Hadoop on Palmetto

CPSC 3620

Linh B. Ngo

Software preparation
(should already be done before class)

- In your home directory on Palmetto, create a directory called *software*:

```
mkdir  ~/software
```

- Copy the following files into the newly created directory

```
cp /scratch1/lngo/classes/cpsc3620/hdp/hdp.tar.gz  ~/software
cp /scratch1/lngo/classes/cpsc3620/hdp/java.tar.gz  ~/software
cp /scratch1/lngo/classes/cpsc3620/hdp/scala.tar.gz  ~/software
cp /scratch1/lngo/classes/cpsc3620/hdp/spark.tar.gz  ~/software
```

- Decompressed the tar files:

```
cd  ~/software
tar -xzf  *.tar.gz
```

- After decompression, in your *software* directory there should be two subdirectories:

```
hadoop-2.2.0.2.1.0.0-92
jdk1.7.0_25
scala-2.10.4
spark-1.4.1-bin-without-hadoop
```

Data preparation
(should already be done before class)

- ssh –X into the first node of your Hadoop cluster
- Create an example directory in your home directory on Palmetto called *mapreduce*, and copy two examples into this directory (should already been done from MapReduce Lab)

```
mkdir   ~/mapreduce
cp -R /scratch1/lngo/classes/cpsc3620/hdp/shakespeare  ~/mapreduce
cp -R /scratch1/lngo/classes/cpsc3620/hdp/airline   ~/mapreduce
```

Management preparation
(should already be done before class)

- Copy the following directory to your **home** directory and decompress

```
cd    ~
cp    /scratch1/lngo/classes/cpsc3620/hdp/hdp2.2.tar.gz    ~
tar   xzf   hdp2.2.tar.gz
```

- Edit your .bashrc file

```
vim    ~/.bashrc
```

- Add the following line to the end of your .bashrc file:

```
source    /home/$USER/hdp-2.2/bin/setenv.sh
```

Start Hadoop on Palmetto

- Go to the **`hdp-2.2`** directory and start up the PBS job for Hadoop cluster

```
cd    ~/hdp-2.2
qsub    start-hadoop.pbs
```

Design of PBS Hadoop

- Reserve a set of nodes from Palmetto through PBS submission script
- Paths to Hadoop deployment directories are configured via environment variables in **setenv.sh**
- First node in the PBS_NODEFILE: NameNode
- Second node in the PBS_NODEFILE: Resource Manager (YARN)
- The remaining nodes: DataNode/NodeManager
- All nodes share the same XML configuration files inside **/home/$USER/hdp-2.2/config**
- The configuration files are populated from a set of templates (**/home/$USER/hdp-2.2/config_templates**) through **bin/pbs-configure.sh**, which is called in the PBS submission script
- Additional Hadoop-based packages can be stacked onto this implementation following similar configuration principle.

lngo@user001:~

```
[lngo@user001 ~]$ qstat -anu lngo

pbs01:

                                                            Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
5164085.pbs01   lngo     bigdata_ myHadoop     7455   4  64  240gb 10:00 R 01:35
   node1679/0*16+node1771/0*16+node1772/0*16+node1745/0*16
[lngo@user001 ~]$
```

qstat –anu $USER

First Node

Second Node

```
[lngo@node1766 AirTraffic]$ hadoop fs
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]
        [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-count [-q] [-h] <path> ...]
        [-cp [-f] [-p | -p[topax]] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] <path> ...]
        [-expunge]
        [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-getfacl [-R] <path>]
        [-getfattr [-R] {-n name | -d} [-e en] <path>]
        [-getmerge [-nl] <src> <localdst>]
        [-help [cmd ...]]
        [-ls [-d] [-h] [-R] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] [-l] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
…
```

File system commands

- Create a new directory on your HDFS cluster

  hadoop fs -mkdir /airline

- View this directory

  hadoop fs -ls /

- Upload data to this directory

  hadoop fs –copyFromLocal ~/mapreduce/airline /airline

- View data

  hadoop fs -cat /airline/data/1987.csv

```
[lngo@node1766 AirTraffic]$ hdfs
Usage: hdfs [--config confdir] COMMAND
     where COMMAND is one of:
  dfs                  run a filesystem command on the file systems supported in Hadoop.
  namenode -format     format the DFS filesystem
  secondarynamenode    run the DFS secondary namenode
  namenode             run the DFS namenode
  journalnode          run the DFS journalnode
  zkfc                 run the ZK Failover Controller daemon
  datanode             run a DFS datanode
  dfsadmin             run a DFS admin client
  haadmin              run a DFS HA admin client
  fsck                 run a DFS filesystem checking utility
  balancer             run a cluster balancing utility
  jmxget               get JMX exported values from NameNode or DataNode.
  mover                run a utility to move block replicas across
                       storage types
  oiv                  apply the offline fsimage viewer to an fsimage
  oiv_legacy           apply the offline fsimage viewer to an legacy fsimage
  oev                  apply the offline edits viewer to an edits file
  fetchdt              fetch a delegation token from the NameNode
  getconf              get config values from configuration
  groups               get the groups which users belong to
  snapshotDiff         diff two snapshots of a directory or diff the
                       current directory contents with a snapshot
  lsSnapshottableDir   list all snapshottable dirs owned by the current user
                                        Use -help to see options
  portmap              run a portmap service
  nfs3                 run an NFS version 3 gateway
  cacheadmin           configure the HDFS cache
  crypto               configure HDFS encryption zones
  storagepolicies      get all the existing block storage policies
  version              print the version
```

HDFS command

- Check for status of your uploaded files

```
hdfs fsck /user/<username>/airline/data -blocks -files
-locations
```

