title: Azure PowerShell Build 2017 QuickStart
date: 2017-05-02 14:08:08

Azure PowerShell Build 2017 QuickStart

This challenge requires using Azure PowerShell. You will not need any other programs or tools besides:

- Windows machine.
- PowerShell 5.0+.
- MSTSC (Remote Desktop Client).

Setup

These steps can also be completed by following the Azure PowerShell Get Started Documentation (https://docs.microsoft.com/en-us/powershell/azure/get-started-azureps).

Install Azure PowerShell

Using the instructions found in the Azure PowerShell Documentation (https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps#step-2-install-azure-powershell), install the `AzureRM` module.

> NOTE: You may need to run PowerShell as an administrator.

```
$ Install-Module AzureRM
```

Login

Login to Azure PowerShell with one command.

```
$ Login-AzureRmAccount
```

A dialog box will appear, prompting you to enter your credentials. Upon completion, you should see some basic information.

```
Environment           : AzureCloud
Account               : your@email.com
TenantId              : ########-####-####-####-############
SubscriptionId        : ########-####-####-####-############
SubscriptionName      : SubName
CurrentStorageAccount :
```

Your "default subscription" is denoted as `SubName` above.

Select subscription

You may want to use a different subscription, so you need to select your preferred subscription. First, get a list of your subscriptions.

```
$ Get-AzureRmSubscription

SubscriptionName : SubName1
SubscriptionId   : ########-####-####-####-############
TenantId         : ########-####-####-####-############
State            : Enabled

SubscriptionName : SubName2
SubscriptionId   : ########-####-####-####-############
TenantId         : ########-####-####-####-############
State            : Disabled
```

The easiest way to select your preferred subscription is by selecting the subscription and piping it to `Select-AzureRmSubscription`.

```
$ Get-AzureRmSubscription -SubscriptionName "SubName1" | Select-
AzureRmSubscription
```

Create a resource group

Create a new resource group

A good first step for exploring Azure is to create a new resource group in which you can experiment. Let's take a look at the resource group commands and find how to create one.

```
$ Get-Command *ResourceGroup

CommandType     Name                                               Version
Source
-----------     ----                                               -------    -
-----
Cmdlet          Export-AzureRmResourceGroup                        3.8.0
AzureRM.Resources
Cmdlet          Find-AzureRmResourceGroup                          3.8.0
AzureRM.Resources
Cmdlet          Get-AzureRmResourceGroup                           3.8.0
AzureRM.Resources
Cmdlet          New-AzureRmResourceGroup                           3.8.0
AzureRM.Resources
Cmdlet          Remove-AzureRmResourceGroup                        3.8.0
AzureRM.Resources
Cmdlet          Set-AzureRmResourceGroup                           3.8.0
AzureRM.Resources
```

In this case, you want to create a new resource group. You can get help for the `New-AzureRmResourceGroup` cmdlet (and any other cmdlet) in a few different ways. First, you can get help straight from the PowerShell prompt.

```
$ Get-Help New-AzureRmResourceGroup

NAME
    New-AzureRmResourceGroup

SYNOPSIS
    Creates an Azure resource group

SYNTAX
    New-AzureRmResourceGroup -Name <String> -Location <String> [-Tag
<Hashtable>] [-Force <SwitchParameter>]
    [<CommonParameters>]

DESCRIPTION
    The New-AzureRmResourceGroup cmdlet creates an Azure resource group and
returns an object that represents the
    resource group.

    If you find an issue with this cmdlet, please create an issue on
https://github.com/Azure/azure-powershell/issues,
    with a label "ResourceManager".

RELATED LINKS
    Online Version: http://go.microsoft.com/fwlink/?LinkID=393048

REMARKS
    To see the examples, type: "get-help New-AzureRmResourceGroup -examples".
    For more information, type: "get-help New-AzureRmResourceGroup -detailed".
    For technical information, type: "get-help New-AzureRmResourceGroup -full".
    For online help, type: "get-help New-AzureRmResourceGroup -online"
```

Alternatively, you can get help information from the [Azure PowerShell Reference Documentation (https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/new-azurermresourcegroup)](https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/new-azurermresourcegroup).

