cloudera

Cloudera Data Science Workbench Installation from Scratch on AWS

Partner Presales @Cloudera:
Filippo Lambiente
Alvin Heib

(greetings to Toby Fergusson)

cloudera

II. Table of Contents

cloudera

I. Introduction

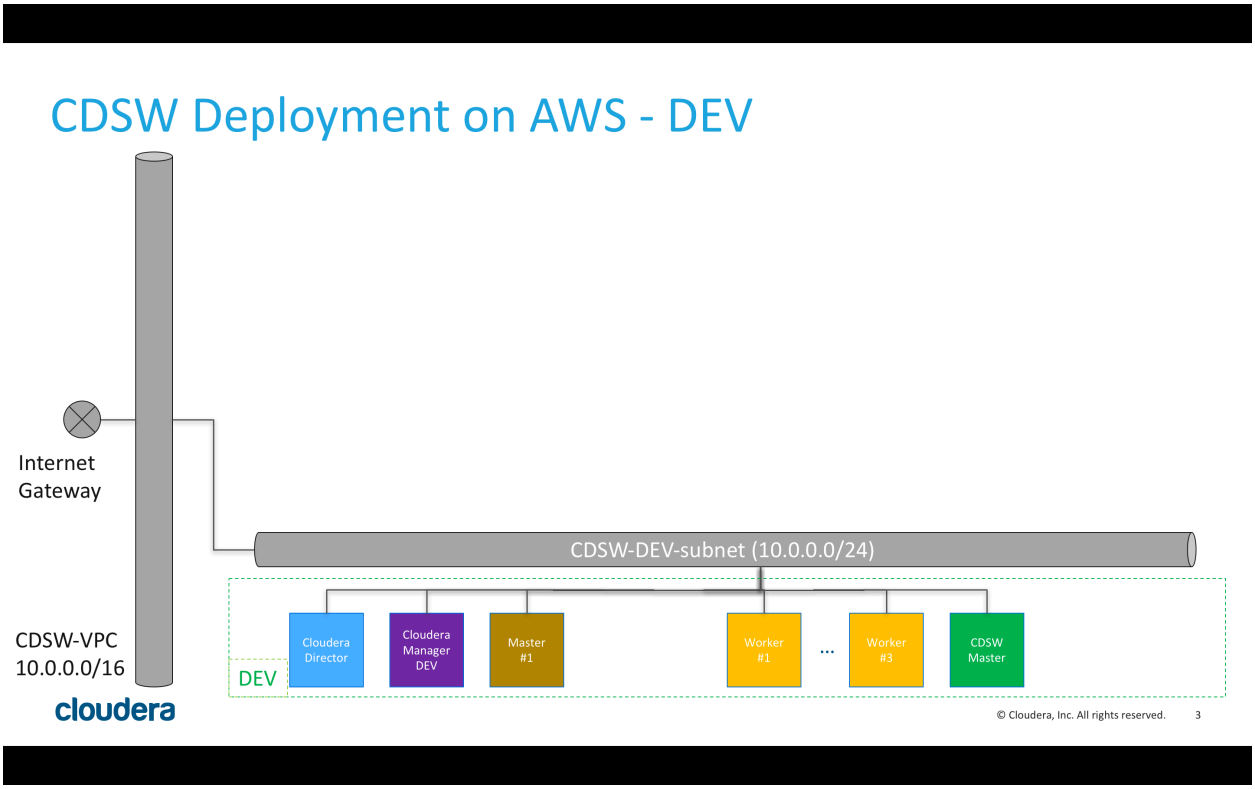
Cloudera Data Science and Engineering is the comprehensive offering for exploratory data science and machine learning at scale. It lives where your datas lives, on-premise and across public clouds. Cloudera gives data science users better access to Hadoop data with familiar and performant tools that address every stage of the modern predictive analytics workflow.

This Enablement Session will focus on creating a fully functional Cloudera Data Science Workbench service based on AWS Public Cloud. The Cloudera Data Engineering cluster will be created through Cloudera Director.

The Operations, Industrialisation and Production/Security readiness will also be covered.

