

Azure Cloud Concepts and Tools

Introduction to Cloud Analytics

Template Version: 2.0

Introduction

During this lab, we will explore the essential components of the Azure Cloud concepts and tools.

The purpose of this lab is to familiarize with the available Azure Tools like Azure Portals, Azure PowerShell, Azure Storage Explorers and so on.

Estimated Time

90 minutes

Objectives

Upon completing this lab, you will be able to:

* Explore Azure Management Portal
* Experience different Azure Tools

Logon Information

Use the following credentials to login into virtual environment

* Username: Administrator
* Password: Password1

Contents

[ 1](#_Toc497687929)

[Azure Cloud Concepts and Tools 1](#_Toc497687930)

[Lab Title 1: Explore Azure Cloud Tools 3](#_Toc497687931)

[Exercise 1: Explore Azure Management Portal 3](#_Toc497687932)

[Exercise 2: Explore other Azure Tools 6](#_Toc497687933)

Lab Title 1: Explore Azure Cloud Tools

Exercise 1: Explore Azure Management Portal

The Azure Management Portal the management environment for creating and managing your Azure assets. In this exercise, you will connect to the Azure Management Portal, explore the classic and new portal user experience, create a storage account and explore the different Hadoop deployment and configuration options.

Tasks

1. Task: Create Storage Account
   1. Open Internet Explorer and navigate to <https://portal.azure.com/>. This portal is the Azure portal.
   2. Login with your subscriptions credential.
   3. By default, you should see the full portal interface.
   4. On the left menu click new. In the “New” blade select Storage, in the “Storage” blade, click Storage Account.
   5. Choose a name for the storage account. (the name must be unique).
   6. Leave Deployment Model, Account Kind, Performance, Storage Service Encryption, Secure transfer required as default for the purpose of the lab.
   7. Select Replication option. For the purpose of the lab, you can use locally redundant.
   8. Choose to create a new resource group or use an existing one.

Resource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization.

* 1. Choose a location.
  2. Click “Create” button to create storage account.
  3. Once created, in the left menu click on all resources and in the search box type the name of the storage account you have just created.
  4. Explore the different menus, dashboard, configure containers and so on.
  5. Under BLOB SERVICE, click on Browse blobs. On the newly opened blade, click on +Container on the top to create a new container. Give it a name and click OK.
  6. Wait for the container to get created. Once it shows up, click on it. On the container blade, click on Container properties button on top.
  7. Copy the URL to notepad. Close the container blade.
  8. Click on the Access Keys on the settings part of the menu for your storage account. You will see the options to copy or regenerate both the primary and secondary access keys. Copy key1 and save it to notepad.
  9. Also copy the storage account name from the same blade and save it to notepad.
  10. Close the storage account blade.

1. Task: Explore Hadoop Cluster Configuration on Azure Management Portal

In this task, you will explore the option for deploying and configuring Hadoop clusters. Note you will only explore the options without finalizing the create/deploy cluster. Later another lesson, you will deploy and configure the HDINSIGHT cluster using Azure PowerShell.

1. Launch Internet Explorer and navigate to: <https://portal.azure.com/>
2. Click on **+ New**. In the next blade, Data + Analytics, then HDInsight.
3. For the purpose of this lab, you will use Quick create.
4. Choose a unique name for the cluster. Click on “Cluster type”, configure as below and choose “Select”.
   1. Cluster type: Spark
   2. Operating system: Linux
   3. Version: Spark 2.1.0 (HDI 3.6)
   4. Cluster tier: Standard
5. Choose a user name and password for the cluster.
6. Choose a username for Secure Shell (SSH) and use the same password as cluster login.
7. Choose a name for a new resource group or use an existing one.
8. Choose a region and click next.
9. Click on Select a Storage account and select the storage account you created in Task 1.
10. Leave everything else as default for the purpose of this lab and click Next.
11. Once the validation is completed, click on Download template and parameters button in the bottom. This will open a Template blade.
12. Explore the Parameters and Resources. You can alternately click on Template and Parameters tabs on top.
13. Click on Download to download the zip file. Extract this zip file. We will use it in a later exercise.
14. Switch back to the Azure portal. Close the Template blade. This will take you back to the Cluster summary blade.
15. Close here without launching the deployment. Choose the X button in the HDInsight blade and click OK to discard the changes.
16. Task: Explore Market Place on Azure Management Portal
    1. Open Internet Explorer and navigate to <https://portal.azure.com/>.
    2. Click “+New” and choose “See all” at the first line of Marketplace. Type Hortonworks and choose Hortonworks Data Platform Standard.
    3. The Deployment of Hortonworks Data Platform Standard is IaaS and in fact can be accessed from the Virtual Machine Menu of the Marketplace. Once explored the different options you can close the window of the Portal.

