# Hadoop on Azure Virtual Machines

## Cloudera CDH Edition

This framework will automate many of the steps for creating a Hadoop cluster on Windows Azure Virtual Machines. This framework is designed to deploy Cloudera CDH on Linux using Cloudera Manager.

This framework will automate the generation of the Azure based infrastructure, and walk through the manual steps to configure the environment. The framework is a blend of Azure CLI, bash scripts coupled with manual configuration.

The following instructions were tested on Ubuntu 12.04 LTS.

This document will be used by those who prefer using Linux or Mac for deploying CDH on Windows Azure Linux VMs.

# Requirements

* Windows Azure Cross Platform Command Line Tools: <http://www.windowsazure.com/en-us/documentation/articles/xplat-cli/>
* SSH client.
* Windows Azure Subscription. Steps below assist with creating the storage account and containers.

# Overview

1. Update the cdhsetup.sh file with information about your CDH setup
2. Execute createmgmtnode.sh
   1. Create the storage account
   2. Create the Affinity Group
   3. Create the Virtual Network
   4. Create the Management Node
3. Execute createmasterimage.sh. This step is only necessary if you are creating a new customer image for your nodes.
   1. Create the Master Node
4. Manually configure the Management
   1. Attach disk
   2. Set root passwords
   3. Set up passwordless SSH between the Management Node and the Master Node
   4. Set various server configurations to meet CDH requirements
   5. Update host files
5. Manually configure the Master Node
   1. Attach disk (Management Node only)
   2. Set root passwords
   3. Set up passwordless SSH between the Management Node and the Master Node
   4. Set various server configurations to meet CDH requirements
   5. Update host files
6. Prepare the Master Node for provisioning
   1. Update waagent.conf (Master Node only)
   2. Run waagent –deprovision (Master Node only)
7. Create the Windows Azure Image
   1. Stop the Master Node
   2. Capture an image
8. Update cdhsetup.sh nodeImageName with the name of the captured image.
9. Execute createclusternode.sh
   1. Creates multiple Windows Azure Virtual Machines using the Master Node image
   2. Creates the script mountdrive.sh which mounts data drives on each node in the cluster
   3. Creates the script updatehosts.sh which updates /etc/hosts file on each node in the cluster
10. Run the script updatehosts.sh to update /etc/hosts
11. Run the script mountdrive.sh
12. Install Cloudera Manager on Management Node
13. Install CDH using Cloudera Manager



# Installing CDH on Windows Azure Virtual Machines

## Preparation

### Windows Azure Cross Platform Command Line Tools

This section provides steps for setting up your development PC to leverage Windows Azure command line tools for deployments.

#### MAC OSX Mavericks

For Mac you need to install brew. It can be installed by running the command shown below from the terminal window. Detailed instructions are here at <http://brew.sh/>

Install node by executing brew install node

Install Azure CLI by executing npm install –g azure-cli

Test that Azure CLI was installed by executing azure –v

You should see version 0.7.4 or greater

ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"

brew update

brew doctor

brew install node

npm install -g azure-cli

azure –v

brew install jq

Jq is a light weight command line json processor available for many platform.

#### Ubuntu

Here are the instructions for install Azure CLI on Ubuntu. For a different flavor of Linux, some of the set up commands will need the corresponding changes.

## Install node.js

|  |
| --- |
| sudo apt-get install -y python-software-properties python g++ make  sudo add-apt-repository ppa:chris-lea/node.js  sudo apt-get update  sudo apt-get install nodejs |

For reference, see

<http://askubuntu.com/questions/49390/how-do-i-install-the-latest-version-of-node-js>

[Installing-Node.js-via-package-manager](https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager).

## Install CLI

|  |
| --- |
| sudo npm install -g azure-cli  # check version to ensure CLI is properly installed  azure -v # should be 0.7.4 or greater |

Once Azure CLI has been installed the rest of the instructions are same for MAC and Linux.