The architecture of the deployment for the Enablement Session:

cloudera



There will be a Virtual Private Cloud (VPC) with a CDSW-DEV-subnet deployed for the Network Layer. This subnet is made public, and will have access to the internet.
An instance of Cloudera Director and Cloudera Manager will be deployed.
The Cloudera Cluster will be a minimal one based on 1 Master and 3 Worker Nodes.
There will be a single Cloudera Data Science Workbench node. (no edge nodes will be deployed).

Here will be the associated sizing for the AWS Deployment:

DEV					SYSTEM DISK				DATA DISK			
NODE TYPE	QTY	INSTANCE TYPE	CPU	RAM	QTY	CAPACITY (GB)	TYPE	TOTAL (GB)	QTY	CAPACITY (GB)	TYPE	TOTAL (GB)
Cloudera Director	1	t2.medium	2	4	1	50	EBS	50				0
Cloudera Manager	1	t2.large	2	8	1	50	EBS	50				0
Master	1	t2.large	2	8	1	50	EBS	50				0
Worker	3	t2.xlarge	4	16	1	50	EBS	50	1	500	EBS	500
CDSW-Master	1	t2.2xlarge	8	32	1	100	GP2	100	2	500	GP2	1 000
			18	68				300				1 500

cloudera

II. Prerequisites

A. AWS Account / Quotas

Please make sur with your AWS administrator that you have an AWS account (with credentials) and sufficient quotas. We will use different Virtual Machine sizing for the different environments DEV / STG / PRD.

The minimum requirement are:
* 1 Virtual Private Cloud (VPC)
* 1 Security Group
* 7 Instances (from t2.medium to t2.2xlarge)
* 18 vCPU total
* 68 GB RAM total
* 300 GB DISK (EBS / GP2) total for OS
* 1.5 TB DISK (EBS / GP2) total for Data Store

B. AWS Command Line Interface installed (CLI)

Please got through this tutorial to correctly install your AWS CLI environment.
Installing the AWS Command Line Interface

C. Knowledge on SSH with Keypairs into a linux terminal

During the session, we will provide a keypairs to ease the Cloudera Staff debuggability. You could find private / public keypairs (**cdsw-admin** / **cdsw-admin.pub**) under this link:

Cloudera Data Science Workbench Installation from scratch on AWS - Github

Special attention for windows users, please make sure sure that you have:

- Installed putty. Putty installation link
- And converted the **cdsw-admin** private key. Putty user guide

cloudera

III. Lab #1: AWS Environment Setup

A. Prepare a directory with files for the Enablement Session

- Identify a storage location to clone or unzip all the scripts & files for our Enablement session.

- Use clone (using `git clone <URL>`) or download (using the Top-Right green download button) the current git repo:
  https://github.com/heibalvin/Cloudera-Data-Science-Workbench-Installation-from-scratch-on-AWS
  You should have a directory named `Cloudera-Data-Science-Workbench-Installation-from-scratch-on-AWS`.

- You shoud have a directory hierarchy very near to:

  Cloudera-Data-Science-Workbench-Installation-from-scratch-on-AWS
  + README.md
  + cdsw-admin
  + cdsw-admin.pub
  + owner_tag.properties
  + provider.properties
  + SECRET.properties
  + ssh.properties
  + cloudera-director-install.sh
  + cloudera-CDH-bootstrap-script.sh

- Special attention is required on the cdsw-admin & cdsw-admin.pub key pairs' permissions. We must protect it from being read by other, and from accidental deletion (no more access to your VMs). On linux, please issue a command `chmod 400 cdsw-admin*`.

- During the Enablement Session we will be intensivelly using linux environment variable. Most of them needs to be updated in the corresponding files:

owner_tag.properties:
[YOUR-USERNAME]

provider.properties:
[YOUR-AWS-ACCESS-KEY-ID]
[YOUR-AWS-DEV-SECURITY-GROUP-ID]
[YOUR-AWS-DEV-SUBNET-ID]

SECRET.properties:
[YOUR-AWS-SECRET-ACCESS-KEY]

As well as URL for the Different Services:
[CLOUDERA-DIRECTOR-PUBLIC-DNS]
[CLOUDERA-MANAGER-PUBLIC-DNS]
[CLOUDERA-CDSW-PUBLIC-DNS]
[CLOUDERA-CDSW-PRIVATE-IP]

B. Log on to AWS Console

- First step is to open your favorite web browser and use the url. Then click on the button **Sign in to the Console**. You will be able to login with your AWS user/password.

