13

Managing Azure

In this chapter, we cover the following recipes:

* Using PowerShell with Azure
* Create core Azure resources
* Exploring the Azure Storage Account
* Creating Azure SMB share
* Creating an Azure web site
* Creating an Azure Virtual Machine

# Introduction

Azure is Microsoft's cloud computing platform and is a competitor to Amazon's Amazon Web Services and other public cloud providers. Azure provides you with access to a vast and constantly growing range of features. Azure enables any organization to move some, most, or even their entire on-premises infrastructure into the cloud.

Azure features come from three levels:

* Infrastructure as a Service (IaaS)
* Platform as a Service (PaaS)
* Software as a Service (SaaS)

IaaS is, in effect, an instant computing infrastructure that you can provision, manage, and use over the internet or via a private network connection. IaaS includes essential computing infrastructure components (servers, storage, networking, firewalls, and security), plus the physical plant you require to run these components (for example, power and air conditioning). In an IaaS environment, the servers are Azure virtual machines (effectively Hyper-V VMs) and interact with the networking, security, and storage components.

PaaS is a complete deployment environment in the cloud, including the operating system, storage, and other infrastructure. One key PaaS offering in Azure is the Azure SQL Database. Things like the OS and SQL server patching, which you would have to deal with if you deploy SQL in an IaaS environment, are all managed by Azure. The Azure offering service a (nearly) complete SQL service, all managed by Azure. You can do a few things in a full SQL server implementation that the SQL PASS offering does not provide. These are generally actions that only the platform owner is allowed to perform. For example, with SQL running inside an IaaS Azure VM, you can not use SQL database mirroring—the SQL PaaS service does not provide that feature for you to use. If you need SQL services that the Azure SQL offering does not provide, you can create a VM and install and manage a SQL server in the VM.

With SaaS, you just use an application that the vendor has placed in the cloud. A key example of SaaS is Office 365 (O365), which bundles Exchange Online, SharePoint Online, Teams, and more. Strictly speaking, Office 365 is not an Azure offering—you purchase it directly from either the Office 365 website or via a Microsoft Partner. In terms of purchase, Office 365 is a single offering with many different plans (combinations of services that include a downloadable version of the Office applications, such as Word and Excel). Using PowerShell to manage Office 365, each of the included applications has a unique operational approach. With Exchange Online, for example, you use PowerShell Implicit Remoting to manage the exchange component of your Office 365 subscription. Other commands run locally but make use of REST API calls to Azure across the Internet.

To provide authentication for software running within Azure and other SaaS applications, you can use the Azure Active Directory (AAD). With AAD, you can create a cloud-only directory or synchronize the AAD with your on-premises Active Directory. AAD can also be used to provide authentication for a range of other third-party SaaS applications. Full details on managing AAD and Office 365 are outside the scope of this chapter.

We begin this chapter with the first recipe: Creating core Azure resources*,* you create an environment that allows you to manage Azure and the Office 365 SaaS components. This recipe also shows how to download the cmdlets you need.

The *Creating core Azure resources* recipe guides you through creating a few of the core resources you need to create and manage other Azure resources. These include a resource group and a storage account. You create all Azure resources within a resource group.

You create and store any required storage, such as VHD files for an Azure VM, in a storage group. While the recipes in this chapter use a single resource group and a single storage account for simplicity. In large-scale Azure deployments, you may require multiple instances of these resources.

In the Creating Azure storage recipe, we look at setting up Azure storage using the Azure storage account we created earlier. The Creating and using an Azure SMB file share recipe shows you how you can create an SMB-3 file share that you can access from client applications across the internet. Instead of having an application point to an on-premises file share, you can now host the share in Azure. This feature might be useful if you use Azure IaaS VM to host an application that utilizes a shared folder for its data. You could also use it as a file share in the cloud.

The Creating and using Azure websitesrecipe shows you how you can set up a simple website. The recipe creates a free Azure App Plan which supports an IIS Web site. With this app plan, you can set up a simple website, say for a short-term marketing campaign. You can scale this to deliver internet-scale websites that you can have Azure scale dynamically according to load.

