Assignment 1

AMS 560 - 2024 Fall

Professor Zhenhua Liu, TAs Yunlong Pan & Xander Barron

Credit for slides: Weihao Wang

Requirements

When you have completed submission #1, please submit a full screen recording (and screenshots as backup) to show the following tasks have been carried out successfully:

1. A cluster has been created on Cloudlab **(5%)**
2. JDK is setup on each node **(5%)**
3. Passwordless SSH has been setup **(5%)**
4. Yarn has been downloaded in each node **(5%)**
5. Successful setting up of the files core-site.xml, hdfs-site.xml, and yarn-site.xml **(20%)**
6. Setting up of fair-scheduler.xml **(10%)**
7. Creating of slaves file in each node **(10%)**
8. Successfully running HDFS and Yarn (See more in the future slides) **(15%)**
9. Correct number of live nodes on 50070 site **(15%)**
10. **Full screen recording of the process (10%)**

Attention!

- Take some time to watch the “Assignment 1 Tutorial” video before you start your trial on Cloudlab.
- Always use local hostnames like ctl, cp-1, cp-2. Never use public hostnames like amd203.utah.cloudlab.us in your xml configuration file. This may cause your instance to be terminated without advanced notice. (In the tutorial video, we mention hostnames like “ctl-0”, you do not to use it this year, just use “ctl” instead.)
- Turn off the YARN service by running Hadoop/sbin/stop-all.sh when you are not using Cloudlab.
- Don’t extend your experiment unless it is really, really necessary.

Basic Linux commands:

- **Navigation:**
  1. pwd: Print working directory, shows the current directory.
  2. ls: List contents of the directory.
  3. cd: Change directory.
- **File Operations:**
  1. touch: Create an empty file.
  2. cp: Copy files or directories.
  3. mv: Move or rename files or directories.
  4. rm: Remove files or directories.
  5. cat: Display contents of a file.
  6. more & less: View the contents of a file one screen at a time.
  7. head: View the beginning of a file.
  8. tail: View the end of a file.

Basic Linux commands:

- **File Permissions:**
  1. chmod: Change file permissions.
  2. chown: Change file owner.
  3. chgrp: Change group ownership.
- **Search:**
  1. find: Search files and directories based on criteria.
  2. grep: Search for a specific pattern within files.
- **Processes:**
  1. ps: Display current processes.
  2. top: Display system's active processes.
  3. kill: Kill a process.

Basic Linux commands:

- **System Information:**
  1. uname: Display system information.
  2. df: Display disk space usage.
  3. du: Display directory space usage.
  4. free: Display memory usage.
- **Networking:**
  1. ifconfig or ip: Display network interface configuration.
  2. netstat: Network statistics.
  3. ping: Test network connectivity.

Basic Linux commands:

- **Package Management (varies by distribution):**
  1. apt-get (Debian/Ubuntu): A tool for handling packages.
  2. yum (Older Red Hat/Fedora): Another package manager.
  3. dnf (Newer Red Hat/Fedora): Replaces yum.
  4. pacman (Arch Linux): Package manager for Arch Linux.
- **Archiving & Compression:**
  1. tar: Archive files.
  2. gzip: Compress files.
  3. gunzip: Decompress files.
  4. zip, unzip: Work with zip archives.

Basic Linux commands:

- **Text Editors:**
  1. nano: Easy-to-use terminal text editor.
  2. vi or vim: Advanced terminal text editor.
  3. emacs: Another powerful text editor.
- **User Management:**
  1. useradd: Add a new user.
  2. userdel: Delete a user.
  3. passwd: Change user password.
  4. groupadd: Add a new group.

Basic Linux commands:

- **System Shutdown & Restart:**
  1. shutdown: Shutdown the system.
  2. reboot: Restart the system.
- **Miscellaneous:**
  1. echo: Display a message.
  2. man: Display the manual page for a command.
  3. history: Display command history.

Agenda

1. Setup a new cluster on Cloudlab
2. Setup Apache Hadoop Yarn
3. What to submit

1. Setup a new cluster on Cloudlab

Prerequisites

1. You must have an account on cloudlab.us
2. Using Linux or Mac OS

Steps

1. Create an SSH key
2. Create a cluster
3. SSH to each node on the cluster

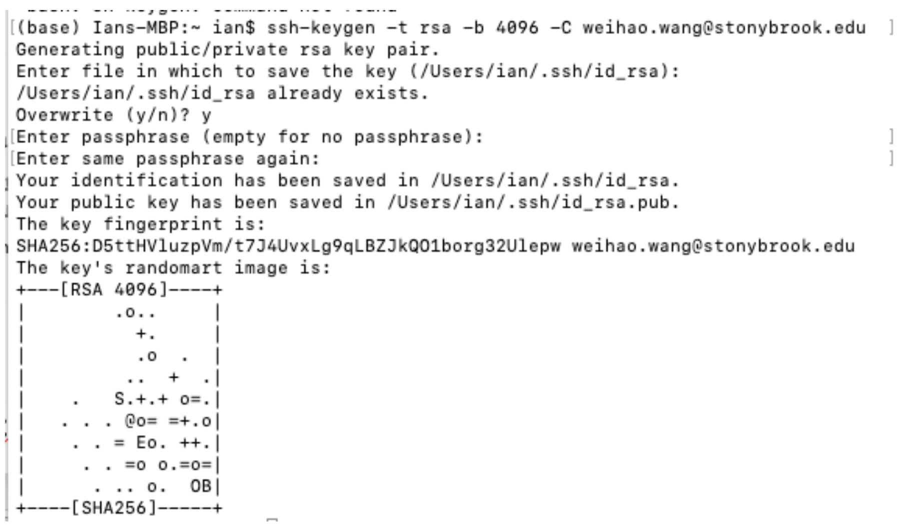
1.1 Create an SSH key

Open the terminal

1. ssh-keygen -t rsa -b 4096 -C your_email@example.com
2. You will be asked for file path and password

   Refer: https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/ for more detail.

   To make it simple, please skip entering the filepath and password.



```
(base) Ians-MBP:~ ian$ ssh-keygen -t rsa -b 4096 -C weihao.wang@stonybrook.edu
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/ian/.ssh/id_rsa):
/Users/ian/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/ian/.ssh/id_rsa.
Your public key has been saved in /Users/ian/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:D5ttHVluzpVm/t7J4UvxLg9qLBZJkQO1borg32Ulepw weihao.wang@stonybrook.edu
The key's randomart image is:
+---[RSA 4096]----+
|        .o..     |
|          +.     |
|          .o  .  |
|         ..  +  .|
|    .   S.+.+ o=.|
|   . . . @o= =+.o|
|    . . = Eo. ++.|
|     . . =o o.=o=|
|      . .. o.  OB|
+----[SHA256]-----+
```

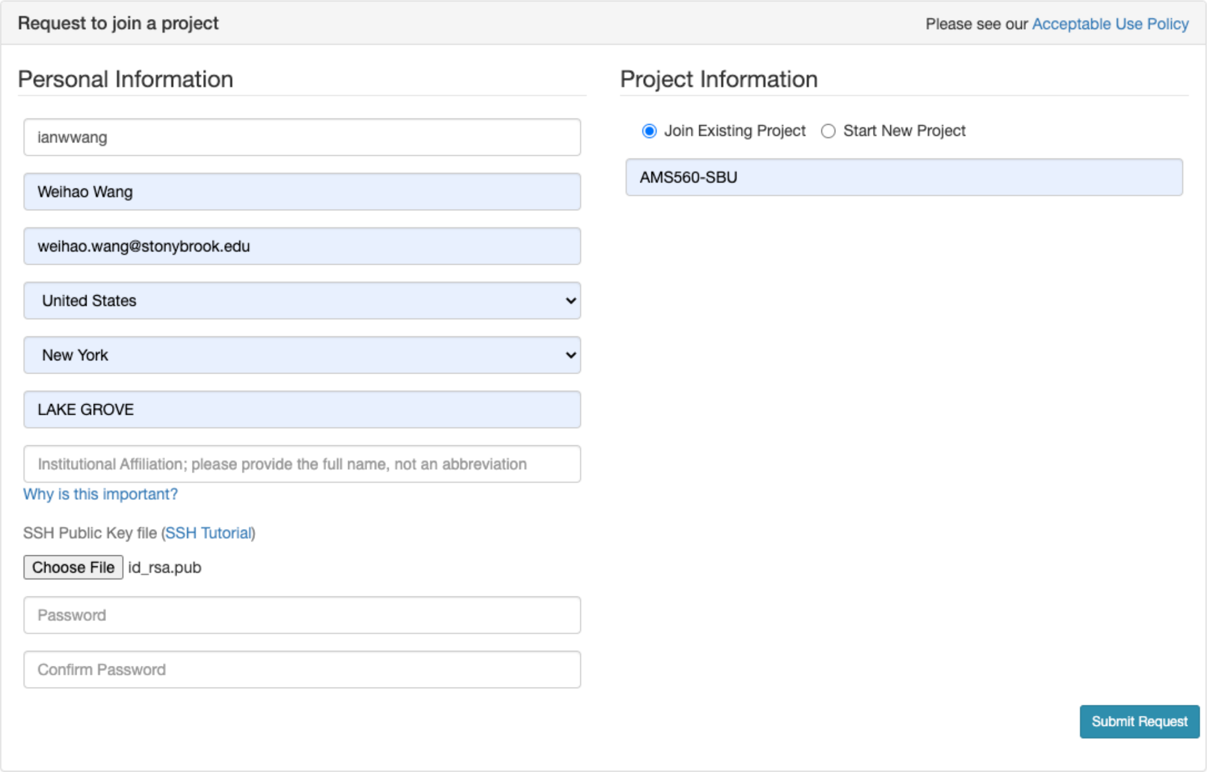
Stony Brook University

1.1 Create an SSH key (cont.)

If you do not have an account on cloudlab.us, go to the website and click ‘sign up’.

Then fill out your information like this:

You can use the SSH key you just created.



Request to join a project

Please see our Acceptable Use Policy

Personal Information

ianwwang

Weihao Wang

weihao.wang@stonybrook.edu

United States

New York

LAKE GROVE

Institutional Affiliation; please provide the full name, not an abbreviation

Why is this important?

