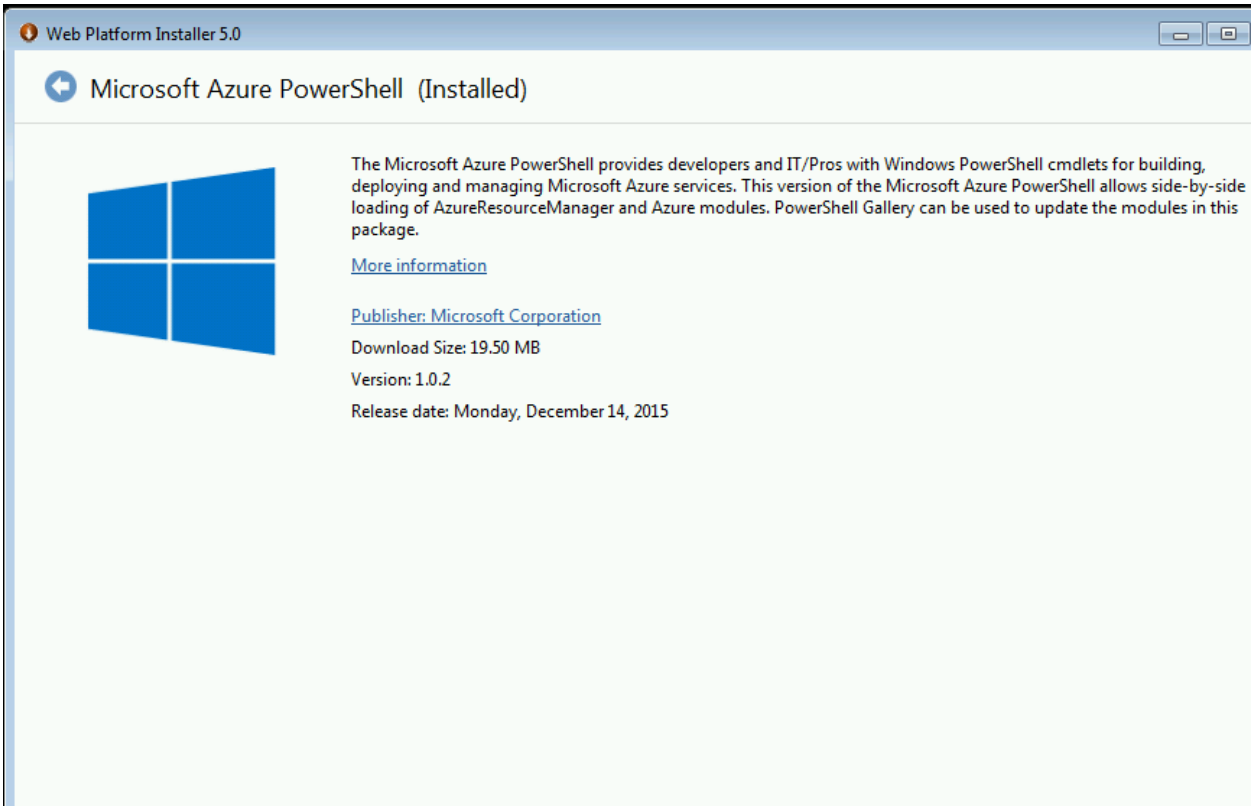
Round up Using Azure PowerShell to manage IaaS VM

Prerequisites

1) We have already installed **Azure PowerShell 1.0** on the VM using Web PI



2) You will need an active azure subscription. A subscription will be provided to you in the lab interface.