The next recipe, Creating and using Azure virtual machines, examines how to create an Azure VM and access it via RDP. Although Azure uses a variant of Hyper-V to run Azure VMs. You can not use the Hyper-V cmdlets yousaw in Chapter 11 to maange Azure VMs.

This chapter is only a taster for using Azure with PowerShell. There is so much more that you can do that could not fit into this book.

## Getting Ready

You run the recipes in this chapter on SRV1, a Windows host or VM with Internet access.

# Creating core Azure resources

There are two fundamental things you need to do before you can start managing Azure features using PowerShell. The first is to obtain an Azure subscription. The second is to obtain the cmdlets you use to access Azure and Office 365's features.

Azure is a commercial service—each feature you use potentially has a real-world cost attached, which Microsoft bases on your usage of Azure resources. For example, with an Azure VM, you would pay to have the VM running, and there are additional charges for the VM’S virtual disk storage and network traffic in and out of your VM.

The charges for Office 365 and other SAAS offerings , on the other hand, are user-based—a given user can use lots of emails, for example, without incurring any additional charges. For details on costs for Azure, see https://azure.microsoft.com/pricing/, and for details of Microsoft 365 charges, see https://www.microsoft.com/microsoft-365/buy/compare-all-microsoft-365-products.

There are many ways you can get an Azure subscription, including via a Visual Studio subscription (https://visualstudio.microsoft.com/vs/pricing/), via an Action Pack subscription (https://docs.microsoft.com/partner-center/mpn-get-action-pack), or outright purchase on a pay as you go basis.

Microsoft also provides a one-month free trial subscription that helps you test out the redcipes in this chapter. The trial subscription provides you with full access to Azure features up to a financial limit, which is $200 US dollars or similar in other currencies at the time of writing. These limits may have changed by the time you read this book. The trial subscription should be sufficient to enable you to learn how to use PowerShell with Azure.

To get a trial subscription, navigate to https://azure.microsoft.com/free/, and fill in the forms. Note that a free trial requires you to submit a credit card number. There is no charge for the subscription; the credit card number is used only for identify verification—and it keeps the lawyers happier.

If you take out an Azure trial and wish to keep your Azure resources running after the trial expires, you have to move it to a pay as you go subscription. You should receive an email shortly before the trial expires to transition it.

To use PowerShell with Azure's various features, you need to obtain cmdlets that Microsoft does not provide in Windows Server 2022/Windows 10, Windows PowerShell 5.0/5.1, or PowerShell 7. You get the relevant modules from the PowerShell Gallery. You use the cmdlets in the PowerShellGet module to find and download the necessary modules.

As a word of warning – these cmdlets change regularly. For the most part, these changes add functions and fix bugs, but you may find that new versions of a module bring breaking changes. These could that could affect your script. The Azure team provide good notice of breaking changes, and you usually have plenty of notice and flexibility over when you deploy any updates.

## Getting Ready

You run this recipe on SRV1, a domain-joined server.

## How to do it...

1. Finding core Az module on the PS Gallery

Find-Module -Name Az |

  Format-Table -Wrap -Autosize

1. Installing Az module

Install-Module -Name Az -Force

1. Discovering Azure modules and how many cmdlets each contains

$HT = @{ Label ='Cmdlets'

         Expression = {(Get-Command -module $\_.name).count}}

Get-Module Az\* -ListAvailable |

    Sort-Object {(Get-command -Module $\_.Name).Count} -Descending |

       Format-Table -Property Name,Version,Author,$HT -AutoSize

1. Finding Azure AD cmdlets

Find-Module AzureAD |

  Format-Table -Property Name, Version, Author -AutoSize -Wrap

1. Installing the Azure AD module

Install-Module -Name AzureAD -Force

1. Discovering Azure AD Module

$FTHT = @{

    Property = 'Name', 'Version', 'Author', 'Description'

    AutoSize = $true

    Wrap     = $true

}

Get-Module -Name AzureAD -ListAvailable |

  Format-Table @FTHT

1. Logging into Azure

$Subscription = Login-AzAccount

1. Getting Azure account name

$Account = $Subscription.Context.Account.Id

"Azure Account   : $Account"

