

PowerShell cmdlets For Azure Resource Manager (ARM)

Run Windows PowerShell ISE as Administrator and use the commands mentioned in the following sections to perform various tasks with ARM. PowerShell ISE shows various parameters that are available for each of these commands.

Login Manually and through Batch Script

This section shows the necessary commands needed to manually and automatically login to ARM.

➤ **Login-AzureRmAccount**

Once this command is run, a Microsoft window pops up where you enter the Azure account email id and then it redirects to the password. Once the credentials are typed and validated, then the Azure account details appear in the PowerShell command line.

➤ **Select-AzureSubscription**

Set the correct subscription in case of multiple subscriptions by providing the SubscriptionId parameter value. If this value is unknown then use the command - **Get-AzureRmSubscription** to show a list of all subscriptions and then select the correct one.

➤ Please follow the commands mentioned in the below link to setup automatic login to Azure using PowerShell commands and these can be added to create batch scripts.

<https://cmatskas.com/automate-login-for-azure-powershell-scripts/>

Creating and managing Resource Groups

Resource groups are used to group and manage multiple resources together. For example, grouping all resources for an application like a storage account, a database instance and a Virtual Machine (VM).

➤ **New-AzureRmResourceGroup -Name 'myResourceGroup' -Location 'westeurope'**
Creates a new resource group with the specific name and location under the current subscription. Name and Location are the required parameters.

➤ **Get-AzureRmResourceGroup**

Gets a list of all resource groups under the current subscription.

Adding ResourceGroupName paramter to the above command returns details of that specific resource group.

- **Get-AzureRmResource** -ResourceName mystoragename -ResourceGroupName myResourceGroup

This command is useful to retrieve details of a resource that exists under a particular resource group. The above example retrieves a storage account details called mystoragename.

- **Set-AzureRmResource -Tag** @{ Dept="IT"; Environment="Test" } -ResourceName mystoragename -ResourceGroupName TestRG1 -ResourceType Microsoft.Storage/storageAccounts

As they are called, tags are used to add more properties to resources which enable several resources from different resource groups to belong to the same category. The above command adds two tags to the storage account (IT and Test). This is a good practice to add tags to resources.

Tags of a resource are updated as a single object. So add a tag to a resource that already has tags, retrieve the existing tags in an object, then add the new tag to that object and then reapply that object to the resource. Below is an example of it:

```
$tags = (Get-AzureRmResource -ResourceName mystoragename -ResourceGroupName TestRG1).Tags
$tags += @{Status="Approved"}
Set-AzureRmResource -Tag $tags -ResourceName mystoragename -ResourceGroupName TestRG1 -ResourceType Microsoft.Storage/storageAccounts
```

- **Find-AzureRmResource**

This command is used to find resources using different parameters as search conditions. Some of them are – ResourceNameContains, TagName, ResourceType.

- **New-AzureRmResourceLock**

This command is useful to prevent any accidental deletion or modification of a resource. It has a parameter called LockLevel with two values – CanNotDelete and ReadOnly. This lock can be removed by using Remove-AzureRmResourceLock.

- **Remove-AzureRmResource** and **Remove-AzureRmResourceGroup**

These commands are used to delete a resource or resource group respectively.

Creating and managing Virtual Networks

Virtual networks are required to be setup in order to add VMs.

- **New-AzureRmVirtualNetwork**

Creates a Virtual Network and requires some parameter values like Location, Name, AddressPrefix and ResourceGroupName. Location can be found by using the command - Get-AzureRmLocation

➤ **Test-AzureRmDnsAvailability**

This command is to test for unique domain name and is returned True if the **DomainNameLabel** value that is passed is unique.

➤ **New-AzureRmPublicIpAddress**

This command creates the public IP address and it uses the domain name that was tested and selected.

➤ **New-AzureRmLoadBalancerFrontendIpConfig**

This command creates a frontend IP configuration for incoming network traffic.

➤ **Get-AzureRmVirtualNetwork**

This command gets a list of virtual networks available within a resource group or can also get information about a single virtual network depending on the parameters that are passed.

➤ **Remove-AzureRmVirtualNetwork**

This command deletes a virtual network from a resource group.

Creating and managing Web Apps and Web Jobs

Web Apps

➤ **New-AzureRmWebApp**

This command is used to create a WebApp. Location, Name and ResourceGroupName are the additional required parameters.

➤ **Remove-AzureRmWebApp**

This command is used to delete a web app. Name and ResourceGroupName parameters are required.

➤ **Get-AzureRmWebApp**

This command gets a list of all web apps under the subscription. Name and ResourceGroupName parameters can be used to get specific web apps.

➤ **Set-AzureRmWebApp**

This command is used to change the settings of a web app. Various parameters are available to make the desired changes to the web app.