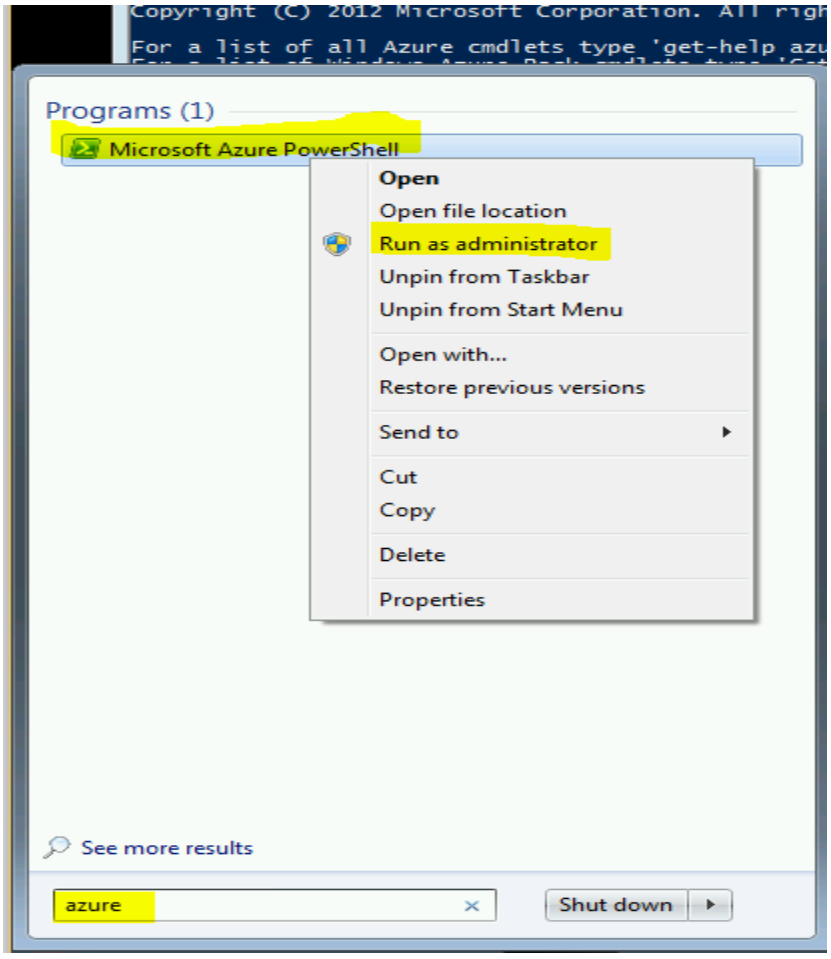
Step 1

High level Steps

- 1.1 Get Started Get-Command
- 1.2 Get-Help

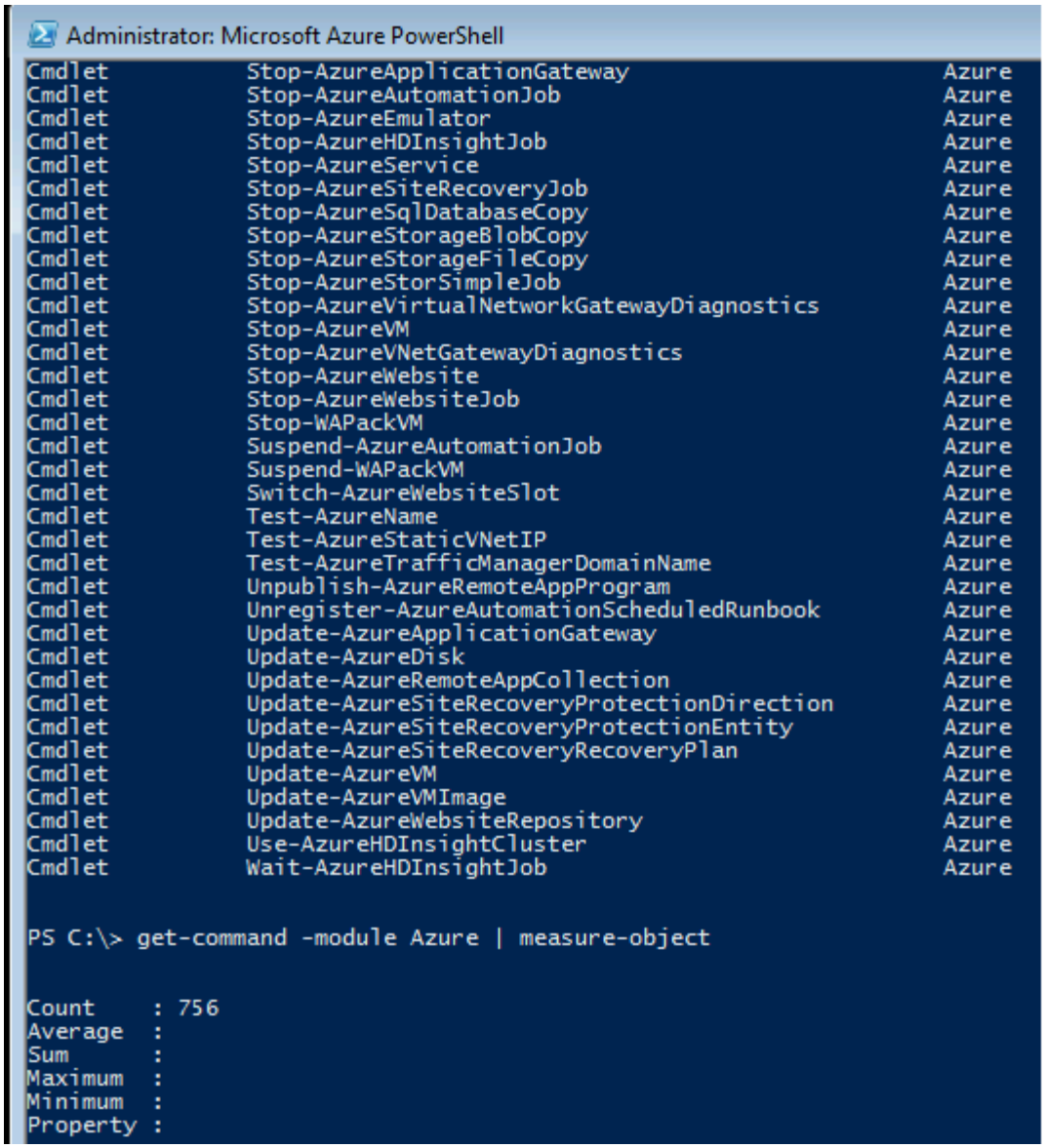
Step 1.1 Get Started Get-Command

Start Azure PowerShell Window as **Administrator** (a shortcut is on the Desktop)



The first thing I always recommend to anyone when exploring a new PowerShell module is to use the **Get-Command -Module Azure** cmdlet. By using this command, you can easily explore all of the cmdlets available to you in that module.

1) **Get-Command -Module Azure : To list available cmdlet in Azure PowerShell**
2) **Get-Command -Module Azure | Measure-Object : To get count of available cmdlet in Azure PowerShell**



As of this writing, there are 756 cmdlets in the Azure module. You've got a lot of opportunity here!

Step 1.2 Get Help