1. Viewing Azure subscription

$SubID = $Subscription.Context.Subscription.Id

Get-AzSubscription -SubscriptionId $SubId |

  Format-List -Property \*

1. Counting Azure locations

$AZL = Get-AzLocation

$LOC = $AZL | Sort-Object Location

"Azure locations:  [{0}]" -f $LOC.Count

1. Viewing Azure locations

$LOC |

  Format-Table Location, DisplayName

1. Getting Azure environments

Get-AzEnvironment |

    Format-Table -Property name, ManagementPortalURL

## How it works...

In step 1, you use the Find-Module to find the Az module on the PS Gallery. The output of this step should resemble this:

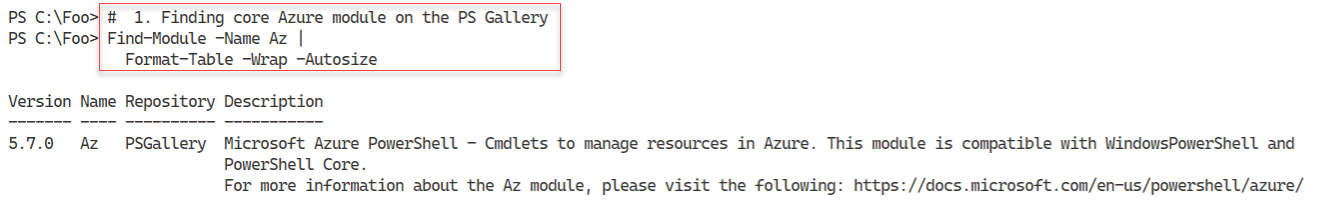


Figure 13.1: Finding the AZ module

**Insert image B42024\_13\_01.png**

In step 2, you install the Az module on SRV1. This step installs all of the individual sub-modules that you can use to manage Azure. Although this step produces no console output, you may see popup progress indicators as PowerShell installs the individual modules.

In step 3, you discover the individual modules and how many cmdlets each one contains. The output looks like this:

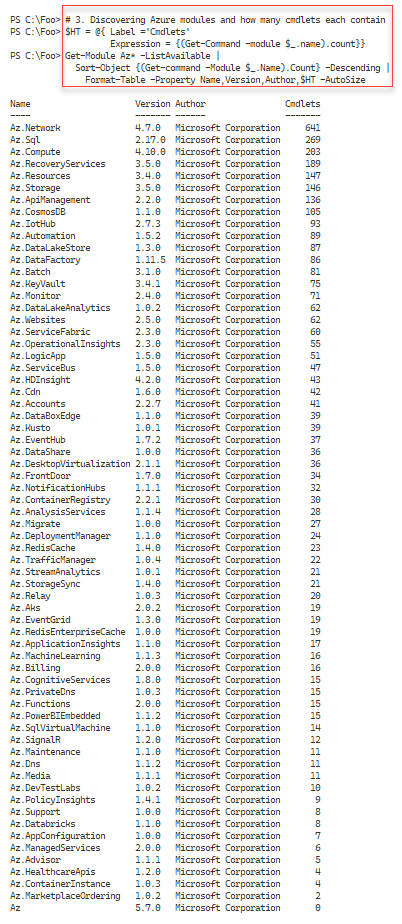


Figure 13.2 Viewing Azure modules

**Insert image B42024\_13\_02.png**

In step 4, you find the Azure AD module on the PS Gallery. The output of this step is as follows:

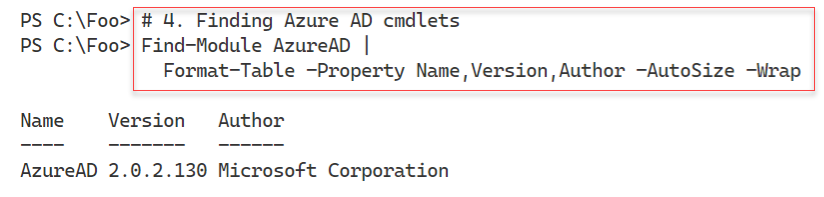


Figure 13.3 Finding the Azure AD module

**Insert image B42024\_13\_03.png**

In step 4, you find more information about the Azure AD module, with output like this:

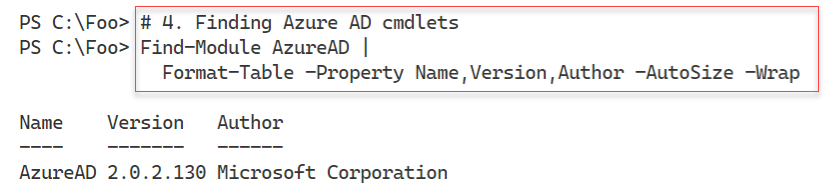


Figure 13.4 Finding details of the Azure AD module

**Insert image B42024\_13\_04.png**

In step 5, you install the Azure AD module, which generates not output to the condole. In step 6, you look at more information about this module.

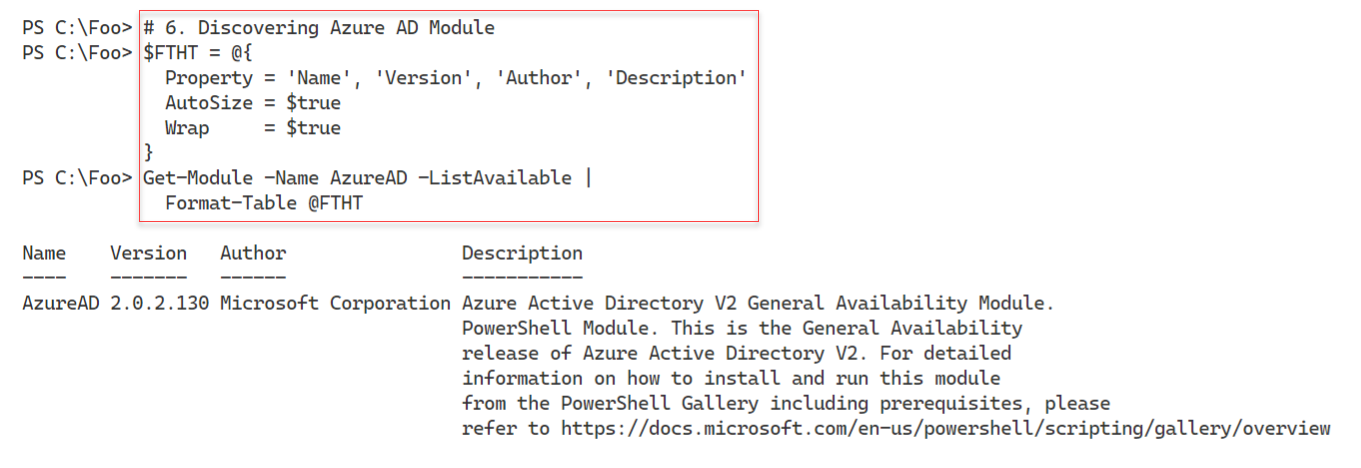


Figure 13.5 Viewing more information on the Azure AD module

**Insert image B42024\_13\_05.png**

In step 7, you use the Connect-AzAccount to login to Azure. The output of this command looks like this:

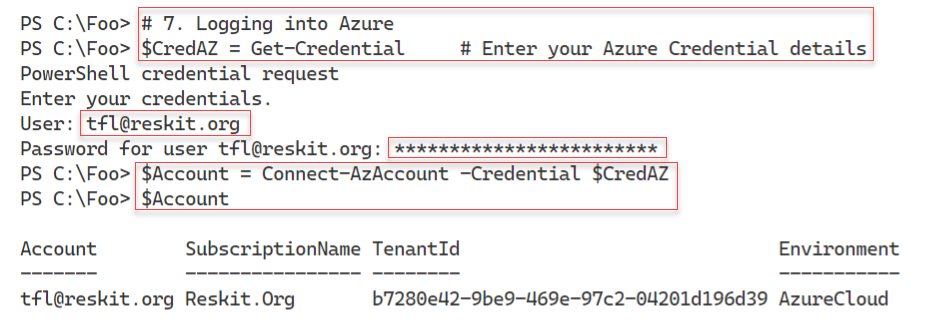


Figure 13.6 Logging in to Azure

**Insert image B42024\_13\_06.png**

Once you have logged in successfully, in step 8, you view the account name for this subscription, with output like this:

