✓ 100 XP ▶

# Describe the options available to create and manage an Azure Virtual Machine

15 minutes

The Azure portal is the easiest way to create resources such as VMs when you are getting started. However, it's not necessarily the most efficient or quickest way to work with Azure, particularly if you need to create several resources together. In our case, we will eventually be creating dozens of VMs to handle different tasks. Creating them manually in the Azure portal wouldn't be a fun task!

Let's look at some other ways to create and administer resources in Azure:

- Azure Resource Manager
- Azure PowerShell
- Azure CLI
- Azure REST API
- Azure Client SDK
- Azure VM Extensions
- Azure Automation Services

## Azure Resource Manager

Let's assume you want to create a copy of a VM with the same settings. You could create a VM image, upload it to Azure, and reference it as the basis for your new VM. This process is inefficient and time-consuming. Azure provides you with the option to create a template from which to create an exact copy of a VM.

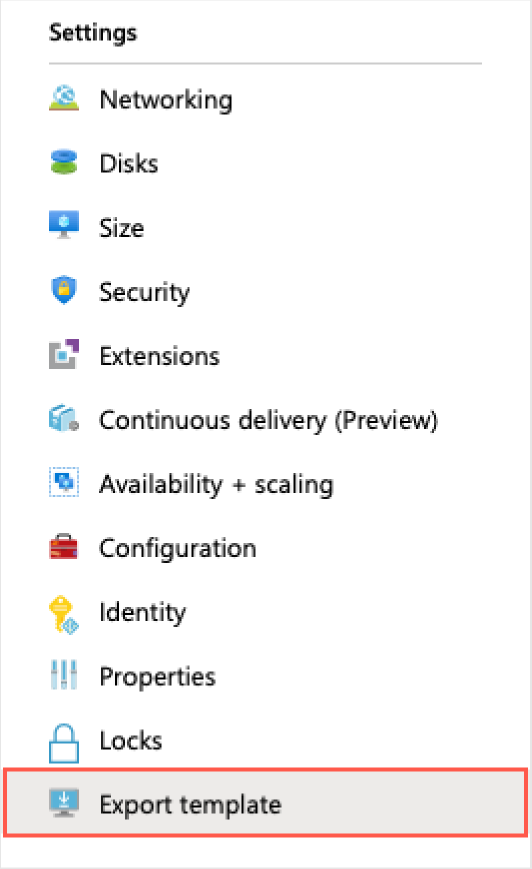
Typically, your Azure infrastructure will contain many resources, many of them related to one another in some way. For example, the VM we created has the virtual machine itself, storage and network interface. **Azure Resource Manager** makes working with these related resources more efficient. It organizes resources into named **resource groups** that let you deploy, update, or delete all of the resources together. When we created the Ubuntu VM site, we identified the resource group as part of the VM creation, and Resource Manager placed the associated resources into the same group.

Resource Manager also allows you to create *templates*, which can be used to create and deploy specific configurations.

## What are Resource Manager templates?

**Resource Manager templates** are JSON files that define the resources you need to deploy for your solution.

You can create resource templates from the **Settings** section for a specific VM by selecting the *Export template* option.



Settings
- Networking
- Disks
- Size
- Security
- Extensions
- Continuous delivery (Preview)
- Availability + scaling
- Configuration
- Identity
- Properties
- Locks
- Export template

You have the option to save the resource template for later use or immediately deploy a new VM based on this template. For example, you might create a VM from a template in a test environment and find it doesn't quite work to replace your on-premises machine. You can delete the resource group, which deletes all of the resources, tweak the template, and try again. If you only want to make changes to the existing deployed resources, you can change the template used to create it and deploy it again. Resource Manager will change the resources to match the new template.

Once you have it working the way you want it, you can take that template and easily re-create multiple versions of your infrastructure, such as staging and production. You can parameterize fields such as the VM name, network name, storage account name, etc., and load the template repeatedly, using different parameters to customize each environment.

You can use automation scripting tools such as the Azure CLI, Azure PowerShell, or even the Azure REST APIs with your favorite programming language to process resource templates, making this a powerful tool for quickly spinning up your infrastructure.

## Azure PowerShell

Creating administration scripts is a powerful way to optimize your workflow. You can automate everyday, repetitive tasks, and once a script has been verified, it will run consistently, likely reducing errors. **Azure PowerShell** is ideal for one-off interactive tasks and/or the automation of repeated tasks.

> **Note**
>
> PowerShell is a cross-platform shell that provides services like the shell window and command parsing. Azure PowerShell is an optional add-on package that adds the Azure-specific commands (referred to as **cmdlets**). You can learn more about installing and using Azure PowerShell in a separate training module.

For example, you can use the `New-AzVM` cmdlet to create a new Azure virtual machine.

PowerShell Copy

```powershell
New-AzVm `
    -ResourceGroupName "TestResourceGroup" `
    -Name "test-wp1-eus-vm" `
    -Location "East US" `
    -VirtualNetworkName "test-wp1-eus-network" `
    -SubnetName "default" `
    -SecurityGroupName "test-wp1-eus-nsg" `
    -PublicIpAddressName "test-wp1-eus-pubip" `
    -OpenPorts 80,3389
```

As shown here, you supply various parameters to handle the large number of VM configuration settings available. Most of the parameters have reasonable values; you only need to specify the required parameters. Learn more about creating and managing VMs with Azure PowerShell in the **Automate Azure tasks using scripts with PowerShell** module.

## Azure CLI

Another option for scripting and command-line Azure interaction is the **Azure CLI**.

