CSC Deutschland GmbH

Deploying a Data Stack for Predictive Maintenance

An instruction of HDP 2.4 deployment with Python & TensorFlow for vibration data analysis in Jupyter Notebook

Autor: Zilong Zhao
✉: zzhao3@csc.com
June 7, 2016

Contents

1 Introduction

Hortonworks Data Platform[1] is an open source Apache Hadoop distribution. Officially Hortonworks offers a Sandbox on their website, which is an interactive Hadoop ecosystem, running in virtual machine. However, the integrated system is CentOS 6, which causes error during the installation of TensorFlow 0.8.0 due to the missing **GNU C Library** `glibc 2.16`.

Hence, in this introduction, we deploy the HDP 2.4 on the latest CentOS Linux distribution, namely Version 7. Furthermore, the Jupyter Notebook will be used as our data science IDE.

Keywords: CentOS 7, HDP 2.4, Spark, TensorFlow, Python, Jupyter Notebook

2 Deploying the HDP

We start our deployment from installation of Cent OS 7 on.

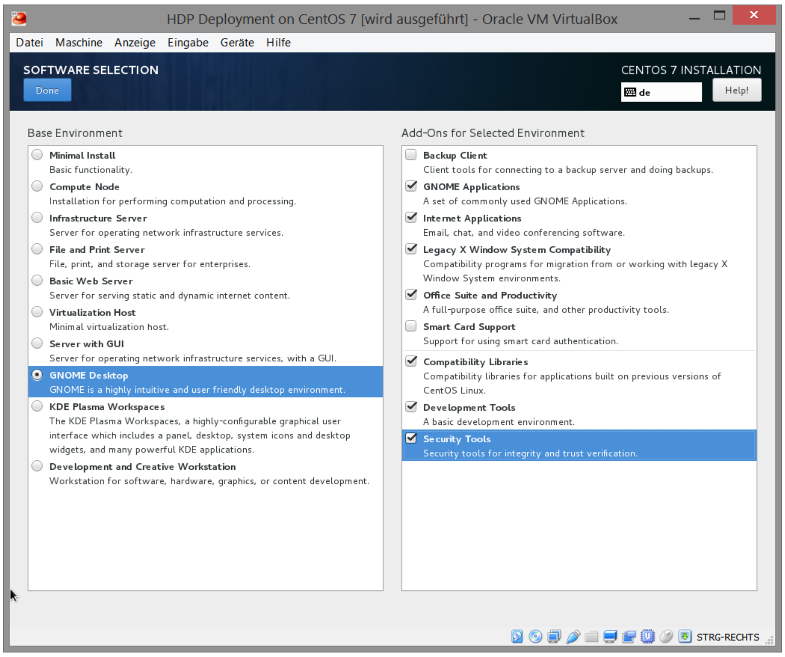
2.1 Create a CentOS 7 VM

The installation of Cent OS 7 should be done in couple of minutes. Meanwhile, we set two user accounts. One is `hdp` as administrator with password `csc`, and `root` with password `hearts4csc`. Applying the CSC proxy settings `webgate.de.emea.csc.com:80` in Applications > System Tools > Settings > Network > Network proxy, we get some text output as above by typing

```
curl www.google.de
```

Also, in Applications > System Tools > Settings > Details > Device name, we set it as `wiesbaden`. First of all, we will install the Anaconda 2.5.0 from `https://repo.continuum.io/archive/index.html`

[1] abbr. HDP



HDP Deployment on CentOS 7 [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
Done
de
Help!
STRG-RECHTS

