# Hands on Lab 5 - Create HBase Cluster using PowerShell

## 1. Overview

Creating and managing services in Microsoft Azure can be achieved through various methods. The previous labs concentrated on the Web Portal experience. This lab will introduce PowerShell for managing the service deployment. While not represented in the course, it should be noted that Microsoft also supports a robust Microsoft Azure command line interface.

In this lab we will;

1. Explore a few common PowerShell cmdlets using the PowerShell Integrated Scripting Environment (ISE).
2. Provision a new HBase cluster using PowerShell.

## 2. Download and Configure PowerShell

The Microsft Azure Cmdlets have been installed on the course VM. If you don’t have the Azure Powershell CmdLets download them from here <http://go.microsoft.com/?linkid=9811175&clcid=0x409>

For information about installing and configuring Windows Azure Powershell visit <http://azure.microsoft.com/en-gb/documentation/articles/install-configure-powershell/>.

1. Open Windows PowerShell ISE by clicking the Windows PowerShell ISE icon in the taskbar.
2. Type the following command to download the publishsettingsfile: PowerShell Get-AzurePublishSettingsFile
3. A browser will open. When prompted, download and save the publishing profile to the C:folder.
4. Return to the PowerShell ISE. Type the following command to import the settings:

* Import-AzurePublishSettingsFile C:\CloudDataCamp\<publishsettingsfilename>.publishsettings

1. Subscription information is displayed when the command completes with success.
2. Execute the following command to set up the PowerShell session to the Azure subscription used in class:

* Select-AzureSubscription -SubscriptionName "<subscription name>"

## 3. Introducing Microsoft Azure PowerShell Integration

Microsoft Azure PowerShell has integrated HDInsight PowerShell cmdlets that are used for managing your cluster.  
At the end of this section you will have learned a few useful Windows Azure PowerShell commands. In addition, you will have created a new HDInsight HBase cluster that will be used for multiple labs in the course.

There are two main environments for executing PowerShell scripts: the command line interface (CLI) and the Windows Azure Powershell Integrated Scripting Environment (ISE). Our lab will be executed in the ISE, which is an environment designed for development and ad-hoc script execution.

1. Open Windows PowerShell ISE by clicking the Windows PowerShell ISE icon in the taskbar.
2. **Get-Help** is the most useful command as you begin learning about PowerShell.
3. Open a new PowerShell tab and type the following to return information about the cmdlet **Get-AzureStorageAccount**. The –Full parameter will return all information about the cmdlet, including examples. Click Run Script (F5) from the toolbar to execute the statement.

* Get-Help Get-AzureStorageAccount -Full

Close the PowerShell tab but keep the ISE open.

1. The command **Get-AzureStorageAccount** will return information about the storage accounts in the subscription. Open a new script tab by clicking the New icon in the upper left of the toolbar. Type the following command and click Run Script (F5) in the toolbar. Review the output, which should be all of the active HDInsight clusters on the subscription:

* Get-AzureStorageAccount

1. Update the command by adding the –StorageAccountName parameter as shown below. Replace with the name of the storage account created in Hands On Lab 1. Review the output, which should be information about your the storage account:

* Get-AzureStorageAccount -StorageAccountName "<storage account name>"

1. Close the PowerShell tab (keep the Windows Azure PowerShell ISE open).

## 4. Create an Azure Virtual Network

### Creating VNets in a new subscription

Only create the vnet using this method if you have NO VNets in your current subscription, uploading the xml document described will delete any existing VNets. Use with care!

1. Using your local file explorer navigate to the assets directory
2. Edit file azureVirtualNetwork.netcfg
3. Modify text Location="North Europe" and set a location appropriate for you, save the file.
4. In PowerShell ISE, navigate to the location of the file.
5. Upload the file using the following command, you need to provide an absolute path to the file otherwise it will generate a error

* set-AzureVNetConfig -configurationpath c:\.......\azureVirtualNetwork.netcfg

1. Continue to Step 5

### Creating VNets in a subscription with other VNets

Skip this step if you created a virtual network by uploading the file detailed above.

We will create a Azure Virtual Network to which we will attach a HBaseCluster. Copy the following script to a new tab in PowerShell ISE.

1. Review and copy the following script to a new tab in the PowerShell ISE. A copy of this script is available on the virtual machine under \*\*C:\*.

* $vnetName = "Vnet1"  
  $subnetName = "Subnet-1"  
  $location = "<location>"  
    
  #Get the current azure config  
  $currentVNetConfig = get-AzureVNetConfig  
  [xml]$workingVnetConfig = $currentVNetConfig.XMLConfiguration  
  $virtNetCfg = $workingVnetConfig.GetElementsByTagName("VirtualNetworkSites")  
    
  #Add a new virtual network  
  $newNetwork = $workingVnetConfig.CreateElement("VirtualNetworkSite","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $newNetwork.SetAttribute("name",$vnetName)  
  $newNetwork.SetAttribute("Location",$location)  
  $Network = $virtNetCfg.appendchild($newNetwork)  
    