SSH Public Key file (SSH Tutorial)

Choose File id_rsa.pub

Password

Confirm Password

Project Information

● Join Existing Project ○ Start New Project

AMS560-SBU

Submit Request

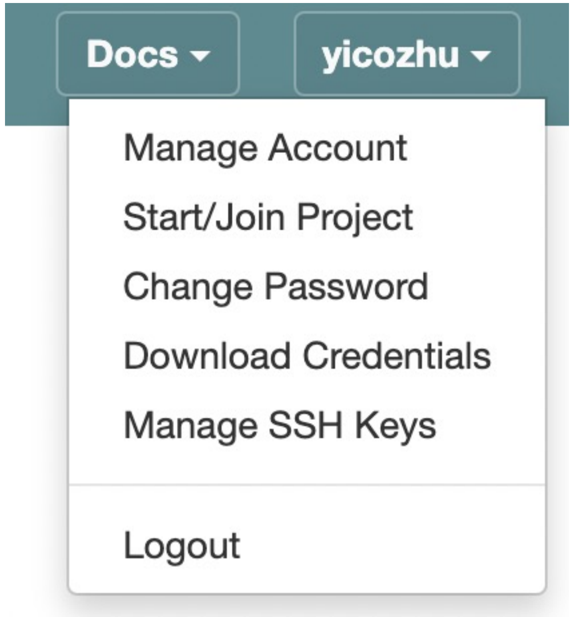
FAR BEYOND

1.1 Create an SSH key (cont.)

1. Log into your account on cloudlab.us

   Go to “Your account name” → “Manage SSH Keys”

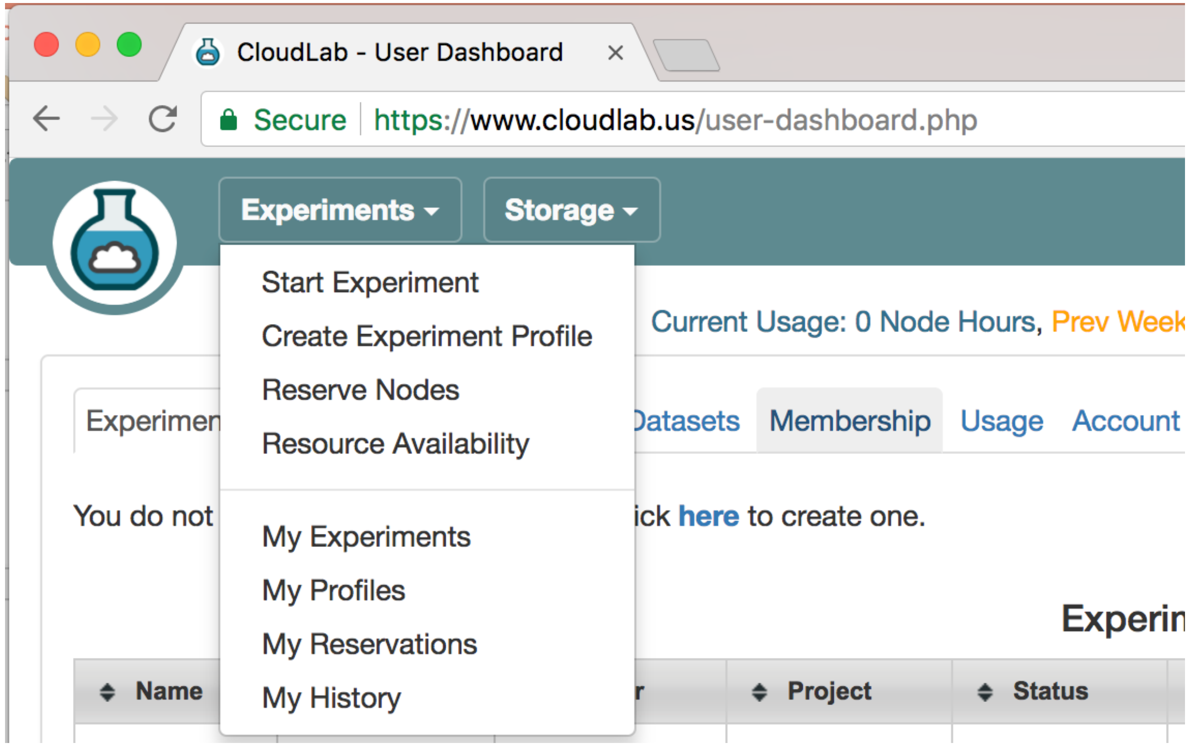
2. In the Add key tab: click “load from file” → load the [filepath]/id_rsa.pub → click “add key”



Hints:
You could use “Command + Shift + .” to check the hidden folders in your system.

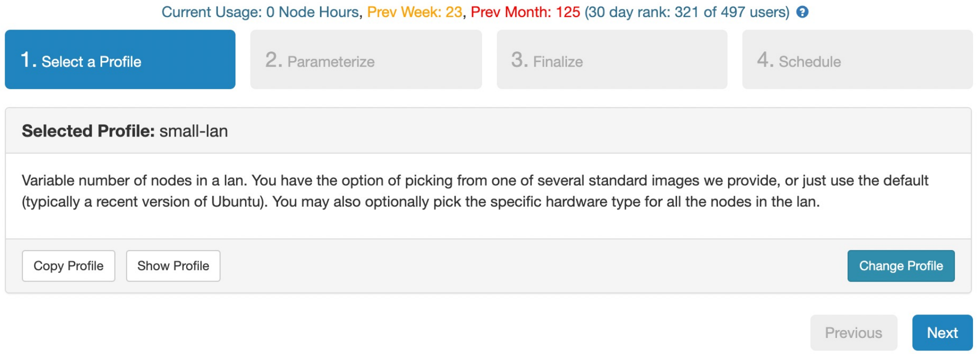
1.2 Create a cluster

1. Start Experiment



1.2 Create a cluster (cont.)

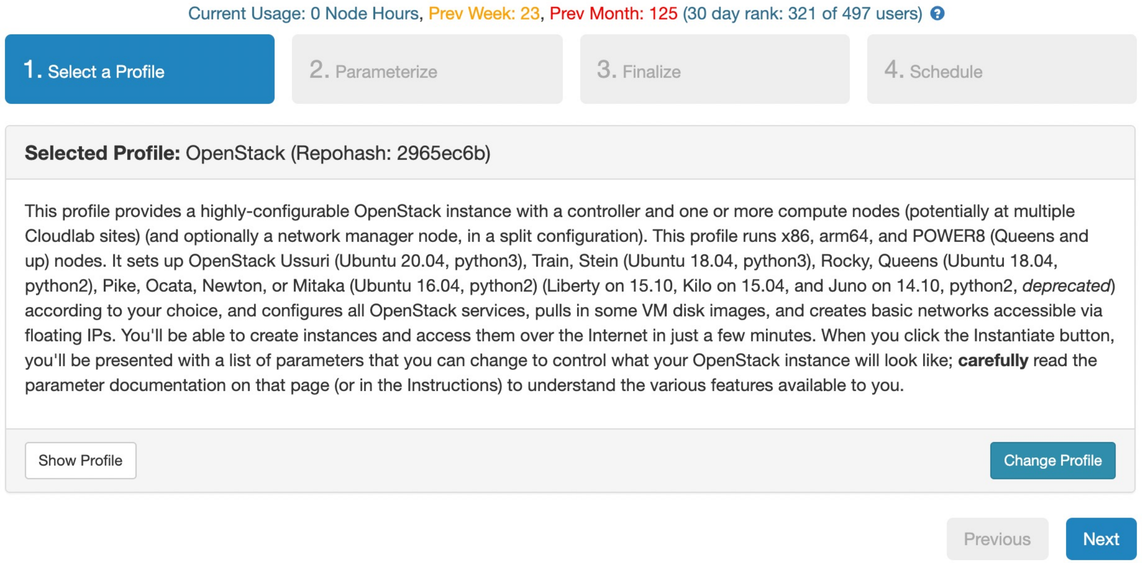
- Select “OpenStack” after click “Change Profile”
- Click Next



Current Usage: 0 Node Hours, Prev Week: 23, Prev Month: 125 (30 day rank: 321 of 497 users)

Selected Profile: small-lan

Variable number of nodes in a lan. You have the option of picking from one of several standard images we provide, or just use the default (typically a recent version of Ubuntu). You may also optionally pick the specific hardware type for all the nodes in the lan.



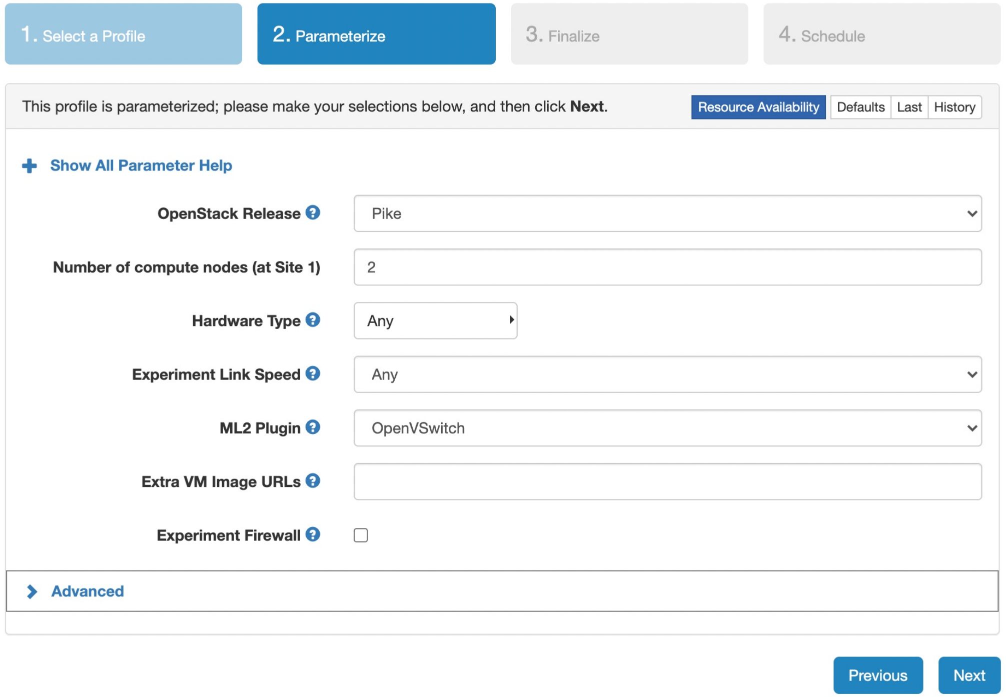
Current Usage: 0 Node Hours, Prev Week: 23, Prev Month: 125 (30 day rank: 321 of 497 users)

Selected Profile: OpenStack (Repohash: 2965ec6b)

This profile provides a highly-configurable OpenStack instance with a controller and one or more compute nodes (potentially at multiple Cloudlab sites) (and optionally a network manager node, in a split configuration). This profile runs x86, arm64, and POWER8 (Queens and up) nodes. It sets up OpenStack Ussuri (Ubuntu 20.04, python3), Train, Stein (Ubuntu 18.04, python3), Rocky, Queens (Ubuntu 18.04, python2), Pike, Ocata, Newton, or Mitaka (Ubuntu 16.04, python2) (Liberty on 15.10, Kilo on 15.04, and Juno on 14.10, python2, *deprecated*) according to your choice, and configures all OpenStack services, pulls in some VM disk images, and creates basic networks accessible via floating IPs. You'll be able to create instances and access them over the Internet in just a few minutes. When you click the Instantiate button, you'll be presented with a list of parameters that you can change to control what your OpenStack instance will look like; **carefully** read the parameter documentation on that page (or in the Instructions) to understand the various features available to you.