## Authenticate using Certificate

The development machine will use a certificate (self-signed) to authenticate against Windows Azure Management services. For details, see [How to install the Windows Azure Cross-Platform Command-Line Interface](http://www.windowsazure.com/en-us/documentation/articles/xplat-cli/#install).

|  |
| --- |
| azure account download |

A browser will pop up and prompt for login. After a successful login, you will be prompted to download and save a file with file name extension .publishsettings. This publish-settings file contains the certificate with private key, and will be used in the next step. Please note that a new management certificate is automatically created using this step. The details of this newly created certificate can also be viewed in the portal under Settings/Management Certificates section.

If the development machine is a server without a UI (user interface), the URL will be displayed on the Unix prompt along with an informational message; see below.

|  |
| --- |
| $ azure account download  info: Executing command account download  info: Launching browser to http://go.microsoft.com/fwlink/?LinkId=254432  help: Save the downloaded file, then execute the command  help: account import <file>  info: account download command OK |

Copy the URL and use and go to a different machine which does have a browser enabled. Login using a browser and download the publish-settings file. In this case, the publish-settings file will have to be copied over to the development machine before proceeding to the next step.

|  |
| --- |
| azure account import <path-to-publish-settings-file> |

If everything goes through well up to this point, you should see a hidden folder .azure in your current working directory on the development machine. This directory should contain a certificate in .pem format and other files for connecting to your Azure subscription. For example:

|  |
| --- |
| ls -la .azure  total 20  drwxrw-r-- 2 azureuser azureuser 4096 Jan 29 21:26 ./  drwxr-xr-x 8 azureuser azureuser 4096 Jan 29 21:26 ../  -rw-rw-r-- 1 azureuser azureuser 105 Jan 29 21:26 config.json  -rw------- 1 azureuser azureuser 2740 Jan 29 21:26 managementCertificate.pem  -rw------- 1 azureuser azureuser 3962 Jan 29 21:26 publishSettings.xml |

If you logon to the Windows Azure Management Portal using a browser, and navigate to Settings 🡪 Management Certificates page, you should see a certificate (with public key) already installed there. At this point the development machine is configured for interacting with the Windows Azure Subscription.

To verify that connectivity has been established, run the command shown below to list the Account/Subscription that will be accessible from this machine.

|  |
| --- |
| azure account list |

### Cluster Configuration

#### Development PC

Get all the scripts and files from GitHub <https://github.com/devopscloudorg/azure-cdh/tree/master/bash>

You will need to edit the cdhsetup.sh in a text editor. This file contains all the settings necessary to create a CDH cluster.

#### Affinity Group

Affinity Group help you deploy your compute and storage account together. You need to specify a name as well as the region where the affinity group should be created.

*#Affinty group helps you keep your storage and compute in the same region*

export affinityGroupName**=**youreastcdhag

*#Name the region where affinity group should be created.*

*#choices are valid values are "East US", "West US", "East Asia", "Southeast Asia", "North Europe", "West Europe"*

export affinityGroupLocation**=**"East US"

#### Azure Storage Account

Define the name of your storage account. This name has be globally unique as it is the public DNS for your storage account.

*#name of the storage account here your virtual machines will be stored.*

export storageAccount**=**yourcdhstorage

#### Management Node Image Name

Windows Azure provides virtual machine images that are supported by Microsoft or other vendors.

If you need to get a list of which images are available you can run the command

Azure vm image list

In this example we are select Oracle Linux 6 image

*#Name of the image you will use to create your virtual machines*

*#export imageName=b39f27a8b8c64d52b05eac6a62ebad85\_\_Ubuntu\_DAILY\_BUILD-precise-12\_04\_3-LTS-amd64-server-20140204-en-us-30GB*

export imageName**=**c290a6b031d841e09f2da759bbabe71f\_\_Oracle-Linux-6

#### Management Node Configuration

This will configure various settings related to management node.