SOFTWARE SELECTION

CENTOS 7 INSTALLATION

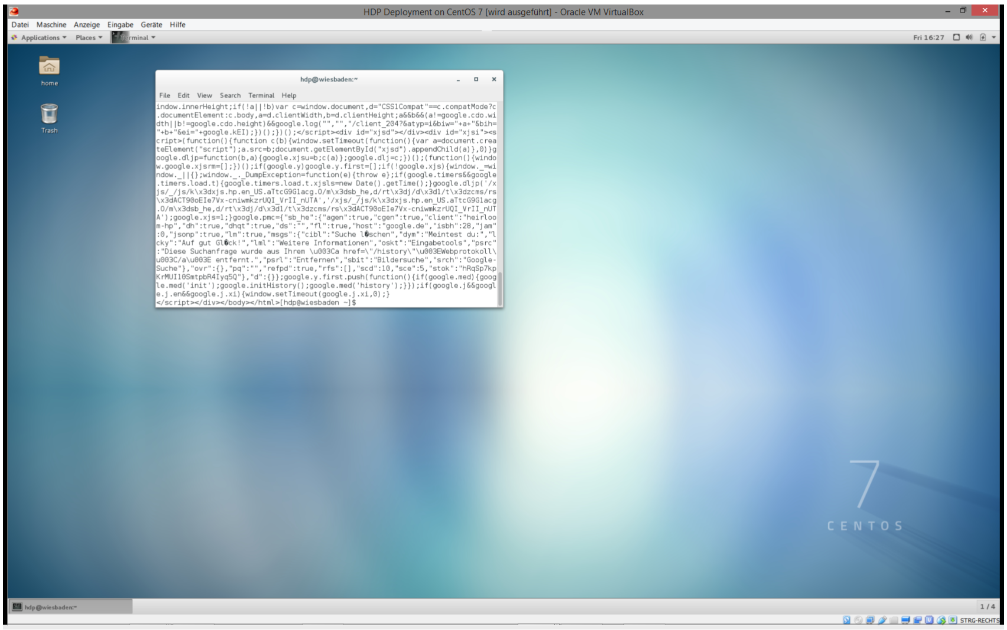
Base Environment

- ○ **Minimal Install**
  Basic functionality.
- ○ **Compute Node**
  Installation for performing computation and processing.
- ○ **Infrastructure Server**
  Server for operating network infrastructure services.
- ○ **File and Print Server**
  File, print, and storage server for enterprises.
- ○ **Basic Web Server**
  Server for serving static and dynamic internet content.
- ○ **Virtualization Host**
  Minimal virtualization host.
- ○ **Server with GUI**
  Server for operating network infrastructure services, with a GUI.
- ● **GNOME Desktop**
  GNOME is a highly intuitive and user friendly desktop environment.
- ○ **KDE Plasma Workspaces**
  The KDE Plasma Workspaces, a highly-configurable graphical user interface which includes a panel, desktop, system icons and desktop widgets, and many powerful KDE applications.
- ○ **Development and Creative Workstation**
  Workstation for software, hardware, graphics, or content development.

Add-Ons for Selected Environment

- □ **Backup Client**
  Client tools for connecting to a backup server and doing backups.
- ☑ **GNOME Applications**
  A set of commonly used GNOME Applications.
- ☑ **Internet Applications**
  Email, chat, and video conferencing software.
- ☑ **Legacy X Window System Compatibility**
  Compatibility programs for migration from or working with legacy X Window System environments.
- ☑ **Office Suite and Productivity**
  A full-purpose office suite, and other productivity tools.
- □ **Smart Card Support**
  Support for using smart card authentication.
- ☑ **Compatibility Libraries**
  Compatibility libraries for applications built on previous versions of CentOS Linux.
- ☑ **Development Tools**
  A basic development environment.
- ☑ **Security Tools**
  Security tools for integrity and trust verification.

Figure 1: The Cent OS 7 package selection



HDP Deployment on CentOS 7 [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
Applications
Places
Terminal
Fri 16:27
home
Trash
hdp@wiesbaden:~
File Edit View Search Terminal Help
7
CENTOS
hdp@wiesbaden:~
1 / 4
STRG-RECHTS

```
indow.innerHeight;if(!a||!b)var c=window.document,d="CSS1Compat"==c.compatMode?c
.documentElement:c.body,a=d.clientWidth,b=d.clientHeight;a&&b&&(a!=google.cdo.wi
dth||b!=google.cdo.height)&&google.log("","","/client_204?&atyp=i&biw="+a+"&bih=
"+b+"&ei="+google.kEI);})();})();</script><div id="xjsd"></div><div id="xjsi"><s
cript>(function(){function c(b){window.setTimeout(function(){var a=document.crea
teElement("script");a.src=b;document.getElementById("xjsd").appendChild(a)},0)}g
oogle.dljp=function(b,a){google.xjsu=b;c(a)};google.dlj=c;})();(function(){windo
w.google.xjsrm=[];})();if(google.y)google.y.first=[];if(!google.xjs){window._=wi
ndow._||{};window._._DumpException=function(e){throw e};if(google.timers&&google
.timers.load.t){google.timers.load.t.xjsls=new Date().getTime();}google.dljp('/x
js/_/js/k\x3dxjs.hp.en_US.aTtcG9G1acg.O/m\x3dsb_he,d/rt\x3dj/d\x3d1/t\x3dzcms/rs
\x3dACT90oEIe7Vx-cniwmkzrUQI_VrII_nUTA','/xjs/_/js/k\x3dxjs.hp.en_US.aTtcG9G1acg
.O/m\x3dsb_he,d/rt\x3dj/d\x3d1/t\x3dzcms/rs\x3dACT90oEIe7Vx-cniwmkzrUQI_VrII_nUT
A');google.xjs=1;}google.pmc={"sb_he":{"agen":true,"cgen":true,"client":"heirloo
m-hp","dh":true,"dhqt":true,"ds":"","fl":true,"host":"google.de","isbh":28,"jam"
:0,"jsonp":true,"lm":true,"msgs":{"cibl":"Suche l�schen","dym":"Meintest du:","l
cky":"Auf gut Gl�ck!","lml":"Weitere Informationen","oskt":"Eingabetools","psrc"
:"Diese Suchanfrage wurde aus Ihrem \u003Ca href=\"/history\"\u003EWebprotokoll\
u003C/a\u003E entfernt.","psrl":"Entfernen","sbit":"Bildersuche","srch":"Google-
Suche"},"ovr":{},"pq":"","refpd":true,"rfs":[],"scd":10,"sce":5,"stok":"hRqSp7kp
KrMUI10SmtpbR4Iyq5Q"},"d":{}};google.y.first.push(function(){if(google.med){goog
le.med('init');google.initHistory();google.med('history');}});if(google.j&&googl
e.j.en&&google.j.xi){window.setTimeout(google.j.xi,0);}
</script></div></body></html>[hdp@wiesbaden ~]$
```

