Machine Learning using Spark:

Module 5, Lesson 2  
Azure Spark Cluster Hands-On Lab

## Overview

In this lab, you will create a Spark cluster in HDInsight. We will then use Jupyter notebook to access Spark and Hive context. To get you familiar with Jupyter, we will run several SQL queries interactively using the Hive context.

## Objectives

In this hands-on lab you will learn how to:

* How to create a SSH public key using PuTTYgen
* Set up a Spark cluster in HDInsight and use the public/private key as credentials
* Launch the Jupyter notebook and launch a PySpark kernel.
* Execute SQL statements using the *%%SQL* magic cell.

## Prerequisites

The following are required to complete this hands-on lab:

* A web browser
* A Microsoft Windows machine on which the [PuTTY](http://www.putty.org/) and PuTTYgen has been installed

Note: The Azure portal is continually improved and changed. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing, but may not match the latest design of portal.

## Exercises

This hands-on lab includes the following exercises:

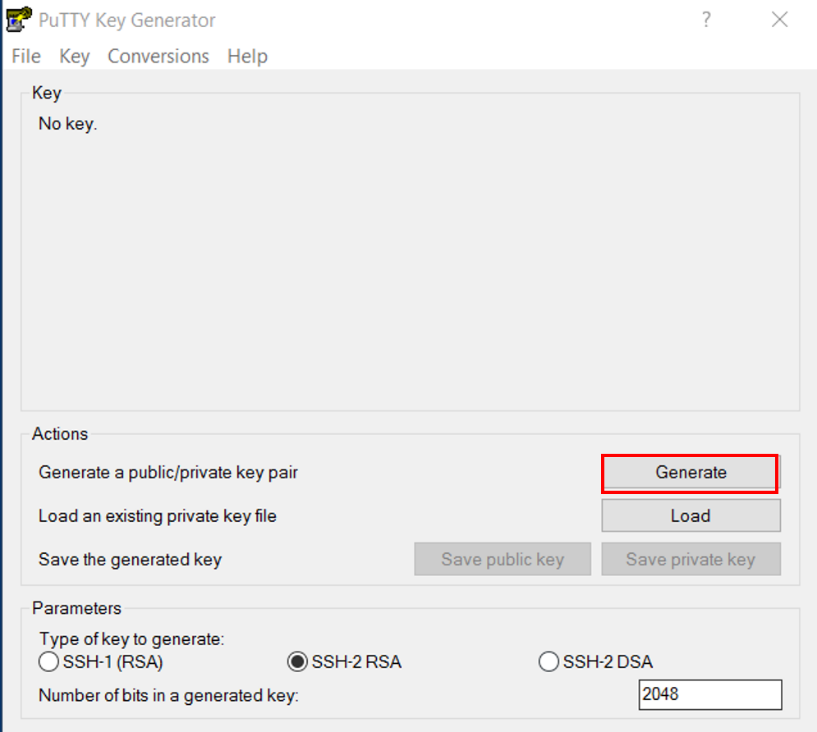
* Exercise 1: Create a public key for SSH
* Exercise 2: Provisioning and configuring a Spark cluster
* Exercise 3: Configure PuTTY to use the private key
* Exercise 4: Run Spark SQL queries from Jupyter notebook

## Exercise 1: Create a Public key for SSH

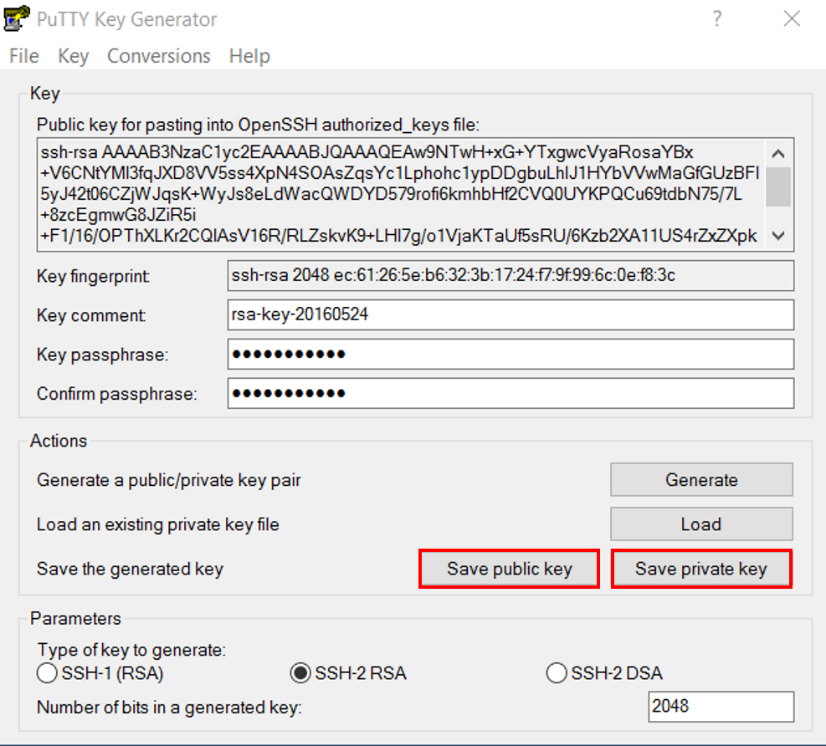
In this exercise, students will learn how create a public key/private key pair using PuttyGen. Once created, the private key can be used to enter a site that was “locked” using the public key.