Another cmdlet you should get familiar with is **Get-Help**. PowerShell has a great help system, and using it allows you to get up to speed quickly on everything the Azure module can do. To use PowerShell's help, simply pass the name of the azure cmdlet you want to learn more about to the Get-Help cmdlet.

```
Get-Help Get-AzureVMImage
```

This example retrieves all help topics about the Get-AzureVMImage cmdlet. Feel free to peruse the cmdlets available to you in the Azure module, and use Get-Help to dig a little deeper to understand their purpose.

Step 2 Associate Windows Azure account with Azure PowerShell

High level Steps

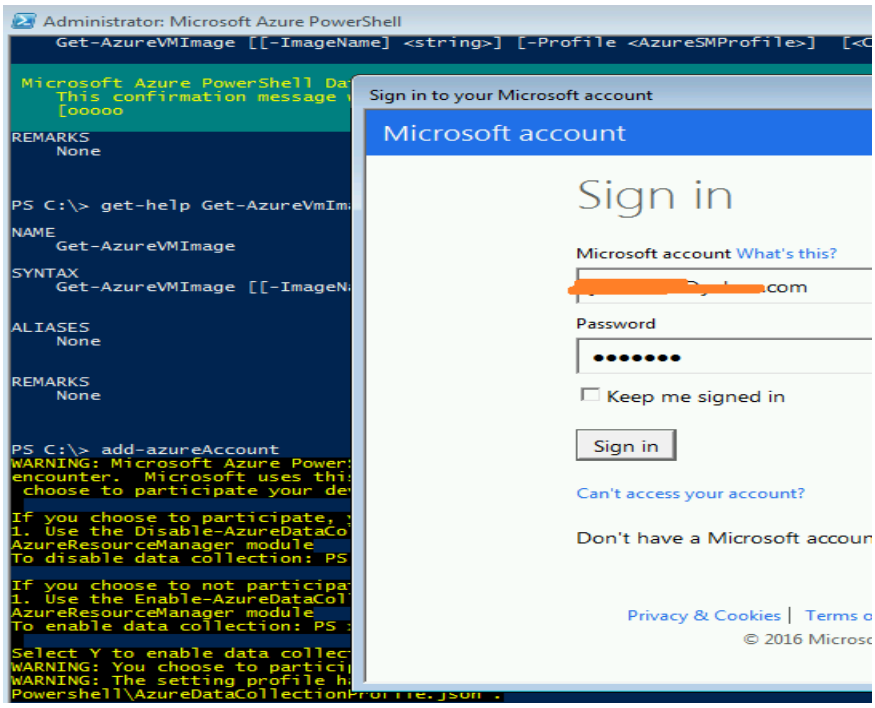
- 2.1 Add Windows Azure Account
- 2.2 Download and Import an Azure Publish Settings File
- 2.3 Get and Select Your Azure Subscription

