

Azure - Cheatsheet

Original Article - <https://wiki.mutschlerhome.com/en/CheatSheets/AzureCheatsheet>

- [Powershell Installation and Modules](#)
 - [Install latest version of Azure CLI on Mac](#)
 - [Install latest version of Azure CLI on Linux](#)
 - [Install PowerShell](#)
 - [Update Powershell](#)
 - [Uninstall Powershell](#)
 - [Install Azure PowerShell module](#)
 - [Setup the AzureRMAlias module](#)
 - [List available modules](#)
 - [Get authenticated in Powershell](#)
 - [List commands in a module](#)
 - [List functions in a module](#)
 - [Authenticate via the CLI](#)
- [Info Gathering](#)
 - [List subscriptions for authenticated account](#)
 - [Get tenant id](#)
 - [Get subscription id](#)
 - [List tenants](#)
 - [List Resource Groups by name](#)
 - [Set Subscription](#)
 - [List all VMs](#)
- [Blob Storage](#)
 - [General](#)
 - [Get storage keys](#)
 - [List storage containers](#)

- [List storage container contents](#)
- [List blob names](#)
- [Azure Kubernetes \(k8s\)](#)
 - [Get available versions of k8s in a region](#)
 - [List managed k8s clusters](#)
 - [Get Resource Group Name for clusters](#)
 - [Get Cluster Name](#)
 - [Configure kubect!](#)
- [Security Auditing](#)
 - [ScoutSuite](#)
 - [PowerZure](#)
 - [Get help for a particular function](#)
 - [Get all content from all KeyVault](#)
 - [MicroBurst](#)
 - [SkyArk](#)
- [Create Azure Objects](#)
 - [Create a new user](#)
- [Azure AD](#)
 - [List all applications](#)
 - [List all service principles](#)
 - [List all groups](#)
- [VMSS](#)
 - [View all VMSS in a subscription](#)
 - [Get VMSS by name and associated resource group](#)
 - [List vms in a VMSS](#)
 - [Get computer name of vms in a VMSS](#)
 - [Run command in VM in a VMSS](#)
- [Metadata Service](#)
 - [Get all instance metadata](#)

- [Get access token](#)
- [Request storage account token](#)
- [Get AKS node IP](#)

- [Roles](#)
 - [Create new role assignment](#)

- [Features](#)
 - [Show registered features in a table format](#)
 - [Find features associated w/ ContainerService](#)

- [Azure Powershell Module Cheatsheet](#)

Powershell Installation and Modules

Powershell Installation and Modules

Install latest version of Azure CLI on Mac

```
brew update && brew install azure-cli
```

[Original Article](#)

Install latest version of Azure CLI on Linux

```
# YOLO  
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Resources: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-linux?pivot=apt>

<https://docs.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-5.5.0>

Powershell Installation and Modules

Install PowerShell

```
brew install --cask powershell-preview
```

Run a powershell terminal with:

```
pwsh-preview
```

Update Powershell

```
brew update
```

```
brew upgrade powershell-preview --cask
```

Uninstall Powershell

```
brew uninstall --cask powershell  
sudo rm -rf /usr/local/bin/pwsh-preview /usr/local/microsoft/powershell
```

Resource: <https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-macos?view=powershell-7.1>

Install Azure PowerShell module

For the current user:

```
if ($PSVersionTable.PSEdition -eq 'Desktop' -and (Get-Module -Name AzureRM -ListAvailable)) {  
    Write-Warning -Message ('Az module not installed. Having both the AzureRM and ' +  
        'Az modules installed at the same time is not supported.')  
} else {  
    Install-Module -Name Az -AllowClobber -Scope CurrentUser  
}
```

Resource: <https://docs.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-5.5.0>

Setup the AzureRMAlias module

This will deal with incompatibilities with older scripts that use AzureRM:

```
Enable-AzureRmAlias
```

List available modules

```
Get-Module -ListAvailable
```

Get authenticated in Powershell

```
Connect-AzAccount %USERNAME
```

List commands in a module

```
Get-Command -Module <module name>
```

List functions in a module

```
Get-Command -Module <module name> -Type Function
```

Authenticate via the CLI

Run this command to get authenticated:

```
az login
```

This will result in a web browser opening, or a URL prompt. Navigating to this url will prompt you for a code, which you've been provided in the command line. Paste it in, click next, and select the proper account.

Resource: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-macos>

Info Gathering

Info Gathering

List subscriptions for authenticated account

az account list

Info Gathering

Get tenant id

```
az account list | jq '.[].tenantId'
```

Info Gathering

Get subscription id

az cli:

```
az account list | jq '[][.id]'
```

Powershell:

```
Get-AzSubscription
```

Resources:

[Powershell docs](#)

[CLI docs](#)

Info Gathering

List tenants

```
az account tenant list
```

Info Gathering

List Resource Groups by name

```
az group list | jq -r '.[].name'
```

Resource: <https://docs.microsoft.com/en-us/cli/azure/group?view=azure-cli-latest>

Info Gathering

Set Subscription

az cli:

```
az account set -s <name or id>
```

Powershell:

```
Set-AzureSubscription -Id [Subscription ID]
```

Resources:

<https://powerzure.readthedocs.io/en/latest/Requirements/requirements.html>

<https://stackoverflow.com/questions/38475104/azure-cli-how-to-change-subscription-default>

Info Gathering

List all VMs

```
az vm list
```

Resource: <https://docs.microsoft.com/en-us/cli/azure/vm?view=azure-cli-latest>

Blob Storage

Blob Storage

General

List all storage accounts and output in a table format:

```
az storage account list -o table
```

List all storage accounts and get storage account names:

```
az storage account list -o json | jq -r '.[].name'
```

You can assign one of the account names to an env var if you'd like:

```
export AZURE_STORAGE_ACCOUNT=<storage account name from output>
```

Blob Storage

Get storage keys

If you set the env var:

```
az storage account keys list -n $AZURE_STORAGE_ACCOUNT
```

You can assign one of the keys to an env var if you'd like:

```
export AZURE_STORAGE_KEY='<your key from the output of the previous command>'
```

Blob Storage

List storage containers

```
az storage container list --account-name $AZURE_STORAGE_ACCOUNT --account-key  
"$AZURE_STORAGE_KEY"
```

Blob Storage

List storage container contents

```
az storage blob list --container-name <name of storage container from previous command> --  
account-name $AZURE_STORAGE_ACCOUNT --account-key $AZURE_STORAGE_KEY
```

Resource: <https://www.secsignal.org/en/news/how-i-hacked-a-domain-controller-in-azure-during-a-penetration-test/>

Blob Storage

List blob names

```
az storage blob list --container-name <name of storage container from previous command> --  
account-name $AZURE_STORAGE_ACCOUNT --account-key $AZURE_STORAGE_KEY | jq '.[].name'
```

Azure Kubernetes (k8s)

Azure Kubernetes (k8s)

Get available versions of k8s in a region

```
REGION=westus2 # This will vary depending on the region you're using  
az aks get-versions --location $REGION -o table
```

Resource: <https://gist.github.com/yokawasa/fd9d9b28f7c79461f60d86c23f615677#aks-cheat-sheet>

Azure Kubernetes (k8s)

List managed k8s clusters

```
az aks list
```

Resource: <https://docs.microsoft.com/en-us/cli/azure/aks?view=azure-cli-latest>

Azure Kubernetes (k8s)

Get Resource Group Name for clusters

```
AZ_RESOURCE_GROUP_NAME=$(az aks list | jq -r '[][.resourceGroup]')
```

Azure Kubernetes (k8s)

Get Cluster Name

```
AZ_CLUSTER_NAME=$(az aks list | jq -r '[][.name]')
```

Azure Kubernetes (k8s)

Configure kubectl

This is pretty awesome, good job Microsoft:

```
az aks get-credentials --resource-group $AZ_RESOURCE_GROUP_NAME --name $AZ_CLUSTER_NAME
```

Resources:

https://www.reddit.com/r/kubernetes/comments/8ohxcz/how_to_connect_kubectl_to_aks/

<https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-deploy-cluster>

Security Auditing

ScoutSuite

<https://github.com/nccgroup/ScoutSuite> will generate an HTML report outlining various issues that exist in the configuration for a given account.

Install:

```
git clone git@github.com:nccgroup/ScoutSuite.git
cd ScoutSuite
pipenv --python 3
pipenv shell
pip install -r requirements.txt
```

Run:

```
python scout.py azure --cli
```

Resources: <https://kalilinuxtutorials.com/scout-suite-multi-cloud-security-auditing-tool/>

PowerZure

```
git clone git@github.com:hausec/PowerZure.git
cd PowerZure
pwsh-preview

# Authenticate
Connect-AzAccount

# Import PowerZure
# impo is shorthand for Import-Module
ipmo ./PowerZure.ps1

# If you have multiple subscriptions, set the one you want to target:
Set-AzureSubscription -Id [Subscription ID]

# Enumerate all roles
Get-AzureRole

# Enumerate resources the current user has access to
Get-AzureTargets

# Show info about current user
Show-AzureCurrentUser
```

Resources:

<https://powerzure.readthedocs.io/en/latest/>

[Fix error when importing AzureADPreview](#)

[Fix missing modules](#)

Show all functions

```
powerzure -h
```

Security Auditing

Get help for a particular function

For example:

```
get-help Get-AzureTargets
```

Security Auditing

Get all content from all KeyVault

```
Show-AzureKeyVaultContent -All
```

Resource:

<https://hausec.com/2020/01/31/attacking-azure-azure-ad-and-introducing-powerzure/>

MicroBurst

```
git clone git@github.com:NetSPI/MicroBurst.git
cd MicroBurst

pwsh-preview

# Authenticate
Connect-AzAccount

# Import MicroBurst
ipmo ./MicroBurst.psml

# Install module for Out-GridView
Install-Module Microsoft.PowerShell.GraphicalTools

# Show commands
Get-Command -Module MicroBurst

# Dump info from an Azure subscription
**Note:** Be sure to click a row in the pop up before clicking **Export**
Get-AzDomainInfo -folder MicroBurst -Verbose

# Look for creds or certificate stores in a number of places and dump them to `secrets.txt`
**Note:** Be sure to click a row in the pop up before clicking **Export**
Get-AzPasswords -Verbose | Out-File -FilePath ./secrets.txt

# Dump Key Vault Keys and Secrets from an Azure subscription
# via Automation Accounts specifically
**Note:** Be sure to click a row in the pop up before clicking **Export**
Get-AzKeyVaultsAutomation -Verbose
```

Resources:

[Where I found out about the tool initially](#)

[Write output to a file](#)

Security Auditing

SkyArk

```
git clone https://github.com/cyberark/SkyArk
cd SkyArk
pwsh-preview
Import-Module .\SkyArk.ps1 -force
Start-AzureStealth
```

Resource: [https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology and Resources/Cloud - Azure Pentest.md](https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Cloud%20-%20Azure%20Pentest.md)

Create Azure Objects

Create Azure Objects

Create a new user

```
New-AzureUser -Username 'test@test.com' -Password reallyAwesomePassword123!
```

Azure AD

Azure AD

List all applications

```
az ad app list --output=table --query='[].{Name:displayName,URL:homepage}'
```

List all service principles

```
az ad sp list --output=table --  
query='[].{Name:displayName,Enabled:accountEnabled,URL:homepage,Publisher:publisherName,MetadataURL:samlMetadataUrl}'
```

Azure AD

List all groups

```
az ad group list --output=json --query='[].{Group:displayName,Description:description}'
```

Resource: <https://www.blackhillsinfosec.com/red-teaming-microsoft-part-1-active-directory-leaks-via-azure/>

VMSS

VMSS

View all VMSS in a subscription

Simply navigate to this page and use the magical Try it button to use the REST API to grab this info. Neat!

CLI

```
az vmss list
```

VMSS

Get VMSS by name and associated resource group

```
az vmss list | jq '.[].name, .[].resourceGroup'
```

VMSS

List vms in a VMSS

```
az vmss list-instances -n $VMSS_NAME -g $RESOURCE_GROUP
```

Resource: <https://docs.microsoft.com/en-us/cli/azure/vmss?view=azure-cli-latest>

https://github.com/andyt530/az2tf/blob/master/scripts/295_azurevm_virtual_machine_scale_set.sh

VMSS

Get computer name of vms in a VMSS