Configuration options

- core-site.xml
- mapred-site.xml
- hdfs-site.xml
- yarn-site.xml

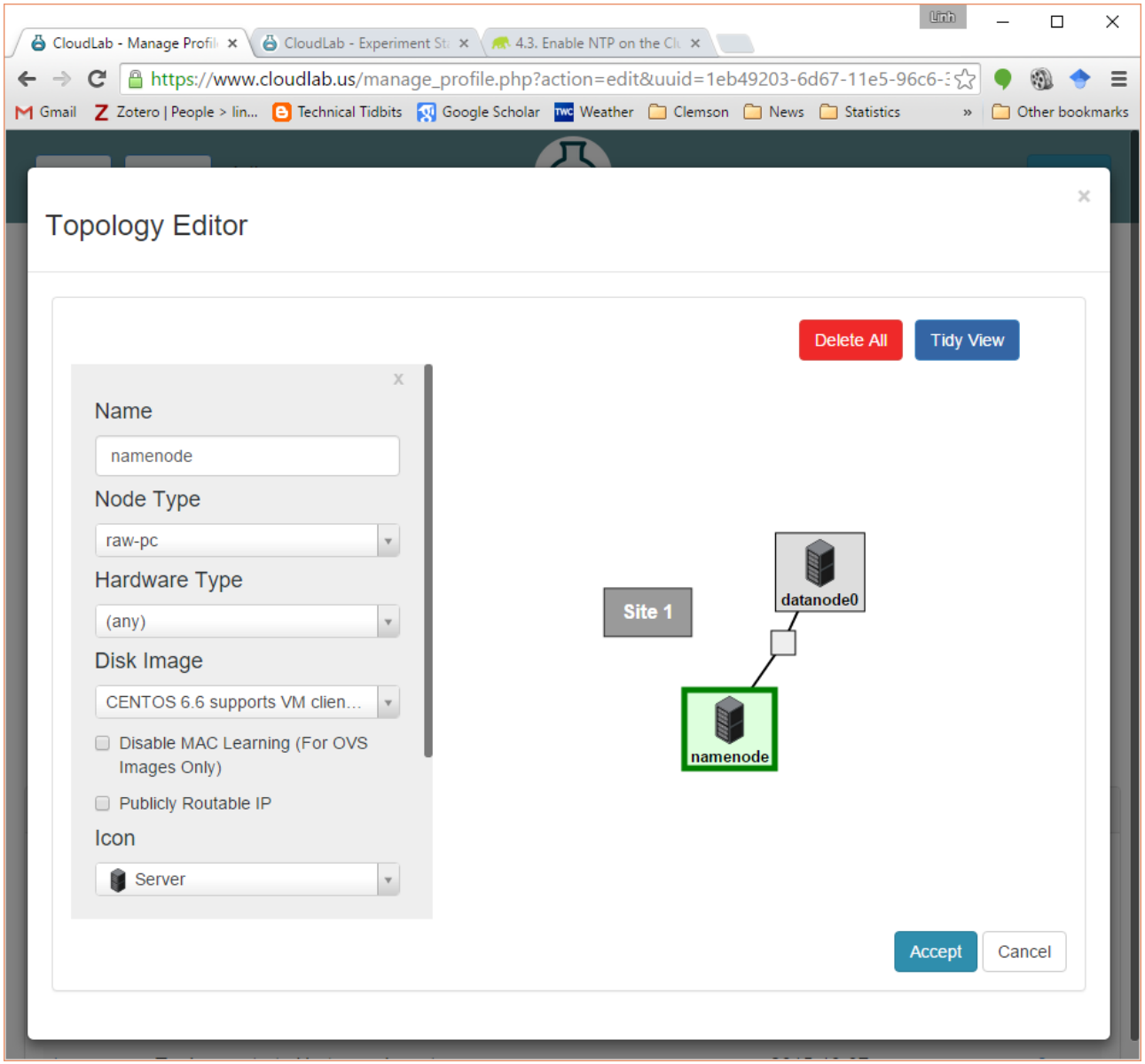
Monitoring Hadoop

- Set up X11 tunneling for SSH: http://citi.clemson.edu/palmetto/userguide/#X11_Tunneling_with_SSH_for_Palmetto
- Must ssh –X into Palmetto
- ssh –X to the first node in your qstat window
- From the headnode, type:
  firefox &
- To monitor HDFS in the X11 Firefox windows, go to URL
  <first node>:50070
- To monitor YARN in the X11 Firefox windows
  <second”>:8088

Hadoop on CloudLab

- Enterprise Hadoop
- Hortonworks
- http://hortonworks.com/hdp/downloads/

Set up 2-node Cluster on CloudLab



On each node

- SSH onto the node from Palmetto
- Change to root:

```
sudo su -
```

- Execute the following commands:

```
chkgconfig -list ntpd
chkconfig ntpd on
service ntpd start
chkconfig iptables off
/etc/init.d/iptables stop
setenforce 0
```

On each node

- Setup Ambari download server

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/2.1.2/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- On namenode

```
yum -y install ambari-server
yum -y install ambari-agent
```

- On datanode

```
yum -y install ambari-agent
```

On namenode

- Set up ambari server:

```
ambari-server setup
```

- Select default for all questions
- Select 1 for JDK version
- When all done, start ambari server

```
ambari-server start
```

On each node

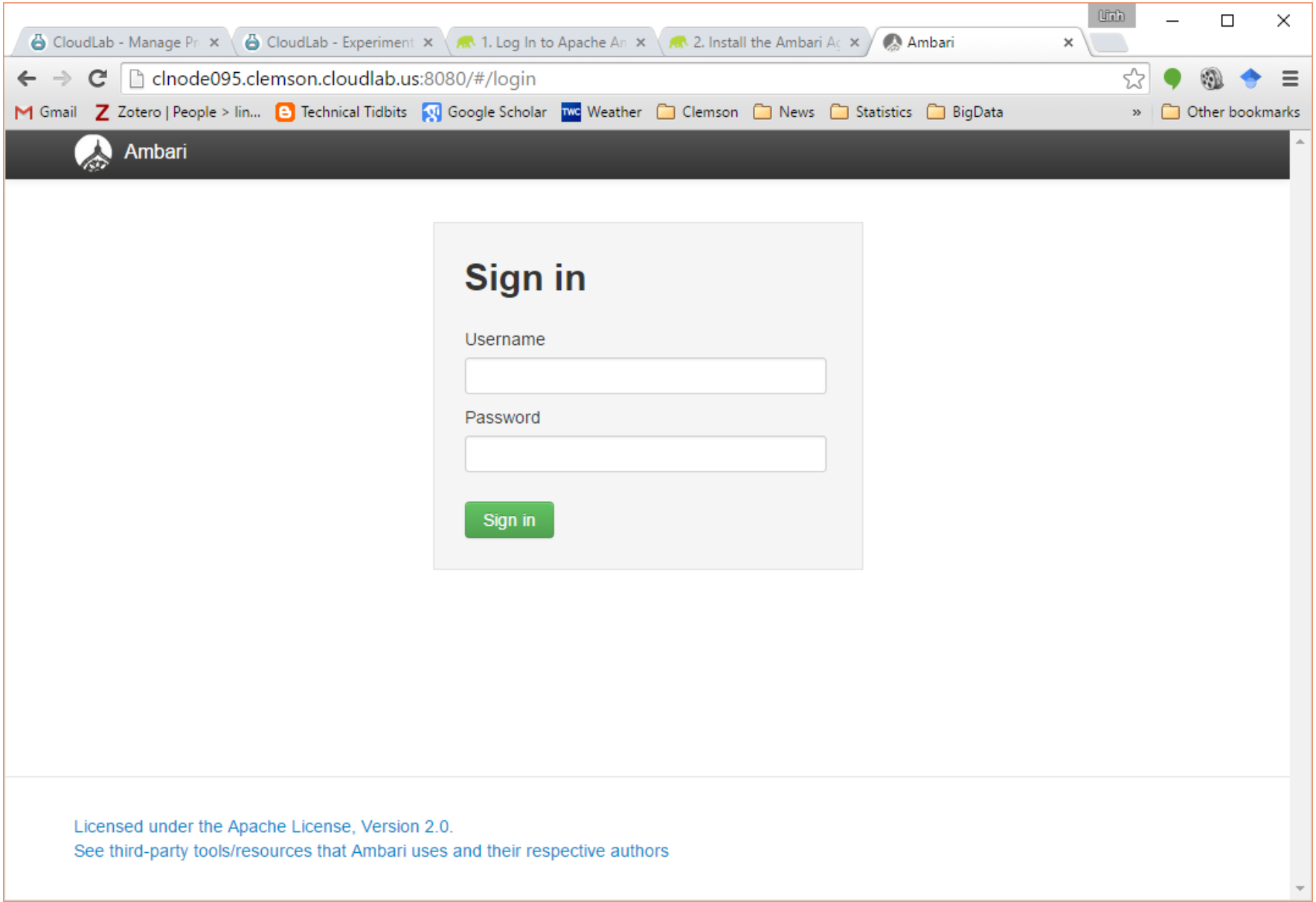
- Using vim to edit `/etc/ambari-agent/conf/ambari-agent.ini`
- Change:

```
hostname=<hostname of namenode as shown in list view of CloudLab>
```