1. Generate a public/private key pair

Start PuTTYgen and click on **Generate** to begin creating a new key pair.



Move the mouse randomly around the designated area to help PuTTYgen generate a key pair. Once finished, the user will get a screen similar to the one below. Putty will recommend that you enter a Key passphrase for additional security. Enter a passphrase if you so choose. Click on **Save public key** and

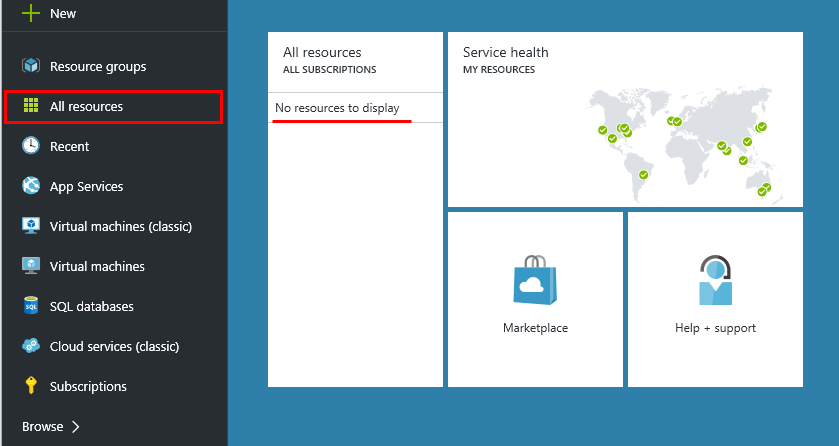


**Save private key** to save the key to a secure location on your Windows computer. Putty will recommend that you enter a Key passphrase for additional security. Enter a passphrase if you so choose. The private key is a \*.ppk file that users will use to access the HDInsight cluster created with the matching public key. The public key is also displayed at the top of the screen as sequence of random text. Users can alternatively cut and paste this to a text file to save the public key.

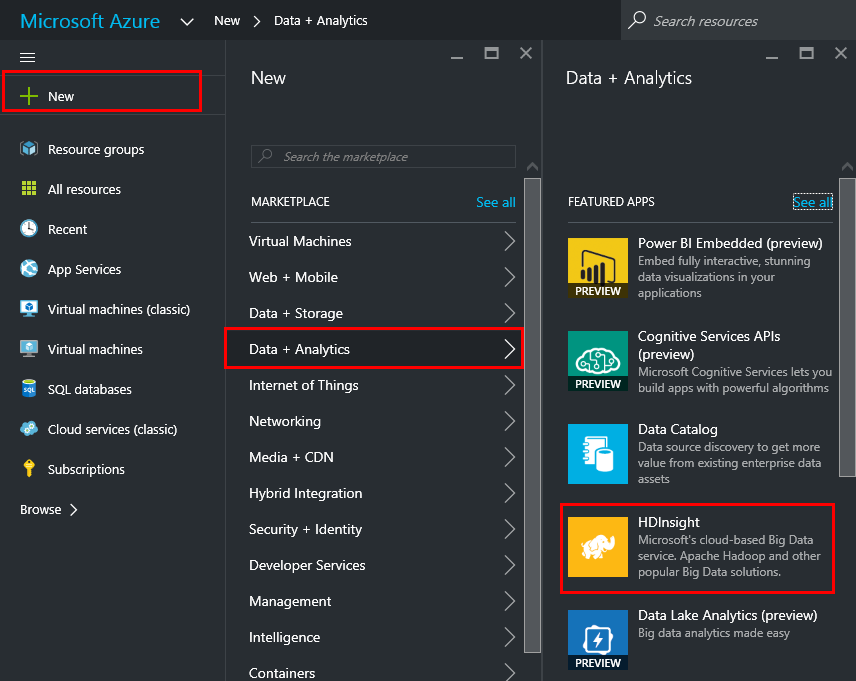
## Exercise 2: Provisioning and configuring a Spark Cluster

The first task you have to perform is to provision a Spark cluster in HDInsight.

1. In a web browser, navigate to <http://portal.azure.com>. Sign into the portal using your subscription.
2. In the Azure portal, click **All resources**, and verify that there are no existing Spark HDInsight clusters in your subscription.



1. In the Hub menu (on the left edge), click **New** (indicated by a +), and in the “**Data + Analytics**”, click HDInsight. Then use New **HDInsight** Cluster section to create a new cluster.