  /https://aws.amazon.com/console/

- For the Enablement Session, we will be using Ireland data-center (eu-west-1). Please select the **EU(Ireland)** datacenter on the menu Top-Right (next to Support)

C. Create a new VPC

Click on Top-Left AWS Search Box, and search for VPC Services.

We will use the **VPC Wizard**, please click on the button to proceed.

For ease of the Enablement Session, we will be creating a single DEV subnet for this VPC. Please select option **VPC with a Single Public Subnet**.

- VPC name = [YOUR-USERNAME]-CDSW-VPC for ex: aheib-CDSW-VPC.
- Subnet name = [YOUR-USERNAME]-CDSW-DEV-subnet for ex: aheib-CDSW-DEV-subnet.

cloudera

D. Retrieve Subnet Information

Make sure you are still in the VPC Services page.

Please select **Subnets** on the left menu.
And search for your newly created subnet [YOUR-USERNAME]-CDSW-DEV-subnet. Under **subnet-id**field, you should find a subnet-id.

This is [YOUR-AWS-DEV-SUBNET-ID] value.

Please update your **provider.properties** file with the new value.

E. Create a new Security Group

Make sure you are still in the VPC Services page.

Please select **Security Groups** on the left menu. And click on the **Create Security Group** button.

- Name tag = [YOUR-USERNAME]-CDSW-DEV-secgroup for ex: aheib-CDSW-DEV-secgroup
- Group name = [YOUR-USERNAME]-CDSW-DEV-secgroup for ex: aheib-CDSW-DEV-secgroup
- Description = Cloudera Director, Manager, Data Science Workbench
- VPC = identify your newly created VPC for ex: vpc-43126424 l aheib-CDSW-VPC

And search for your newly created security group [YOUR-USERNAME]-CDSW-DEV-secgroup (for ex: aheib-CDSW-DEV-secgroup). Under **group-id** field, you should find a security group id.

This is [YOUR-AWS-DEV-SECGROUP-ID] value.

Please update your **provider.properties** file with the new value.

cloudera

Then, you will need to add Security Group Rules for inbound traffic. Please select **Inbound** Tab, Edit and add rules for each entry below:

- Type = SSH, Port = 22, Source = Anywhere (for SSH services)
- Type = Custom TCP, Port = 7189, Source = Anywhere (for Cloudera Director Services)
- Type = Custom TCP, Port = 7180, Source = Anywhere (for cloudera Manager Services)
- Type = Custom TCP, Port = 80, Source = Anywhere (for Cloudera Data Science Workbench Services)
- Type = All Traffic, Port = 0-65535, Source = 10.0.0.0/24 (for CDH services, no firewall on local network)

F. Import your KeyPair

Click on Top-Left AWS Search Box, and search for EC2 Services.

- Please select **Key Pairs** on the left menu. And click on the **Import Key Pair** button. Load public key from file = **cdsw-admin.pub** which is in your folder.
- Key Pair name = **cdsw-admin**

cloudera

IV. Lab #2: Cloudera Director Setup

A. Deploy your Cloudera Director Instance

Make sure you are still in the EC2 Services page.

Please select **EC2 Dashboard** on the left menu. And click on the **Launch Instance** button.

- Choose AMI.
  We will be using CentOS 7 type of images for this Enablement Session.
  Please select **AWS Marketplace** on left-menu.
  Then search for **centos** images.
  For this Enablement Session we will be using exclusively **CentOS 7 (x86_64) - with Updates HVM**.

- Choose Instance
  Type. For instance type, you will need to select suitable virtual machine for Cloudera Director. Please select **t2.medium** (for a DEV environment)

- Configure Instance.
  Select the correct Network and Subnet. It is important to set to Enable the flag Auto-assign Public IP.

Before moving to the next step, please make sure to add the Cloudera Director install scripts.
To do so, please open up Advanced Details and import file : **cloudera-director-install.sh**

- Add Storage
  Please modify the storage capacity from 8 -> 30 GB.

