* 5 minutes

Suppose you need to choose a tool to administer the Azure resources you’ll use to test your Customer Relationship Management (CRM) system. Your tests require you to create resource groups and provision virtual machines (VMs).

You want something that is easy for administrators to learn, but powerful enough to automate the installation and setup of multiple virtual machines, or script a full application environment. There are multiple tools available, and you need to find the best one for your people and tasks.

Azure provides three administration tools:

* The Azure portal
* The Azure CLI
* Azure PowerShell

They all offer approximately the same amount of control; any task that you can do with one of the tools, you can likely do with the other two. All three are cross-platform, running on Windows, macOS, and Linux. They differ in syntax, setup requirements, automation support.

Here, we will describe each of the three options and provide some guidance on how to decide among them.

## What is the Azure portal?

The Azure portal is a website that lets you create, configure, and alter the resources in your Azure subscription. The portal is a Graphical User Interface (GUI) that makes it convenient to locate the resource you need and execute any required changes. It also guides you through complex administrative tasks by providing wizards and tooltips.

The portal does not provide any way to automate repetitive tasks. For example, to set up 15 VMs, you would need to create them one by one, completing the wizard for each VM. This can be time consuming, and is error prone for complex tasks.

## What is the Azure CLI?

The Azure CLI is a cross-platform command-line program to connect to Azure and execute administrative commands on Azure resources. For example, to create a VM, you would use a command like the following:

az vm create \  
 --resource-group CrmTestingResourceGroup \  
 --name CrmUnitTests \  
 --image UbuntuLTS  
 ...

The Azure CLI is available two ways: inside a browser via the Azure Cloud Shell, or with a local install on Linux, Mac, or Windows. In both cases, you can use it interactively, or use it with scripts to automate tasks. For interactive use, you’d first launch a shell such as cmd.exe on Windows, or Bash on Linux or macOS, then issue the commands at the shell prompt. To automate repetitive tasks, you’d assemble the commands into a shell script using the script syntax of your chosen shell, then execute the script.

## What is Azure PowerShell?

Azure PowerShell is a module you add to PowerShell to let you connect to your Azure subscription and manage resources. Azure PowerShell requires PowerShell to function. PowerShell provides services like the shell window, command parsing, and so on. The Azure Az PowerShell module adds Azure-specific commands.

For example, Azure PowerShell provides the **New-AzVM** command, which creates a virtual machine for you in your Azure subscription. To use it, you would launch the PowerShell application, then issue a command like the following:

New-AzVm `  
 -ResourceGroupName "CrmTestingResourceGroup" `  
 -Name "CrmUnitTests" `  
 -Image "UbuntuLTS"  
 ...

Azure PowerShell is also available two ways: inside a browser via the Azure Cloud Shell, or with a local install on Linux, Mac, or Windows. In both cases, you have two modes to choose from. You can use it in interactive mode, in which you manually issue one command at a time, or in scripting mode, where you execute a script that consists of multiple commands.

There is approximate parity between the portal, the Azure CLI, and Azure PowerShell with respect to the Azure objects they can administer and the configurations they can create. They are also all cross-platform. This means you will typically consider several other factors when making your choice:

* **Automation**: Do you need to automate a set of complex or repetitive tasks? Azure PowerShell and the Azure CLI support this, while Azure portal does not.
* **Learning curve**: Do you need to complete a task quickly without learning new commands or syntax? The Azure portal does not require you to learn syntax or memorize commands. In Azure PowerShell and the Azure CLI, you must know the detailed syntax for each command you use.
* **Team skillset**: Does your team have existing expertise? For example, your team may have used PowerShell to administer Windows. If so, they will quickly become comfortable using Azure PowerShell.

## Example

Recall that you are choosing an administrative tool to create the test environments for your CRM application. Your administrators have two specific Azure tasks they need to complete:

1. Create one resource group for each category of testing (unit, integration, and acceptance)
2. Create multiple VMs in each resource group before every round of testing

To create the resource groups, the Azure portal is a reasonable choice. These are one-off tasks, so you don’t need scripts to complete them.

Finding the best tool to create the VMs is a more challenging decision. You need to create several VMs, and you need to create them repeatedly, likely several times each week. This means you’ll want automation, so the Azure portal is not a good choice. In this case, either Azure PowerShell or the Azure CLI will meet your needs. If your team members have some existing PowerShell knowledge, Azure PowerShell will likely be the best match. Azure PowerShell is available on the operating systems your admin team uses, it supports automation, and should be quick for your team to learn.

Most administrators’ first experience with Azure is in the Portal. It’s a great place to start as it provides a clean, well-structured graphical interface, but provides limited options for automation. When you need automation, Azure gives you two options: Azure PowerShell for admins with PowerShell experience and the Azure CLI for everyone else.

In practice, businesses typically have a mix of one-off and repetitive tasks. This means it is common to use both the Portal and a scripting solution. In our CRM example, it’s appropriate to create the resource groups via the Portal and automate the VM creation with PowerShell.

The rest of this module focuses on installing and using Azure PowerShell.