

Processing Big Data with Hadoop in Azure HDInsight

Lab 1B – Using PowerShell with HDInsight

Overview

In this lab, you will use PowerShell to provision an HDInsight cluster. You will then use a PowerShell script to upload source data to Azure storage, run a MapReduce job, and download the results. Finally, you will use PowerShell to delete your cluster and its storage.

What You'll Need

To complete the labs, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Microsoft Windows computer with the following software installed:
 - Microsoft Azure PowerShell
- The lab files for this course

Note: To set up the required environment for the lab, follow the instructions in the **Setup** document for this course. Specifically, you must have signed up for an Azure subscription, installed and configured Azure PowerShell, and imported the publisher settings for your Azure subscription into PowerShell.

When working with cloud services, transient network errors can occasionally cause scripts to fail. If a script fails, and you believe that you have entered all of the required variables correctly; wait a few minutes and run the script again.

Using PowerShell to Provision an HDInsight Cluster

You can use PowerShell to script the provisioning of an HDInsight cluster and its associated Azure storage. The ability to use a PowerShell script can be useful when you need to regularly provision and delete clusters, or when you want to create an automated batch processing solution that can be scheduled for unattended execution.

Use PowerShell to Verify Your Existing Azure Configuration

1. On the desktop, in the taskbar, right-click the PowerShell icon and click **Windows PowerShell ISE**.

2. In the PowerShell command prompt pane, enter the following command to verify the connection to your Azure subscription:

```
Get-AzureSubscription
```

Note: If your Azure subscription details are not displayed, make sure you have installed Azure PowerShell and completed the PowerShell configuration steps in the Setup guide for this course.

3. Enter the following command to information about HDInsight capabilities in your subscription:

```
Get-AzureHDInsightProperties
```

4. If a **ClusterCount** of **1** or more is returned, enter the following command to view details about existing HDInsight clusters in your Azure subscription (if there are none, no information will be displayed):

```
Get-AzureHDInsightCluster
```

5. Think of a name you'd like to use for your HDInsight cluster (for example *myhdcluster*), and then enter the following command (replacing *myhdcluster* with the name you want to use) to determine whether it is already in use:

```
Test-AzureName -Name "myhdcluster" -Service
```

A value of **True** indicates that the name is already in use; in which case repeat this step with different names until you identify an available name (indicated by the value **False**).

6. Think of a name you'd like to use for the storage account used by your HDInsight cluster (for example *myhdstore*), and then enter the following command (replacing *myhdstore* with the name you want to use) to determine whether it is already in use:

```
Test-AzureName -Name "myhdstore" -Storage
```

A value of **True** indicates that the name is already in use; in which case repeat this step with different names until you identify an available name (indicated by the value **False**).

7. Note the available HDInsight cluster and storage names you have identified. You will use them in the next task.

Run a PowerShell Script to Provision an HDInsight Cluster

1. In the PowerShell Interactive Scripting Environment (ISE) tool, if the script pane is not visible, on the **View** menu, select **Show Script Pane**.
2. In the C:\HDILabs\Lab01B folder, open **Provision HDInsight.txt** in Notepad, and then copy and paste the entire contents of the text file into the script pane in the PowerShell ISE. The script should look like this:

```
$storageAccountName = "uniquestoragename"
$location = "West Europe"
$containerName = "containername"
$clusterName = "uniqueclustername"
$clusterNodes = 2
$username = "username"
$password = ConvertTo-SecureString "SecurePa`$`$w0rd" -AsPlainText -
Force

# Create a storage account
```

```

Write-Host "Creating storage account..."
New-AzureStorageAccount -StorageAccountName $storageAccountName -
Location $location

# Create a Blob storage container
Write-Host "Creating container..."
$storageAccountKey = Get-AzureStorageKey $storageAccountName | %{
$.Primary }
$destContext = New-AzureStorageContext -StorageAccountName
$storageAccountName -StorageAccountKey
$storageAccountKey
New-AzureStorageContainer -Name $containerName -Context $destContext

# Create a cluster
Write-Host "Creating HDInsight cluster..."
$credential = New-Object System.Management.Automation.PSCredential
($userName, $password)
New-AzureHDInsightCluster -Name $clusterName -Location $location -
DefaultStorageAccountName
"$storageAccountName.blob.core.windows.net" -DefaultStorageAccountKey
$storageAccountKey -
DefaultStorageContainerName $containerName -ClusterSizeInNodes
$clusterNodes -Credential $credential -Version 3.2
Write-Host "Finished!"