- Add Tags.
  We will be adding 2 tags, which represents the Name of the Instance and the Owner of the instance.
  - Name = [YOUR-USERNAME]-CDSW-DEV-Director for ex: aheib-CDSW-DEV-Director
  - owner = [YOUR-USERNAME] for ex: aheib

cloudera

- Configure Security Group
  Please select and existing Security Group.
  And select the Security Group `[YOUR-USERNAME]-CDSW-DEV-secgroup`.

- Review.
  You will then see a pop-up window asking you the correct Key Pair to be used. Please select the previously imported cdsw-admin keypair.

B. Identify your Cloudera Director Instance

Make sure you are still in the EC2 Services page and select **Instances** on the left menu.

Search for your newly created Director Instance **[YOUR-USERNAME]-CDSW-DEV-Director** for ex: aheib-CDSW-DEV-Director.

Once you have clicked on the instance, you will find in bottom view an entry called Public DNS (IPv4).

This is your [CLOUDERA-DIRECTOR-PUBLIC-DNS] environment variable.

C. Log onto Cloudera Director

First step is to open your favorite web browser and use the url:

http://[CLOUDERA-DIRECTOR-PUBLIC-DNS]:7189/

Then, you will then be asked to accept the Cloudera Director End User License T&C. Please proceed.

Finally, you will be able to login with the default login/password (**admin**/**admin**).

D. Modify your Default Password

On the welcome page, you will have a complete access to Cloudera Director options. Please click on **Admin Menu** on top-right and select **Change Password**.

For the Enablement debuggability purpose, please use this password:

password = Cloudera_123

cloudera

V. Lab #3: Cloudera Data Engineering Cluster Setup

A. Create your AWS-DEV Environment

Please click on **Environment** and select **Add Environment**.

We will then enter information related to our AWS account.

Key	Value	Comments
Environment Name	AWS-CDSW-DEV	in our example
Cloud Provider	AWS	in our example (could be also Azure / GCE)
Access Key Id	[YOUR-AWS-ACCESS-KEY-ID]	for ex: AKIAIOSFODNN7EXAMPLE
Secret Access Key	[YOUR-AWS-SECRET-ACCESS-KEY]	for ex: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
EC2 Region	eu-west-1	in our example
RDS Region	eu-west-1	in case we are using a managed AWS DB for Cloudera Manager, not covered in our example
Username	centos	username for the centos 7 image on AWS with root priviliedges **"/!\ It is not your AWS username /!"**
Private Key	cdsw-admin	in our example

B. Create Node Templates

We will not proceed with the Cloudera Manager creation directly, instead we will create templates for Cloudera Manager / Master / Worker and CDSW nodes.

cloudera

Select the newly created environment in menu **Environment** and select **AWS-CDSW-DEV**.

Select **Templates** menu, on top menu row. Then click on **Create Instance Template**.

- Create a Cloudera Manager Node template.
  Which is the longest part, since we will use Cloudera Directors Template copy feature to create all the others (Master, Worker, etc ...)

Key	Value	Comments
Template Name	Cloudera Manager	in our example
Instance Type	t2.large	for a DEV environment
Image (AMI ID)	ami-7abd0209	which is the AMI ID for eu-west-1 region for CentOS 7
Tags	(owner, [YOUR-USERNAME])	for ex: (owner, aheib)
Security Group Id	[YOUR-AWS-DEV-SECURITY-GROUP-ID]	for ex: sg-27e5c95f
Subnet Id	[YOUR-AWS-DEV-SUBNET-ID]	for ex: subnet-3b64085c
Instance Name Prefix	CM	this will help in identifying the node type under AWS Console
Bootstrap scripts	cloudera-CDH-bootstrap-script.sh	From your Cloudera-Data-Science-Workbench-Installation-from-scratch-on-AWS folder

Once completed you will obtain a first node template. And then, we could start copying this template and create the following ones.

cloudera

- Create a Master Node template.
  Copy Template from the Cloudera Manager Node template, and proceed to the specific updates.

Key	Value	Comments
Template Name	Master	in our example
Instance Name Prefix	Master	this will help in identifying the node type under AWS Console

- Create a Worker Node template.
  Copy Template from the Cloudera Manager Node template, and proceed to the specific updates.