Figure 2: The Cent OS 7 with gnome interface

2.2 Anaconda Python & TensorFlow

Anaconda is an open source distribution of the Python programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. The advantage is we have not only the `pip` package management pipeline, it gives us also a `conda` package management system to connect with the Anaconda cloud service.

```
[hdp@wiesbaden ~]$ cd ~/Downloads/
[hdp@wiesbaden Downloads]$ wget https://repo.continuum.io/archive/Anaconda2-2.5.0-Linux-x86_64.sh
```

To makes the file executable by everyone, we type

```
[hdp@wiesbaden Downloads]$ chmod +x Anaconda2-2.5.0-Linux-x86_64.sh
```

Now by running

```
[hdp@wiesbaden Downloads]$ ./Anaconda2-2.5.0-Linux-x86_64.sh
```

we start to install Anaconda Python. During the installation we accept the default path setting for anaconda2 and for `bash` we have

```
Python 2.7.11 :: Continuum Analytics, Inc.
creating default environment...
installation finished.
Do you wish the installer to prepend the Anaconda2 install location
to PATH in your /home/hdp/.bashrc ? [yes|no]
[no] >>> yes
```

After reopening a new terminal, we install Google TensorFlow with

```
[root@wiesbaden hdp]# pip install -U --ignore-installed setuptools
[root@wiesbaden hdp]# pip install --upgrade
   ↪    https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl
```

2.3 Configure networking

Now our system is ready for Hortonworks Data Platform deployment. However, in the deploying process, we need a proxy free network.

We create two network adapters, "Adapter 1" set to "Host-only Adapter" and "Adapter 2" using "NAT" (If you haven't already created a "Host-only Network", you should do that first, under the VirtualBox Preferences Network tab).

Log in as root. In a "Basic Server" install the nework adapters aren't enabled by default so we need to do this. Modify the /etc/sysconfig/network-scripts/ifcfg-enp0s3 and /etc/sysconfig/network-scripts/ifcfg-enp0s8 files. Change the boot setting to ONBOOT=yes for both files so the adapters are enabled when we boot. When this is done just enter the reboot command. Once you log back in you should see both adapters available when entering the ifconfig command. Make note of the IP address of the enp0s8 interface, we will need that to access the VM from the host machine and we will also use that network for our Ambari configuration.

```
[root@wiesbaden hdp]# hostname
wiesbaden
[root@wiesbaden hdp]# ifconfig
...
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe9a:7044  prefixlen 64  scopeid 0x20<link>
...
```

We'll add enp0s8 IP address and hostname to /etc/hosts:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.56.102   wiesbaden
```

2.4 Verify SSH settings

Now we need to verify SSH settings to check that the sshd service is running and enabled on boot:

```
[root@wiesbaden ~]#  service sshd status
[root@wiesbaden ~]#  chkconfig --list sshd
```

If it is not then run these commands:

```
[root@wiesbaden ~]#  chkconfig sshd on
[root@wiesbaden ~]#  service sshd start
```

2.5 Connect with SSH

Now we can log out and SSH to the server from a terminal on the host machine.

```
[hdp@wiesbaden ~]$ ssh root@192.168.56.102
```

As we log in as root through SSH the first time, we should see

```
[hdp@wiesbaden ~]$ ssh root@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is e3:92:52:2a:54:5b:8e:0b:b1:61:9a:7b:6f:b8:72:a8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
root@192.168.56.102's password:
Last login: Tue Jun  7 10:45:49 2016
ABRT has detected 1 problem(s). For more info run: abrt-cli list --since 1465289149
[root@wiesbaden ~]#
```

2.6 Update OpenSSL

Next step we will prepare for Ambari installation. Ambari is a cluster management for Hortonworks Data Platform. We update the installed OpenSSL package using:

```
[root@wiesbaden ~]#  yum -y update openssl
```

2.7 Set up Password-less SSH

It is a good idea to creat a password-less SSH, in order to do this we need to generate and store our SSH certificates in the .ssh/autorized_keys file on the local system. We can use ssh-keygen for this. We creat a password free SSH for the `root` account:

```
[root@wiesbaden hdp]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id\_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id\_rsa.
Your public key has been saved in /root/.ssh/id\_rsa.pub.
The key fingerprint is:
28:f6:d9:2b:d7:37:3b:b9:7b:30:68:a4:77:e0:bb:1f root@wiesbaden
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|       .  o      |
|    o . S+ o     |
|   . o o. = +    |
|      o .+ oE+   |
|      . ..o =..  |
|       o. .+=B   |
+-----------------+
[root@wiesbaden hdp]# cd /root/.ssh/
[root@wiesbaden .ssh]# cat /root/.ssh/id_rsa.pub >> authorized_keys
```

We show the private key value with

```
[root@wiesbaden hdp]# cat /root/.ssh/id_rsa
```

and use the key as an option during the installation.

Next, we do the same process for `hdp` account.