1. Create a new cluster with configuration value. The Cluster Name is unique name for cluster. You have to remember this name. Then, click the **Select cluster type** for more detailed information. Select the value below as:

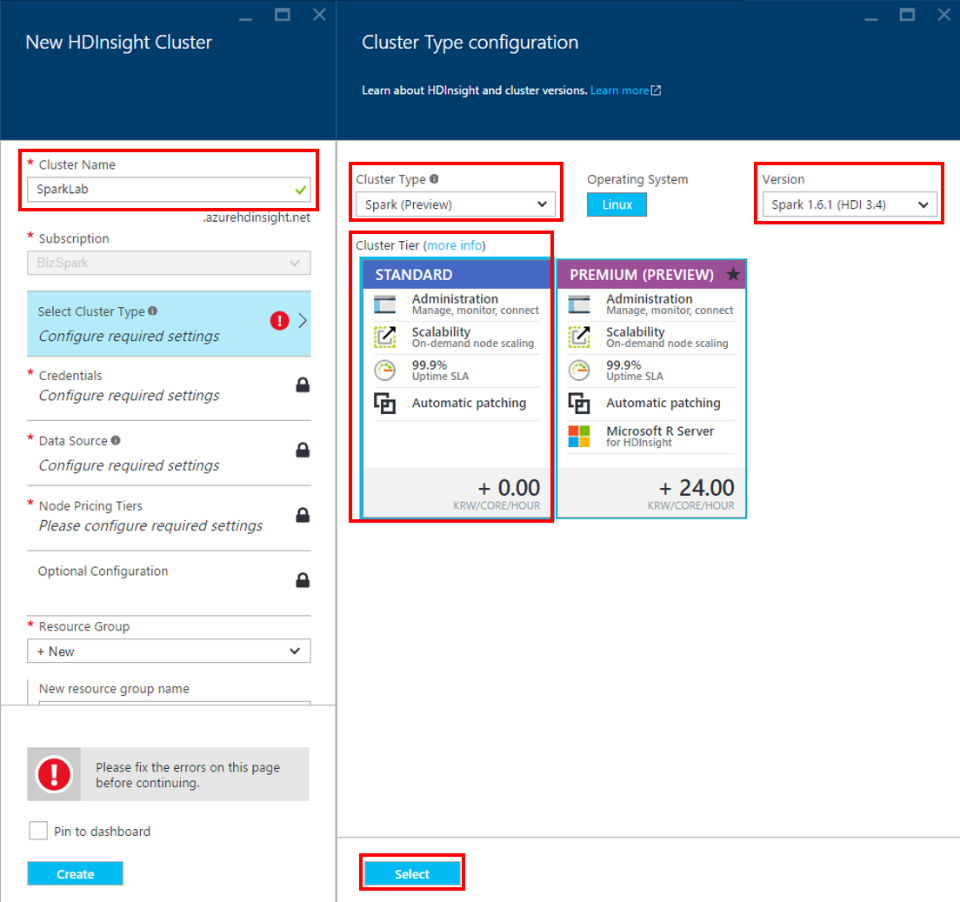