1.2 Create a cluster (cont.)

- OpenStack Release: Select “Pike”
- Number of compute nodes: depends on your choice, could use 2, 4, …



This profile is parameterized; please make your selections below, and then click **Next**.

OpenStack Release: Pike

Number of compute nodes (at Site 1): 2

Hardware Type: Any

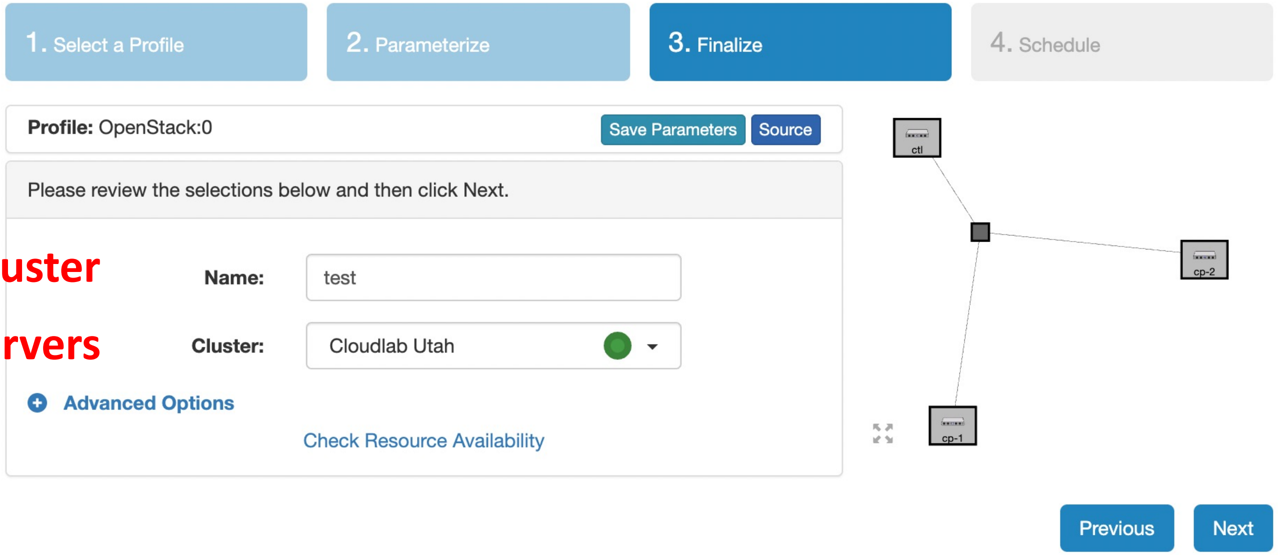
Experiment Link Speed: Any

ML2 Plugin: OpenVSwitch

Extra VM Image URLs:

Experiment Firewall: □

1.2 Create a cluster (cont.)



Profile: OpenStack:0

Please review the selections below and then click Next.

Name: test

Cluster: Cloudlab Utah ●

Input the name of cluster

Select the site of servers

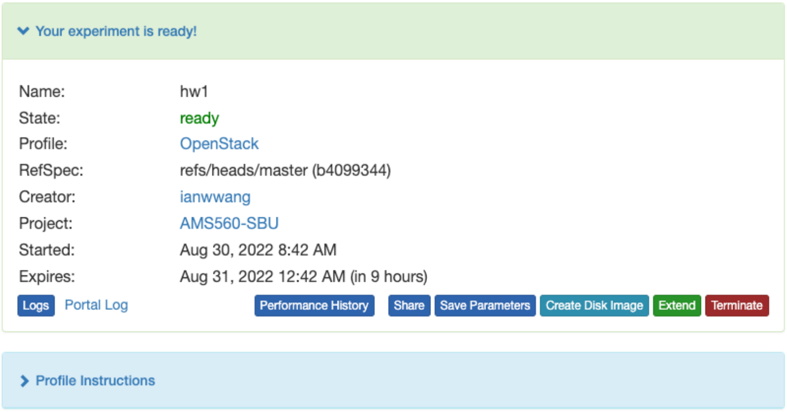
Click Finish and wait

Hints:

1. Please find more details in “Docs”: http://docs.cloudlab.us/
2. Check how to use OpenStack on Cloudlab: http://docs.cloudlab.us/openstack-tutorial.html

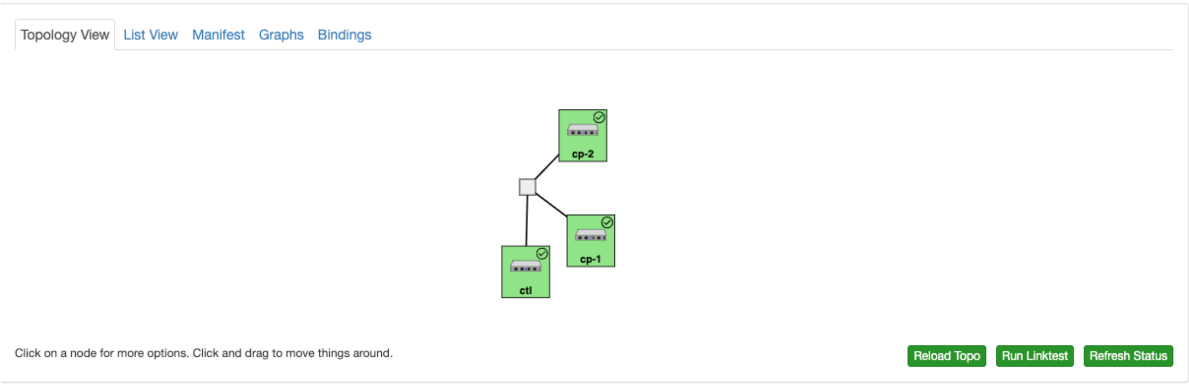
1.2 Create a cluster (cont.)

- It will take around 40 min to finalize the cluster.
- You will see the “cluster” is ready to use message below.
- Make sure all the nodes status are “done”: use Topology View to check.

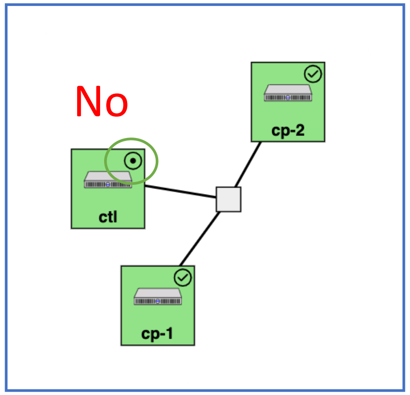


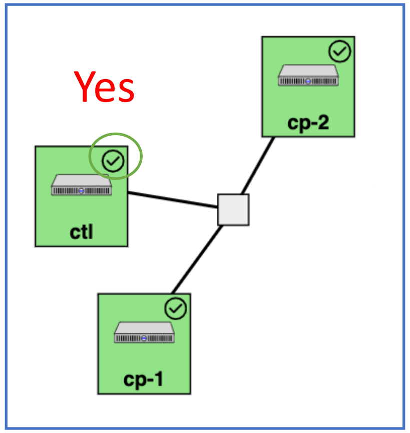
Your experiment is ready!

Name: hw1
State: ready
Profile: OpenStack
RefSpec: refs/heads/master (b4099344)
Creator: ianwwang
Project: AMS560-SBU
Started: Aug 30, 2022 8:42 AM
Expires: Aug 31, 2022 12:42 AM (in 9 hours)



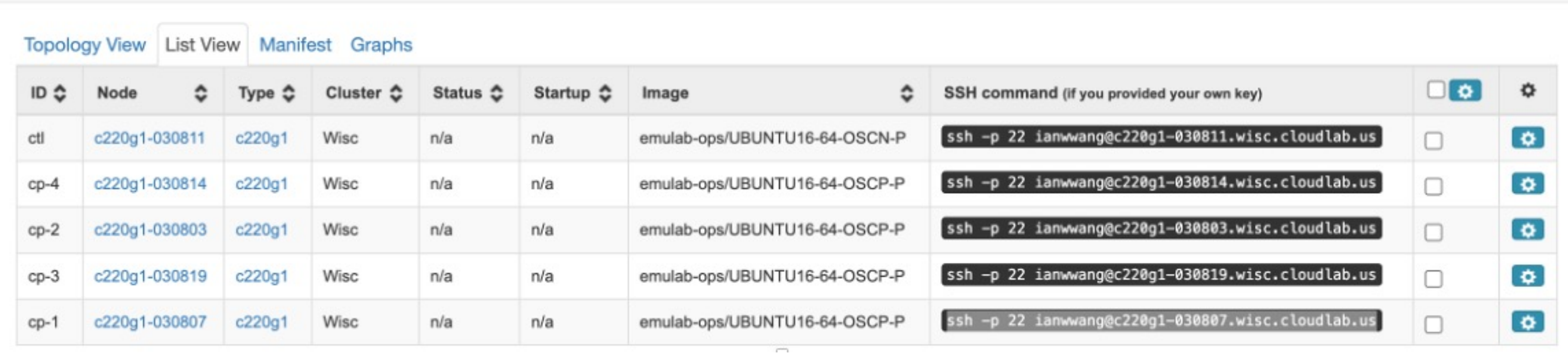
Click on a node for more options. Click and drag to move things around.





