbooks

- Three authoring status – **Newly created**, **Published** and **In edit**
- Testing a runbook
  1. Select the appropriate Automation Account, and then select Runbooks.
  2. Select any runbook that you want to test.
  3. In the Runbook blade, select Edit.
  4. In the selected runbook, select Test pane.

# Implement Azure Automation with DSC

## What is LCM?

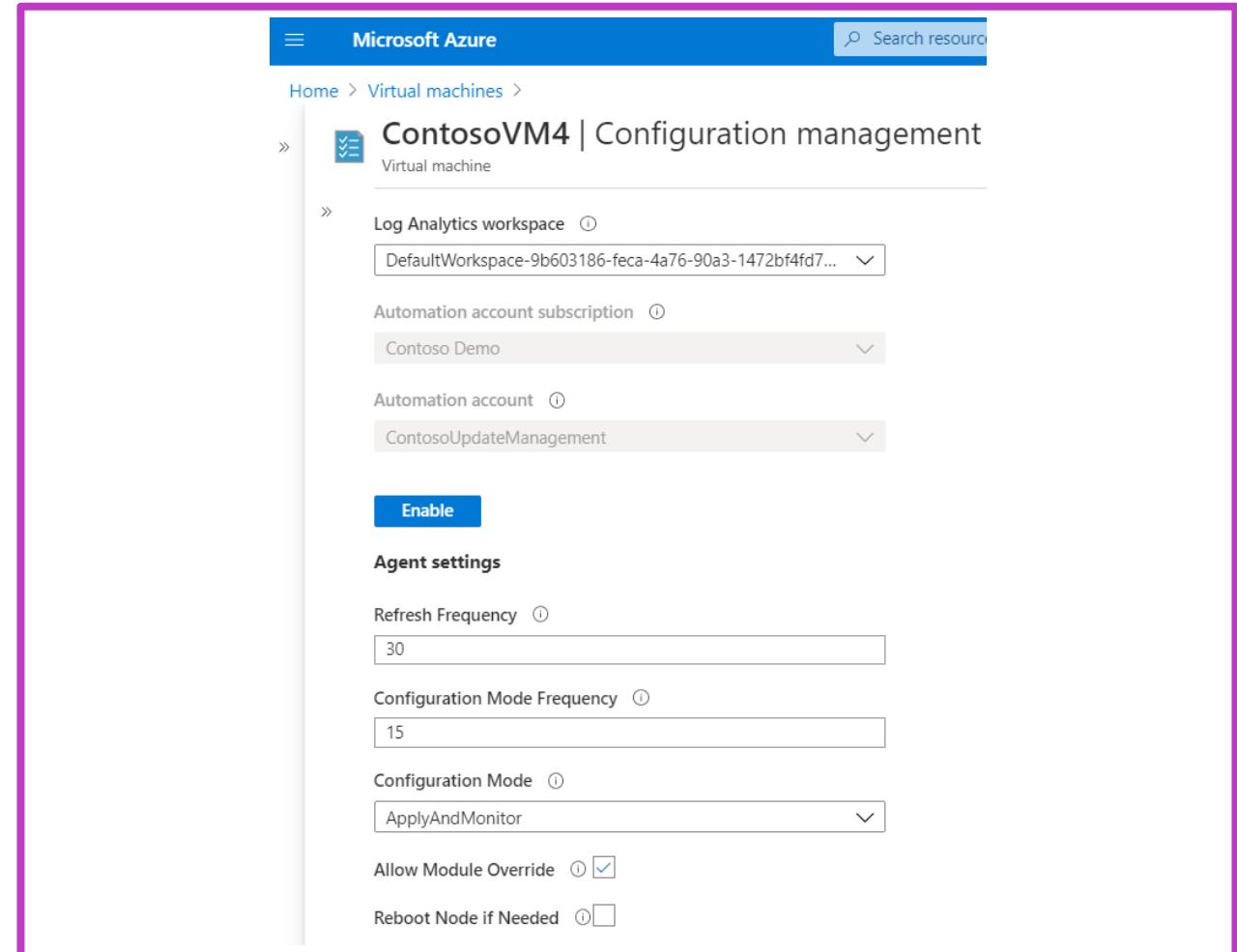
- Windows-based DSC relies on the **Local Configuration Manager** (LCM) component. This component serves as the execution engine of the Windows PowerShell DSC scripts.