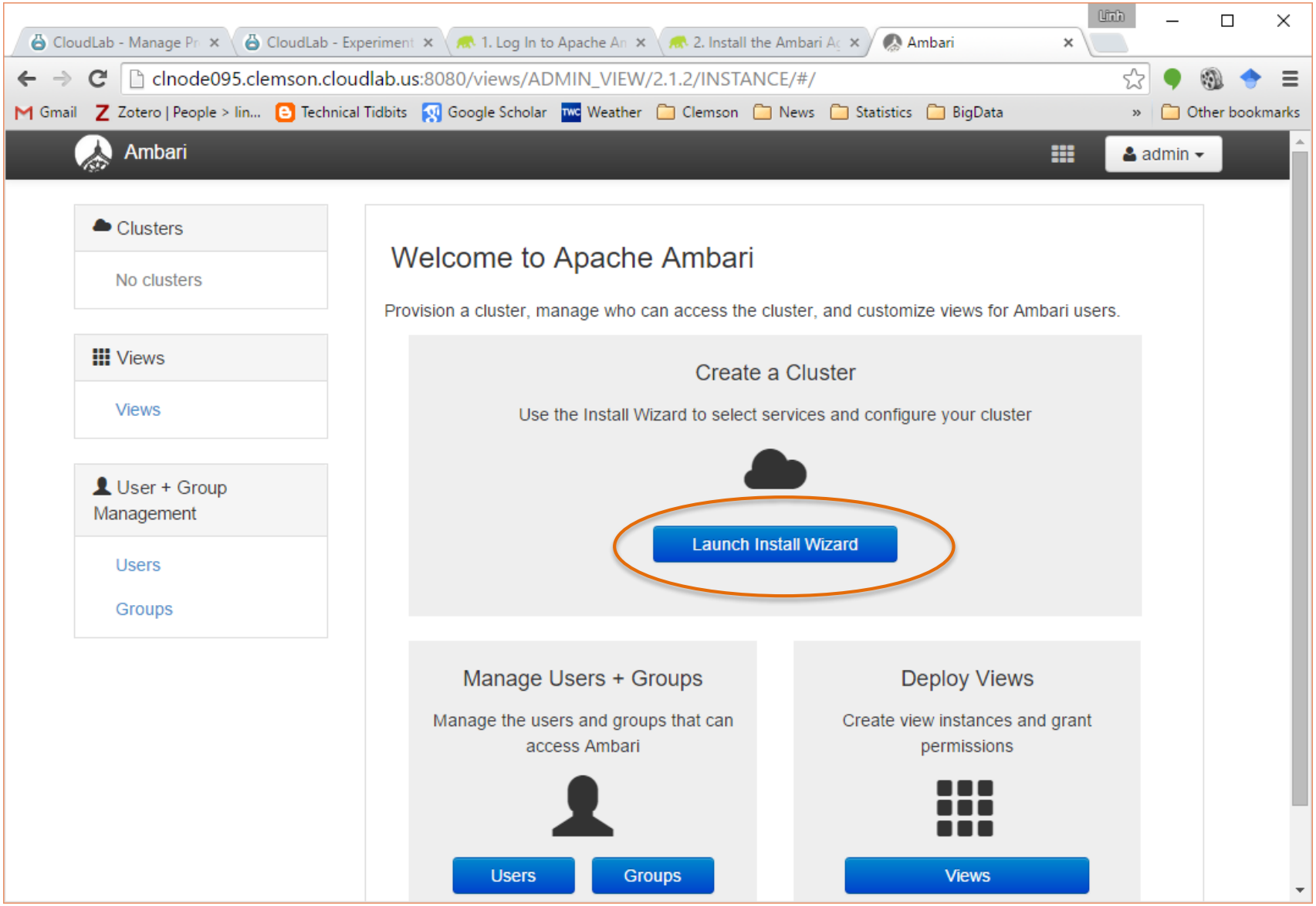
- Start Ambari Agent

```
ambari-agent start
```