Cluster type: Spark

Cluster Operating System: Linux

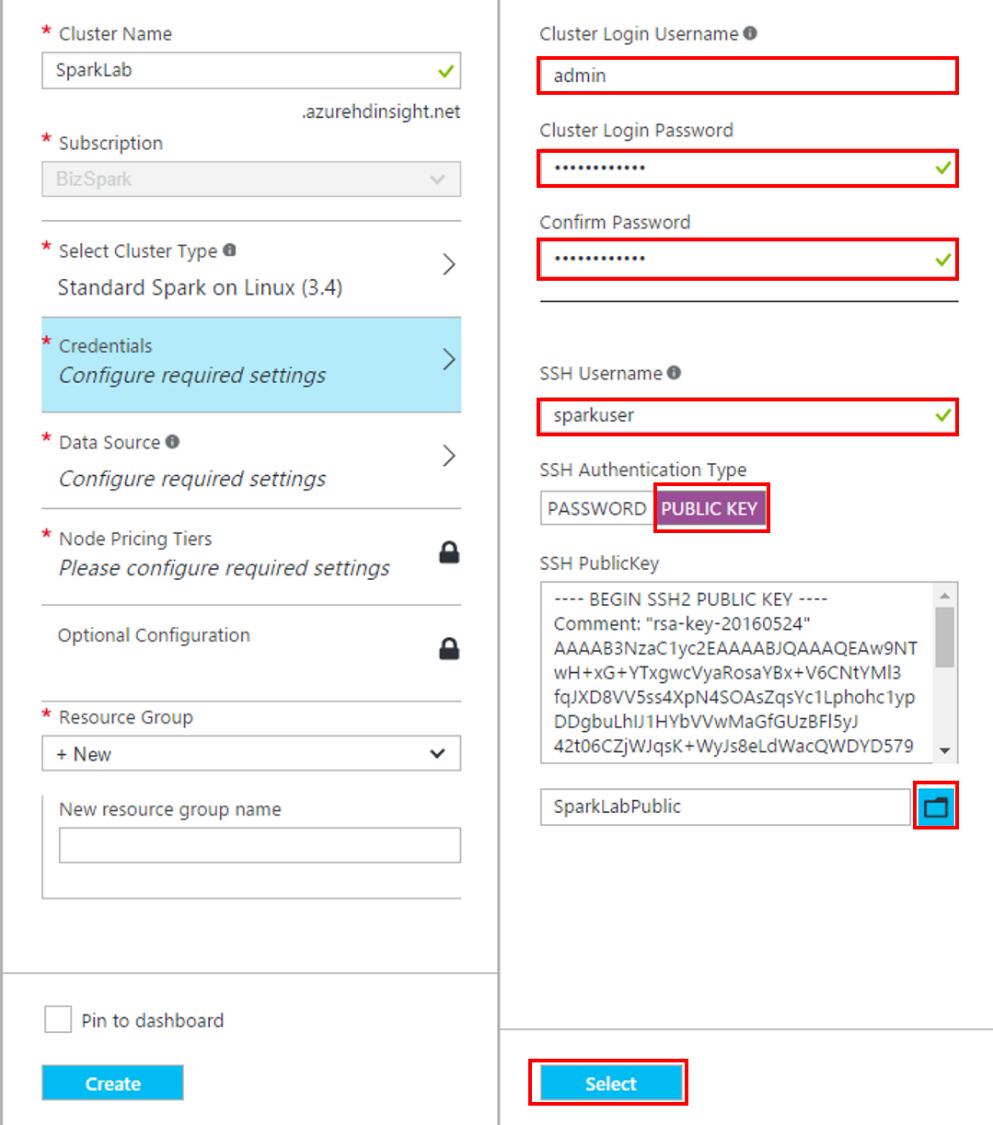
Version: Choose the latest version of Hadoop. Ex) Spark 1.6.1 (HDI 3.4)

Cluster Tier: Standard

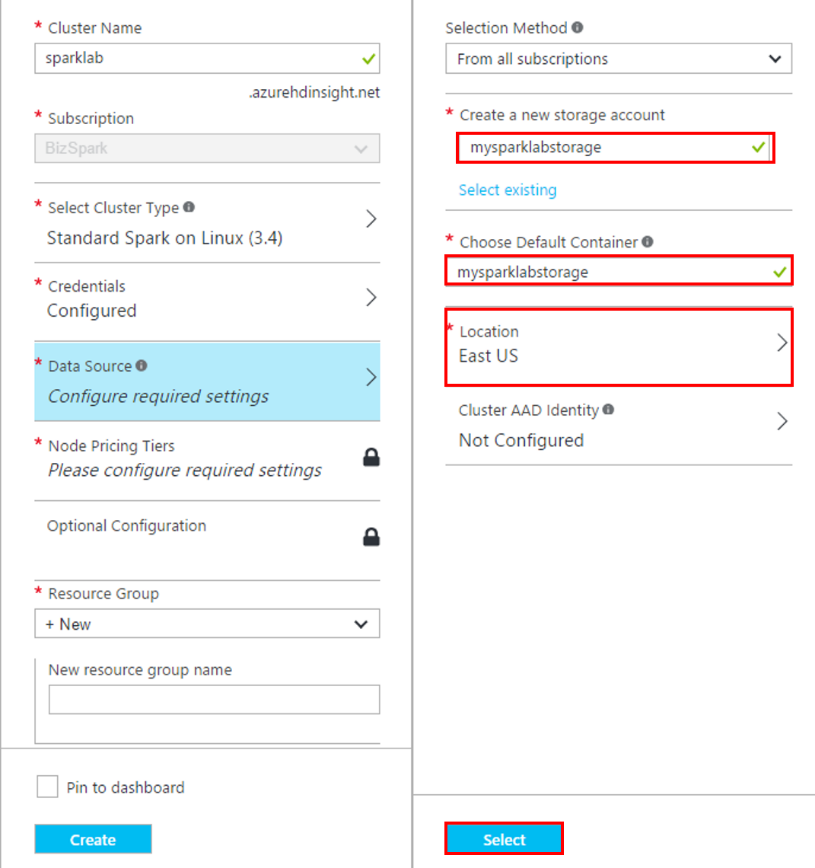
And, click the **Select** button.