In this case, we want to create a resource group in the `West US 2` location.

```
$ New-AzureRmResourceGroup -Name MyQuickStartRg -Location WestUS2

ResourceGroupName : MyQuickStartRg
Location          : westus2
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/########-####-####-####-
############/resourceGroups/MyQuickStartRg
```

Create a Windows virtual machine

Each application uses different Azure services, but let's start with the common task of creating a VM. This quickstart challenge uses a Windows VM so that you can remote into the VM. All subsequent calls follow an [Azure PowerShell Sample Script (https://docs.microsoft.com/en-](https://docs.microsoft.com/en-)

us/azure/virtual-machines/scripts/virtual-machines-windows-powershell-sample-create-vm?toc=%2fpowershell%2fmodule%2ftoc.json) and could be run as part of a `*.ps1` file. For this challenge, you will walk through a few steps at a time.

Helpful variables

You should first set a few variables in PowerShell to make issuing the cmdlets easier.

```
$ $resourceGroup = "MyQuickStartRg"
$ $location = "WestUS2"
$ $vmName = "MyQuickStartVm"
$ $dnsName = "$vmName$(Get-Random)"
```

Setup network configuration

Create a subnet.

```
$ $subnetConfig = New-AzureRmVirtualNetworkSubnetConfig -Name mySubnet -
AddressPrefix 192.168.1.0/24
```

Create a virtual network.

```
$ $vnet = New-AzureRmVirtualNetwork -ResourceGroupName $resourceGroup -Location
$location -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
```

Create a public IP address and give the VM a public DNS name.

```
$ $pip = New-AzureRmPublicIpAddress -ResourceGroupName $resourceGroup -Location
$location -Name  -AllocationMethod Static -IdleTimeoutInMinutes 4
```

Create an inbound network security rule for port 3389 (this allows RDP).

```
$ $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 $vnet.Subnets[0].Id -
PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id
```

Create the VM configuration

First, you must create a VM configuration `PSObject`.

```
$ $vmConfig = New-AzureRmVMConfig -VMName $vmName -VMSize Standard_DS1_v2
```

Now, set the operating system values.

```
$ $vmConfig = $vmConfig | Set-AzureRmVMOperatingSystem -Windows -ComputerName
$vmName -Credential $(Get-Credential -Message "Enter a username and password
for the virtual machine.")
```

Set the source image.

```
$ $vmConfig = $vmConfig | Set-AzureRmVMSourceImage -PublisherName
MicrosoftWindowsServer -Offer WindowsServer -Skus 2016-Datacenter -Version
latest
```

Add the network interface.

```
$ $vmConfig = $vmConfig | Add-AzureRmVMNetworkInterface -Id $nic.Id
```

Create the VM

At this point, you can issue the statement to create the VM.

```
$ New-AzureRmVM -ResourceGroupName $resourceGroup -Location $location -VM
$vmConfig
```

> NOTE: This step may take a few minutes to complete. Check out the Azure PowerShell Documentation (https://docs.microsoft.com/en-us/powershell/azure/overview) while you wait!

Remote into VM

Start the RDP session by initiating the connection from the command-line.

```
$ mstsc /v:$($pip.IpAddress)
```

Cleanup

To prevent extraneous cost, clean up your resource group (unless you plan to interact with this VM at a later date).

```
$ Remove-AzureRmResourceGroup -Name MyQuickStartRg
```

Send feedback

If you've followed along, you should now be aware of another tool available for you. Azure PowerShell is open source (https://github.com/Azure/azure-powershell) and available for Windows (coming to Mac/Linux soon).

Your final step to complete this challenge: provide us feedback!

```
$ Send-Feedback
```

Conclusion

For more information, visit us online:

- Azure PowerShell Documentation (https://docs.microsoft.com/en-us/powershell/azure).
- Our GitHub Repo (https://github.com/Azure/azure-powershell).