1.3 SSH to a node in cluster

- At the List View, you can see the SSH command to access any node
- At your local terminal, you can ssh to the node you want



ID ⇕	Node ⇕	Type ⇕	Cluster ⇕	Status ⇕	Startup ⇕	Image ⇕	SSH command (if you provided your own key)	□ ⚙	⚙
ctl	c220g1-030811	c220g1	Wisc	n/a	n/a	emulab-ops/UBUNTU16-64-OSCN-P	ssh -p 22 ianwwang@c220g1-030811.wisc.cloudlab.us	□	⚙
cp-4	c220g1-030814	c220g1	Wisc	n/a	n/a	emulab-ops/UBUNTU16-64-OSCP-P	ssh -p 22 ianwwang@c220g1-030814.wisc.cloudlab.us	□	⚙
cp-2	c220g1-030803	c220g1	Wisc	n/a	n/a	emulab-ops/UBUNTU16-64-OSCP-P	ssh -p 22 ianwwang@c220g1-030803.wisc.cloudlab.us	□	⚙
cp-3	c220g1-030819	c220g1	Wisc	n/a	n/a	emulab-ops/UBUNTU16-64-OSCP-P	ssh -p 22 ianwwang@c220g1-030819.wisc.cloudlab.us	□	⚙
cp-1	c220g1-030807	c220g1	Wisc	n/a	n/a	emulab-ops/UBUNTU16-64-OSCP-P	ssh -p 22 ianwwang@c220g1-030807.wisc.cloudlab.us	□	⚙

2. Apache Hadoop Yarn

Prerequisites

1. Install and setup Java JDK 1.8

Steps

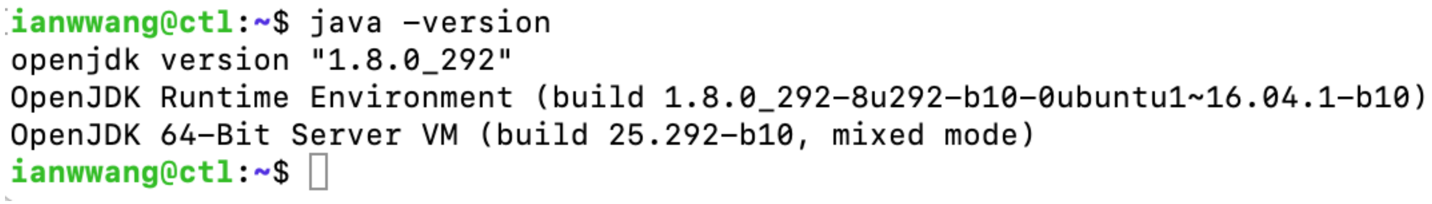
1. Download Yarn 2.7.2
2. Setup passwordless SSH
3. Setup HDFS
4. Setup Yarn

2.0 Apache Hadoop Yarn – Setup OPEN JDK 8

1. ssh to master node, then enter:

   $ sudo apt-get install -y openjdk-8-jdk

   $ java –version Check if the JDK 8 install successfully



```
ianwwang@ctl:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
ianwwang@ctl:~$
```

2. Go to other worker nodes (cp-1, cp-2) and repeat the “install JDK 8” procedures

   Hints:
   You could use “exit” to shut down the connection to the current node, then ssh to another node.

2.1 Setup passwordless SSH

1. From localhost, upload the keys to $HOME/.ssh/ of a cloudlab_node (using DNS name)

```
localhost$ scp ~/.ssh/id_rsa* [cloudlab_node]:~/.ssh/
```

For me: scp ~/.ssh/id_rsa* ianwwang@c220g1-030811.wisc.cloudlab.us:~/.ssh/

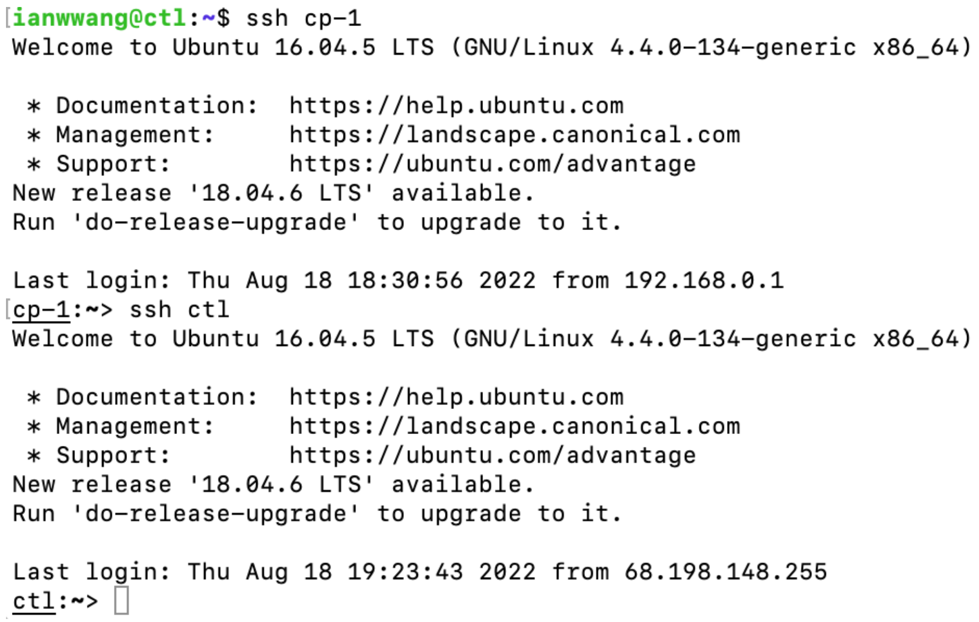
2. SSH to the cloud lab node and enter the following commands

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys;
chmod 0600 ~/.ssh/id_rsa*;
chmod 0600 ~/.ssh/authorized_keys;
rm -rf ~/.ssh/known_hosts;
echo 'StrictHostKeyChecking no' >> ~/.ssh/config
```

Repeat the procedures for each node

2.1 Setup passwordless SSH (cont.)

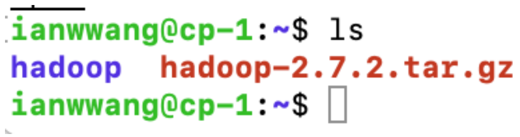
- You can ssh from a node to another without using password



```
[ianwwang@ctl:~$ ssh cp-1
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-134-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Aug 18 18:30:56 2022 from 192.168.0.1
[cp-1:~> ssh ctl
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-134-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Aug 18 19:23:43 2022 from 68.198.148.255
ctl:~> 
```

2.2 Download Yarn 2.7.2

- SSH to a cloud lab node first
- Download Apache Hadoop
  - wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.2/hadoop-2.7.2.tar.gz
- Extract Hadoop & change the folder name
  - tar -xvzf hadoop-2.7.2.tar.gz; mv hadoop-2.7.2 hadoop
- You could use ls to check the content in the folder.



```
ianwwang@cp-1:~$ ls
hadoop  hadoop-2.7.2.tar.gz
ianwwang@cp-1:~$ 
```

Download & extract Hadoop for other nodes as well
Make sure you have the Hadoop folder in each mode

2.3 Setup HDFS

- **SSH to each Cloudlab node and enter the following commands**
- **setup Hadoop paths (if you get the message ‘export: command not found’, ran ‘bash’ first)**

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/ >> .bashrc
```

```
echo export HADOOP_PREFIX=~/hadoop >> .bashrc;echo export HADOOP_YARN_HOME=~/hadoop >> .bashrc;echo export HADOOP_HOME=~/hadoop >> .bashrc;echo export HADOOP_CONF_DIR=~/hadoop/etc/hadoop >> .bashrc;echo export YARN_CONF_DIR=~/hadoop/etc/hadoop >> .bashrc;source .bashrc
```

- **Setup “export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64” into the first line of hadoop/etc/hadoop/hadoop-env.sh**

```
cd hadoop/etc/hadoop/

vim hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Hints: Check how to use vim to edit files in Linux system: https://www.howtoforge.com/vim-basics
Or you can use nano instead. (nano hadoop-env.sh)

2.3 Setup HDFS (cont.)

- On Cloudlab each node, create HDFS folder, e.g., /dev/hdfs and grant permissions to read & write.

sudo mkdir /dev/hdfs; sudo chmod 777 /dev/hdfs

- cd hadoop/etc/hadoop/, we will need to setup several xml files in this folder on each node:
  1. core-site.xml
  2. hdfs-site.xml
  3. yarn-site.xml
  4. slaves

2.3 Setup HDFS (cont.)

- On Cloudlab each node, edit file hadoop/etc/hadoop/core-site.xml

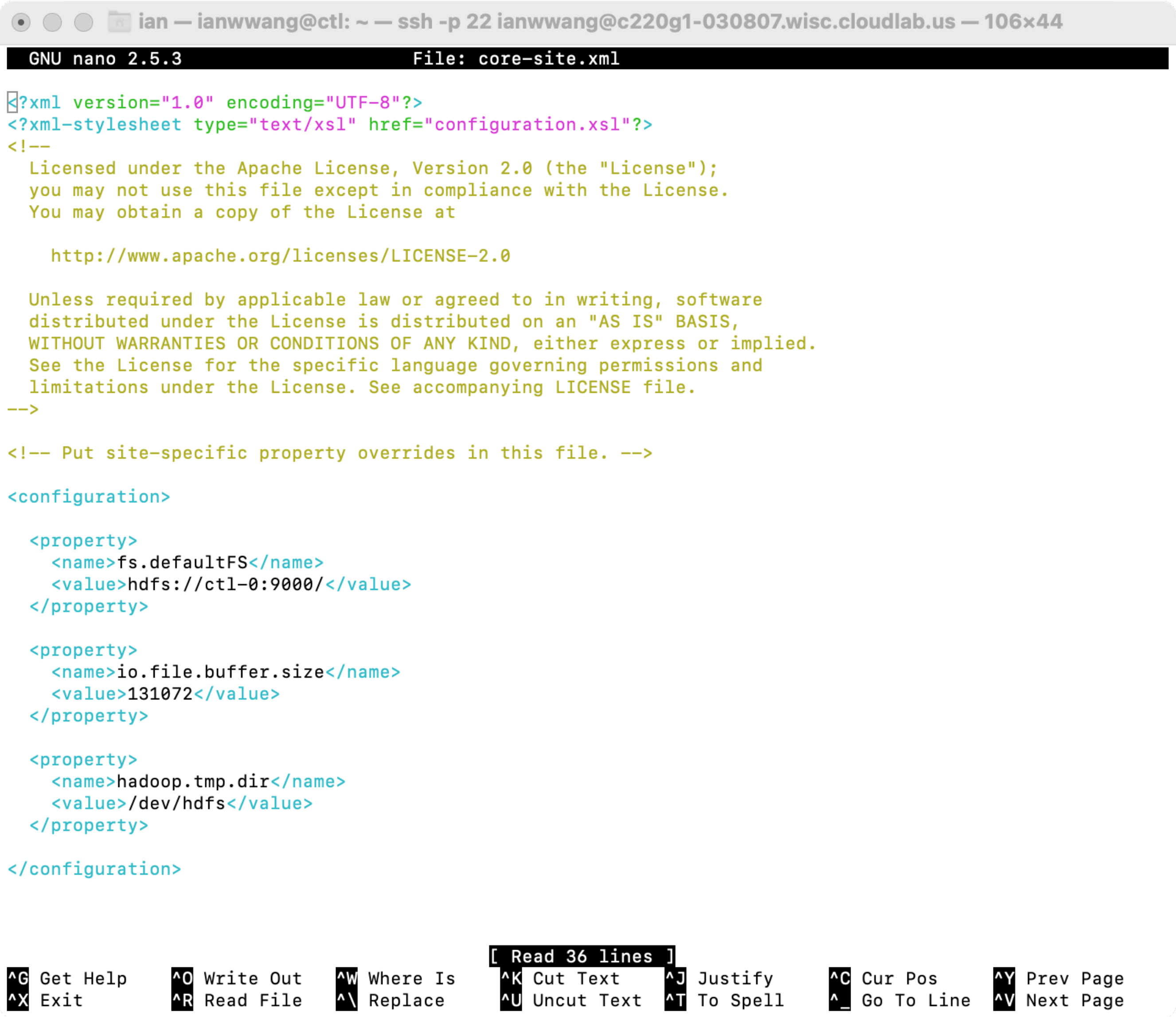
    ctl : hostname of the master node

    /dev/hdfs : hdfs folder

    You can use “vim”, press “i” key into “INSERT” mode to edit, then save it by “:wq” + enter

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl“ href="configuration.xsl"?>

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://ctl:9000/</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/dev/hdfs</value>
  </property>
</configuration>
```