You can deploy DSC configuration in either push mode or pull mode

## Requirements for DSC with Windows computers

- Operating system
- Windows Remote Management
- Network

To control a VM with DSC, you must first onboard the VM, or node, with Azure Automation



# How to use DSC

Requirements for DSC with Windows computers:

- Windows 8.1 or later, or Windows Server 2012 R2 or later.
- Windows Remote Management
- Network. Access to Internet, TCP 443 Global URL: \*.azure-automation.net

With DSC, you can define the desired state of:

- An operating system
- An application

You can use a script to define your DSC or you can create a DSC by using the Azure portal

The screenshot shows the Microsoft Azure portal interface for an Automation Account named "ContosoUpdateManagement". The main area displays a summary of configuration status: 1 node, 0 Failed, 0 Pending, 0 Not compliant, 0 In progress, 0 Unresponsive, and 1 Compliant. Below this summary, a table lists nodes and their status. A sidebar on the left provides navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration Management (Inventory, Change tracking, State configuration (DSC)), and Update management.

Node	Status	Node configuration
ContosoVM4	Compliant	

# Available PowerShell commands

## What is the Run Command?

- The **Run Command** feature enables you to run scripts inside VMs
- Use to manually correct configuration drift
- Uses the VM agent to run PowerShell scripts within an Azure Windows VM
- The commands available for a general machine or application management
- Running a command requires the Microsoft.Compute/virtualMachines/runCommand/action permission.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a blue header bar with the Microsoft Azure logo, a search bar containing 'Search resources, services, and docs (G+)', and several icons for notifications, help, and settings. Below the header, the URL 'Home > Virtual machines > ContosoVM4 | Run command' is visible. To the left, there's a sidebar with navigation links: 'Virtual machine' (selected), 'Inventory', 'Change tracking', 'Configuration management (...)', 'Policies', 'Run command' (selected), 'Monitoring' (with sub-links: 'Insights', 'Alerts', 'Metrics', 'Diagnostic settings', 'Logs', 'Connection monitor'), and 'Logs'. The main content area is titled 'ContosoVM4 | Run command' and contains a description: 'Run Command uses the VM agent to let you run a script inside this virtual machine. This can be helpful for troubleshooting and recovery, and for general machine and application maintenance. Select a command to see details.' Below this, a table lists various PowerShell commands with their descriptions:

Name	Description
RunPowerShellScript	Executes a PowerShell script
DisableNLA	Disable Network Level Authentication
DisableWindowsUpdate	Disable Windows Update Automatic Updates
EnableAdminAccount	Enable administrator account
EnableEMS	Enable EMS
EnableRemotePS	Enable remote PowerShell
EnableWindowsUpdate	Enable Windows Update Automatic Updates
IPConfig	List IP configuration
RDPSettings	Verify RDP Listener Settings
ResetRDPCert	Restore RDP Authentication mode to defaults
SetRDPPort	Set Remote Desktop port

# Custom Script Extensions

Ansible

## The usage of Custom Script Extension

- Scripts used for post deployment configuration and software installation.
- Extension helps to download and run scripts in Azure VMs.
- The scripts can be run using Resource Manager templates, Azure Command-Line Interface (Azure CLI), REST API, PowerShell, or the Azure portal.
- Requirements to use custom script extensions Windows 10 or later, Windows Server 2008 R2 or later

 **Custom Script Extension**

Publisher: Microsoft Corp.