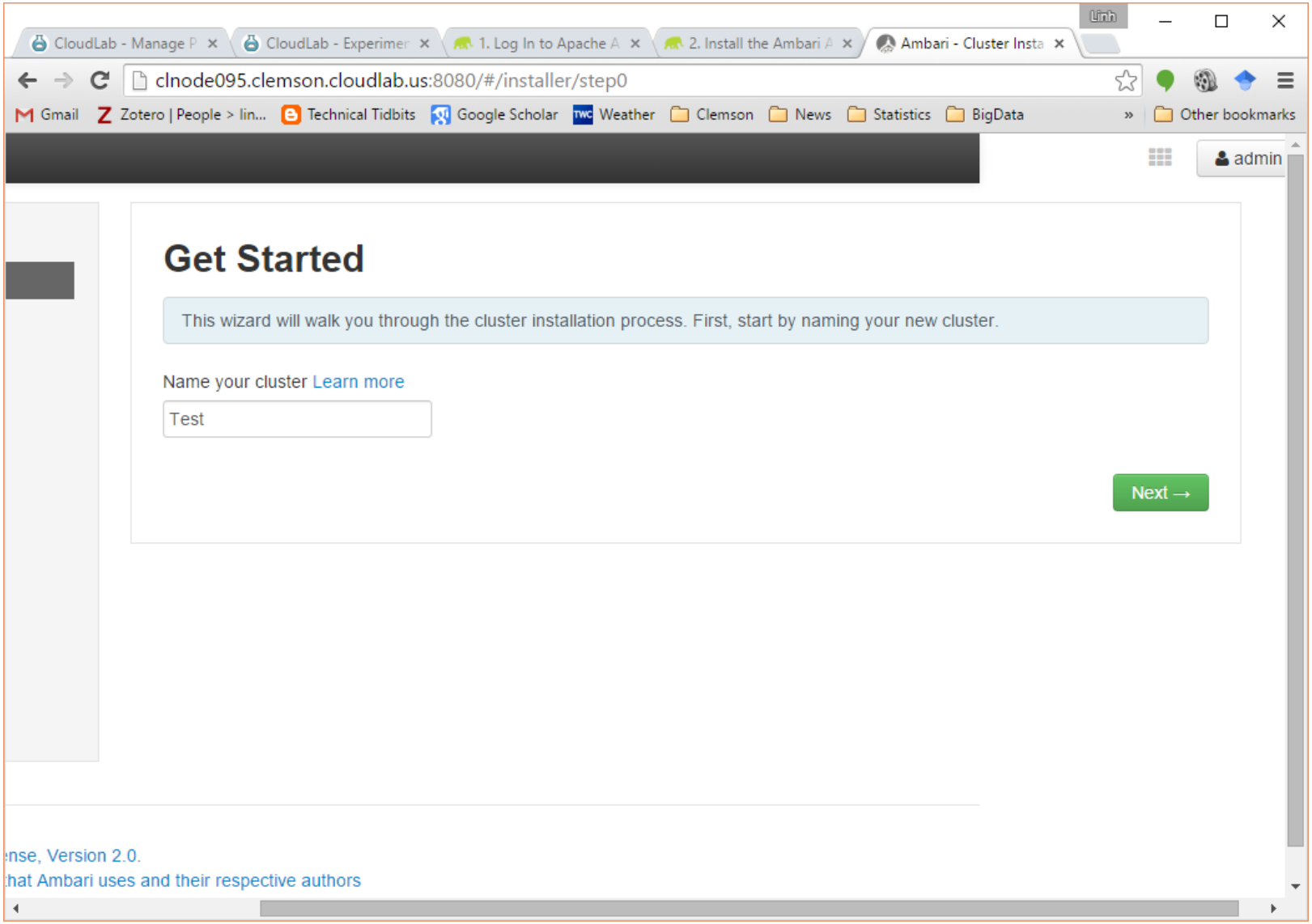
Ambari Server (admin/admin)



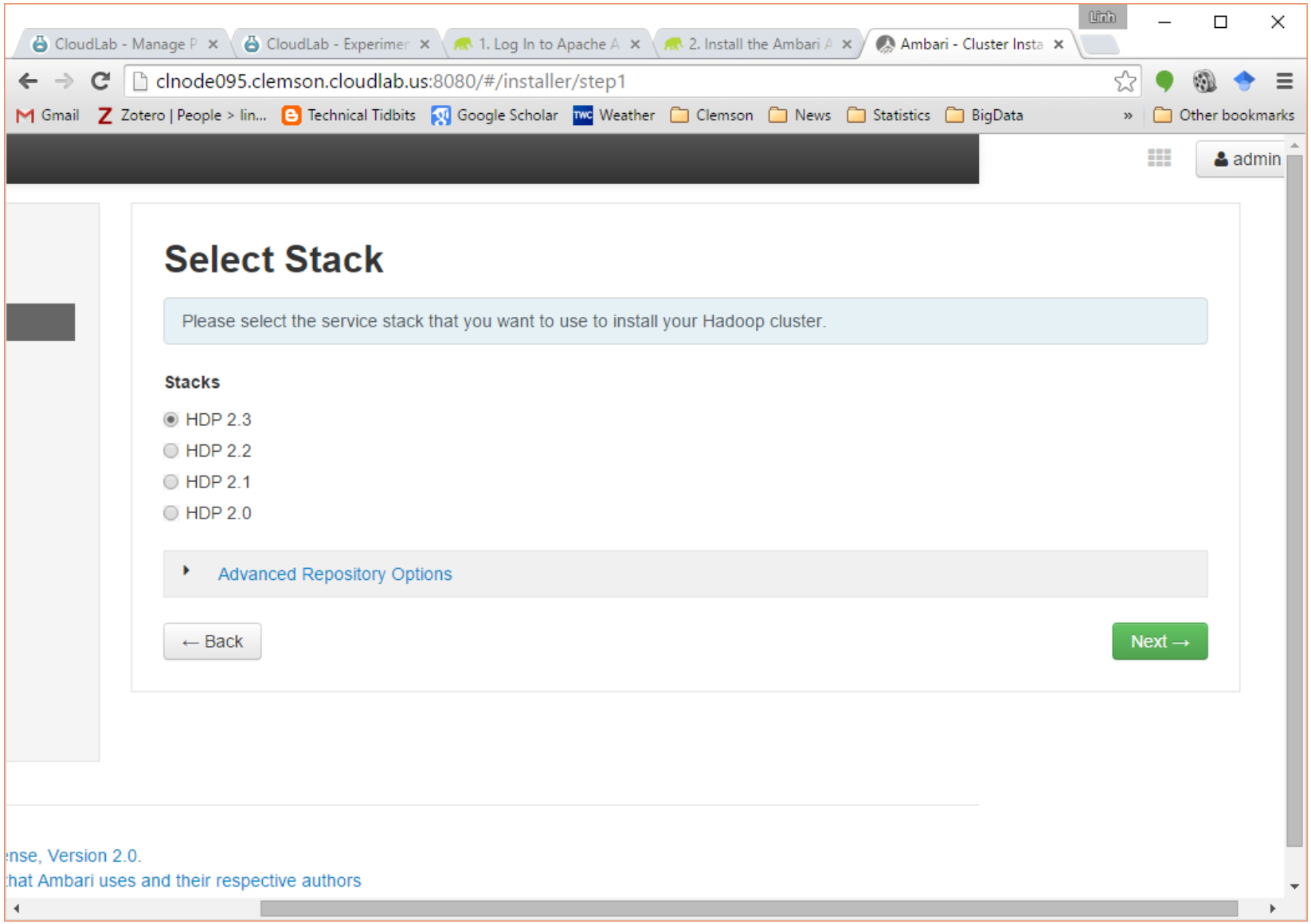
Ambari Server (admin/admin)