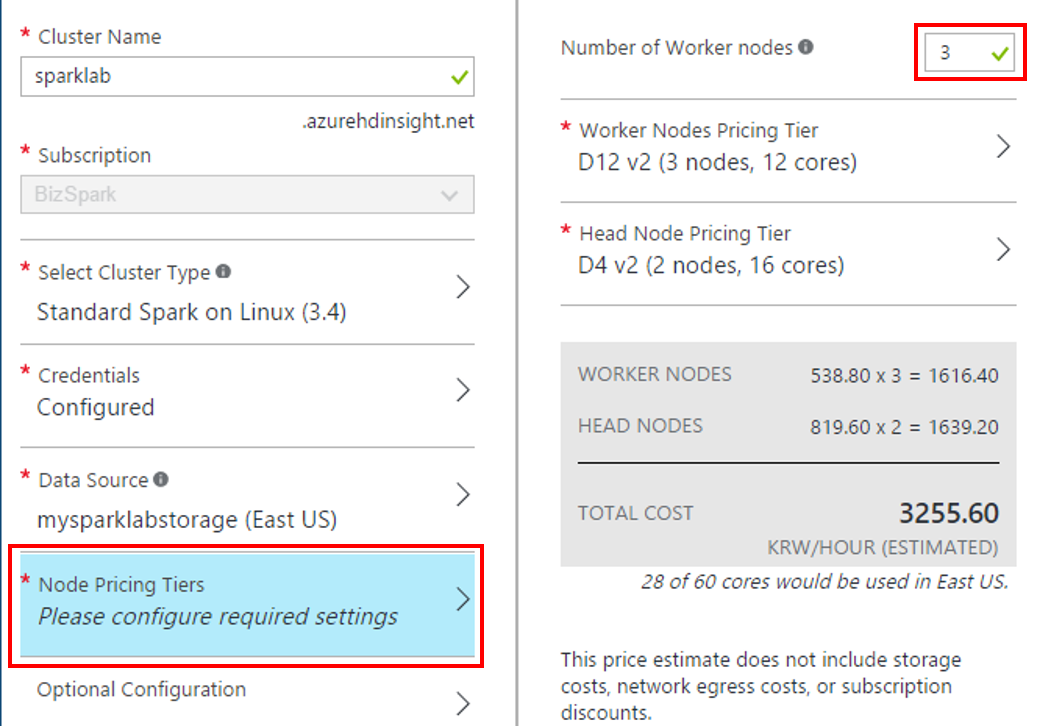
1. Click the **Credentials** section. Enter a user name of your choice. (default is “admin”). Enter and confirm a strong password. The password must be at least 10 characters in length and must contain at least one digit, one non-alphanumeric, and one upper or lower case letter. SSH Username is different from Username. For this cluster, we are going to using the Public key that we created in Exercise 1 to authenticate. Instead of PASSWORD, click on **PUBLIC KEY** and click on the folder icon on the bottom right. Navigate to the folder where the public/private key pair was saved and choose the public key. This will upload the public key that we created earlier. Alternatively, users can also cut and paste the public key saved to a text file in Exercise 1 into the window under **SSH Public Key.** Once finished, click the **Select** button.



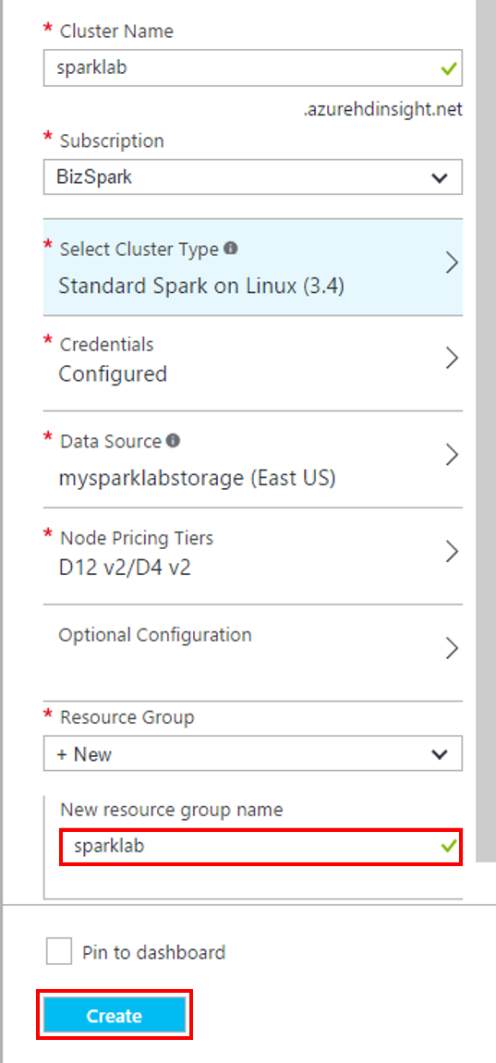
1. Click the **Data Source** section. Enter a unique name consisting of lower-case letters and numbers only in storage account and choose default container. Select any available region in Location field. And click the **Select** button.



1. In Node Pricing Tiers, enter 3 for the **Number of Worker nodes**. For Worker nodes and Head Node, View all and choose the smallest available size. And, click the **Select** button.