Step 2.1 Add Windows Azure Account

Once PowerShell for Windows Azure is installed, the first thing that must be done is to associate PowerShell with the Windows Azure account and subscription to be managed. Begin by opening an elevated PowerShell session and adding your Windows Azure account using the following command:

Add-AzureAccount

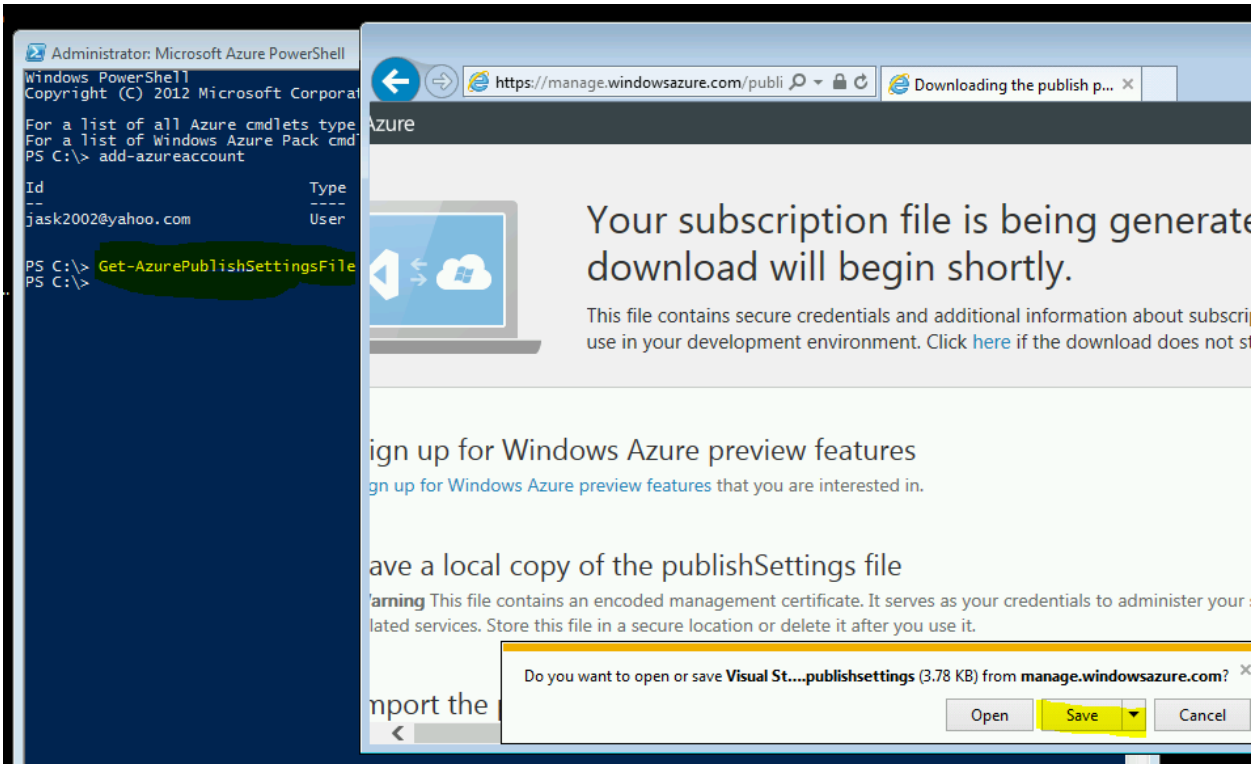
You'll be prompted to enter the e-mail address of the account you wish to sign in with. [Ask coordinator to provide you with azure demo account]