Ambari Server (admin/admin)



Ambari Server (admin/admin)



Ambari Server (admin/admin)

CloudLab - Experiment St | 1. Log In to Apache Amba | 2. Install the Ambari Agen | Ambari - Cluster Install Wi

clnode095.clemson.cloudlab.us:8080/#/installer/step2

Gmail | Zotero | People > lin... | Technical Tidbits | Google Scholar | Weather | Clemson | News | Statistics | BigData | Other bookmarks

admin

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

```
clnode095.clemson.cloudlab.us
clnode092.clemson.cloudlab.us
```

Host Registration Information

Provide your SSH Private Key to automatically register hosts

Choose File No file chosen

ssh private key

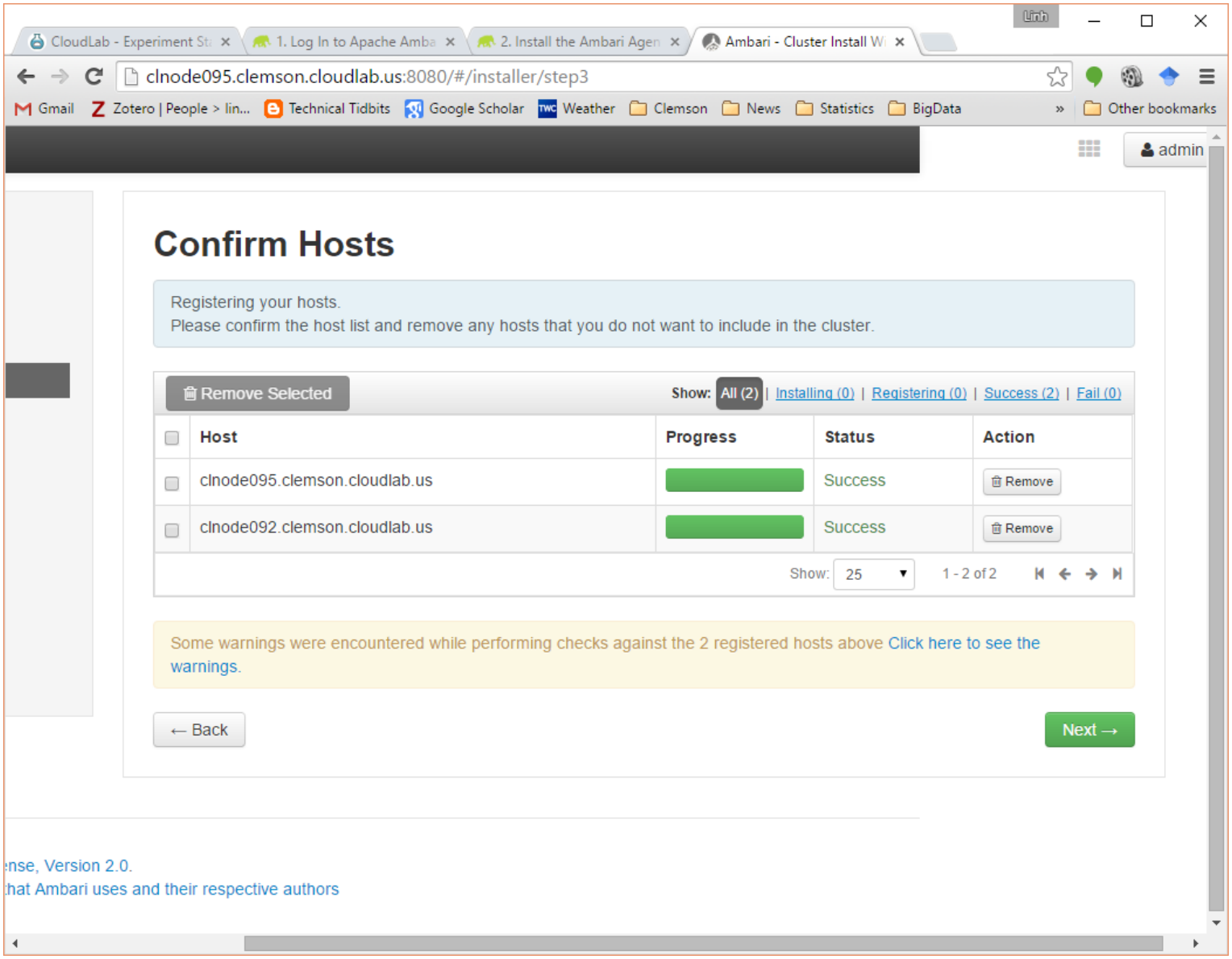
SSH User Account root

Perform manual registration on hosts and do not use SSH

← Back

Register and Confirm →

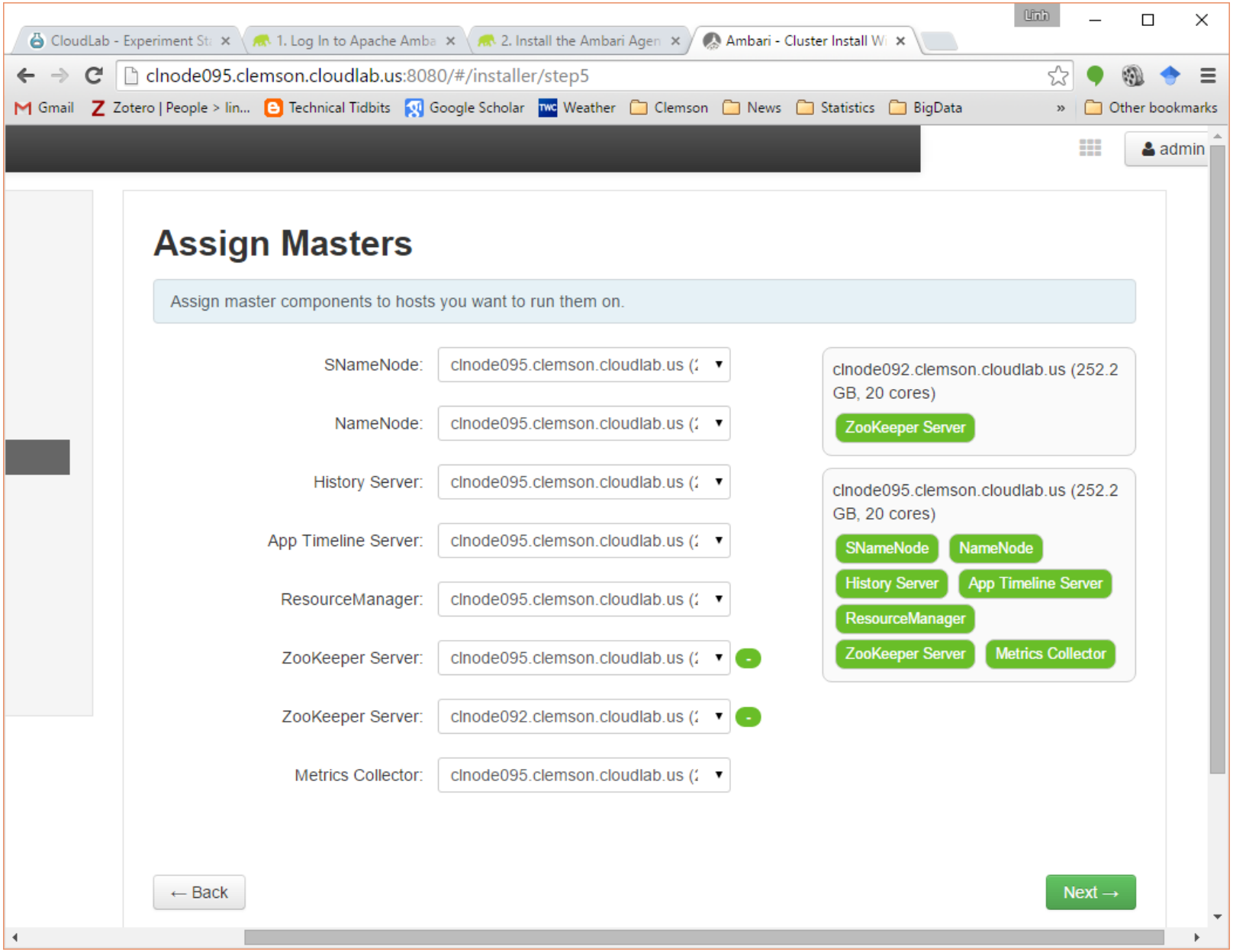
Assuming you had ambari agents up and running …