*#Size of the Virtual machine. Valid sizes are extrasmall, small, medium, large, extralarge, a5, a6, a7*

export instanceSize**=extralarge**

*#Size of the data disk you want to attach to the VM you are creating. You will typically attach at least 1 disk*

export diskSizeInGB**=**5

*#Number of disks you want to attach. Small VM can have 2 disks, medium can have 4, large can have 8 and extralarge can have 8 data disks*

export numOfDisks**=**1

*#virtual machine settings. We will generate names of all the VM from these names*

export vmNamePrefix**=**yourbdcdh

export cloudServicePrefix**=**yourbdcdh

*#user name and password for the virtual machine you are creating*

export adminUserName**=**azureuser

*#Azure CLI enforces strong passwords uppercase, lower case and special characters*

export adminPassword**=**Password.1!

#### Virtual Network Setting

Virtual machines in a Hadoop cluster need to communicate with each other. We recommend putting all the virtual machines in the same virtual network. Virtual network can have one of more subnets.

You can learn more about Azure Virtual Network here <http://www.windowsazure.com/en-us/documentation/services/virtual-network/>

vnetName defines a name of the virtual network

vnetAddressSpace defines the address space for the entire virtual network. We chose IP address range 172.16.0.0/16

You can create one of more subnets in a virtual network. In this example we configured one subnet named mysubnet. The address space used by this subnet is 172.16.1.0/24

*#setting related to virtual network*

export vnetName**=**youreastcdhvnet

*#address space allows 192.168.0.0, 10.0.0.0 and 172.16.0.0 ip address ranges*

*#virtual network faq is here http://msdn.microsoft.com/en-us/library/windowsazure/dn133803.aspx*

export vnetAddressSpace**=**172.16.0.0

export vnetCidr**=**16

export subnetName**=**mysubnet

export subnetAddressSpace**=**172.16.1.0

export subnetCidr**=**24

#### Cluster Node Settings

You will need to create customized image for the nodes in your cluster. After you have created the image you will need to update the setting nodeImageName with the name of your customized image.

nodeCount is the number of nodes in your CDH cluster.

nodeSize is the size of each node. Size will depend on the type of your workload. For compute intensive workloads you may need to select a larger instance size.

*#These settings are for nodes in the CDH cluster*

*#Name of the custom image you will use to create your cluster nodes*

*#After you have create your master node image replace the value of nodeImageName wiht the image you created*

export nodeImageName**=**c290a6b031d841e09f2da759bbabe71f\_\_Oracle-Linux-6

*#Number of nodes in your CDH cluster*

export nodeCount**=**10

*#Size of the nodes in the hadoop cluster. Valid sizes are extrasmall, small, medium, large, extralarge, a5, a6, a7*

export nodeSize**=**small

## Create the Infrastructure

From your PC run the bash script create createmgmtnode.sh and createmasternode.sh

This script will use the settings defined in cdhsetup.sh to create your virtual machine. Upon successful completion this script will display detailed information about management node.

#### Sample Execution Script

#On your Development PC

createmgmtnode.sh

createmasterimage.sh

## Manually configure the Management Node

The Management Node is named $vmNamePrefix followed by 0. You will use ssh client to configure this node.

Use your ssh client to log into the management node. In our example we used

ssh azureuser@yourbdcdh0.cloudapp.net

You will be prompted for password. Enter the value of $adminPassword you defined in cdhsetup.sh file.

### Set root passwords

The first steps will be to set the root passwords on both the Management and Master Nodes. You are currently connected to the Management Node.

Elevate to root. Enter the password for the $adminUserName when prompted. Update the root password. Enter the new password when prompted.

**sudo –s**

**passwd**

Install wget

**yum install wget**

### Update Kernel-Header

If you are using the gallery image “Oracle Linux 6.4.0.0.0”, the following steps must be executed for a successful installation of CDH.

Update the yum conf file. Comment out the line “exclude=kernel-uek-headers”

**vi /etc/yum.conf**

**# exclude=kernel-uek-headers**

Update kernel headers.