Hints:
Be really careful when editing files, e.g. double quotations,…

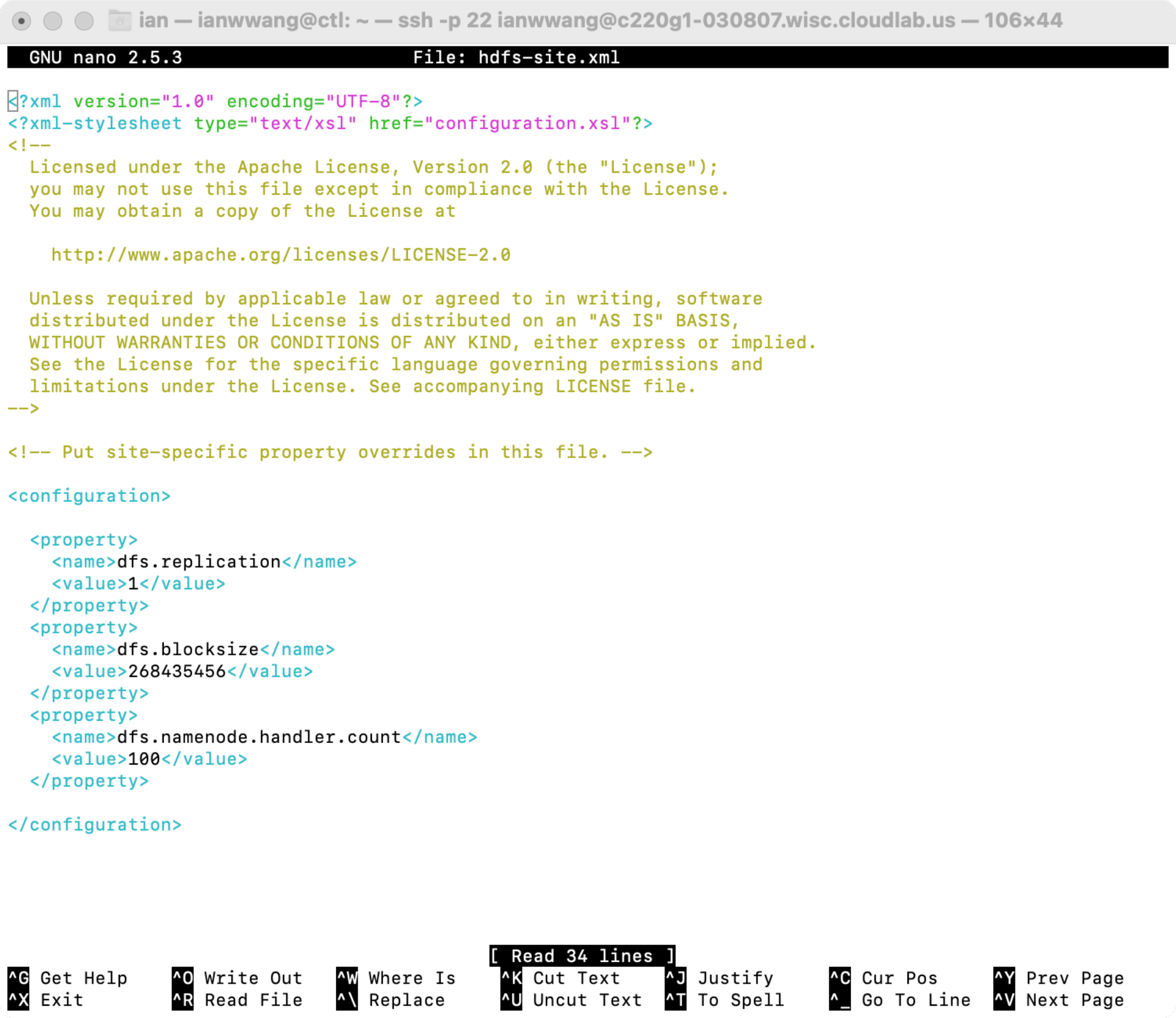

ian — ianwwang@ctl: ~ — ssh -p 22 ianwwang@c220g1-030807.wisc.cloudlab.us — 106×44

```
  GNU nano 2.5.3                         File: core-site.xml

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://ctl-0:9000/</value>
  </property>

  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>

  <property>
    <name>hadoop.tmp.dir</name>
    <value>/dev/hdfs</value>
  </property>

</configuration>

                                             [ Read 36 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next Page
```

2.3 Setup HDFS (cont.)

- On Cloudlab each node, edit the file hadoop/etc/hadoop/hdfs-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.blocksize</name>
    <value>268435456</value>
  </property>
  <property>
    <name>dfs.namenode.handler.count</name>
    <value>100</value>
  </property>
</configuration>
```


ian — ianwwang@ctl: ~ — ssh -p 22 ianwwang@c220g1-030807.wisc.cloudlab.us — 106×44

```
  GNU nano 2.5.3                   File: hdfs-site.xml

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.blocksize</name>
    <value>268435456</value>
  </property>
  <property>
    <name>dfs.namenode.handler.count</name>
    <value>100</value>
  </property>