➤ **Restart-AzureRmWebapp, Stop-AzureRmWebapp, Start-AzureRmWebapp**

These commands are respectively used to restart, stop and start a web app.

Web Jobs

The following is a very useful link that describes how to use desired PowerShell scripts to run as Web Jobs on Azure.

<https://blogs.msdn.microsoft.com/nicktrog/2014/01/22/running-powershell-web-jobs-on-azure-websites/>

Creating and managing Virtual Machines

Before creating a VM, several resources related to the Virtual Network have to be setup and some of them have been mentioned above under Virtual Network section. Below are the related commands for setting up VM.

➤ **New-AzureRmVirtualNetworkSubnetConfig**

This command creates a subnet config.

➤ **New-AzureRmVirtualNetwork**

This command creates a Virtual Network where various parameters are passed in along with the above subnet config.

➤ **New-AzureRmPublicIpAddress**

This command creates a public IP Address as also mentioned in the earlier section.

➤ **New-AzureRmNetworkSecurityRuleConfig**

This command is used to create network security rules for inbound and outbound access as well as for remote desktop and web access. Accordingly various parameters are setup.

➤ **New-AzureRmNetworkSecurityGroup**

This command creates a network security group using the above security rules.

➤ **New-AzureRmNetworkInterface**

This command creates a network card which connects the VM to a subnet, network security group, and public IP address.

➤ **New-AzureRmVMConfig**

This command creates a VM configuration which includes various settings from the above commands that are required for the VM deployment. During this command there is a prompt to enter credentials which are configured to login to t

➤ **New-AzureRmVM**

This command creates a VM and uses the VM configuration from the above command as parameter value.

➤ **Get-AzureRmPublicIpAddress**

This command gets the public IP address of the VM which can be used to test web connectivity and remote desktop connection with the VM.

➤ **Install-WindowsFeature -name Web-Server -IncludeManagementTools**

This command is used to install IIS on the VM and enable local firewall rule to allow web traffic.

➤ **Remove-AzureRmResourceGroup**

This command deletes the resource group and all its resources and is useful to delete VM and its resources when it's no longer needed.

Creating and managing Storage Accounts

Azure Storage helps to store various types of data that are useful for applications and works like a local storage place to access various files.

➤ **New-AzureStorageAccount**

This command creates a storage account for a specific Azure account and Azure subscription. It takes the storage name and location as parameters.

➤ **Get-AzureStorageAccount**

This command gets a list of storage accounts under the current subscription.

➤ **Get-AzureStorageKey**

This command gets the primary storage access key for the storage account.

➤ **New-AzureStorageContext**

This command creates a storage context which authenticates further storage related requests without having to pass the account details and access key each time.

➤ **New-AzureStorageContainer** and **Set-AzureStorageBlobContent**

These commands are used to create a storage container and to upload a blob to the container. They are used to store large amount of unstructured data like text.

➤ **New-AzureStorageTable**

This command creates a table in the Azure storage using the previously created storage context. They are used to store large number of NoSQL data.

➤ **New-AzureStorageQueue**

This command create queues in Azure storage using the previously created storage context. They are used to store large number messages.

Configure alerts and diagnostics

These commands work under a specific Azure account and subscription.

➤ **Get-AzureRmLog**

This command retrieves activity logs for various scenarios depending on the parameters that are passed such as date, time, caller. It can also be used to view all alert events.

➤ **Get-AzureRmAlertHistory**

This command is used to retrieve alert rule history and related information depending on the parameters that are passed.

➤ **Get-AzureRmAlertRule**

This command is used to retrieve information on various properties of alert rules and it takes different parameters.

There are several other alert rule commands and can be used as needed.

➤ **Get-AzureRmDiagnosticSetting**

This command is used to get diagnostic log settings for a resource group.

➤ **Set-AzureRmDiagnosticSetting**

This command is used to change the diagnostic settings using various parameters.

Working with availability sets and load balancers

Load Balancers

The following commands help to create an internal load balancer for azure VMs inside a cloud service or virtual networks within a specific region. These commands are for the same resource group.

➤ Create a subnet config and virtual network as mention in the virtual network section.

➤ **New-AzureRmLoadBalancerRuleConfig**

This command is used to create a load balancer rule which takes other config rules as parameters like **New-AzureRmLoadBalancerInboundNatRuleConfig** (NAT config rule), **New-AzureRmLoadBalancerProbeConfig** (probe config rule).

➤ **New-AzureRmLoadBalancer**

This command creates a load balancer by adding all the above created config rules as parameters.

Availability Sets

➤ **AzureAvailabilitySet**

This command is used as part of a parameter with the **Get-AzureVM** command to add the availability set to a VM.

Creating and managing Network Security Groups

Network Security Groups (NSGs) contain access control rules that provide traffic control to VMs, NICs or subnets in a virtual network based various criteria such as traffic direction, protocol, address.