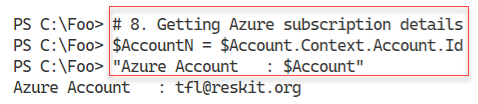


Figure 13.7 Viewing the Azure account name

**Insert image B42024\_13\_07.png**

In step 9, you view details of the Azure subscription, with output like this:



Figure 13.8 Viewing details of the Azure subscription

**Insert image B42024\_13\_08.png**

In step 10, you use the Get-AzLoction cmdlet to discover the Azure locations of Azure data centers around the world. The output, at least at the time of writing, looks like this:

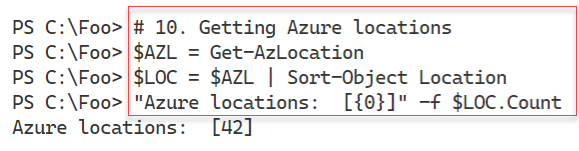


Figure 13.9 Counting Azure global locations

**Insert image B42024\_13\_09.png**

In step 11, you view the Azure locations, with output like this:



Figure 13.10 Viewing Azure global locations

**Insert image B42024\_13\_10.png**

Microsoft has created several independent Azure environments, each with an independent management portal and sets of services. The Az cmdlets work with any environment you can access, although some services may not exist in all environments. In step 12, you view the current Azure environments, with output like this:

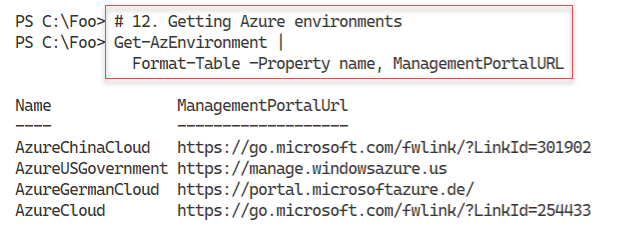


Figure 13.11 Viewing Azure cloud environments

**Insert image B42024\_13\_11.png**

## There's more...

In step 3, you enumerate the Azure modules loaded in the previous step, and for each one, you display the name and the number of commands in the module. Some of the modules and are large and contain a wealth of commands; others are small. While the AZ.Network module contains over 600 commands, the Az.MarketPlaceOrdering module contains just 2 commands. By the time you read this book, the number of modules and the commands in each has probably changed. The core modules and cmdlets you use in this chapter should not have changed, but you never know.

In step 7, you log in to Azure. In production, using a simple user and password, event a long one is not sufficiently secure. A best practice is to use multi-factor authentication (MFA). In which case, you could use the Connect-AzAccount without parameters and login via Microsoft’s GUI.

In step 10 and step 11, you count and view the Azure locations around the world. Each Azure location is one, and sometimes more than one physical data center delivering Azure services. Not every Azure location delivers all Azure service offerings, especially as Microsoft rolls out new features. Microsoft is constantly investing in new locations, so there may be even more locations by the time you read this chapter.

Microsoft maintains several separate and independently-run and operated Azure environments. In addition to the public Azure Cloud environment, Microsoft provides three additional parallel envivoronments: China, Germany, and the US government. In step 12, you view the publically acknowledged environments – there may be more (or not).

# Exploring the Azure Storage Account

## Getting Ready

This recipe uses SRV2, a recently added workgroup host. By default, this host is a DHCP client.

## How to do it...

## How it works...

## There's more...

# Creating an Azure SMB share

## Getting Ready

This recipe uses SRV1, a recently added workgroup host. By default, this host is a DHCP client.

## How to do it...

## How it works...

## There's more...

# Configuring an Azure web site

## Getting Ready

This recipe uses SRV2, a recently added workgroup host. By default, this host is a DHCP client.

## How to do it...

## How it works...

## There's more...

# Creating an Azure Virtual Machine

## Getting Ready

This recipe uses SRV1, a recently added workgroup host. By default, this host is a DHCP client.

## How to do it...

## How it works...

## There's more...