</configuration>

                             [ Read 34 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line ^V Next Page
```

2.4 Setup YARN

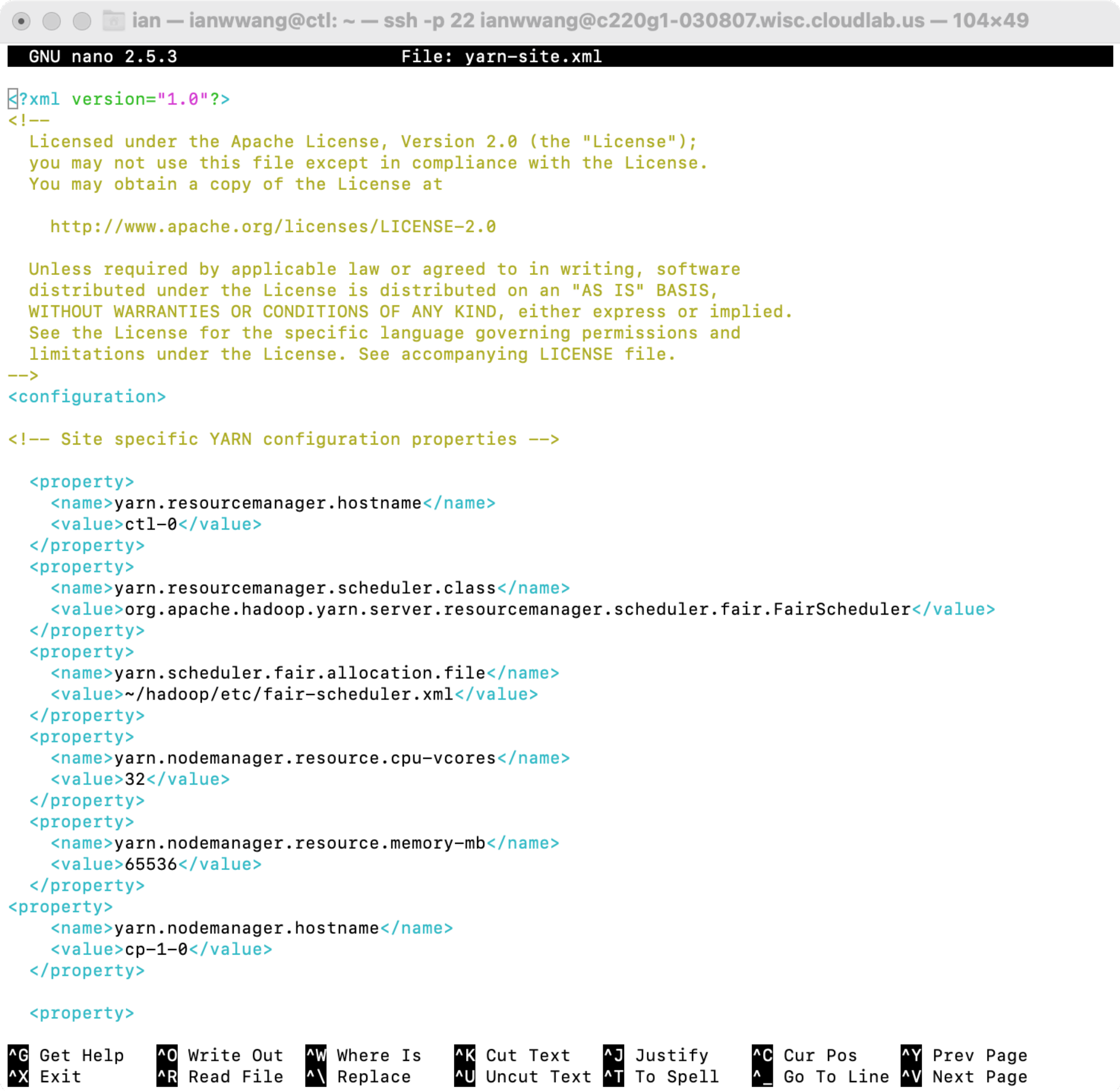
On Cloudlab each node, edit the file hadoop/etc/hadoop/yarn-site.xml

1. hostname: the hostname of the master node, e.g, ctl
2. [nodemanger hostname]: the hostname of the node where the file is located, change it accordingly to: ctl, cp-1, cp-2, …
3. yarn.nodemanager.resource.cpu-vcores: Number of cpu-cores on the node
4. yarn.nodemanager.resource.memory-mb: Amount of memory on the node in MB

Hints:

1. If you are in ctl, change the [nodemanger hostname] to ctl-0; if you are in cp-1, change the [nodemanger hostname] to cp-1-0
2. Check the default setting here: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

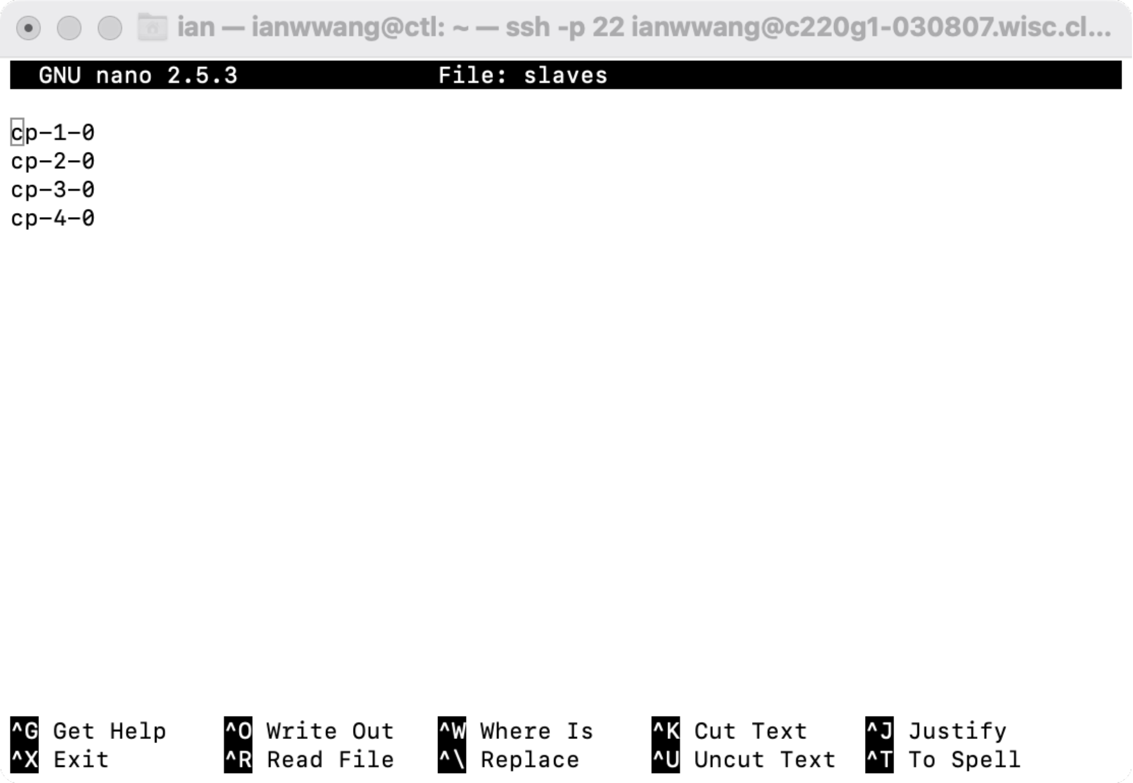
```xml
<?xml version="1.0"?>
<configuration>
 <property>
  <name>yarn.resourcemanager.hostname</name>
  <value>[master note]</value>
 </property>
 <property>
  <name>yarn.resourcemanager.scheduler.class</name>
  <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>
 </property>
 <property>
  <name>yarn.scheduler.fair.allocation.file</name>
  <value>~/hadoop/etc/fair-scheduler.xml</value>
 </property>
 <property>
  <name>yarn.nodemanager.resource.cpu-vcores</name>
  <value>32</value>
 </property>
 <property>
  <name>yarn.nodemanager.resource.memory-mb</name>
  <value>65536</value>
 </property>
 <property>
  <name>yarn.nodemanager.hostname</name>
  <value> [nodemanger hostname]</value>
 </property>
 <property>
  <name>yarn.nodemanager.bind-host</name>
  <value> [nodemanger hostname]</value>
 </property>
</configuration>
```



```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>ctl-0</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.class</name>
    <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>
  </property>
  <property>
    <name>yarn.scheduler.fair.allocation.file</name>
    <value>~/hadoop/etc/fair-scheduler.xml</value>
  </property>
  <property>
    <name>yarn.nodemanager.resource.cpu-vcores</name>
    <value>32</value>
  </property>
  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>65536</value>
  </property>
<property>
    <name>yarn.nodemanager.hostname</name>
    <value>cp-1-0</value>
  </property>

  <property>
```

2.4 Setup YARN (cont.)

- On Cloudlab each node (including ctl), create file hadoop/etc/hadoop/slaves
- We try to indicate the compute nodes
  - In this demonstration, I have 4 compute nodes (cp-1, cp-2, cp-3, cp-4)
  - If you only have two, then use (cp-1, cp-2)



ian — ianwwang@ctl: ~ — ssh -p 22 ianwwang@c220g1-030807.wisc.cl...

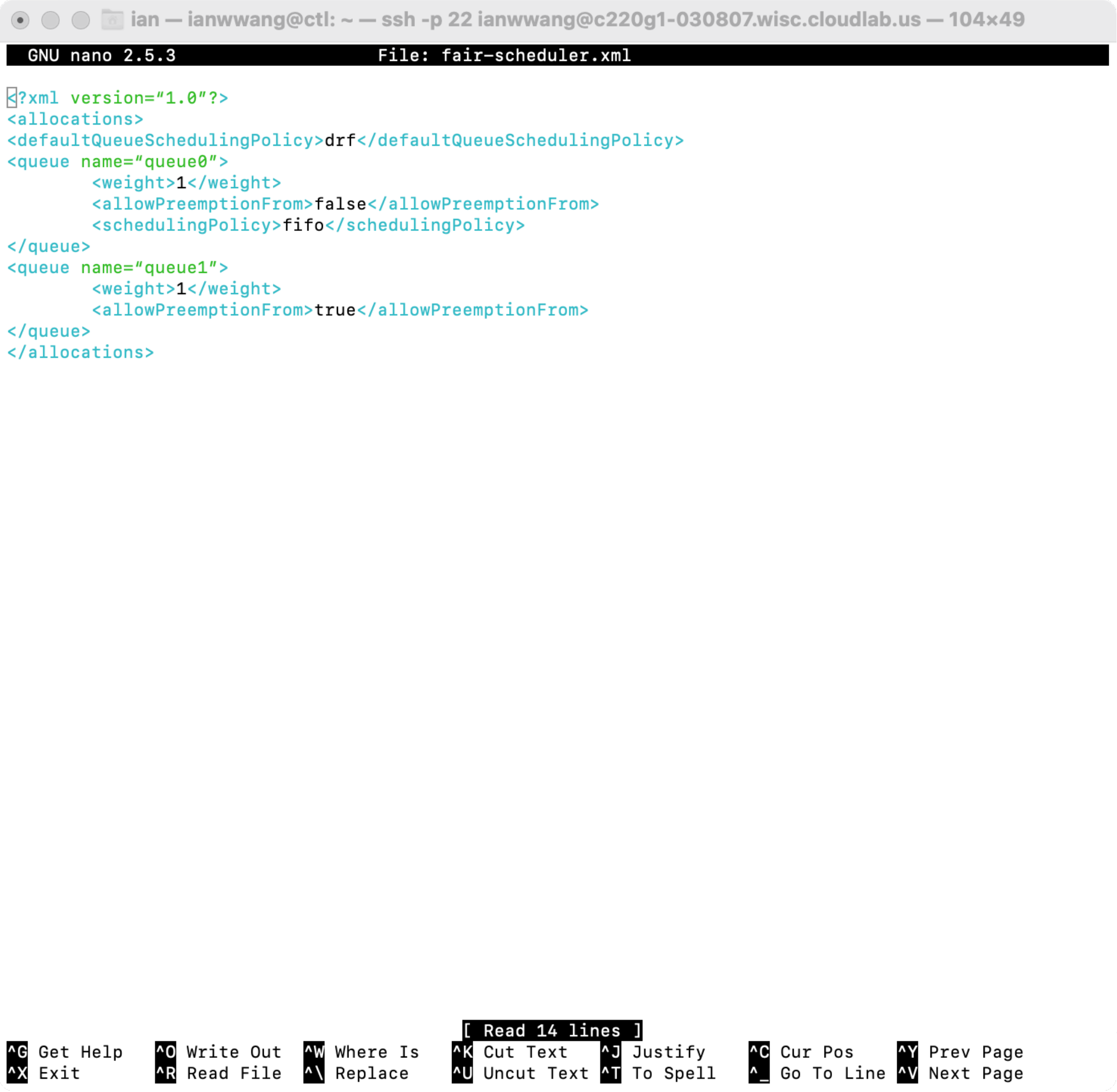
```
  GNU nano 2.5.3                 File: slaves

cp-1-0
cp-2-0
cp-3-0
cp-4-0
```

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify
^X Exit   ^R Read File   ^\ Replace   ^U Uncut Text ^T To Spell

2.4 Setup YARN (cont.)

- On Cloudlab each node, setup Fair-scheduler
  - Create hadoop/etc/fair-scheduler.xml
    - 2 queues: queue0 & queue1
    - DRF is the resource allocation polity used between two queues.
      - The internal scheduling policy for queue0 is FIFO.