```
[hdp@wiesbaden ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hdp/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hdp/.ssh/id_rsa.
Your public key has been saved in /home/hdp/.ssh/id_rsa.pub.
The key fingerprint is:
1f:2f:1b:d4:de:26:f0:83:4f:13:82:f5:be:83:af:9c hdp@wiesbaden
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|          .      |
|         o o     |
|        S = +    |
|         o X o   |
|          =.X o  |
|         ..B.*   |
|          Eo+.   |
+-----------------+
[hdp@wiesbaden ~]$ su
Password:
[root@wiesbaden hdp]# cat /home/hdp/.ssh/id_rsa.pub >> ~/.ssh/authorizedkeys
[root@wiesbaden hdp]# chmod 600 ~/.ssh/authorized_keys
```

We creat a password free SSH for the `root` account:

```
[root@wiesbaden hdp]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id\_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id\_rsa.
Your public key has been saved in /root/.ssh/id\_rsa.pub.
The key fingerprint is:
28:f6:d9:2b:d7:37:3b:b9:7b:30:68:a4:77:e0:bb:1f root@wiesbaden
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|       .  o      |
|    o . S+ o     |
|   . o o. = +    |
|      o .+ oE+   |
|      . ..o =..  |
|       o. .+=B   |
+-----------------+
[root@wiesbaden hdp]# cd /root/.ssh/
[root@wiesbaden .ssh]# cat /root/.ssh/id_rsa.pub >> authorized_keys
```

We show the private key value with

```
[root@wiesbaden hdp]# cat /root/.ssh/id_rsa
```

and use the key as an option during the installation.

We should now be able to create an SSH connection to wiesbaden without being prompted for a password (if you get a prompt for adding the host to the known hosts file, just enter yes):

```
[hdp@wiesbaden ~]$ ssh root@192.168.56.102
Last login: Mon Jun  6 15:41:18 2016 from wiesbaden
[root@wiesbaden ~]# exit
logout
Connection to 192.168.56.102 closed.
```

2.8 Enable ntpd

Next, Ambari needs time to be synchronized so we need to enable ntpd.

```
[root@wiesbaden hdp]# chkconfig ntpd on
Note: Forwarding request to 'systemctl enable ntpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to
    ↪   /usr/lib/systemd/system/ntpd.service.
[root@wiesbaden hdp]# ntpdate pool.ntp.org
 6 Jun 15:44:25 ntpdate[7998]: adjust time server 131.188.3.221 offset 0.208360 sec
[root@wiesbaden hdp]# service ntpd start
Redirecting to /bin/systemctl start  ntpd.service
```

2.9 Disable iptables

Then we disable iptables. A number of network ports need to be open on the VM for Ambari during setup. The easiest way to open the ports is to disable the iptables process:

```
# service iptables stop
# chkconfig iptables off
```

2.10 Disable SELinux and Check umask Value

Now we disable SELinux and Check umask Value. To disable SELinux during the Ambari setup:

```
# setenforce 0
```

To permanently disable SELinux for the VM modify /etc/sysconfig/selinux and change the config to SELINUX=disabled. Also, make sure umask is set to 0022 (it should be for a new install)

```
# umask
0022
```

2.11 Disable IPv6

Then we need to disable IPv6, log in as root and cut-and-paste the following commands into your terminal window to disable IPv6:

```
mkdir -p /etc/sysctl.d
( cat > /etc/sysctl.d/99-hadoop-ipv6.conf <<-’EOF’
## Disabled ipv6
## Provided by Ambari Bootstrap
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
EOF
    )
sysctl -e -p /etc/sysctl.d/99-hadoop-ipv6.conf
```

2.12 Disable Transparent Huge Pages (THP)

When installing Ambari, one or more host checks may fail if you have not disabled Transparent Huge Pages on all hosts.

To disable THP log in as root and add the following commands to your /etc/rc.local file:

```
if test -f /sys/kernel/mm/redhat_transparent_hugepage/defrag;
  then echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
fi
if test -f /sys/kernel/mm/redhat_transparent_hugepage/enabled;
  then echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
fi
```

To confirm, reboot the host and then run the command:

```
[root@wiesbaden hdp]# cat /sys/kernel/mm/transparent_hugepage/enabled
[always] madvise never
```

2.13 Install httpd

We need to have the web server running so log in as root and install it with the following commands:

```
[root@wiesbaden hdp]# yum -y install httpd
```

We'll set the ServerName to be ${hostname}:80 in /etc/httpd/conf/httpd.conf

```
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName ${hostname}:80
```

Now we can start the httpd server.

```
chkconfig httpd on
service httpd start
```

Congratulations! We finally move to step for Ambari installation.

2.14 Set up Ambari yum repo

Download the Ambari repository file to a directory on your installation host.

```
[root@wiesbaden hdp]# wget -nv
   ↪    http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.2.2.0/ambari.repo -O
   ↪    /etc/yum.repos.d/ambari.repo
```

Confirm that the repository is configured by checking the repo list.

```
[root@wiesbaden hdp]# yum repolist
Loaded plugins: fastestmirror, langpacks
Updates-ambari-2.2.2.0                                   | 2.9 kB     00:00
Updates-ambari-2.2.2.0/primary_db                          | 6.3 kB   00:00
Loading mirror speeds from cached hostfile
 * base: wftp.tu-chemnitz.de
 * extras: ftp.plusline.de
 * updates: centos.schlundtech.de
repo id                             repo name                              status
Updates-ambari-2.2.2.0              ambari-2.2.2.0 - Updates                   8
base/7/x86_64                       CentOS-7 - Base                        9,007
extras/7/x86_64                     CentOS-7 - Extras                        305
updates/7/x86_64                    CentOS-7 - Updates                     1,683
repolist: 11,003
```

2.15 Install Ambari server packages

Install the Ambari packages. This also installs the default PostgreSQL Ambari database. Enter y when prompted to to confirm transaction and dependency checks.