1. We will not set anything in the **Optional Configuration** section. To complete creating the cluster, leave the pull down menu under **Resource Group** to **+New** and provide a unique name in **New resource group name** for Resource Group. Finally click the **Create** button to provision your Spark cluster in HDInsight.



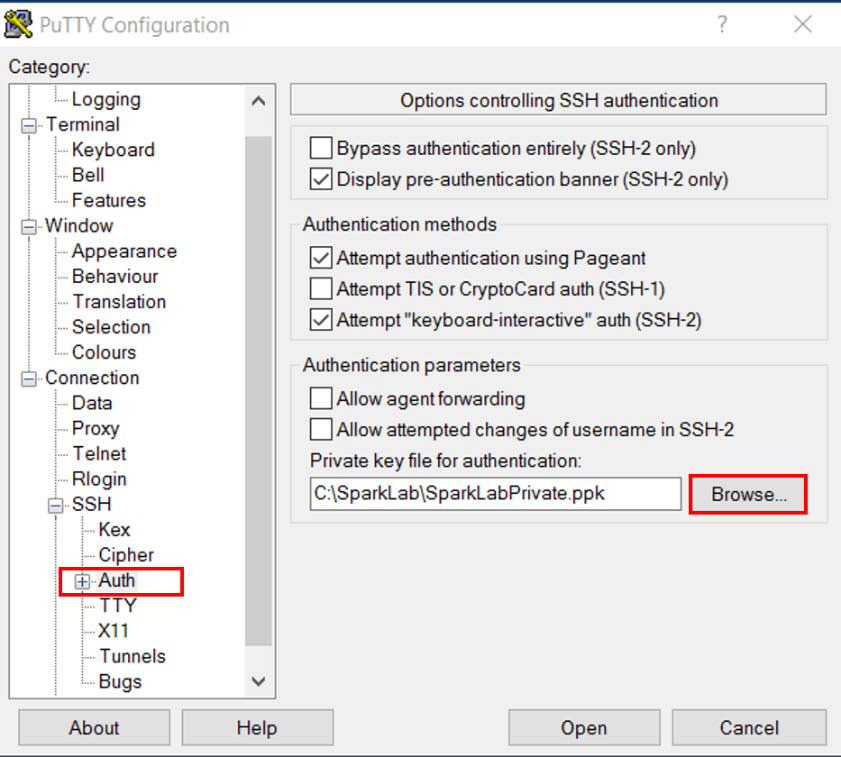
1. It may take 10 to 20 minutes for the cluster to be provisioned. As soon as an HDInsight cluster is running, the credit in your subscription will start to be charged. At the end of the lab, delete your cluster in order to avoid using your credit unnecessarily.

## Exercise 3: Configure PuTTY to use the private key

In this exercise, students will learn how use the private key generated in Exercise 1 to enter the newly created Spark Cluster from Exercise 2.

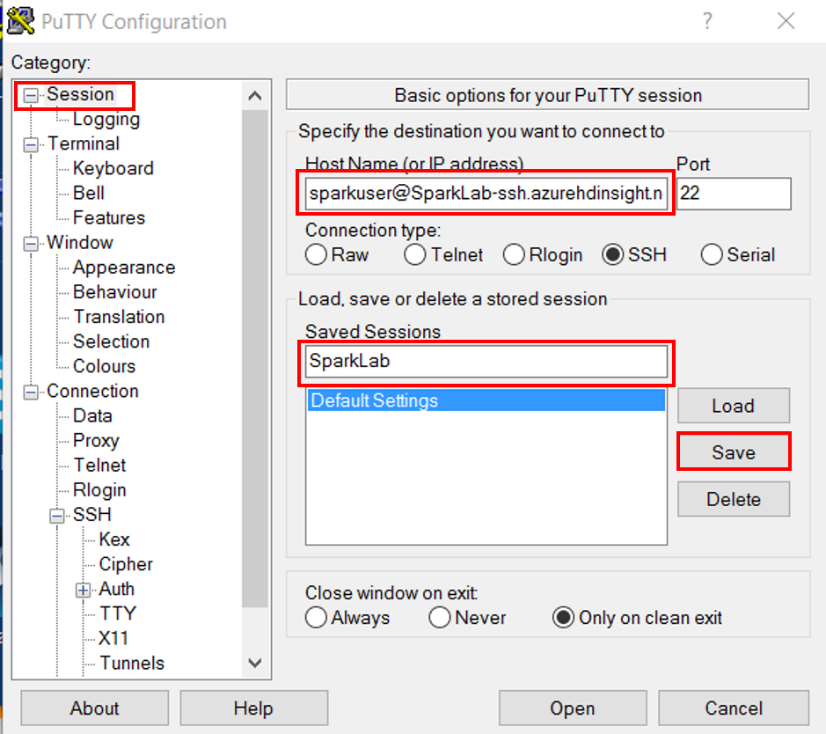
1. Configure PuTTY to use private key

Start PuTTY and select **Auth** from the **Category** section on the left. Select the saved private key by clicking on **Browse** and navigating to the Windows directory where the user saved the public/private key pair from Exercise 1. Do NOT click Open yet.