```
<?xml version="1.0"?>
<allocations>
<defaultQueueSchedulingPolicy>drf</defaultQueueSchedulingPolicy>
<queue name="queue0">
      <weight>1</weight>
      <allowPreemptionFrom>false</allowPreemptionFrom>
      <schedulingPolicy>fifo</schedulingPolicy>
</queue>
<queue name="queue1">
      <weight>1</weight>
      <allowPreemptionFrom>true</allowPreemptionFrom>
</queue>
</allocations>
```



ian — ianwwang@ctl: ~ — ssh -p 22 ianwwang@c220g1-030807.wisc.cloudlab.us — 104×49

```
  GNU nano 2.5.3                File: fair-scheduler.xml

<?xml version=“1.0”?>
<allocations>
<defaultQueueSchedulingPolicy>drf</defaultQueueSchedulingPolicy>
<queue name=“queue0”>
        <weight>1</weight>
        <allowPreemptionFrom>false</allowPreemptionFrom>
        <schedulingPolicy>fifo</schedulingPolicy>
</queue>
<queue name=“queue1”>
        <weight>1</weight>
        <allowPreemptionFrom>true</allowPreemptionFrom>
</queue>
</allocations>

                                            [ Read 14 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line ^V Next Page
```

2.4 Setup YARN (cont.)

At the master node (ctl),

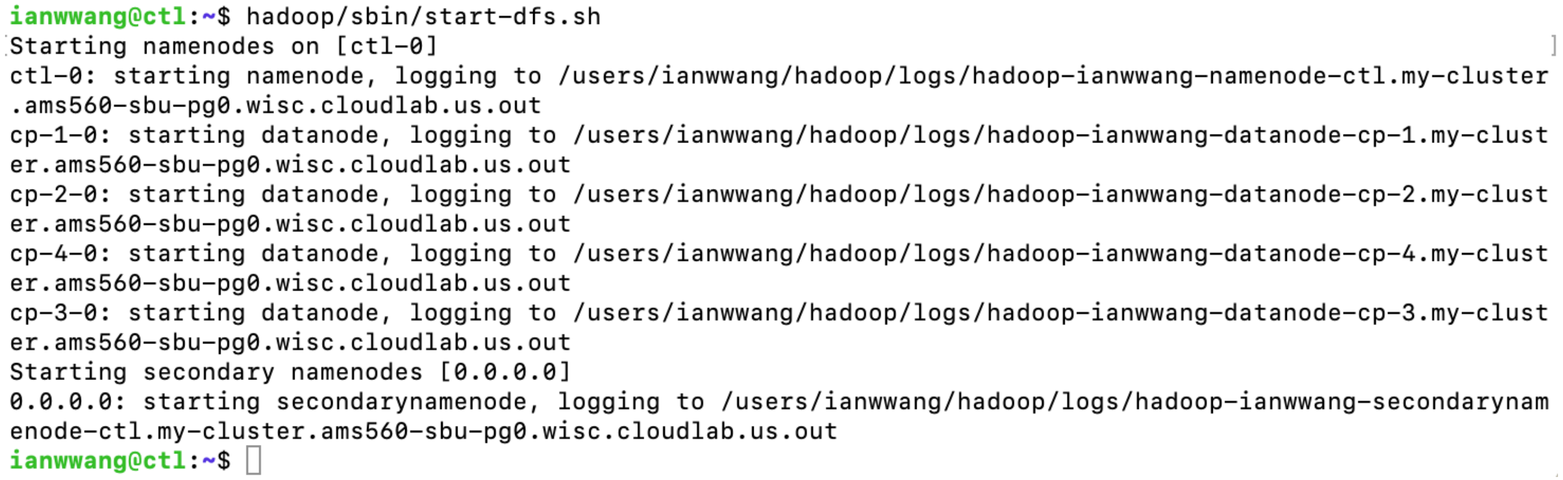
format HDFS first:

```
$ hadoop/bin/hdfs namenode -format hdfs
```

2.4 Setup YARN (cont.)

At the master node (ctl),

- Start HDFS:
  - $ hadoop/sbin/start-dfs.sh



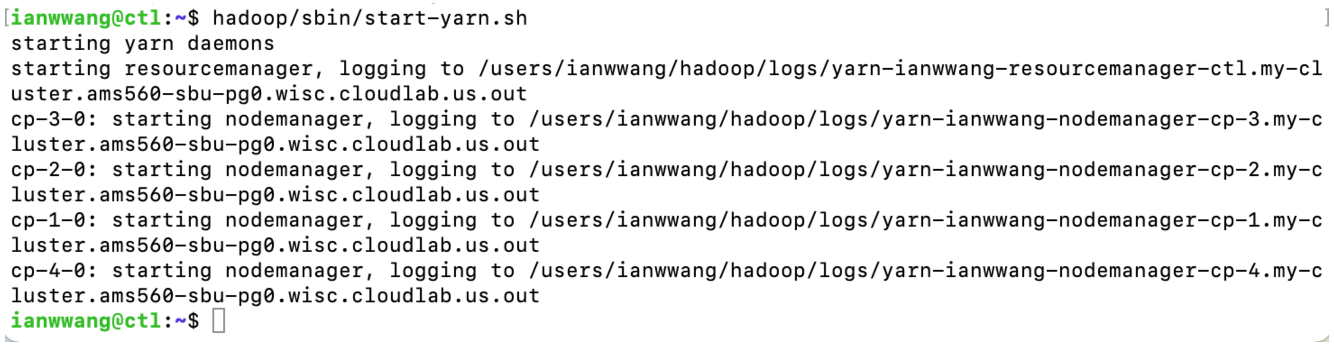
```
ianwwang@ctl:~$ hadoop/sbin/start-dfs.sh
Starting namenodes on [ctl-0]
ctl-0: starting namenode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-namenode-ctl.my-cluster
.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-1-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-1.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-2-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-2.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-4-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-4.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-3-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-3.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-secondarynam
enode-ctl.my-cluster.ams560-sbu-pg0.wisc.cloudlab.us.out
ianwwang@ctl:~$ 
```

2.4 Setup YARN (cont.)

At the master node (ctl),

Start Yarn:

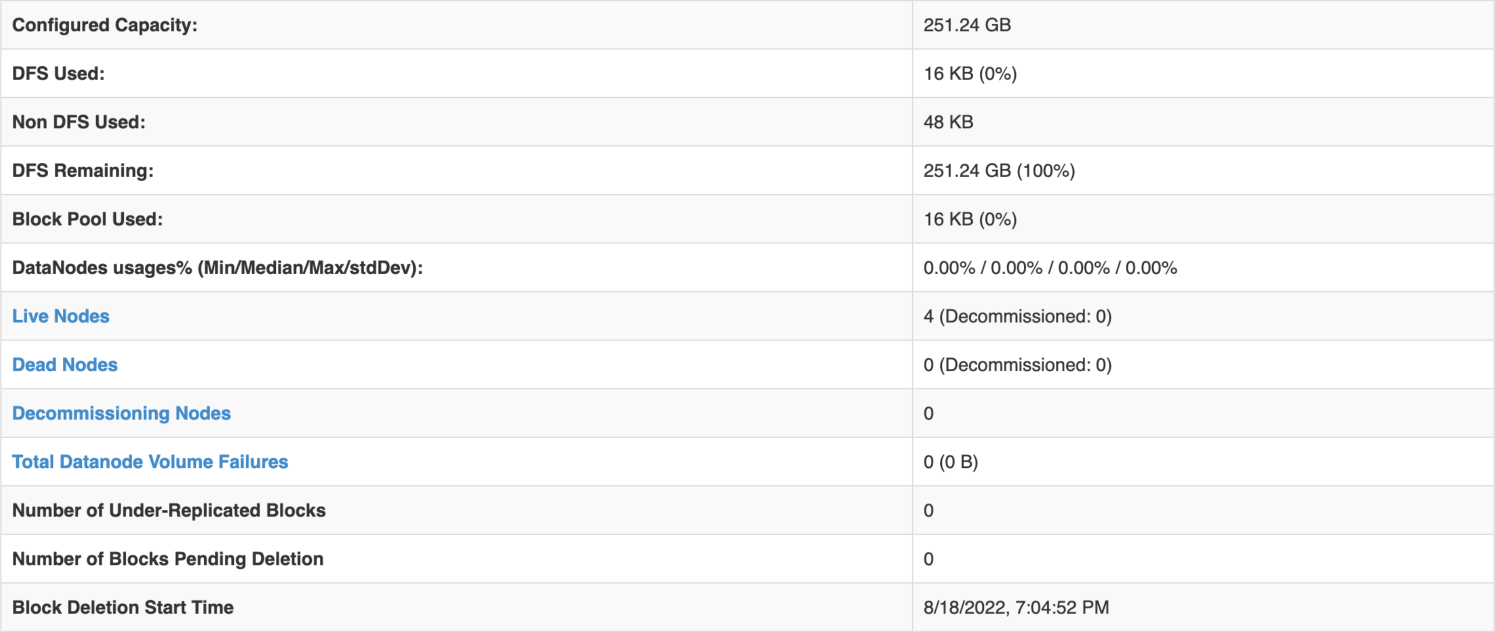
$ hadoop/sbin/start-yarn.sh



```
[ianwwang@ctl:~$ hadoop/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-resourcemanager-ctl.my-cl
uster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-3-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-3.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-2-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-2.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-1-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-1.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-4-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-4.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
ianwwang@ctl:~$ 
```

2.5 Test YARN

- Visit http://[master node's hostname]:50070 & check the HDFS status
  - The capacity and the number of live nodes are greater than zero.
  - You could also check by
    - $ hadoop/bin/hdfs dfsadmin -report



Configured Capacity:	251.24 GB
DFS Used:	16 KB (0%)
Non DFS Used:	48 KB
DFS Remaining:	251.24 GB (100%)
Block Pool Used:	16 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	4 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	8/18/2022, 7:04:52 PM

2.5 Test YARN (cont.)

check the Yarn status