Exercise 2: Explore other Azure Tools

The purpose of this lab is to Familiarize with the available Azure Tools like Azure PowerShell, Azure Storage Explorers and so on.

Tasks

1. Task: Explore Azure Powershell
2. Start Windows PowerShell with Elevated Privileges (Run as Administrator)
3. Install and configure Azure PowerShell as described in the documentation:  
   <https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps>
4. Type **Add-AzureRmAccount**. On the Sign in to Microsoft Azure window, type in the email associated to your Azure Subscription and follow the authentication process.
5. If you are using your corporate account, additional Sign in authentication may be required (phone, smart card). Chose the preferred method and continue with the authentication.
6. Once authenticated successfully the prompt will be returned with additional information, for example the default Subscription selected if you have multiple subscriptions and how to select a different one.
7. Type **Get-AzureRmStorageAccount**. Note the name of the account you would like to use.
8. Type **Get-AzureRmStorageKey**. When requested type in the Resource Group and Account Name obtained in the previous step. Copy and save the storage key.
9. Task: Explore Microsoft Azure Storage Explorer
10. Open your browser and navigate to <http://storageexplorer.com/> .
11. Download the Microsoft Azure Storage Explorer.
12. Execute the setup file to install the tool and follow the instructions
13. Open Microsoft Azure Storage Explorer. Right click Storage Accounts and choose Connect to Azure Storage.
14. Choose Use a storage account name and key and click Next
15. Type your account name and primary key from Access Keys (if you need to remember how to get it, review Task 1 of Exercise 1.) and click Next.
16. Click Connect.
17. Explore the stored data under blobs, file shares, queues or tables if any.
18. Navigate to Blob Containers. Then click on the container you created in the first exercise.
19. In the center pane, click on Upload and notice that you see two options – Upload Folder and Upload Files.
20. Click on Upload Folder and upload any sample folder from your local machine. Alternately you can upload the Flights folder by downloading and extracting it from the workshop repo on github.
21. Task: Explore AZCopy
22. Download AZCopy by navigating to <http://aka.ms/downloadazcopy> in your web browser.
23. Execute the setup and accept the defaults.
24. By default, AzCopy is installed to %ProgramFiles(x86)%\Microsoft SDKs\Azure\AzCopy.
25. Open command prompt as an Administrator and navigate to the path above.
26. We will now use AzCopy to upload the Flights data to the container created in exercise 1. If you had already uploaded the data to the container, you can delete it from either the Azure portal or by using the Azure Storage Explorer.
27. In CMD, type the following command with the appropriate values. Some of these like the storage key and path (i.e. URL) will come from the values you stored in notepad in exercise1.

AzCopy /Source:”local path of the Flights folder” /Dest:azure storage account URL /DestKey:storageaccountkey

1. Once the copy finishes, use either Azure Storage Explorer or the Azure Portal and review the files that were copied to the container. Confirm that the folder 201210 and the file inside it were not copied to the container. This is because we did not add the /S parameter. When you add the /S parameter, AZCopy recursively upload the files. When downloading, it does the same.
2. Rerun the command by adding /S at the end and observe the outcome.

AzCopy /Source:”local path of the Flights folder” /Dest:azure storage account URL /DestKey:storageaccountkey /S

1. Now that the files are copied to the container, let’s try to download a specific file to the local machine. To do this, you need to use the /S parameter along with the /Pattern parameter and specify the file name that you want to download. This will force AZCopy to recursively search for that file and download it to the local machine at the path specified. Also note that we now specify the /SourceKey since the blob storage is our source.

AzCopy /Source:Azure storage account URL /Dest:"local path where you want to download the file" /SourceKey:storage account key /S /Pattern:"201212.csv"

1. Task: Explore ARM Templates
2. Navigate to the extracted templates folder from the Hadoop Cluster Configuration exercise.
3. The folder will have two json files (template.json and parameter.json) and four deployment files (deploy.ps1, deploy.sh, deployer.rb and DeploymentHelper.cs).
4. Review the json and PowerShell files. We will now use these files to deploy the HDInsight cluster to azure.
5. The parameters.json file already has the values for al the parameters except the clusterLoginPassword and the sshPassword. We can either edit this file and add the password as plaintext or we can provide those when we run the PowerShell script. We will choose the more secure option.
6. The template.json file has the parameters defined and the resources that will be created or used.
7. Right click deploy.ps1 and click on Edit. This should open the file in PowerShell ISE.
8. Add the following parameter to the list of parameters at the beginning of the script. You can add two parameters if you want separate password for the cluster admin and ssh user. Here we will use the same password for both.