Ambari Server (admin/admin)

- HDFS
- YARN+MapReduce2
- Tez
- ZooKeeper
- Ambari Metrics

Ambari Server (admin/admin)



Ambari Server (admin/admin)

CloudLab - Experiment St | 1. Log In to Apache Amba | 2. Install the Ambari Agen | Ambari - Cluster Install Wi

clnode095.clemson.cloudlab.us:8080/#/installer/step6

Gmail | Zotero | People > lin... | Technical Tidbits | Google Scholar | Weather | Clemson | News | Statistics | BigData | Other bookmarks

admin

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✱.
"Client" will install HDFS Client, MapReduce2 Client, YARN Client, Tez Client and ZooKeeper Client.

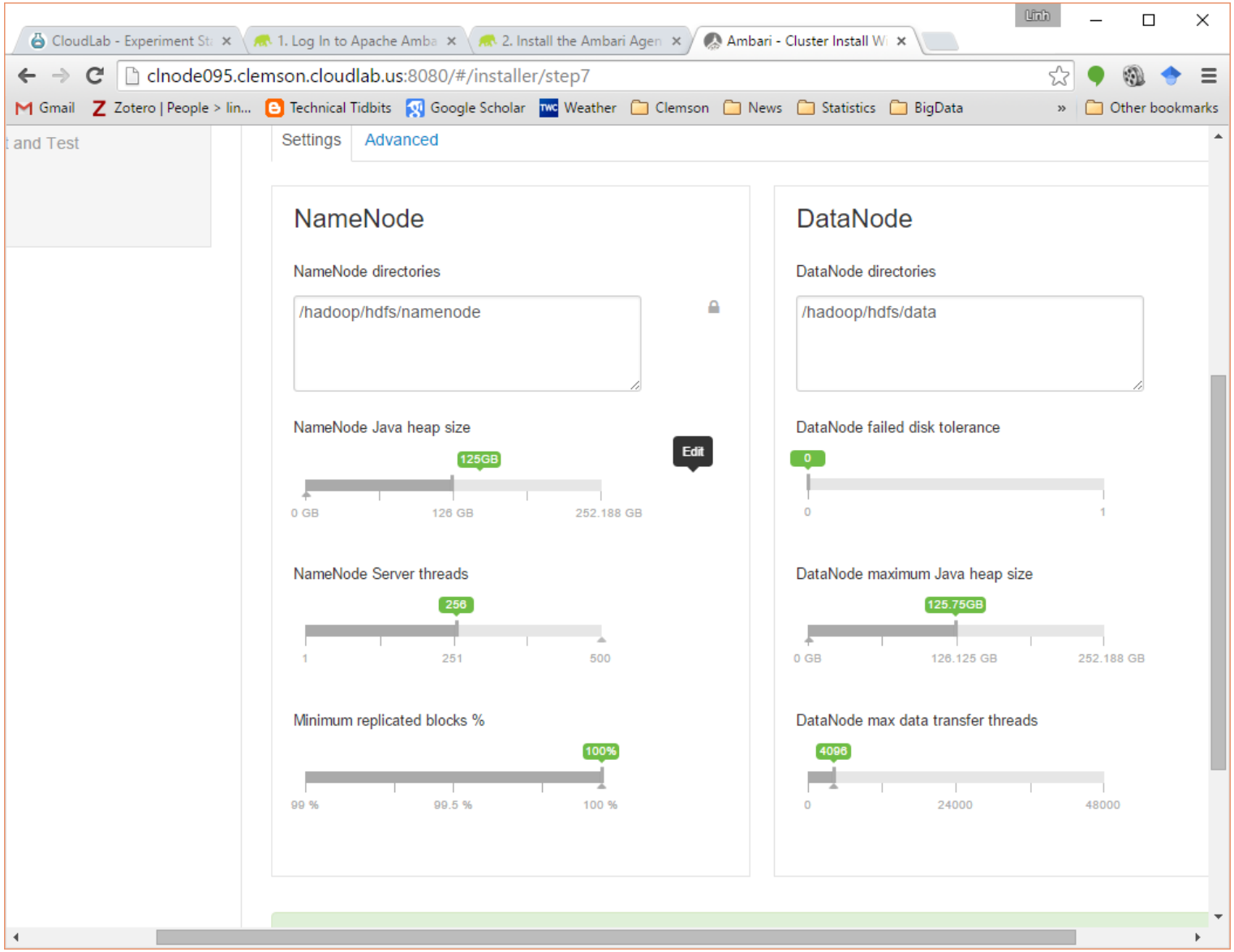
Host	all | none	all | none	all | none	all | none
clnode095.clemson.cloudl... ✱	☐ DataNode	☐ NFSGateway	☐ NodeManager	☑ Client
clnode092.clemson.cloudl... ✱	☑ DataNode	☐ NFSGateway	☑ NodeManager	☐ Client