Key	Value	Comments
Template Name	Worker	in our example
Instance Type	t2.xlarge	for a DEV environment
Instance Name Prefix	Worker	this will help in identifying the node type under AWS Console
EBS Volume Count	1	we will need a 1 * 500 GiB HDFS disk
EBS Volume Size	500	we will need a 1 * 500 GiB HDFS disk

- Create a CDSW Master Node template.
  Copy Template from the Cloudera Manager Node template, and proceed to the specific updates.

Key	Value	Comments
Template Name	CDSW-Master	in our example
Instance Type	t2.2xlarge	for a DEV environment
Instance Name Prefix	CDSW-Master	this will help in identifying the node type under AWS Console
EBS Volume Count	2	we will need a 2 * 500 GiB HDFS disk

cloudera

Key	Value	Comments
EBS Volume Size	500	we will need a 2 * 500 GiB HDFS disk
EBS Volume Type	gp2	in our example
Root Volume Size	100	in our example

C. Deploy your first Cloudera Manager Instance

Make sure you are still in the **AWS-CDSW-DEV** environment, then select **Add Cloudera Manager**.

Key	Value	Comments
Cloudera Manager Name	CM-DEV-0	in our example
Instance Template	Cloudera Manager	the one we created on previous steps
License Type	Cloudera Enterprise Trial	in our example, and expires within 30 days
Database Server	Embedded Database	in our example, should be an external DB for production
Cloudera Manager Admin Username	admin	please use this username for enablement session debugging ease
Cloudera Manager Admin Username	Cloudera_123	please use this password for enablement session debugging ease

D. Deploy your first Cloudera Cluster

You will automatically reach the wizard to create your first cluster.

Key	Value	Comments

cloudera

Key	Value	Comments
Cluster Name	EDH-DEV-0	
Services	Core Hadoop with Spark on Yarn	Spark is needed for CDSW, all Services is also possible

On the same page, you will need to define the cluster architecture (number of master, slave, CDSW-master nodes).

- for **masters** group name, choose **Master** node template and set the instance count to **1**.
- for **workers** group name, choose **Worker** node template and set the instance count to **3**
- for **gateways** group name, choose **CDSW-Master** node template and set the instance count to **1**.

Your cluster (Cloudera Manager, Master, Workers and CDSW-Master) should be deploying in background. You should observe after 15 min, that the entire cluster is deployed.

E. Identify your Cloudera Manager

You have a complete view of your cluster deployed from Cloudera Director.

Click on your newly deployed Cloudera Manager called **Manager**.

When you extend the **View Properties**, you will identify your **[CLOUDERA-MANAGER-PUBLIC-DNS]**.

cloudera

VI. Lab #4: Upgrade Cluster to Spark 2

A. Install Spark2 Add-On Service on Cloudera Manager

- First, we need to install new packages on Cloudera Manager for Spark 2 libraries. Connect to your Cloudera Manager instance through a terminal window, using command:

```
ssh -i cdsw-admin centos@[CLOUDERA-MANAGER-PUBLIC-DNS]
```

- Update the JAVA_HOME environment variable for cloudera manager server config file:

```
sudo sh -c "echo export JAVA_HOME=/usr/java/jdk1.8.0_121-cloudera >>
/etc/default/cloudera-scm-server"
```

- Then you will be able to download appropriate the official CSD file:

```
sudo wget --directory-prefix=/opt/cloudera/csd/
http://archive.cloudera.com/spark2/csd/SPARK2_ON_YARN-
2.2.0.cloudera1.jar
```

```
sudo chmod 644 /opt/cloudera/csd/SPARK2_ON_YARN-2.2.0.cloudera1.jar
```

```
sudo chown cloudera-scm:cloudera-scm /opt/cloudera/csd/SPARK2_ON_YARN-
2.2.0.cloudera1.jar
```

```
sudo systemctl restart cloudera-scm-server
```

B. Restart Cloudera Management Services

- First step is to open your favorite web browser and use the below url. you will be able to login with the default login/password (**admin/Cloudera_123**).

  http://[CLOUDERA-MANAGER-PUBLIC-DNS]:7180/

cloudera

- We will need to setup JAVA_HOME environment variable. On the Cloudera Manager search bar please enter JAVA_HOME and set the environment variable to `/usr/java/jdk1.8.0_121-cloudera.`
- We can now restart Cloudera Management Service and Cluster Stalled Services