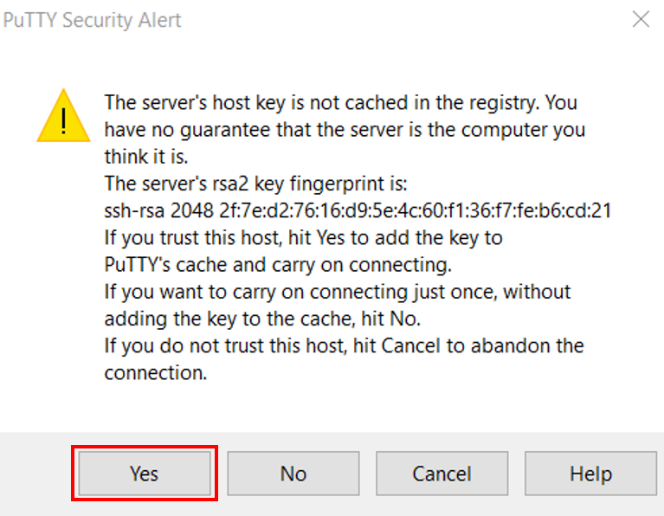
1. Configure login information and save session

Select **Session** from the **Category** on the left and enter the **HostName**. The format for this entry is *ssh-username@ClusterName***-ssh.azurehdinsight.net**. The SSH user name was configured in step 5 of Exercise 2 (configured as sparkuser in this example). The ClusterName was configured in step 4 of Exercise 2(configured as SparkLab in this example). In this example, this would be sparkuser@SparkLab-ssh.azurehdinsight.net. In **SavedSession**, provide a name for the session and click **Save**.



1. Login to Spark cluster using private key

Now click on **Open** to begin the secure shell session. When you first login, Putty will display an alert that the host key is not cached in the registry. Click **Yes** to confirm that you trust the host.



If the user chose to enter a passphrase when saving the private key in Exercise 1, the user will now be prompted to enter that passphrase to complete the login process.

## Exercise 4: Run Spark SQL queries from Jupyter notebook

In this exercise, students will use Jupyter notebook to run Spark SQL queries against the Spark cluster. HDInsight Spark clusters provide two kernels that can be used with the Jupyter notebook. They are PySpark for application written in Python and Spark for applications written in Scala. Students will use the PySpark kernel to run the rest of the exercise.

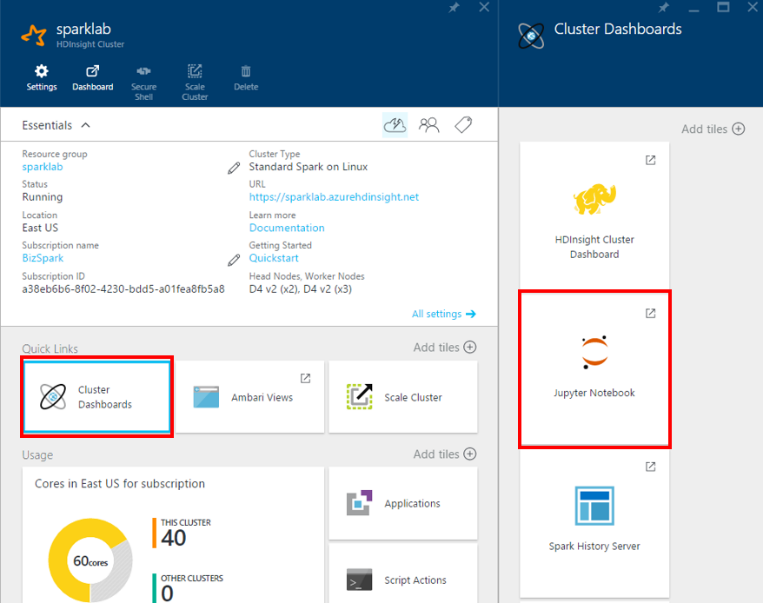
There are a number of advantages to using a PySpark kernel.

* The user does not need to worry about the spark or hive context. These are automatically set for the user.
* Users can use cell magics such as *%%sql*, to directly run SQL or Hive queries without any need for preceding setup code snippets
* The output of the SQL and Hive queries are automatically visualized on the notebook

1. Create Jupyter notebook with PySpark kernel

From the Azure Portal, navigate to the Spark cluster that the student just created. It can be found under **Browse All>HDInsight Clusters**.

From QuickLinks, select **Cluster Dashboards** and then select **Jupyter Notebook**. If prompted enter the admin credentials for the cluster. This will open up a browser window loaded with Jupyter notebook



Create **New** and select a PySpark kernel.