**yum install kernel-headers**

### Mount Disks Script

The disks will be mounted based on a startup script.

Copy the makefilesystem.sh script from your development PC to Management node. You can use scp to copy this file as shown below.

**scp makefilesystem.sh** [**azureuser@management\_node\_hostname.cloudapp.net:**](mailto:azureuser@management_node_hostname.cloudapp.net:)**makefilesystem.sh**

**On your management node.**

**mkdir /root/scripts**

**mv makefilesystem.sh /root/scripts**

**chmod 755 /root/scripts/makefilesystem.sh**

### Update Server Configuration Settings

Type the following commands to configure the server settings for CDH prerequisites.

**#disable iptables**

**chkconfig iptables off**

**/etc/init.d/iptables stop**

**setenforce 0**

**#start ntp service**

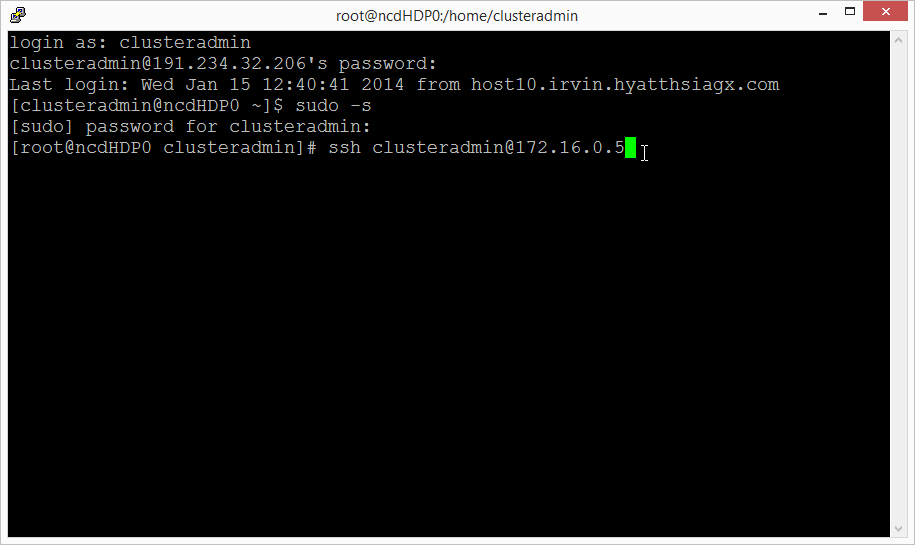
**chkconfig ntpd on**

**ntpdate pool.ntp.org**

## Manually Configure Master Node

Enter the following to move to the Master Node. Enter the password for the adminUserName when prompted.

**ssh <adminUserName>@<Master Node Hostname>.cloudapp.net**



Elevate to root. Enter the password for the $adminUserName when prompted.

Update the root password. Enter the new password when prompted.

**sudo –s**

**passwd**

With both root passwords set we can now configure the nodes for the cluster.

### Update Kernel-Header

If you are using the gallery image “Oracle Linux 6.4.0.0.0”, the following steps must be executed for a successful installation of CDH.

Update the yum conf file. Comment out the line “exclude=kernel-uek-headers”

**vi /etc/yum.conf**

**# exclude=kernel-uek-headers**

Update kernel headers.

**yum install kernel-headers**

### Mount Disks Script

The disks will be mounted based on a startup script.

Copy the makefilesystem.sh script from your development PC to Management node. You can use scp to copy this file as shown below.

**scp makefilesystem.sh** [**azureuser@master\_node\_hostnane.cloudapp.net:**](mailto:azureuser@master_node_hostnane.cloudapp.net:st.pl)**makefilesystem.sh**

**On your management node.**

**mkdir /root/scripts**

**mv makefilesystem.sh /root/scripts**

**chmod 755 /root/scripts/makefilesystem.sh**

### Update Server Configuration Settings

Configure the server settings for CDH prerequisites.