```
az vmss list-instances -n $VMSS_NAME -g $RESOURCE_GROUP | jq '.[].osProfile.computerName'
```

Run command in VM in a VMSS

This will run commands in the instance with an id of 0. See the above commands for how to get the id that corresponds to the instance you want to work with.

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'echo $1 $1' --parameters hello world
```

Run whoami:

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'whoami'
```

Run download and run a binary as a background job:

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'bash -c "cd /tmp && wget https://example.com/binary && chmod +x binary && ./binary &"'
```

Resources:

[https://docs.microsoft.com/en-us/powershell/module/az.compute/add-](https://docs.microsoft.com/en-us/powershell/module/az.compute/add-azvmsshpublickey?view=azps-5.5.0)

[azvmsshpublickey?view=azps-5.5.0](https://docs.microsoft.com/en-us/powershell/module/az.compute/add-azvmsshpublickey?view=azps-5.5.0)

[Run command on a VMSS instance](#)

Metadata Service

Metadata Service

Get all instance metadata

```
curl -s -H Metadata:true --no-proxy "*" "http://169.254.169.254/metadata/instance?api-version=2020-09-01"
```

Get access token

```
curl -s -H Metadata:true --no-proxy "*"
"http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-
01&resource=https://management.azure.com"
```

Resources:

[Official docs](#)

[Accessible endpoints](#)

[Get access token](#)

Request storage account token

```
# Get OAuth Token
TOKEN=$(curl -s "http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com" -H Metadata:true | jq -r '.access_token')

# Get subscription id
SUB=$(curl -s -H Metadata:true --no-proxy "*" "http://169.254.169.254/metadata/instance?api-version=2020-09-01" | jq -r '.compute.subscriptionId')

# Get list of storage accounts
curl -s -H "Authorization: Bearer $TOKEN" -H Metadata:true
"https://management.azure.com/subscriptions/$SUB/providers/Microsoft.Storage/storageAccounts?api-version=2021-06-01"
```

Resource:

[Powershell script I used](#)

Metadata Service

Get AKS node IP

```
curl -s -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2017-08-01" |  
jq -r .network.interface[].ipv4.ipAddress[].privateIpAddress
```

Resource: <https://itnext.io/how-a-naughty-docker-image-on-aks-could-give-an-attacker-access-to-your-azure-subscription-6d05b92bf811>

Roles

Roles

Create new role assignment

This will try to assign the assignee the owner role:

```
az role assignment create --assignee <user or service principal> --role "owner"
```

Resource: <https://www.xmcyber.com/privilege-escalation-and-lateral-movement-on-azure-part-1/>

Features

Features

Show registered features in a table format

```
az feature list -o table
```

Features

Find features associated w/ ContainerService

```
az feature list -o table --query "[?contains(name,  
'Microsoft.ContainerService')].{Name:name,State:properties.state}"
```

Resource: <https://heranonazure.wordpress.com/2019/09/02/secure-api-server-using-authorized-ip-address-ranges/>

Azure Powershell Module Cheatsheet

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
1	Syntax	az		PowerShell uses a verb-noun pair for the names of cmdlets	
2	Log in to Azure	az login	az login -u johndoe@contoso.com -p VerySecret	Az.Accounts	Connect-AzAccount
3	Manage Azure subscription information	az account	az account list --output table	Az.Accounts	Get-AzSubscription
4	Manage private registries with Azure Container Registries	az acr	az acr list -g MyResourceGroup -o table	Az.ContainerRegistry	Get-AzContainerRegistry -ResourceGroupName "MyResourceGroup"
5	Manage Azure Container Services	az acs	az acs list-locations --subscription	Az.Compute	Get-AzContainerService -ResourceGroupName "myRG"
6	Manage Azure Active Directory	az ad	az ad group show --group	Az.Resources	Get-AzADGroup -First 100
7	Manage Azure Advisor	az advisor	az advisor recommendation list --category Performance	Az.Advisor	Get-AzAdvisorRecommendation -Category Performance
8	Manage Azure Kubernetes Services	az aks	az aks get-versions --location westus2	Az.Aks	Get-AzAksVersion -Location westus
9	Manage Azure Media Services resources	az ams	az ams account list --resource-group --subscription	Az.Media	Get-AzMediaService -ResourceGroupName "myRG"