You may also reach the Jupyter Notebook for your cluster by opening the following URL in your browser. https://CLUSTERNAME.azurehdinsight.net/jupyter (Replace CLUSTERNAME with the name of student cluster)

1. Name the PySpark kernel with a user friendly name

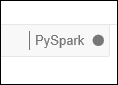


1. Import SQL types

The PySpark kernel includes the context for Spark and Hive. The student does not have to create these contexts explicitly.

To begin the exercise, import the types required for this scenario. To do so, paste the following code snippet in a cell and press SHIFT + ENTER. Pressing SHIFT + ENTER executes the entry in the current cell and moves the cursor to the next cell.

from pyspark.sql.types import \*

When you execute an entry in the cell, the PySpark kernel will indicate that it is busy executing this statement. This will be shown on the right corner of the web browser. It will show a solid circle next to PySpark and will change back to a hollow circle when finished executing the cell statement.

1. Load sample data into a temporary table.

When a Spark cluster is created in HDInsight, a sample data file – hvac.csv – is copied to the associated Azure blob storage account under wasb:///HdiSamples/HdiSamples/SensorSampleData/hvac/HVAC.csv. In an empty cell, paste the following code example and press SHIFT + ENTER.

# Load the data

hvacText = sc.textFile("wasb:///HdiSamples/HdiSamples/SensorSampleData/hvac/HVAC.csv")

# Create the schema

hvacSchema = StructType([StructField("date", StringType(), False),StructField("time", StringType(), False),StructField("targettemp", IntegerType(), False),StructField("actualtemp", IntegerType(), False),StructField("buildingID", StringType(), False)])

# Parse the data in hvacText

hvac = hvacText.map(lambda s: s.split(",")).filter(lambda s: s[0] != "Date").map(lambda s:(str(s[0]), str(s[1]), int(s[2]), int(s[3]), str(s[6]) ))

# Create a data frame

hvacdf = sqlContext.createDataFrame(hvac,hvacSchema)

# Register the data frame as a table to run queries against

hvacdf.registerTempTable("hvac")

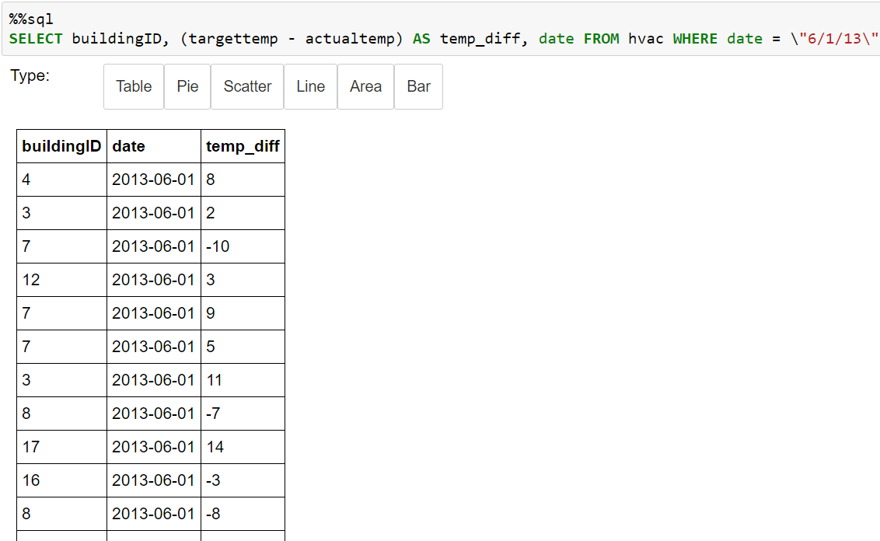
1. Run a SQL query using *%%SQL*

The PySpark kernel provides the *$$SQL* magic interface. This allows users to run a SQL query on the temporary table hvac that the student just created. In an empty cell, paste the following code example and press SHIFT + ENTER.

%%sql

SELECT buildingID, (targettemp - actualtemp) AS temp\_diff, date FROM hvac WHERE date = \"6/1/13\"

Once the code is fully executed, the data will be presented in tabular format by default. Users can also see the results in other visualization formats by selecting one of the other output types.



Try visualizing the data is other formats. For example, here is a sample output in Bar format. The **Bar** type is first selected. This presents a set of parameters for this visualization type. Change the X axix to buildingID, Y to temp\_diff, and the FUNC to Avg. This will visualize the average temperature difference for each of the buildings associated with that buildingID.



1. Shutdown the Jupyter notebook.

After trying the various visualization types, shut down the notebook to release any resources. Click the **File** menu on the notebook and **Close and Halt.**

1. Delete the Spark Cluster

HDInsight clusters are billed on a per minute basis. In order to avoid costly charges, delete your cluster after you have finished using it.

## Summary

In this hands-on lab, you learned how to:

* Create a public/private key pair to use on SSH
* Create a Spark Cluster on HDInsight
* Launch a Jupyter notebook and run SQL queries on it