* 15 minutes

Recall our original scenario: creating VMs to test our CRM software. When a new build is available, we want to spin up a new VM so we can test the full install experience from a clean image. When we’re finished, we want to delete the VM.

Let’s try the commands you would use to create a VM.

## Create a Linux VM with Azure PowerShell

Because we’re using the Azure sandbox, you won’t have to create a resource group. Instead, use the resource group **[sandbox resource group name]**. In addition, be aware of the location restrictions.

Let’s create a new Azure VM with PowerShell.

1. Use the New-AzVm cmdlet to create a VM.
   * Use the resource group **[sandbox resource group name]**.
   * Give the VM a name. Typically, you want to use something meaningful that identifies the purposes of the VM, location, and (if there’s more than one) instance number. We’ll use “testvm-eus-01” for “Test VM in East US, instance 1”. Come up with your own name based on where you’ll place the VM.
   * Select a location close to you from the following list, available in the Azure sandbox. Make sure to change the value in the following example command if you’re using copy and paste.
     + westus2
     + southcentralus
     + centralus
     + eastus
     + westeurope
     + southeastasia
     + japaneast
     + brazilsouth
     + australiasoutheast
     + centralindia
   * Use “UbuntuLTS” for the image; this is Ubuntu Linux.
   * Use the Get-Credential cmdlet and feed the results into the Credential parameter.
   * Important
   * See the [Linux VM FAQ](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/faq#what-are-the-username-requirements-when-creating-a-vm) for username and password limitations. Passwords must be 12 - 123 characters in length, and meet three of the following four complexity requirements:
     + Have lowercase characters
     + Have uppercase characters
     + Have a digit
     + Have a special character (Regex match [\W\_])
   * Add the -OpenPorts parameter and pass “22” as the port. This port will let us SSH into the machine.
   * Create a public IP address name. You’ll use this name to create and find your static IP address to sign in to the machine.

* New-AzVm -ResourceGroupName [sandbox resource group name] -Name "testvm-eus-01" -Credential (Get-Credential) -Location "East US" -Image UbuntuLTS -OpenPorts 22 -PublicIpAddressName "testvm-01"
* Tip
* You can use the **Copy** button to copy commands to the clipboard. To paste, right-click on a new line in the Cloud Shell window and select **Paste**, or use the Shift+Insert keyboard shortcut (⌘+V on macOS).

1. Create a username and password, then press Enter. PowerShell will start creating your VM.
2. The VM creation takes a few minutes to complete. After completion, you can query it and assign the VM object to a variable ($vm).

* $vm = (Get-AzVM -Name "testvm-eus-01" -ResourceGroupName [sandbox resource group name])

1. Query the value to dump out the information about the VM.

* $vm
* You should see something like the following output:
* ResourceGroupName : [sandbox resource group name]  
  Id : /subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/[sandbox resource group name]/providers/Microsoft.Compute/virtualMachines/testvm-eus-01  
  VmId : 00000000-0000-0000-0000-000000000000  
  Name : testvm-eus-01  
  Type : Microsoft.Compute/virtualMachines  
  Location : eastus  
  Tags : {}  
  HardwareProfile : {VmSize}  
  NetworkProfile : {NetworkInterfaces}  
  OSProfile : {ComputerName, AdminUsername, LinuxConfiguration, Secrets}  
  ProvisioningState : Succeeded  
  StorageProfile : {ImageReference, OsDisk, DataDisks}

1. You can reach into complex objects through a dot (“.”) notation. For example, to see the properties in the VMSize object associated with the HardwareProfile section, run the following command:

* $vm.HardwareProfile

1. Or, to get information on one of the disks, run the following command:

* $vm.StorageProfile.OsDisk

1. You can even pass the VM object into other cmdlets. For example, running the following command will show you all available sizes for your VM:

* $vm | Get-AzVMSize

1. Now, run the following command to get your public IP address:

* Get-AzPublicIpAddress -ResourceGroupName [sandbox resource group name] -Name "testvm-01"

1. With the IP address, you can connect to the VM with SSH. For example, if you used the username “bob”, and the IP address is “205.22.16.5”, running this command would connect to the Linux machine:

* ssh bob@205.22.16.5
* Log out by entering exit.

## Delete a VM

To try out some more commands, let’s delete the VM. We’ll shut it down first:

Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName

Now, let’s delete the VM by running the Remove-AzVM cmdlet:

Remove-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName

Run this command to list all the resources in your resource group:

Get-AzResource -ResourceGroupName $vm.ResourceGroupName | Format-Table

You should see a bunch of resources (disks, virtual networks, and so on) that all still exist.

Microsoft.Compute/disks  
Microsoft.Network/networkInterfaces  
Microsoft.Network/networkSecurityGroups  
Microsoft.Network/publicIPAddresses  
Microsoft.Network/virtualNetworks

The Remove-AzVM command *just deletes the VM*. It doesn’t clean up any of the other resources. At this point, we’d likely just delete the resource group itself and be done with it. However, let’s run through the exercise to clean it up manually. You should see a pattern in the commands.

1. Delete the network interface:

* $vm | Remove-AzNetworkInterface –Force

1. Delete the managed OS disks and storage account:

* Get-AzDisk -ResourceGroupName $vm.ResourceGroupName -DiskName $vm.StorageProfile.OSDisk.Name | Remove-AzDisk -Force

1. Next, delete the virtual network:

* Get-AzVirtualNetwork -ResourceGroupName $vm.ResourceGroupName | Remove-AzVirtualNetwork -Force

1. Delete the network security group:

* Get-AzNetworkSecurityGroup -ResourceGroupName $vm.ResourceGroupName | Remove-AzNetworkSecurityGroup -Force

1. And finally, delete the public IP address:

* Get-AzPublicIpAddress -ResourceGroupName $vm.ResourceGroupName | Remove-AzPublicIpAddress -Force

We should’ve caught all the created resources. Check the resource group just to be sure. We performed many manual commands here, but a better approach would have been to write a *script*. Then we could reuse this logic later to create or delete a VM. Let’s look at scripting with PowerShell.

## Next unit: Create and save scripts in Azure PowerShell

[Continue](https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/7-create-resource-using-script/)

Need help? See our [troubleshooting guide](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.automate-azure-tasks-with-powershell.6-exercise-create-resource-interactively&documentId=77ba3b53-ba0c-d53e-d5df-24d1b9bef7b0&versionIndependentDocumentId=f12a42d4-7b4b-99c1-df77-181b7533bf12&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fazure%2Fautomate-azure-tasks-with-powershell%2F6-exercise-create-resource-interactively.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fautomate-azure-tasks-with-powershell%2F6-exercise-create-resource-interactively&author=mirobb) or provide specific feedback by [reporting an issue](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.automate-azure-tasks-with-powershell.6-exercise-create-resource-interactively&documentId=77ba3b53-ba0c-d53e-d5df-24d1b9bef7b0&versionIndependentDocumentId=f12a42d4-7b4b-99c1-df77-181b7533bf12&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fazure%2Fautomate-azure-tasks-with-powershell%2F6-exercise-create-resource-interactively.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fautomate-azure-tasks-with-powershell%2F6-exercise-create-resource-interactively&author=mirobb#report-feedback).