**#disable iptables**

**chkconfig iptables off**

**/etc/init.d/iptables stop**

**setenforce 0**

**#start ntp service**

**chkconfig ntpd on**

**ntpdate pool.ntp.org**

## Set up passwordless SSH between the Management Node and the Master Node

In the Management Node log in as root and generate the key.

**ssh-keygen**

Accept the default file location when prompted (press enter). Press enter to create the key without a passphrase. The public key is stored in .ssh/id\_rsa.pub, and the private key is id\_rsa. You will use the private key later during the CDH installation. At this time you will copy the public key to the Master Node to enable passwordless ssh.

Copy the key to the Master node and Management Node (self-referencing), enter the root password when prompted.

**ssh-copy-id -i /root/.ssh/id\_rsa.pub root@<Master Node Hostname>.cloudapp.net**

**ssh-copy-id -i /root/.ssh/id\_rsa.pub root@<Management Node Hostname>.cloudapp.net**



To test the keys were set up correctly, type the following and validate that you are not prompted for a password.

**ssh <Management Node Hostname>.cloudapp.net**

**ssh <Master Node Hostname>.cloudapp.net**

## Create Image of the Master Node

### Windows Azure Linux Agent

Set up the master node virtual machine for provisioning as an image. Open the waagent.conf file.

**vi /etc/waagent.conf**

Change the following settings:

**Provisioning.DeleteRootPassword=n**

**Provisioning.RegenerateSshHostKeyPair=n**

Run the Windows Azure Linux Agent.

**waagent –deprovision**

Open the Windows Azure Management Portal and navigate to the dashboard of the Master Node virtual machine. Shut down the machine. After the machine is stopped, click Capture to create an image.

Update the cdhsetup.sh file nodeImageName with the value of image you created.

## Create the Cluster

On your development PC run the bash script createclusternodes.sh. It create the virtual machines. It also creates hosts.txt and mountdrives.sh.

SCP updatehosts.sh, hosts.txt and mountdrives.sh to the management node.

Execute UpdateHosts.sh as it reads the hosts.txt and update /etc/hosts file.

Mountdrives.sh executes makefilesystem.sh on each node in the cluster to mount the data drives on them. Also updates /etc/hosts

#### Sample Execution Script

#On your dev PC

createclusternodes.sh

scp hosts.txt root@**<Management Node Hostname>.cloudapp.net:hosts.txt**

scp mountdrive.sh root@**<Management Node Hostname>.cloudapp.net:mountdrive.sh**

scp updatehosts.sh root@**<Management Node Hostname>.cloudapp.net:updatehosts.sh**

scp cdhsetup.sh root@**<Management Node Hostname>.cloudapp.net:cdhsetup.sh**

#On your management node logged in as root

chmod 755 mountdrive.sh

chmod 755 updatehosts.sh

#Reads the hosts.txt and updates /etc/hosts file on the management node

UpdateHosts.sh

#This script executes st.pl on all the nodes in CDH cluster to mount the drive

#It also update the hosts file on each node in the cluster

mountdrive.sh

To test the virtual machines were generated correctly verify that password less ssh is working correctly.

**ssh <Machine Hostname>.cloudapp.net**

## Install Ambari

Open an SSH session in PuTTY to the Management Node.

Download and install Ambari:

**wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.4.2.104/ambari.repo**

**cp ambari.repo /etc/yum.repos.d**

**yum install ambari-server**

After the installation has completed, run the setup:

**ambari-server setup**

After the management node is restarted, verify Ambari is started. Connect to the Management Node in PuTTY and execute the following command:

**ambari-server start**

## Install CDH

Open the browser and navigate to http://<Management Node Hostname>:8080

When prompted, log in as admin with password admin.

In general, follow the prompts to install. The only tricky part I would point out is in the install screen to input hosts and the key. You will paste the private key from the Management Node:

**cat .ssh/id\_rsa**

Highlight the full key and press enter to copy to the clipboard. You will then paste this in the web interface.