```
[root@wiesbaden hdp]# yum install ambari-server
...
  Installing : postgresql-libs-9.2.15-1.el7_2.x86_64                        1/4
  Installing : postgresql-9.2.15-1.el7_2.x86_64                             2/4
  Installing : postgresql-server-9.2.15-1.el7_2.x86_64                      3/4
  Installing : ambari-server-2.2.2.0-460.x86_64                             4/4
  Verifying  : postgresql-libs-9.2.15-1.el7_2.x86_64                        1/4
  Verifying  : postgresql-server-9.2.15-1.el7_2.x86_64                      2/4
  Verifying  : ambari-server-2.2.2.0-460.x86_64                             3/4
  Verifying  : postgresql-9.2.15-1.el7_2.x86_64                             4/4

Installed:
  ambari-server.x86_64 0:2.2.2.0-460

Dependency Installed:
  postgresql.x86_64 0:9.2.15-1.el7_2
  postgresql-libs.x86_64 0:9.2.15-1.el7_2
  postgresql-server.x86_64 0:9.2.15-1.el7_2

Complete!
```

Note: Accept the warning about trusting the Hortonworks GPG Key. That key will be automatically downloaded and used to validate packages from Hortonworks. You will see the following message:

```
Retrieving key from http://public-repo-1.hortonworks.com/ambari/centos7/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
Importing GPG key 0x07513CAD:
 Userid     : "Jenkins (HDP Builds) <jenkin@hortonworks.com>"
 Fingerprint: df52 ed4f 7a3a 5882 c099 4c66 b973 3a7a 0751 3cad
 From       : http://public-repo-1.hortonworks.com/ambari/centos7/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
Is this ok [y/N]: y
```

2.16 Configure and start the Ambari server

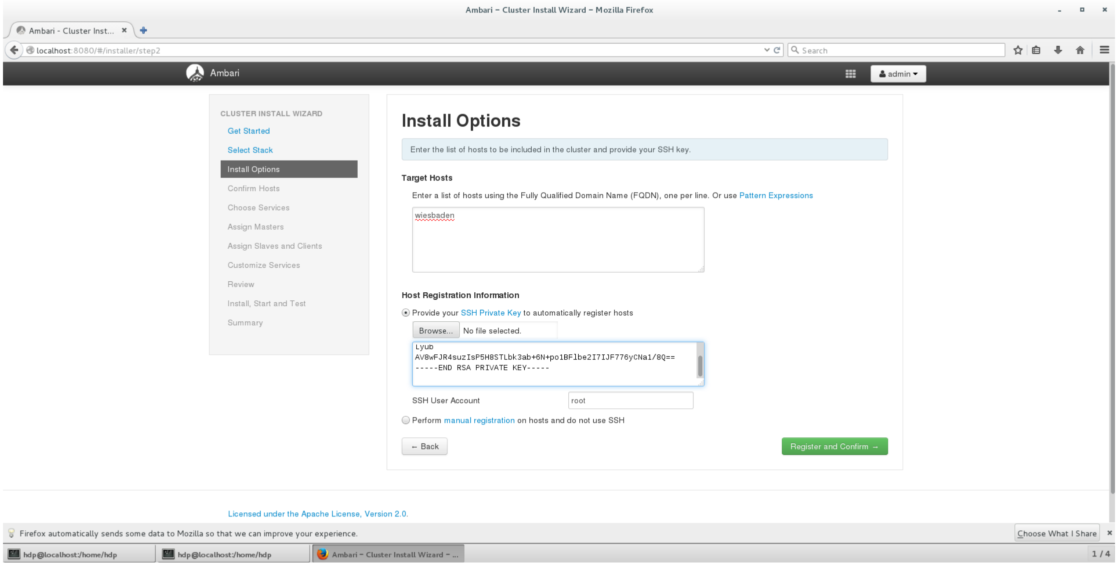
Before starting the Ambari Server, you must set it up. Setup configures Ambari to talk to the Ambari database, installs the JDK and allows you to customize the user account the Ambari Server daemon will run as. The ambari-server setup command manages the setup process. Run the following command on the Ambari server host to start the setup process. You may also append Setup Options to the command.

```
[root@wiesbaden hdp]# ambari-server setup
Using python  /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'enabled'
SELinux mode is 'permissive'
WARNING: SELinux is set to 'permissive' mode and temporarily disabled.
OK to continue [y/n] (y)? y
Customize user account for ambari-server daemon [y/n] (n)? n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Redirecting to /bin/systemctl status  iptables.service

Checking JDK...
[1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
[2] Oracle JDK 1.7 + Java Cryptography Extension (JCE) Policy Files 7
[3] Custom JDK
==============================================================================
Enter choice (1): 1
To download the Oracle JDK and the Java Cryptography Extension (JCE) Policy Files you must accept the
    ↪   license terms found at http://www.oracle.com/technetwork/java/javase/terms/license/index.html and
    ↪   not accepting will cancel the Ambari Server setup and you must install the JDK and JCE files
    ↪   manually.
Do you accept the Oracle Binary Code License Agreement [y/n] (y)? y
Downloading JDK from http://public-repo-1.hortonworks.com/ARTIFACTS/jdk-8u60-linux-x64.tar.gz to
    ↪   /var/lib/ambari-server/resources/jdk-8u60-linux-x64.tar.gz
jdk-8u60-linux-x64.tar.gz... 100% (172.8 MB of 172.8 MB)
Successfully downloaded JDK distribution to /var/lib/ambari-server/resources/jdk-8u60-linux-x64.tar.gz
Installing JDK to /usr/jdk64/
Successfully installed JDK to /usr/jdk64/
Downloading JCE Policy archive from http://public-repo-1.hortonworks.com/ARTIFACTS/jce_policy-8.zip to
    ↪   /var/lib/ambari-server/resources/jce_policy-8.zip