  #Add an address space  
  $newAddressSpace = $workingVnetConfig.CreateElement("AddressSpace","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $AddressSpace = $Network.appendchild($newAddressSpace)  
  $newAddressPrefix = $workingVnetConfig.CreateElement("AddressPrefix","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $newAddressPrefix.InnerText="10.0.0.0/8"  
  $AddressSpace.appendchild($newAddressPrefix)  
    
  #Add a subnet  
  $newSubnets = $workingVnetConfig.CreateElement("Subnets","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $Subnets = $Network.appendchild($newSubnets)  
  $newSubnet = $workingVnetConfig.CreateElement("Subnet","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $newSubnet.SetAttribute("name",$subnetName)  
  $Subnet = $Subnets.appendchild($newSubnet)  
  $newAddressPrefix = $workingVnetConfig.CreateElement("AddressPrefix","http://schemas.microsoft.com/ServiceHosting/2011/07/NetworkConfiguration")  
  $newAddressPrefix.InnerText="10.0.0.0/11"  
  $Subnet.appendchild($newAddressPrefix)  
    
  #Write to file and use that file  
  $tempFileName = $env:TEMP + "\azurevnetconfig.netcfg"  
  $workingVnetConfig.save($tempFileName)  
    
  set-AzureVNetConfig -configurationpath $tempFileName

1. Substitute the following variables.

* $location

1. Save the changes to the script. Click Run Script (F5) in the toolbar to begin provisioning the virtual network.

## 5. Provision an HDInsight HBase Cluster Using PowerShell

In a previous lab the HDInsight cluster was provisioned using the Management Portal. Most application developers and administrators will create a management script using PowerShell or command line to operationalize the generation of clusters. The following steps will create a new HBase cluster that will be used for multiple labs in the course.

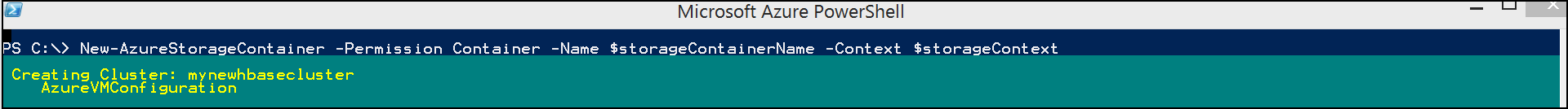
1. Review and copy the following script to a new tab in the PowerShell ISE. A copy of this script is available on the virtual machine under \*\*C:\_CreateHBaseCluster.ps1\*\*.

* $hbaseClusterName = "<HBase cluster name>"  
  $clusterSize = 1  
  $location = "<location>" #i.e. "West US"  
  $storageAccountName = "<storage account name>"  
  $storageContainerName = "<new storage container name>"  
  $vnetName = "Vnet1"  
  $subnetName = "Subnet-1"  
    
  # Check if the storage account exists. If not create the storage account.  
  If (!(Test-AzureName -Service $storageAccountName))   
  {  
   New-AzureStorageAccount -StorageAccountName $storageAccountName -Location $Location   
  }  
    
  # Set the storage account context  
  $storageContext = New-AzureStorageContext -StorageAccountName $storageAccountName -StorageAccountKey (Get-AzureStorageKey $storageAccountName ).Primary  
    
  # Check if the container exists. If not, create the new container.  
  if (!(Get-AzureStorageContainer -Context $storageContext -Name $storageContainerName))  
  {New-AzureStorageContainer -Permission Container -Name $storageContainerName -Context $storageContext}  
    
  # Create the HBase cluster  
  New-AzureHDInsightCluster -Name $hbaseClusterName `  
   -ClusterType HBase `  
   -Version 3.1 `  
   -Location $location `  
   -ClusterSizeInNodes $clusterSize `  
   -DefaultStorageAccountName "$storageAccountName.blob.core.windows.net" `  
   -DefaultStorageAccountKey (Get-AzureStorageKey $storageAccountName ).Primary `  
   -DefaultStorageContainerName $storageContainerName `  
   -VirtualNetworkId (Get-AzureVNetSite $vnetName).Id `  
   -SubnetName $subnetName

1. Substitute the values for the following variables.
   * The HBaseClusterName must be a new unique cluster name.
   * Enter the name of the Virtual Network that was created in the previous step.
   * Enter the subnet name for the Virtual Network created in the previous step.
   * Use the same storage account from the Hands on Lab 1.
   * Input a new container named **hbase** to create a new container for this cluster.

* $hbaseClusterName = "<HBase cluster name>"  
  $location = "<HBaseClusterLocation>"  
  $vnetID = "<AzureVirtualNetworkID>"  
  $subNetName = "<AzureVirtualNetworkSubNetName>"  
  $storageAccountName = "<AzureStorageAccountName>"   
  $storageContainerName = "hbase"

1. Save the changes to the script. Click Run Script (F5) in the toolbar to begin provisioning the cluster.
2. Enter the cluster administrator username and password when prompted.
3. The progress of the cluster creation will be shown in the PowerShell cmd interface, wait until the cluster has been created. Provisioning will complete in 10-20 minutes.

* 

1. Once complete the cluster details will be printed in the output window.

* 