The Azure CLI is Microsoft's cross-platform command-line tool for managing Azure resources such as virtual machines and disks from the command line. It's available for macOS, Linux, and

Windows, or in the browser using the Cloud Shell. Like Azure PowerShell, the Azure CLI is a powerful way to streamline your administrative workflow. Unlike Azure PowerShell, the Azure CLI does not need PowerShell to function.

For example, you can create an Azure VM with the `az vm create` command.

Azure CLI Copy

```
az vm create \
    --resource-group TestResourceGroup \
    --name test-wp1-eus-vm \
    --image win2016datacenter \
    --admin-username jonc \
    --admin-password aReallyGoodPasswordHere
```

The Azure CLI can be used with other scripting languages, for example, Ruby and Python. Both languages are commonly used on non-Windows-based machines where the developer might not be familiar with PowerShell.

Learn more about creating and managing VMs in the **Manage virtual machines with the Azure CLI tool** module.

# Programmatic (APIs)

Generally speaking, both Azure PowerShell and Azure CLI are good options if you have simple scripts to run and want to stick to command-line tools. When it comes to more complex scenarios, where the creation and management of VMs form part of a larger application with complex logic, another approach is needed.

You can interact with every type of resource in Azure programmatically.

## Azure REST API

The Azure REST API provides developers with operations categorized by resource as well as the ability to create and manage VMs. Operations are exposed as URIs with corresponding HTTP methods (`GET`, `PUT`, `POST`, `DELETE`, and `PATCH`) and a corresponding response.

The Azure Compute APIs give you programmatic access to virtual machines and their supporting resources. With this API, you have operations to:

- Create and manage availability sets
- Add and manage virtual machine extensions
- Create and manage managed disks, snapshots, and images
- Access the platform images available in Azure

- Retrieve usage information of your resources
- Create and manage virtual machines
- Create and manage virtual machine scale sets

## Azure Client SDK

Even though the REST API is platform and language agnostic, most often developers will look toward a higher level of abstraction. The Azure Client SDK encapsulates the Azure REST API, making it much easier for developers to interact with Azure.

The Azure Client SDKs are available for a variety of languages and frameworks, including .NET-based languages such as C#, Java, Node.js, PHP, Python, Ruby, and Go.

Here's an example snippet of C# code to create an Azure VM using the `Microsoft.Azure.Management.Fluent` NuGet package:

C# Copy

```csharp
var azure = Azure
    .Configure()
    .WithLogLevel(HttpLoggingDelegatingHandler.Level.Basic)
    .Authenticate(credentials)
    .WithDefaultSubscription();
// ...
var vmName = "test-wp1-eus-vm";

azure.VirtualMachines.Define(vmName)
    .WithRegion(Region.USEast)
    .WithExistingResourceGroup("TestResourceGroup")
    .WithExistingPrimaryNetworkInterface(networkInterface)
    .WithLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer", "2012-R2-Datacenter")
    .WithAdminUsername("jonc")
    .WithAdminPassword("aReallyGoodPasswordHere")
    .WithComputerName(vmName)
    .WithSize(VirtualMachineSizeTypes.StandardDS1)
    .Create();
```

Here's the same snippet in Java using the **Azure Java SDK**:

Java Copy

```java
String vmName = "test-wp1-eus-vm";
// ...
VirtualMachine virtualMachine = azure.virtualMachines()
    .define(vmName)
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("TestResourceGroup")
    .withExistingPrimaryNetworkInterface(networkInterface)
    .withLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer", "2012-R2-Datacenter")
    .withAdminUsername("jonc")
    .withAdminPassword("aReallyGoodPasswordHere")
    .withComputerName(vmName)
    .withSize("Standard_DS1")
    .create();
```

## Azure VM Extensions

Let's assume you want to configure and install additional software on your virtual machine after the initial deployment. You want this task to use a specific configuration, monitored and executed automatically.

**Azure VM extensions** are small applications that allow you to configure and automate tasks on Azure VMs after initial deployment. **Azure VM extensions** can be run with the Azure CLI, PowerShell, Azure Resource Manager templates, and the Azure portal.

You bundle extensions with a new VM deployment or run them against an existing system.

## Azure Automation Services

Saving time, reducing errors, and increasing efficiency are some of the most significant operational management challenges faced when managing remote infrastructure. If you have a lot of infrastructure services, you might want to consider using higher-level services in Azure to help you operate from a higher level.

**Azure Automation** allows you to integrate services that allow you to automate frequent, time-consuming, and error-prone management tasks with ease. These services include **process automation**, **configuration management**, and **update management**.

- **Process Automation**. Let's assume you have a VM that is monitored for a specific error event. You want to take action and fix the problem as soon as it's reported. Process automation allows you to set up watcher tasks that can respond to events that may occur in your datacenter.

- **Configuration Management**. Perhaps you want to track software updates that become available for the operating system that runs on your VM. There are specific updates you may want to include or exclude. Configuration management allows you to track these updates and take action as required. You use **Microsoft Endpoint Configuration Manager** to manage your company's PC, servers, and mobile devices. You can extend this support to your Azure VMs with Configuration Manager.

- **Update Management**. This is used to manage updates and patches for your VMs. With this service, you're able to assess the status of available updates, schedule installation, and review deployment results to verify updates applied successfully. Update management incorporates services that provide process and configuration management. You enable update management for a VM directly from your **Azure Automation** account. You can also allow update management for a single virtual machine from the virtual machine pane in the portal.

As you can see, Azure provides a variety of tools to create and administer resources so that you can integrate management operations into a process *that works for you*. Let's examine some of the other Azure services to make sure your infrastructure resources are running smoothly.

## Next unit: Manage the availability of your Azure VMs

Continue >