Step 2.2 Download and Import an Azure Publish Settings File

After you've authenticated successfully, download the Azure Publish Settings file using the following PowerShell command:

Get-AzurePublishSettingsFile



Note: This will open IE and prompt you to save publishsetting file. When you save your Publish Settings file, specify a short easily-typed file name to save your file. You need to type the complete path to this file name in the next step.

Import the downloaded Azure Publish Settings file into your Windows PowerShell environment. This updates your Windows PowerShell environment with the connection and certificate information that is needed to connect and authenticate to your Azure subscription: Run following

Import-AzurePublishSettingsFile -PublishSettingsFile "full path to downloaded file"

For example

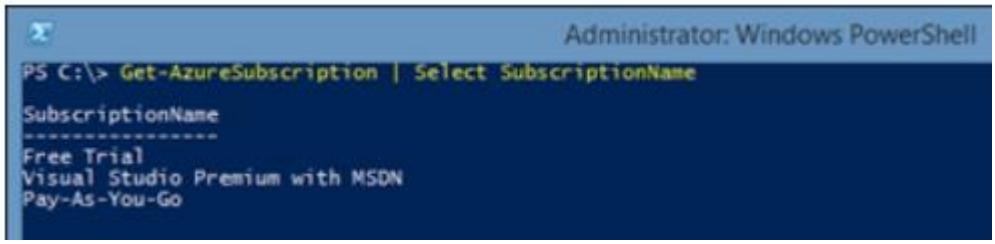
```
PS C:\> Import-AzurePublishSettingsFile -PublishSettingsFile "C:\Users\admin\Downloads\Visual Studio Ultimate with MSDN-
1-12-2016-credentials.publishsettings"


Id          : cbecf419-2974-4845-956b-adf47abb0897
Name        : Visual Studio Ultimate with MSDN
Environment : AzureCloud
Account     : EA1F9930DD979D78E5C08F6470EFC9373D6D2E8A
Properties  : {}
```

Step 2.3 Get and Select Your Azure Subscription

After you complete the steps in this section, Windows PowerShell is configured to connect to your Azure subscription. To test this connection, enter the following cmdlet to connect to Azure and enumerate your subscription information:

Get-AzureSubscription | Select SubscriptionName



Next, select an Azure Subscription [from above output] you want to manage the resources for by executing this command:

Select-AzureSubscription -SubscriptionName "subscription name here"

Step 3 Manage Resources: Build a new virtual machines in Azure using PowerShell

High Level Steps

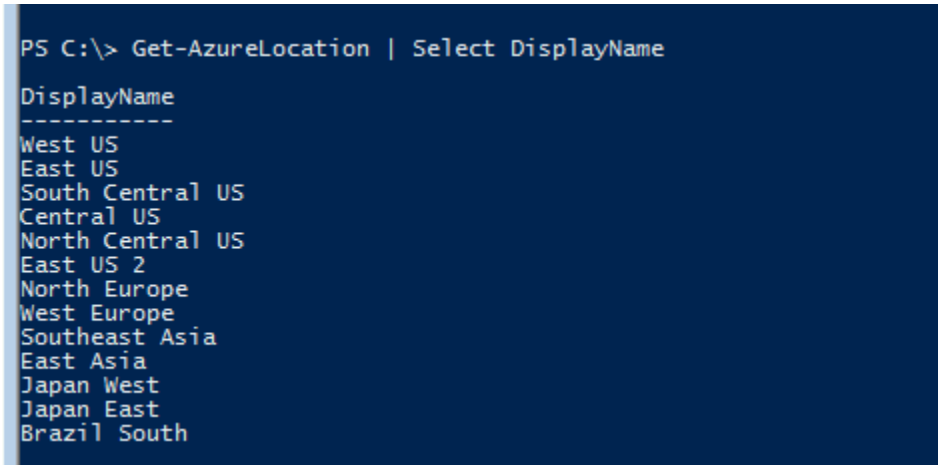
- 3.1 Select location for deployment and associated to subscription
- 3.2 Generate a list of Win 2012 Datacenter Azure platform images for new virtual machines.
- 3.3 Set the common parameters for creating virtual machine
- 3.4 Create VM in Azure Cloud

To start building a new virtual machine in the cloud, we need to select an appropriate virtual machine image, and then use that image to build a new virtual machine.