Show: 25 | 1 - 2 of 2

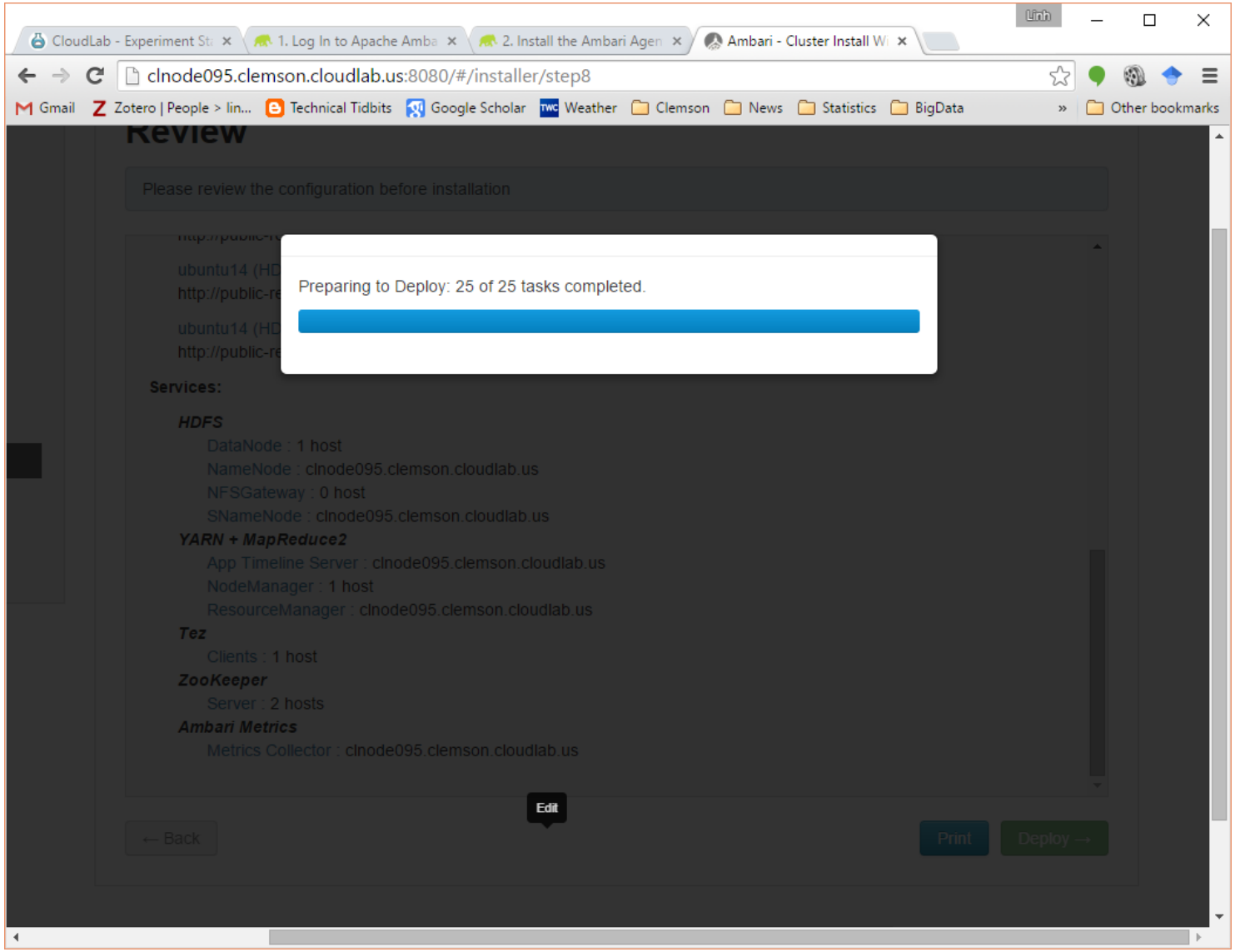
← Back | Next →

ense, Version 2.0.
hat Ambari uses and their respective authors

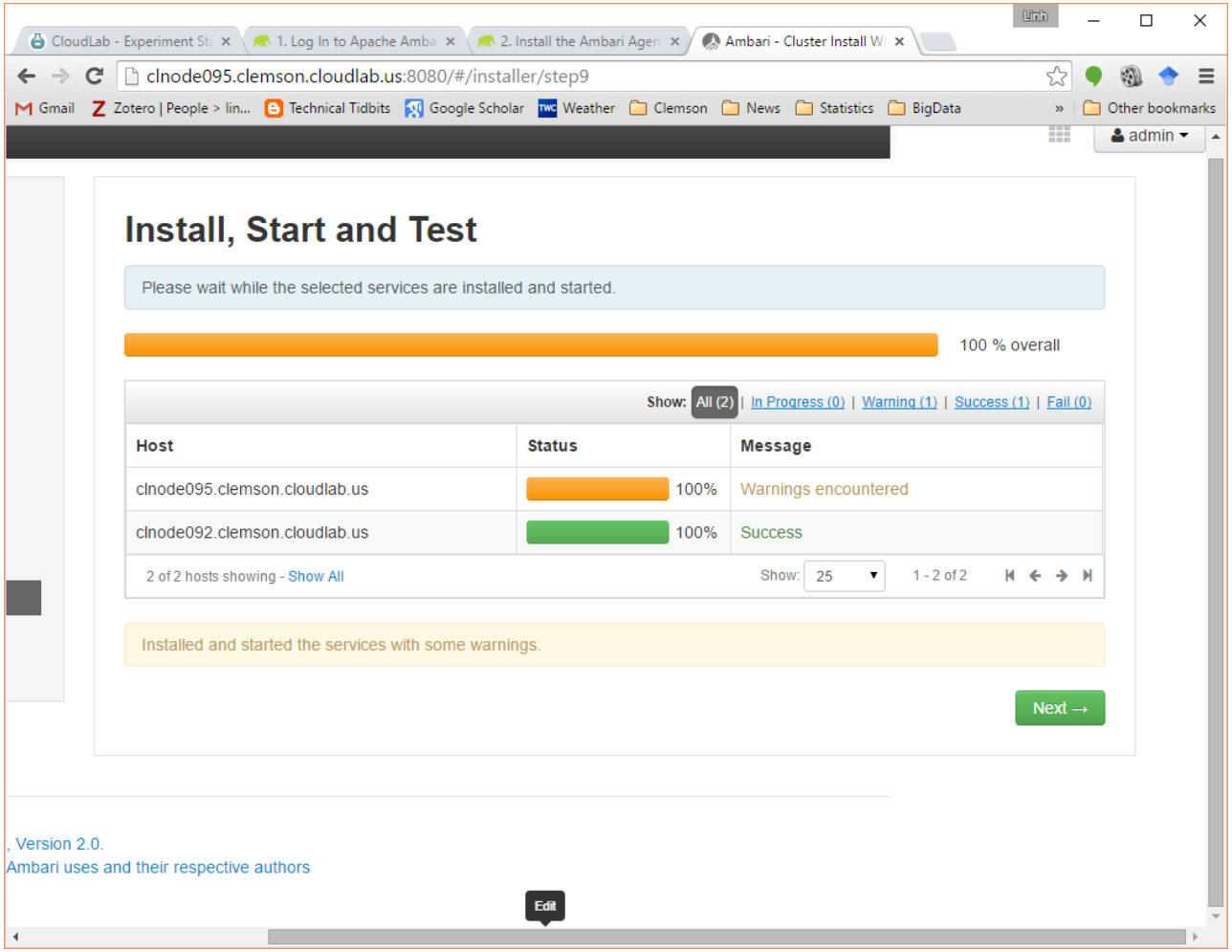
Edit configuration as you see fit



Deploy …


CloudLab - Experiment St
1. Log In to Apache Amba
2. Install the Ambari Agen
Ambari - Cluster Install Wi
clnode095.clemson.cloudlab.us:8080/#/installer/step8
Gmail
Zotero | People > lin...
Technical Tidbits
Google Scholar
Weather
Clemson
News
Statistics
BigData
Other bookmarks
Review
Please review the configuration before installation
Preparing to Deploy: 25 of 25 tasks completed.
Services:
HDFS
DataNode : 1 host
NameNode : clnode095.clemson.cloudlab.us
NFSGateway : 0 host
SNameNode : clnode095.clemson.cloudlab.us
YARN + MapReduce2
App Timeline Server : clnode095.clemson.cloudlab.us
NodeManager : 1 host
ResourceManager : clnode095.clemson.cloudlab.us
Tez
Clients : 1 host
ZooKeeper
Server : 2 hosts
Ambari Metrics
Metrics Collector : clnode095.clemson.cloudlab.us
Edit
← Back
Print
Deploy →

Warning due to lack of space and failed checks (ignore)



HDFS

CloudLab - Experiment S | 1. Log In to Apache Amb | 2. Install the Ambari Age | Ambari - Test | Namenode information | All Applications

clnode095.clemson.cloudlab.us:50070/dfshealth.html#tab-overview

Gmail | Zotero | People > lin... | Technical Tidbits | Google Scholar | Weather | Clemson | News | Statistics | BigData | Picasa Web Albums ... | Conferences | Other bookmarks

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'clnode095.clemson.cloudlab.us:8020' (active)

Started:	Wed Oct 07 23:34:32 EDT 2015
Version:	2.7.1.2.3.2.0-2950, r5cc60e0003e33aa98205f18bccaeaf36cb193c1c
Compiled:	2015-09-30T18:08Z by jenkins from (HEAD detached at 5cc60e0)