Successfully downloaded JCE Policy archive to /var/lib/ambari-server/resources/jce_policy-8.zip
Installing JCE policy...
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? n
Configuring database...
Default properties detected. Using built-in database.
Configuring ambari database...
Checking PostgreSQL...
Running initdb: This may take upto a minute.
Initializing database ... OK


About to start PostgreSQL
Configuring local database...
Connecting to local database...done.
Configuring PostgreSQL...
Restarting PostgreSQL
Extracting system views...
ambari-admin-2.2.2.0.460.jar
......
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.
```

Now, start the Ambari server:

```
[root@wiesbaden hdp]# ambari-server start
```

We can now connect to Ambari server on `localhost:8080`, username is `admin` and password is `admin`. Now it's the time to create our own Hadoop Cluster. From the Ambari Dashboard click the Launch Install Wizard link. We configured our single-node VMs hostname as the only host. We name our cluster as `xip`



Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions

wiesbaden

Host Registration Information

● Provide your SSH Private Key to automatically register hosts

No file selected.

```
Lyub
AV8wFJR4suzIsP5H8STLbk3ab+6N+po1BFlbe2I7IJF776yCNa1/8Q==
-----END RSA PRIVATE KEY-----
```

SSH User Account: root

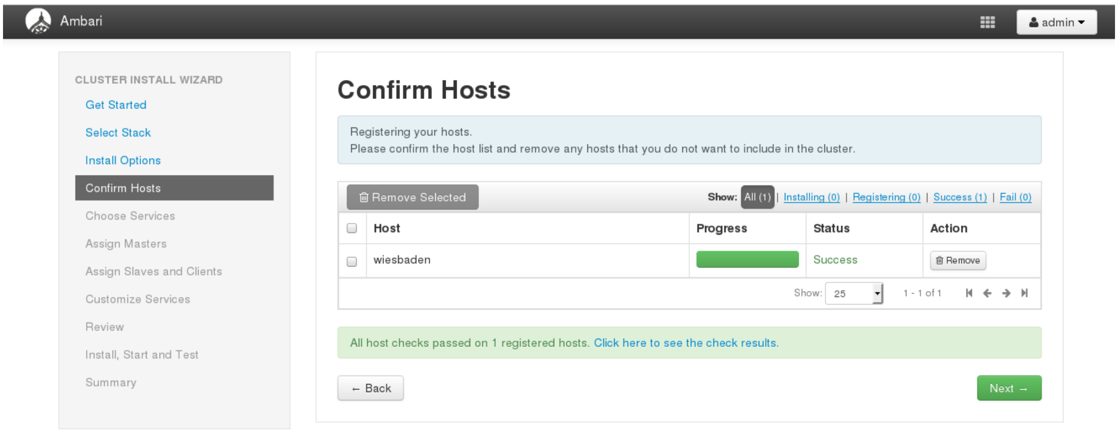
○ Perform manual registration on hosts and do not use SSH

Licensed under the Apache License, Version 2.0.

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 3: Install Options

With the correct setting in Fig. 3, we get the Fig. 4



Ambari

admin

CLUSTER INSTALL WIZARD
- Get Started
- Select Stack
- Install Options
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test
- Summary

Confirm Hosts

Registering your hosts.
Please confirm the host list and remove any hosts that you do not want to include in the cluster.

Remove Selected

Show: All (1) | Installing (0) | Registering (0) | Success (1) | Fail (0)

□	Host	Progress	Status	Action
□	wiesbaden	🟩	Success	Remove

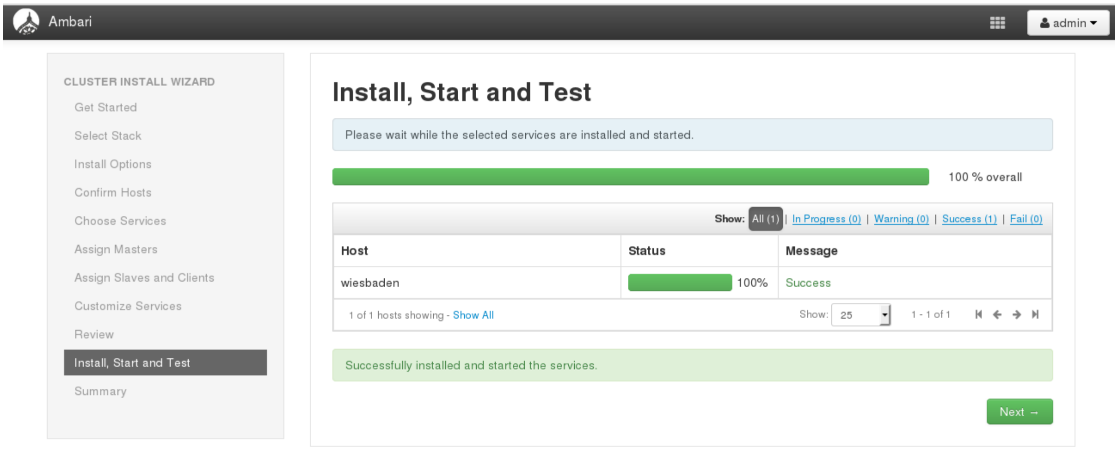
Show: 25 1 - 1 of 1