C. Install Spark2 / Anaconda Parcels

- Add new repos for Anaconda parcels (Spark 2 parcel is already existing).
  Click on the icon top left looking like a delivery package.
  Then, click on **Configure** and add new **Remote Parcel Repository URLs** for Spark2 & Anaconda repos.:

  https://repo.continuum.io/pkgs/misc/parcels/

- Sequentially click on Download (to Cloudera Manager), Distribute (to all hosts) and then Activate (on all hosts) both packages.

- Deploy new Spark2 services on the Cluster. Choose Spark2 only for HDFS / YARN services. History Server will be set on the Master Node and Spark2 Gateway on all of the nodes.

- Coming back to Cloudera Manager front page, you will be asked to restart all services (Spark services has stalled configurations to be taken into account).

cloudera

VII. Lab #5: Cloudera Data Science Workbench Setup

A. Deactivate Firewalling on CDSW

- You first need to connect to your CDSW-Master instance using:
`ssh -i cdsw-admin centos@[CLOUDERA-CDSW-PUBLIC-DNS]`

- Deactivate Temporarily the SELinux service.
Check you SELinux status using command: `sudo sestatus`. The command should return a similar output.
`SELinux status:                 enabled`

  Deactivate temporarly the SELinux service using command: `sudo setenforce 0`.

  Verify that you have successfully temporarily deactivate SELinux.

- Deactivate Permanently the SELinux service.
Check you SELinux status using command: `grep SELINUX= /etc/selinux/config`. The command should return a similar output.

  `# SELINUX= can take one of these three values:`
  `SELINUX=disabled`.

  If SELinux is enabled in the config file, please update the file.

B. Unmount both disk (/dev/xvdf and /dev/xvdg)

.

Un-mount both disk on your CDSW-Master instance:

cloudera

Check the number of disks attached to the virtual machine using `sudo mount | grep xvd` command. You should have something similar to:

```
/dev/xvda1 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
/dev/xvdf on /data1 type ext4 (rw,noatime,seclabel,data=ordered)
/dev/xvdg on /data2 type ext4 (rw,noatime,seclabel,data=ordered)
```

(xvda1=Operating System, xvdf=block device for Docker Containers, and xvdg=block device for cdsw user-data)

Now, you need to un-mount both disks **/dev/xvdf** and **/dev/xvdg**.

```
sudo umount /dev/xvdf /dev/xvdg
```

Verify that you have successfully un-mounted both disks using `sudo mount | grep xvd` command.

- permanently un-mount both disk on your CDSW-Master instance

You could observe that in file **/etc/fstab** you will still find some enrtries for **xvdf** and **xvdg**. We will need to remove these entries in case of CDSW-Master reboot operations.

Check that both entries are in the file **/etc/fstab** using command `sudo cat /etc/fstab | grep xvd`. You should have something similar to:

```
/dev/xvdf /data1 ext4 defaults,noatime 0 0
/dev/xvdg /data1 ext4 defaults,noatime 0 0
```

Now, please remove both entries in the file. Please backup the file to avoid any manual errors. A handy linux command to do it is using `sudo sed -i.bak '/xvd/d' /etc/fstab` command.

Verify that you have successfully permanently un-mounted both disks using `sudo cat /etc/fstab | grep xvd` command.

C. Ensure IPv6 is Disabled

Check that IPv6 is enabled using `sudo cat /etc/sysctl.conf | grep ipv6` command. You should have something similar to:

```
net.ipv6.conf.all.disable_ipv6=1
```

cloudera

Now, please disable the IPv6 feature by setting the value to 0. Please backup the file to avoid any manual errors. A handy linux command to do it is using `sudo sed -i.bak 's/net.ipv6.conf.all.disable_ipv6=1/net.ipv6.conf.all.disable_ipv6=0/g' /etc/sysctl.conf` command.

Verify that you have successfully deactivated IPv6 using command `sudo cat /etc/sysctl.conf | grep ipv6` command.