Cluster ID:	CID-c31ab4be-d8d4-42a9-b456-7745974a681d
Block Pool ID:	BP-1609302222-130.127.133.104-1444275269044

Summary

Security is off.

Safemode is off.

38 files and directories, 11 blocks = 49 total filesystem object(s).

Heap Memory used 15.41 GB of 123.44 GB Heap Memory. Max Heap Memory is 123.44 GB.

Non Heap Memory used 60.15 MB of 61.21 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	14.75 GB
DFS Used:	256.25 MB (1.7%)

YARN

CloudLab - Experiment S | 1. Log In to Apache Amb | 2. Install the Ambari Age | Ambari - Test | Namenode information | All Applications

clnode095.clemson.cloudlab.us:8088/cluster

Gmail | Zotero | People > lin... | Technical Tidbits | Google Scholar | Weather | Clemson | News | Statistics | BigData | Picasa Web Albums ... | Conferences | Other bookmarks

hadoop

All Applications

- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lo Nod
4	0	0	4	0	0 B	220 GB	0 B	0	16	0	1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum
Capacity Scheduler	[MEMORY]	<memory:56320, vCores:1>	<memory:225280, vCores:

Show 20 entries Search

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Progres
application_1444275343859_0004	ambari-qa	DistributedShell	YARN	default	Wed Oct 7 23:40:32 -0400 2015	Wed Oct 7 23:40:40 -0400 2015	FINISHED	SUCCEEDED	N/A	
application_1444275343859_0003	ambari-qa	OrderedWordCount	TEZ	default	Wed Oct 7 23:39:35 -0400 2015	Wed Oct 7 23:40:33 -0400 2015	FINISHED	FAILED	N/A	
application_1444275343859_0002	ambari-qa	OrderedWordCount	TEZ	default	Wed Oct 7 23:38:33 -0400 2015	Wed Oct 7 23:39:31 -0400 2015	FINISHED	FAILED	N/A	
application_1444275343859_0001	ambari-qa	OrderedWordCount	TEZ	default	Wed Oct 7 23:37:29 -0400 2015	Wed Oct 7 23:38:29 -0400 2015	FINISHED	FAILED	N/A	

Showing 1 to 4 of 4 entries First