```
$ hadoop/bin/yarn node -list
```

3. What to submit

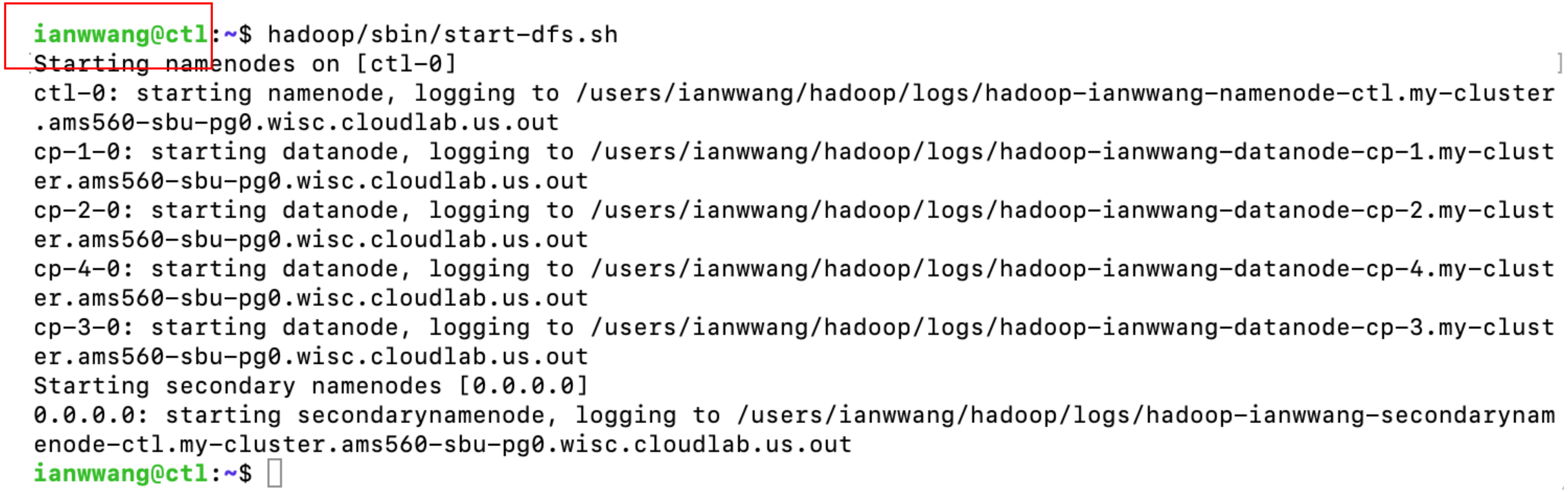
According to the requirements in Assignment #1, you are to submit a screen recording for each step:

1. A cluster has been created on Cloudlab **(5%)**
2. JDK is setup on each node **(5%)**
3. Passwordless SSH has been setup **(5%)**
4. Yarn has been downloaded in each node **(5%)**
5. Successful setting up of the files core-site.xml, hdfs-site.xml, and yarn-site.xml **(20%)**
6. Setting up of fair-scheduler.xml **(10%)**
7. Creating of slaves file in each node **(10%)**
8. Successfully running HDFS and Yarn (See more in the future slides) **(15%)**
9. Correct number of live nodes on 50070 site **(15%)**
10. **Full screen recording of the process (10%)**

You may submit screenshots, but you will not earn point 10 which is 10% of this assignment's grade

3. What to submit (cont.)

1. Use screenshot to show that you can successfully start hdfs.



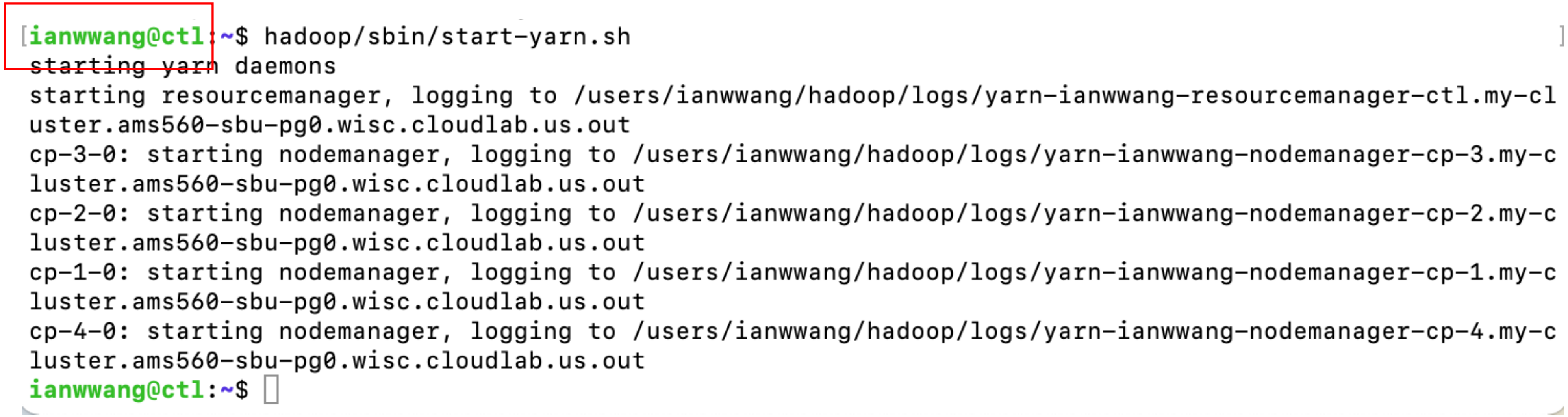
Your Name

```
ianwwang@ctl:~$ hadoop/sbin/start-dfs.sh
Starting namenodes on [ctl-0]
ctl-0: starting namenode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-namenode-ctl.my-cluster
.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-1-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-1.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-2-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-2.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-4-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-4.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-3-0: starting datanode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-datanode-cp-3.my-clust
er.ams560-sbu-pg0.wisc.cloudlab.us.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /users/ianwwang/hadoop/logs/hadoop-ianwwang-secondarynam
enode-ctl.my-cluster.ams560-sbu-pg0.wisc.cloudlab.us.out
ianwwang@ctl:~$ 
```

3. What to submit (cont.)

1. Use screenshot to show that you can successfully start yarn.

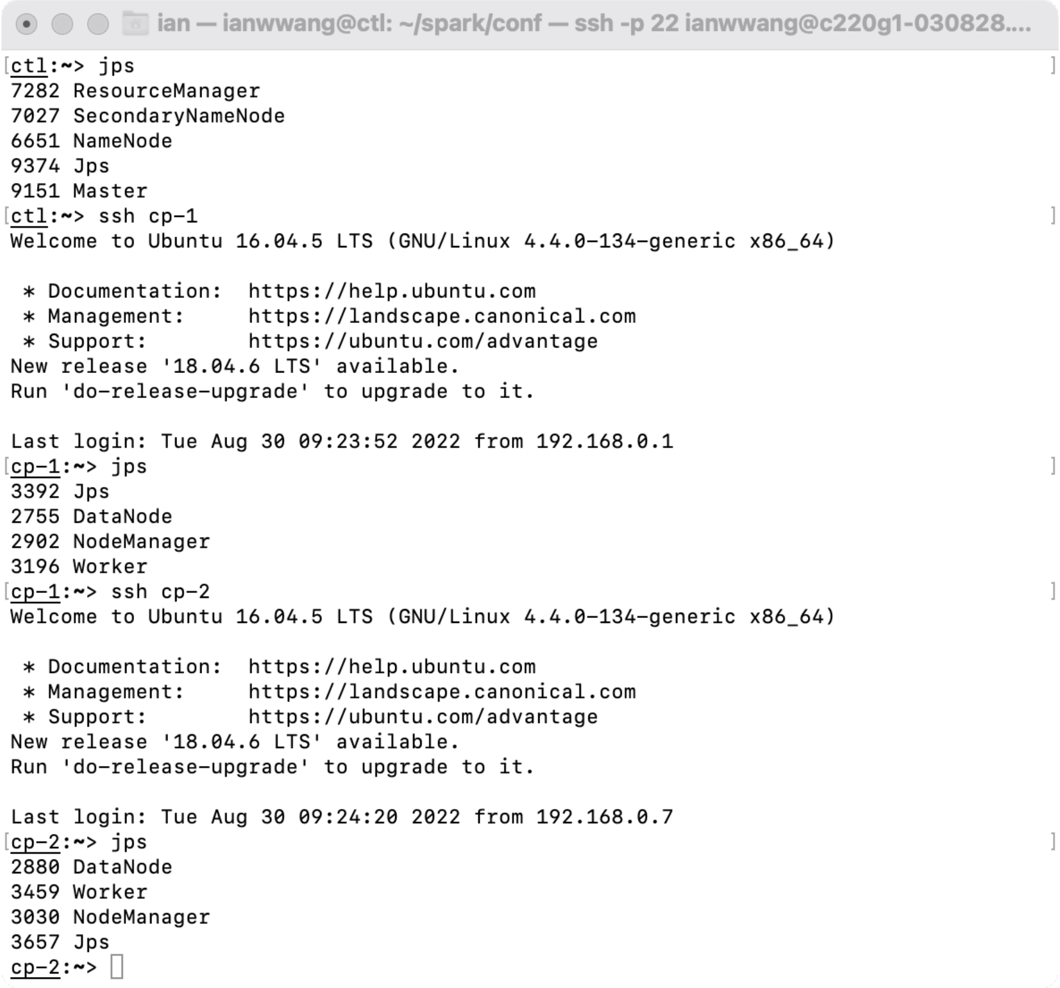
Your Name



```
ianwwang@ctl:~$ hadoop/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-resourcemanager-ctl.my-cl
uster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-3-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-3.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-2-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-2.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-1-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-1.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
cp-4-0: starting nodemanager, logging to /users/ianwwang/hadoop/logs/yarn-ianwwang-nodemanager-cp-4.my-c
luster.ams560-sbu-pg0.wisc.cloudlab.us.out
ianwwang@ctl:~$
```

3. What to submit (cont.)

- Use “jps” command to show that all your nodes works well (use screenshot).
- For me: ianwwang@ctl:~$ jps



ian — ianwwang@ctl: ~/spark/conf — ssh -p 22 ianwwang@c220g1-030828....

```
ctl:~> jps
7282 ResourceManager
7027 SecondaryNameNode
6651 NameNode
9374 Jps
9151 Master
ctl:~> ssh cp-1
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-134-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Aug 30 09:23:52 2022 from 192.168.0.1
cp-1:~> jps
3392 Jps
2755 DataNode
2902 NodeManager
3196 Worker
cp-1:~> ssh cp-2
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-134-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Aug 30 09:24:20 2022 from 192.168.0.7
cp-2:~> jps
2880 DataNode
3459 Worker
3030 NodeManager
3657 Jps
cp-2:~> 
```