```

3. In the first line of the code, replace *uniquestoragename* with the name you identified for your storage account in the previous task.
4. Optionally, in the second line of the code, change the location from *West Europe* to the Azure datacenter nearest your location (for example, *East US* or *Southeast Asia*) – for a full list of available locations, in the PowerShell command line pane, enter the command **Get-AzureLocation**.
5. In the third line of the code, replace *containername* with a suitable name for the container in your storage account where HDInsight will store its files (for example *hdfiles*) – this must be unique within the storage account, but does not need to be globally unique.
6. In the fourth line of the code, replace *uniqueclustername* with the name you identified for your HDInsight cluster in the previous task.
7. In the sixth line of the code, replace *username* with the user name you want to use when connecting to your HDInsight cluster over HTTP (for example, your first name).
8. In the seventh line of the code, replace *SecurePa`\$`\$w0rd* with the password you want to assign to the HTTP user (note that you must prefix reserved PowerShell characters such as **\$** and **#** with a grave (```) character, so the value *SecurePa`\$`\$w0rd* assigns the password *SecurePa\$\$w0rd*).
9. Examine the rest of the code, and note that it performs the following actions:
 - a. Creates an Azure storage account
 - b. Creates a blob container in the storage account, retrieving the primary storage access key from your account in order to do so.
 - c. Creates an HDInsight version 3.2 cluster with two data nodes.
10. Save the script as **Provision HDInsight.ps1** in the C:\HDILabs\Lab01B folder. Then on the toolbar, click **Run Script**.
11. Wait for the script to finish running – it may take around 15 minutes. You can watch the progress in the PowerShell pane. It will display *Finished!* when complete.
12. Enter the following command to verify that your HDInsight cluster has been created:

```
Get-AzureHDInsightCluster
```

13. Keep the PowerShell ISE open for the next exercise.

Using PowerShell to Run a MapReduce Job

Now that you have provisioned an HDInsight cluster, you can use it to process data. You can use PowerShell to upload the source data you want to process, run a job, and then download the output generated by the job.

View Source Data

1. In the C:\HDILabs\Lab1B\Reviews folder, open **reviews1.txt** in Notepad and note that the file contains some sample text representing product reviews that have been posted on a web site by customers. The files named **reviews2.txt** and **reviews3.txt** include similar data.
2. Close Notepad without saving any changes.

Run a PowerShell Script to Process the Data

1. C:\HDILabs\Lab01B folder, rename **MapReduce.txt** to **MapReduce.ps1** (you may need to modify the *View* options for the folder to see file extensions). Then right-click **MapReduce.ps1** and click **Edit** to open the script file in the PowerShell ISE.
2. Change the values assigned to the **\$clusterName**, **\$storageAccountName**, and **\$containerName** variables to match the names of the HDInsight cluster, storage account, and container you created in the previous exercise.
3. Examine the rest of the script, noting that it performs the following tasks:
 - a. Removes any output left over from previous execution of the job.
 - b. Uploads the contents of the **Reviews** subfolder to a folder named **reviewprocessing** in your Azure blob container.
 - c. Runs the **wordcount** MapReduce code in the **hadoop-mapreduce-examples.jar** executable in the Azure blob store.
 - d. Waits for the job to complete, and displays the job status output.
 - e. Downloads the results file generated by the job to a local folder.
 - f. Uses the **cat** command to display the contents of the downloaded results file.
4. Save the script. Then on the toolbar, click **Run Script**.
5. Observe the information displayed in the PowerShell command line pane as the script runs. When the script finishes, the words and their counts are displayed.
6. Keep the PowerShell ISE open for the next exercise.

Using PowerShell to Delete an HDInsight Cluster

Now that you have finished using the HDInsight cluster, you can delete it to reduce costs.

Run a PowerShell Script to Delete the HDInsight Cluster and its Storage

1. In the C:\HDILabs\Lab01B folder, rename **Delete HDInsight.txt** to **Delete HDInsight.ps1** (you may need to modify the *View* options for the folder to see file extensions). Then right-click **Delete HDInsight.ps1** and click **Edit** to open the script file in the PowerShell ISE.
2. Change the values assigned to the **\$clusterName** and **\$storageAccountName** variables to match the names of the storage account and HDInsight cluster you created in the first exercise of this lab.
3. Examine the rest of the script, noting that it performs the following tasks:
 - a. Deletes the HDInsight cluster
 - b. Deletes the storage account
4. Save the script. Then on the toolbar, click **Run Script**.

14. When the script has finished, enter the following command to verify that your HDInsight cluster is no longer present in your subscription:

```
Get-AzureHDInsightCluster
```

5. Close the PowerShell ISE window.