All host checks passed on 1 registered hosts. Click here to see the check results.

← Back Next →

Figure 4: Confirm Hosts

After choosing the services we want to install on our cluster, we assign the masters all with `wiesbaden` due to the single node. With several times `Next` clicks, we are finally able to deploy our single node cluster.



Ambari

admin

CLUSTER INSTALL WIZARD
- Get Started
- Select Stack
- Install Options
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test
- Summary

Install, Start and Test

Please wait while the selected services are installed and started.

100 % overall

Show: All (1) | In Progress (0) | Warning (0) | Success (1) | Fail (0)

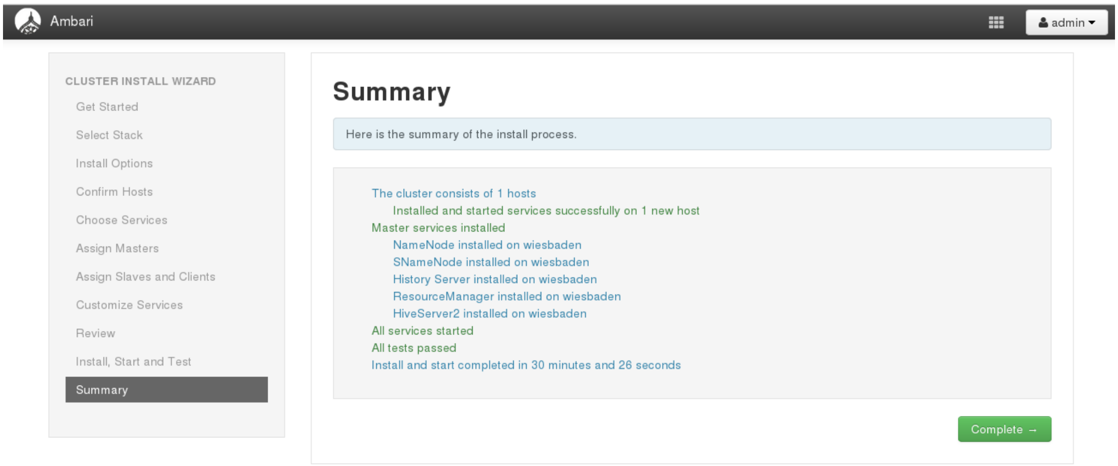
Host	Status	Message
wiesbaden	🟩 100%	Success

1 of 1 hosts showing - Show All Show: 25 1 - 1 of 1

Successfully installed and started the services.

Next →

Figure 5: Install, Start and Test



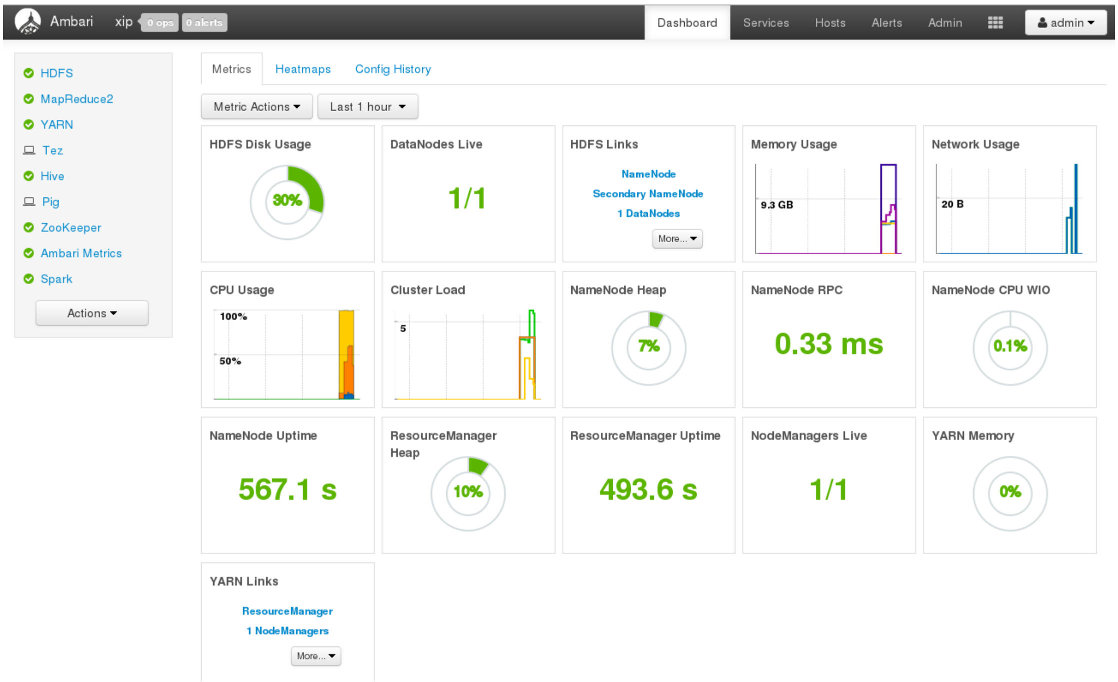
Summary

Here is the summary of the install process.

The cluster consists of 1 hosts
    Installed and started services successfully on 1 new host
Master services installed
    NameNode installed on wiesbaden
    SNameNode installed on wiesbaden
    History Server installed on wiesbaden
    ResourceManager installed on wiesbaden
    HiveServer2 installed on wiesbaden
All services started
All tests passed
Install and start completed in 30 minutes and 26 seconds

Figure 6: Summary

With a successful install process we get a summary: all services started and all tests passed. The following screenshot shows how the Ambari dashboard looks like:



HDFS Disk Usage 30%
DataNodes Live 1/1
HDFS Links
Memory Usage
Network Usage
CPU Usage
Cluster Load
NameNode Heap 7%
NameNode RPC 0.33 ms
NameNode CPU WIO 0.1%
NameNode Uptime 567.1 s
ResourceManager Heap 10%
ResourceManager Uptime 493.6 s
NodeManagers Live 1/1
YARN Memory 0%
YARN Links

Figure 7: The Ambari Dashboard

2.17 Configuring Jupyter Notebook

Since we want to use IPython with Apache Spark we have to use the Python interpreter which is built with Apache Spark, `pyspark`, instead of the default Python interpreter. We begin with upgrading the Jupyter modules