**Overview**

Custom Script Extension is a tool that can be used to automatically launch and execute VM customization tasks post configuration. When this Extension is added to a Virtual Machine, it can download Powershell scripts and files from Azure storage and launch a Powershell script on the VM which in turn can download additional software components. Custom Script Extension tasks can also be automated using the Azure Powershell cmdlets.

**Configure Custom Script Extension Extension** ...

[Create](#) [Review + create](#)

Script file (Required) \*  [Browse](#)

Arguments (Optional)

# Demonstration – Configure a virtual machine by using DSC

In Cloud Shell create a bash script to build the environment

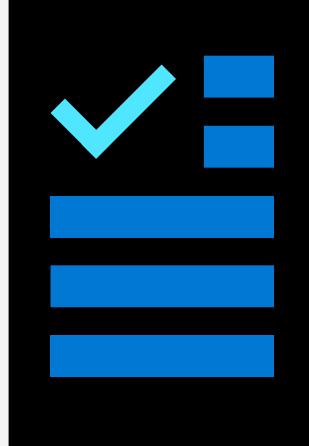
Deploy the VM and RG using Azure CLI commands.

Create and configure custom script extension from Azure CLI commands

Check results of execution custom script on the VM

# Learning recap – Automate the configuration of Windows Server IaaS virtual machines

## Module assessment



### Microsoft Learn Modules ([docs.microsoft.com/Learn](https://docs.microsoft.com/Learn))

Automate the configuration of Windows Server IaaS Virtual Machines

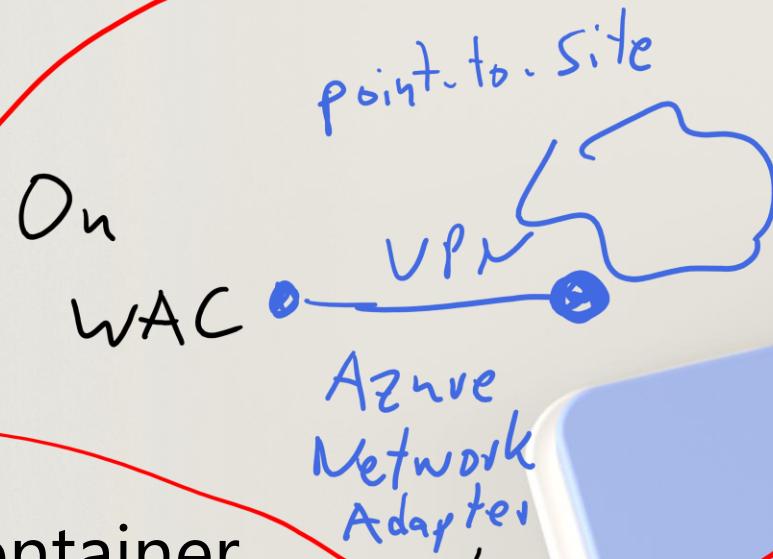
# Lab 06 – Deploying and configuring Windows Server on Azure VMs

Lab 3 WAC on Prem

Lab 4 WAC Hybrid *40min*

Lab 5 Hyper-V und Windows Container

Lab 6 Azure VMs



# Lab 06: Deploying and configuring Windows Server on Azure VMs



## Lab scenario

In order to address concerns regarding your outdated operational model, limited use of automation, and Information Security team concerns, you need to develop and implement an automated deployment process for Azure VMs running Windows Server. You will deploy Azure VMs running Windows Server by using ARM templates and use JIT functionality to restrict VMs to public IPs associated with the London headquarters.

## Objectives

- Author ARM templates for an Azure VM deployment.
- Modify ARM templates to include VM extension-based configuration.
- Deploy Azure VMs running Windows Server by using ARM templates.
- Configure administrative access to Azure VMs running Windows Server.
- Configure Windows Server security in Azure VMs.
- Deprovision the Azure environment.

# End of presentation