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
10	Manage App Configurations	az appconfig	az appconfig list -g MyResourceGroup	Az.AppConfigurat ion	Get-AzAppConfigurati onStore
11	Manage App Service plans	az appservice	az appservice ase list	Az.Websites	Get-AzAppServicePla n -Location "West US"
12	Manage Azure Backups	az backup	az backup container show..	Az.RecoveryServi ces	Get-AzRecoveryServi cesBackupJob
13	Manage Azure Batch	az batch	az batch account list [--resource-group] [--subscription]	Az.Batch	Get-AzBatchAccount - AccountName "pfuller"
14	Manage Azure Billing	az billing	az billing period show --name --subscription	Az.Billing	Get-AzBillingInvoice - Latest
15	Manage Azure Content Delivery Networks	az cdn	az cdn origin list	Az.Cdn	Get-AzCdnOrigin
16	Manage Azure Cognitive Services accounts	az cognitiveservices	az cognitiveservices account list -g MyResourceGroup	Az.CognitiveServi ces	Get-AzCognitiveServi cesAccount
17	Manage Azure CLI configuration	az configure	az configure --defaults group=myRG web=myweb vm=myvm	Az.Accounts	Set-AzContext - SubscriptionId "xxxx-xxxx-xxxx- xxxx"
18	Manage Azure Container Instances.	az container	az container logs --name MyContainerGroup --resource-group MyResourceGroup	Az.ContainerInsta nce	Get-AzContainerGroup - ResourceGroupN ame demo - Name mycontainer
19	Manage Azure Cosmos DB database	az cosmosdb	az cosmosdb list [--resource-group] [--subscription]	Az.CosmosDB	Get-AzCosmosDBAcc ount - ResourceGroupN ame {resourceGroupN ame} -Name

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
20	Manage ARM template deployment at subscription scope	az deployment	az deployment group list -g testrg	Az.DeploymentManager	Get-AzDeploymentManagerArtifactSource -InputObject \$artifactSourceObject
21	Manage Azure Managed Disks	az disk	az disk list [--resource-group] [--subscription]	Az.Compute	Get-AzDisk -ResourceGroupName 'ResourceGroup01' -DiskName 'Disk01'
22	Manage Data Lake Analytics	az dla	az dla account list [--resource-group] [--subscription]	Az.DataLakeAnalytics	Get-AzDataLakeAnalyticsAccount -Name "ContosoAdlAccount"
23	Manage Data Lake Store	az dls	az dls account list --resource-group myRG	Az.DataLakeStore	Get-AzDataLakeStoreAccount -Name "ContosoADL"
24	Manage Azure Data Migration Service	az dms	az dms check-name --location westus2 --name MyService	Az.DataMigration	Get-AzDataMigrationProject -InputObject \$myService
25	Manage Azure Event Grid	az eventgrid	Get-AzEventGridDomain -ResourceGroup myRG -Name Domain1	Az.EventGrid	az ams account list --resource-group --subscription
26	Manage Azure Event Hubs	az eventhubs	az eventhubs eventhub list --resource-group myRG --namespace-name mynamespace	Az.EventHub	Get-AzEventHub -ResourceGroup myRG -NamespaceName MyNamespaceName
27	Manage resource provider features	az feature	az feature list	Az.Resources	Register-AzResourceProvider -ProviderNamespace Microsoft.Network

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
28	Find commands	az find	az find "az storage"	Az.Accounts	Get-Command -Verb Get -Noun AzVM* -Module Az.Compute
29	Manage function apps	az functionapp	az functionapp list --query "[?state=='Running']"	Az.Functions	Get-AzFunctionApp
30	Manage resource groups and template deployments	az group	az group create -l westus -n myRG	Az.Resources	Get-AzResourceGroup -Name "EngineerBlog"
31	Manage HDInsight resources.	az hdinsight	az hdinsight list [--resource-group] [--subscription]	Az.HDInsight	Get-AzHDInsightCluster
32	Managed Service Identities.	az identity	az identity list-operations [--subscription]	Az.ManagedServiceIdentity	Get-AzUserAssignedIdentity -ResourceGroupName PSRG -Name ID1
33	Manage custom virtual machine images.	az image	az image builder show --name mytemplate --resource-group my-group	Az.ManagedServiceIdentity	Get-AzImageBuilderTemplate
34	Manage KeyVault	az keyvault	az keyvault list [--resource-group] [--subscription]	Az.KeyVault	Get-AzKeyVault
35	Manage Azure Kusto resources	az kusto	az kusto cluster list --resource-group myRG	Az.Kusto	Get-AzKustoCluster -ResourceGroupName testrg
36	Manage Azure locks	az lock	az lock list	Az.Resources	Get-AzResourceLock -ResourceGroupName "myRG" -AtScope
37	Manage assignments and definitions	az managedservices	az managedservices definition list	Az.ManagedServices	Get-AzManagedServicesAssignment