[Parameter(Mandatory=$True)]

[System.Security.SecureString]

$password

1. At the very end of the script, update the New-AzureRmResourceGroupDeployment cmdlet by adding the following parameters in both the if and else conditions.

-clusterLoginPassword $password -sshPassword $password

1. Save the file.
2. Now either from the terminal window in PowerShell ISE or from a separate PowerShell window, navigate to the folder where the template scripts reside and launch the PowerShell script as follows.

.\deploy.ps1

1. The script will ask for the subscription ID of your azure subscription. You can retrieve this by going to the Azure portal, navigating to Subscriptions in the left-most blade, clicking on your subscription and then copying the Subscription ID.
2. Provide the subscription ID and other details such as the name of the resource group where you want the cluster to be deployed, name for this deployment (you can give any name), the password and optionally the region where you want the resource group created.
3. The script will also launch a prompt for you to authenticate your Azure credentials before it starts creating the resources.
4. Once all the details are entered successfully, you will see the message Starting deployment… which would indicate that the deployment has started.
5. The deployment should last about 15 minutes. However, if you switch back to the Azure portal and navigate to the resource group name you had provided, you should be able to see an HDInsight resource right away.
6. Once the deployment finishes successfully, you will see the summary of the deployment in the PowerShell window. Review it and close the PowerShell window.
7. Task: Explore Azure Automation Runbook
8. We may not want to keep the cluster running for long since it can burn through Azure credits very quickly. Now, instead of navigating to the Azure portal and deleting the resource group manually, we will try to use the Azure Automation service and deploy a runbook that will automatically delete the resource group at a given time.
9. In the Azure portal, click +, type Automation and press Enter.
10. Select Automation -> Create.
11. Provide the automation name, select the subscription.
12. Under Resource Group, click on Create new and provide a name for the new resource group. Select the Azure region where you want to create this.
13. For Create Azure Run As account, select Yes. This allows us to use Run As account to authenticate to Azure and run jobs automatically.
14. Click on Create.
15. Once the automation account is ready to use, navigate to it.
16. Under PROCESS AUTOMATION, click on Runbooks.
17. Click on +Add a runbook.
18. Select Quick Create – Create a new runbook.
19. Enter a name for the runbook.
20. Under Runbook type, select PowerShell **Workflow**. A workflow allows us to simultaneously perform an action against multiple devices and automatically recover from failures.
21. Click on Create.
22. Once the runbook is created, it should open in the Edit window. If it doesn’t, click on Edit on top.
23. Replace the code inside the curly braces { } with the following code. Replace HDIResourceGroupName with the name of the resource group where we deployed the HDInsight cluster in the earlier task.

$Conn = Get-AutomationConnection -Name AzureRunAsConnection

Add-AzureRMAccount -ServicePrincipal -Tenant $Conn.TenantID -ApplicationId $Conn.ApplicationID -CertificateThumbprint $Conn.CertificateThumbprint

Remove-AzureRMResourceGroup -Name "HDIResourceGroupName" -Force

1. The script is using the AzureRunAsConnection credential that we had created during the runbook creation and using it to authenticate to Azure and run the Remove-AzureRMResourceGroup cmdlet.
2. Click on Save.
3. Click on Publish to publish the runbook. Click Yes if/when prompted.
4. When you are back on the Runbook blade, click on Schedule -> Link a schedule to your runbook -> Create a new schedule.
5. Give a name to the schedule. Under Starts, select the date time and timezone to be 6 minutes ahead of your local time. (You will receive an error if the start time is less than 5 minutes ahead of your current time. So to be on a safer side, we go with 6).
6. Leave Recurrence to Once.
7. Click Create. Then click OK.
8. Once the schedule is successfully created, go back to the Runbook blade and click on Jobs.
9. Wait until the start time you had provided and click on Refresh.
10. You should see a message that the job has completed successfully. This would indicate that the resource group has been deleted. Confirm this by clicking on Resource groups in the left-most blade and checking if the HDI resource group still exists.