Step 3.1: Select location for deployment and associated to subscription

To display a list of locations, execute the following PowerShell command:

Get-AzureLocation | Select DisplayName



Once you've decided on a location to create the storage account. Next, execute the following PowerShell command to create affinity group and storage account:

New-AzureAffinityGroup -Name pslab-group -Location "East US"

New-AzureStorageAccount -StorageAccountName "yournameintials" -AffinityGroup pslab-group

Our new Azure storage account is provisioned, but we want to set this new storage account as our default storage location for our subscription by using the following command:

Get-AzureSubscription

Set-AzureSubscription –SubscriptionName "YourSubscriptionName" -CurrentStorageAccount "yournameinitials"

```
PS C:\> Get-AzureSubscription


SubscriptionId             : cbecf419-2974-4845-956b-adf47abb0897
SubscriptionName           : Visual Studio Ultimate with MSDN
Environment                : AzureCloud
DefaultAccount             : [redacted]com
IsDefault                  : True
IsCurrent                  : True
TenantId                   : 22d6e69d-339d-4ed9-8eb0-4e0f9276bdd2
CurrentStorageAccountName :


PS C:\> Set-AzureSubscription -SubscriptionName "Visual Studio Ultimate with MSDN" -CurrentStorageAccount "yournameintia
ls"
```

Step 3.2 Generate a list of Win 2012 Datacenter Azure platform images for new virtual machines:

Now let's look at the list of available virtual machine images for Win 2012 available to us. Use the following PowerShell command:

The snippet below will return all the Windows Server 2012 Datacenter images

Get-AzureVMImage -Verbose:$false | Where-Object {$_.label -like "Windows Server 2012 Datacenter*"} | Format-Table Label, PublishedDate -AutoSize

```
PS C:\> Get-AzureVMImage -Verbose:$false | Where-Object {$_.label -like "Windows Server 2012 Datacenter*"} | Format-Tabl
e Label, PublishedDate -AutoSize

Label                                        PublishedDate
-----                                        -------------
Windows Server 2012 Datacenter, August 2015    8/24/2015 12:00:00 AM
Windows Server 2012 Datacenter, September 2015 9/16/2015 12:00:00 AM
Windows Server 2012 Datacenter, October 2015   10/22/2015 12:00:00 AM
Windows Server 2012 Datacenter, November 2015  11/20/2015 12:00:00 AM
Windows Server 2012 Datacenter, December 2015  12/14/2015 12:00:00 AM
```

Run this command now

$VMImage = @(Get-AzureVMImage | Where-Object -Property Label -Match "Windows Server 2012 Datacenter, December 2015").ImageName

Perfect! Now $VMImage is storing the **ImageName** for our selected platform image, which we can use when provisioning a new virtual machine on the Azure cloud platform.

This will return a long list of virtual machine images to choose from. Once you've decided on a location and image, you can create the VM

Step 3.3 Set the common parameters for creating virtual machine

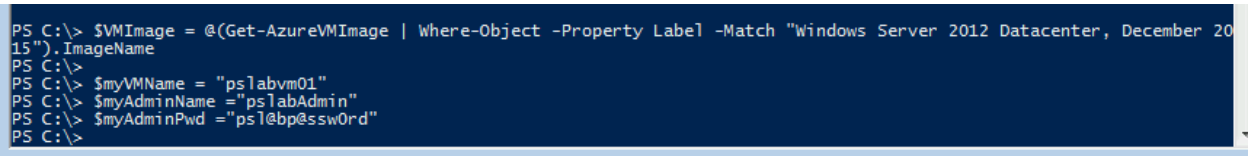
Prior to provisioning our new virtual machine, let's set a few Windows PowerShell variables for some of the common parameters that you may want to customize, such as the virtual machine name, Admin user name, and Admin password:

Run following one by one

$myVMName = "pslabvm01"

$myAdminName ="pslabAdmin"

$myAdminPwd ="psl@bp@ssw0rd"



Step 3.4 Create VM in Azure Cloud

Now, we're ready to build our new virtual machine in Azure. To configure on our new virtual machine during the provisioning process, we use the **New-AzureQuickVM** cmdlet to specify the virtual machine image, virtual machine name, and local Admin credentials. In addition, we'll specify the Azure affinity group that we configured earlier so our storage and virtual machines will be organized in a common affinity group to improve performance.