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
38	Manage Azure Maps	az maps	az maps account show --name MyMapsAccount --resource-group myRG	Az.Maps	Get-AzMapsAccount - ResourceGroupName myRG
39	Manage Azure Database for MariaDB servers	az mariadb	az mariadb db list -g testgroup -s testsvr	Az.MariaDb	Get-AzMariaDbServer
40	Manage the Azure Monitor Service.	az monitor	az monitor action-group list [--resource-group] [--subscription]	Az.Monitor	Get-AzActionGroup
41	Manage Azure Database for MySQL servers.	az mysql	az mysql db list -g testgroup -s testsvr	Az.MySql	Get-AzMySqlServer
42	Manage Azure Network resources.	az network	az network nic list --query "[?dnsSettings.internalDomainNameSuffix={dnsSuffix}]"	Az.Network	Get-AzNetworkInterface
43	Manage resource policies.	az policy	az policy definition show --name MyPolicyDefinition	Az.Resources	Get-AzPolicyDefinition
44	Manage Azure Database for PostgreSQL servers.	az postgres	az postgres db list -g testgroup -s testsvr	Az.PostgreSql	Get-AzPostgreSqlServer
45	Manage dedicated Redis caches for your Azure applications.	az redis	az redis list [--resource-group] [--subscription]	Az.RedisCache	Get-AzRedisCache - Name "myexists"
46	Manage Azure Reservations	az reservations	az reservations reservation list --reservation-order-id [--subscription]	Az.Reservations	Get-AzReservation - ReservationOrderId "1111aaaa"
47	Manage Azure resources.	az resource	az resource list --location westus	Az.Resources	Get-AzResource ft

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
48	Manage user roles for access control with AAD and service principals	az role	az role assignment list [--all]	Az.Resources	Get-AzADServicePrincipal
49	Manage Azure Search services	az search	az search service list --resource-group [--subscription]	Az.Search	Get-AzSearchService - ResourceGroupName felixwa-01
50	Manage security with Azure Security Center	az security	az security alert list	Az.Security	Get-AzDiscoveredSecuritySolution
51	Manage shared image gallery	az sig	az sig list [--resource-group] [--subscription]	Az.ImageBuilder	Get-AzImageBuilderTemplate
52	Manage snapshots	az snapshot	az snapshot list [--resource-group] [--subscription]	Az.Compute	Get-AzSnapshot
53	Manage Azure SQL Databases and Data Warehouses.	az sql	az sql db list --resource-group myRG --server myserver	Az.Sql	Get-AzSqlDatabase - ResourceGroupName "myRG" - ServerName "server01"
54	Manage Azure Storage resources	az storage	az storage account list -g myRG]	Az.Storage	Get-AzStorageAccount - ResourceGroupName "RG01"
55	Manage resource tags	az tag	az tag list [--subscription]	Az.Resources	Get-AzTag -Name "Department"
56	Show the versions of Azure CLI modules	az version	az version [--subscription]	PowerShellGet	Get-InstalledModule Azure - AllVersions
57	Manage Linux or Windows virtual machines	az vm	az vm list -g myRG	Az.Compute	Get-AzVM - ResourceGroupName "myRG" - Name "VirtualMachine07"

#	Azure Product	Azure CLI	Example CLI	PowerShell Module	Example PS
58	Manage Virtual Machine Scale Set (VMSS)	az vmss	az vmss list --resource-group myRG	Az.Compute	Get-AzVmss -ResourceGroupName "Group001" -VMScaleSetName "VMSS001"
59	Manage web apps	az webapp	az webapp list --query "[?state=='Running']"	Az-WebApp	Get-AzWebApp -myRG "Default-Web-WestUS" -Name "ContosoSite"