D. Enable/Activate Rpcbind on Start-Up

Check rpcbind service status by using `sudo systemctl status rpcbind` command. You should have something similar to:

```
rpcbind.service - RPC bind service
Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; indirect; vendor preset: enabled)
Active: inactive (dead)
```

Now, please enable the rpcbind service at startup using below commands:

```
sudo systemctl enable rpcbind
```

```
sudo systemctl start rpcbind
```

Verify that you have successfully deactivated IPv6 using command `sudo systemctl status rpcbind` command.

E. Remove IPtables service blacklisting

First you will need to remove the blocklist file using `sudo rm -f /etc/modprobe.d/iptables-blacklist.conf` command.

Then, you will need to activate iptables module `sudo modprobe ip_tables`.

Finally, you will need to load iptables filters modules `sudo modprobe iptable_filter`.

F. Download and Install Cloudera Data Science Workbench

cloudera

Now that the Instance is ready, we could start downloading the Cloudera Data Science Manager packages and install.

```
sudo yum install -y wget
```

```
sudo wget --directory-prefix=/etc/yum.repos.d/
https://archive.cloudera.com/cdsw/1/redhat/7/x86_64/cdsw/cloudera-cdsw.repo
```

```
sudo yum install -y cloudera-data-science-workbench
```

G. Configure Cloudera Data Science Workbench

- Identify the CDSW configuration file at location and please have a look using command: `sudo cat /etc/cdsw/config/cdsw.conf`

- Update the CDSW-Master public ip using the XIP trick.
  The [CLOUDERA-CDSW-PUBLIC-IP] should be something like x.x.x.x.
  A handy linux command to do it is using `sudo sed -i $.bak 's/DOMAIN="cdsw.company.com"/DOMAIN="cdsw.[CLOUDERA-CDSW-PUBLIC-IP].xip.io"/g /etc/cdsw/config/cdsw.conf` command.

- Update the CDSW-Master private local DNS.
  The [CLOUDERA-CDSW-PRIVATE-IP] should be something like ip-10-251-50-12.ec2.internal. A handy linux command to do it is using `sudo sed -i $.bak 's/MASTER_IP=""/MASTER_IP="[CLOUDERA-CDSW-PRIVATE-IP]"/g /etc/cdsw/config/cdsw.conf` command.

- Update the CDSW Block Device to store Docker images (DOCKER_BLOCK_DEVICES="/dev/xvdf")
- Update the CDSW Block Device to store Application Data (APPLICATION_BLOCK_DEVICE="/dev/xvdg")
- Update the JAVA_HOME environment variable JAVA_HOME=/usr/java/jdk1.8.0_121-cloudera

H. Launch Cloudera Data Science Workbench

cloudera

Start Cloudera Data Science Workbench using command: `sudo cdsw init`
You could check Cloudera Data Science Workbench status using command `sudo cdsw status`.

You can now use Cloudera Data Science Workbench at URL:

http://cdsw.[CLOUDERA-CDSW-PUBLIC-IP].xip.io/

I. Create special Spark user hdfs_super

Since we have not setup a LDAP or AD, we will not be having a correspondance between HDFS and CDSW user. Only a single user will be able to launch Spark / HDFS jobs.

```
sudo groupadd supergroup
```

```
sudo useradd -G supergroup -u 12354 hdfs_super
```

```
sudo su -c "echo Cloudera_123 | passwd --stdin hdfs_super"
```

```
sudo su hdfs -c "hadoop dfs -mkdir /user/hdfs_super"
```

```
sudo su hdfs -c "hadoop dfs -chown hdfs_super:hdfs_super /user/hdfs_super"
```

VIII. Lab#6: Cloudera Data Science Workbench Operations

A. Login as Administrator

B. Setup HADOOP_USER_NAME

C. Setup Docker Container types

cloudera

D. Setup Users / Teams

E. Block External Sign-Ups

IX. Lab #7: Industrialisation: Cloudera Director CLI

A. Log onto Cloudera Director

First, you will need some installation scripts coming from my friend Toby. Use clone (using `git clone <URL>`) or download (using the Top-Right green download button) the current git repo:
https://github.com/TobyHFerguson/cdsw_install
You should have a directory named `cdsw_install`.

Copy files ….

Launch command
cloudera-director bootstrap-remote aws.conf –lca….