Run following command

New-AzureQuickVM -ImageName $VMImage -Windows -Name $myVMName -ServiceName $myVMName -AdminUsername $myAdminName -Password $myAdminPwd -AffinityGroup pslab-group

```
PS C:\> $VMImage = @(Get-AzureVMImage | Where-Object -Property Label -Match "Windows Server 2012 Datacenter, December 2015").ImageName
PS C:\> $myVMName = "pslabvm011"
PS C:\> $myAdminName ="pslabAdmin"
PS C:\> $myAdminPwd ="psl@bp@sswOrd"
PS C:\> New-AzureQuickVM -ImageName $VMImage -Windows -Name $myVMName -ServiceName $myVMName -AdminUsername $myAdminName -Password $myAdminP
tyGroup pslab-group
WARNING: No deployment found in service: 'pslabvm011'.

OperationDescription                    OperationId                             OperationStatus
--------------------                    -----------                             ---------------
New-AzureQuickVM                        d526649f-9ab3-6485-9f62-75b7488c17ab    Succeeded
New-AzureQuickVM                        0e5c017c-7081-63ba-a335-4a8b40fec411    Succeeded
```

After a few minutes…Presto! Our new virtual machine is created in the Azure cloud (as depicted in the previous screenshot) with an **OperationStatus** of **Succeeded**.

Note

If you received following error try changing the $myVMName and rerun the New-AzureQuickVM

```
tyGroup pslab-group
WARNING: No deployment found in service: 'pslabvm01'.
New-AzureQuickVM : Service already exists, AffinityGroup cannot be specified.
At line:1 char:1
+ New-AzureQuickVM -ImageName $VMImage -Windows -Name $myVMName -ServiceName $myVM ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : CloseError: (:) [New-AzureQuickVM], ApplicationException
    + FullyQualifiedErrorId : Microsoft.WindowsAzure.Commands.ServiceManagement.IaaS.PersistentVMs.NewQuickVM
```

Run **get-azurevm** to see the status

```
PS C:\> get-azurevm

ServiceName                             Name                                    Status
-----------                             ----                                    ------
addidas                                 nike                                    StoppedDeallocated
addidas                                 nike5                                   StoppedDeallocated
addidas                                 nike6                                   StoppedDeallocated
ciojas                                  ciojas                                  StoppedDeallocated
clone2nd                                clone2nd                                StoppedDeallocated
DevTestJas                              DevTest                                 StoppedDeallocated
FirsLinux                               FirsLinux                               StoppedDeallocated
forjas                                  forsuhas                                StoppedDeallocated
jaski1vm                                jaski1vm                                StoppedDeallocated
jassql                                  jassql                                  StoppedDeallocated
nike2                                   nike2                                   StoppedDeallocated
nike3                                   nike3                                   StoppedDeallocated
pslabvm011                              pslabvm011                              RoleStateUnknown

PS C:\> get-azurevm

ServiceName                             Name                                    Status
-----------                             ----                                    ------
addidas                                 nike                                    StoppedDeallocated
addidas                                 nike5                                   StoppedDeallocated
addidas                                 nike6                                   StoppedDeallocated
ciojas                                  ciojas                                  StoppedDeallocated
clone2nd                                clone2nd                                StoppedDeallocated
DevTestJas                              DevTest                                 StoppedDeallocated
FirsLinux                               FirsLinux                               StoppedDeallocated
forjas                                  forsuhas                                StoppedDeallocated
jaski1vm                                jaski1vm                                StoppedDeallocated
jassql                                  jassql                                  StoppedDeallocated
nike2                                   nike2                                   StoppedDeallocated
nike3                                   nike3                                   StoppedDeallocated
pslabvm011                              pslabvm011                              Provisioning
```