➤ **New-AzureRmNetworkSecurityRuleConfig**

This command is used to create config rule for NSG by adding various parameters as criteria as mentioned above. Multiple rules are created depending on the requirement such as number of subnets and so on.

➤ **New-AzureRmNetworkSecurityGroup**

This command is used to create an NSG by adding above created config rules as security rules parameter.

➤ **Get-AzureRmNetworkSecurityGroup**

This command gets a list of all NSGs available in a subscription or resource group based on the parameters used. It also gets all the rules and associations of the NSG.

➤ **Add-AzureRmNetworkSecurityRuleConfig**

This command is used to add a new rule to NSG.

➤ **Set-AzureRmNetworkSecurityGroup**

This command is used to update or make new changes to NSG. So for example, this command is run after the above command to save the new changes.

➤ **Set-AzureRmNetworkSecurityRuleConfig**

This command along with **Set-AzureRmNetworkSecurityGroup** command is used to make changes to existing rules of NSG.

➤ **Remove-AzureRmNetworkSecurityRuleConfig**

This command along with **Set-AzureRmNetworkSecurityGroup** command is used to delete a rule.

Working with Azure Active Directory (Azure AD)

Azure AD is Microsoft's multi-tenant, cloud based directory and identity management service. It provides various access control rules for applications.

There is a separate PowerShell module for Azure AD that can be installed so as to use all the latest PowerShell cmdlets for Azure AD.

➤ **Connect-AzureAD**

This command checks for Azure account authentication and connects to Azure AD.

➤ **New-AzureADDevice**

This command creates a device in Azure AD.

➤ **Confirm-AzureADDomain**

This command confirms the ownership of a domain in Azure AD.

➤ **Get-AzureADUser** and **New-AzureADUser**

These commands get an Azure AD user and creates a new user respectively.

➤ **New-AzureADServicePrincipal**

This command creates a service principal.

Working sample scripts

Working with Web Apps

```
$appdirectory="<Replace with your app directory>"
$webappname="mywebapp$(Get-Random)"
$location="West Europe"

# Create a resource group.
New-AzureRmResourceGroup -Name myResourceGroup -Location $location

# Create an App Service plan in `Free` tier.
New-AzureRmAppServicePlan -Name $webappname -Location $location `
-ResourceGroupName myResourceGroup -Tier Free

# Create a web app.
New-AzureRmWebApp -Name $webappname -Location $location -AppServicePlan $webappname `
-ResourceGroupName myResourceGroup
```


Working with VMs

```
# Variables for common values
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
New-AzureRmResourceGroup -Name $resourceGroup -Location $location

# Create a subnet configuration
$subnetConfig = New-AzureRmVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix
192.168.1.0/24

# Create a virtual network
$vnNet = New-AzureRmVirtualNetwork -ResourceGroupName $resourceGroup -Location $location `
-Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

# Create a public IP address and specify a DNS name
$pip = New-AzureRmPublicIpAddress -ResourceGroupName $resourceGroup -Location $location `
-Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4

# Create an inbound network security group rule for port 3389
$nsgRuleRDP = New-AzureRmNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -
Protocol Tcp `
-Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -
DestinationAddressPrefix * `
-DestinationPortRange 3389 -Access Allow

# Create a network security group
$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location `
-Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP

# Create a virtual network card and associate with public IP address and NSG
$nic = New-AzureRmNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location
$location `
-SubnetId $vnNet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration
$vmConfig = New-AzureRmVMConfig -VMName $vmName -VMSize Standard_D1 | `
Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `
Set-AzureRmVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus
2016-Datacenter -Version latest | `
Add-AzureRmVMNetworkInterface -Id $nic.Id
```

```
# Create a virtual machine
New-AzureRmVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig
```

Working with Azure AD

```
# Get credential for current tenant and store it in a variable.
$Credential = Get-Credential

# Connects the authenticated account to Azure AD.
Connect-AzureAD -Credential $Credential

# Create a new application in Azure AD.
New-AzureADApplication -DisplayName "My new application" -IdentifierUri
"http://mynewapp.contoso.com"
```

Working with storage accounts

```
# Update with the name of your subscription.
$SubscriptionName = "YourSubscriptionName"

# Give a name to your new storage account. It must be lowercase!
$StorageAccountName = "yourstorageaccountname"

# Choose "West US" as an example.
$Location = "West US"

# Add your Azure account to the local PowerShell environment.
Add-AzureAccount

# Set a default Azure subscription.
Select-AzureSubscription -SubscriptionName $SubscriptionName -Default

# Create a new storage account.
New-AzureStorageAccount -StorageAccountName $StorageAccountName -Location $Location
```

Thanks for checking out this 10-page study guide for the key Powershell cmdlets when working with Azure. If you found it useful, you'll find Azure courses available at HeroCourses.com or SoftwareArchitect.ca.