```
[root@wiesbaden hdp]# pip install --upgrade jupyter-client
[root@wiesbaden hdp]# pip install --upgrade jupyter-console
[root@wiesbaden hdp]# pip install --upgrade jupyter-core
```

Let's create a Jupyter Notebook profile for pyspark

```
[root@wiesbaden hdp]# ipython profile create pyspark
```

Next generate a jupyter config file:

```
[root@wiesbaden hdp]# jupyter notebook --generate-config
```

You should see the following output:

```
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
```

Now open your preferred editor. I'm using `vi` to edit `jupyter_notebook_config.py`. Let's change some settings as following:

```
# The directory to use for notebooks and kernels.
c.NotebookApp.notebook_dir = u'/usr/hdp/current/spark-client/'

# The port the notebook server will listen on.
c.NotebookApp.port = 8889

# The IP address the notebook server will listen on.
c.NotebookApp.ip = '127.0.0.1'
```

Next we are going to create a shell script to set the appropriate values every time we want to start Jupyter Notebook with `pyspark` enviroment. We create a shell script with the following command:

```
[root@wiesbaden hdp]# vi ~/start_ipython_notebook.sh
```

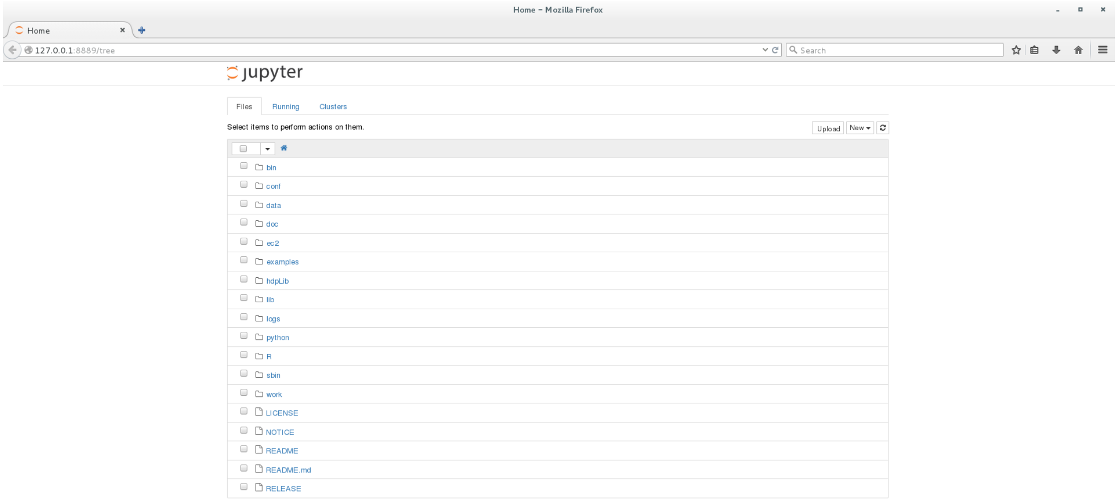
Then copy the following lines into the file:

```
#!/bin/bash
IPYTHON_OPTS="notebook" pyspark
```

Running

```
[root@wiesbaden hdp]# chmod +x ~/start_ipython_notebook.sh
[root@wiesbaden hdp]# ~/start_ipython_notebook.sh
```

A browser will be automatically opened on your host machine and the URl `http://127.0.0.1:8889` shows up and you should see the screen below:



Select items to perform actions on them.

- bin
- conf
- data
- doc
- ec2
- examples
- hdpLib
- lib
- logs
- python
- R
- sbin
- work
- LICENSE
- NOTICE
- README
- README.md
- RELEASE

Figure 8: Jupyter Notebook

Voila! We have just configured jupyter notebook with Apache Spark on the Hortonworks Data Platform.

3 Summary

In this document, we have seen how to deploy the Hortonworks Data Platform on a single node cluster for vibration data analysis. In our future works, we would like to scale out the cluster for more intensive computing and storage tasks.