CSCI 4180 – Tutorial 2

VM Management and Hadoop Setup

Zhao, Jia

jzhao@cse.cuhk.edu.hk

2022.09.21

Outline

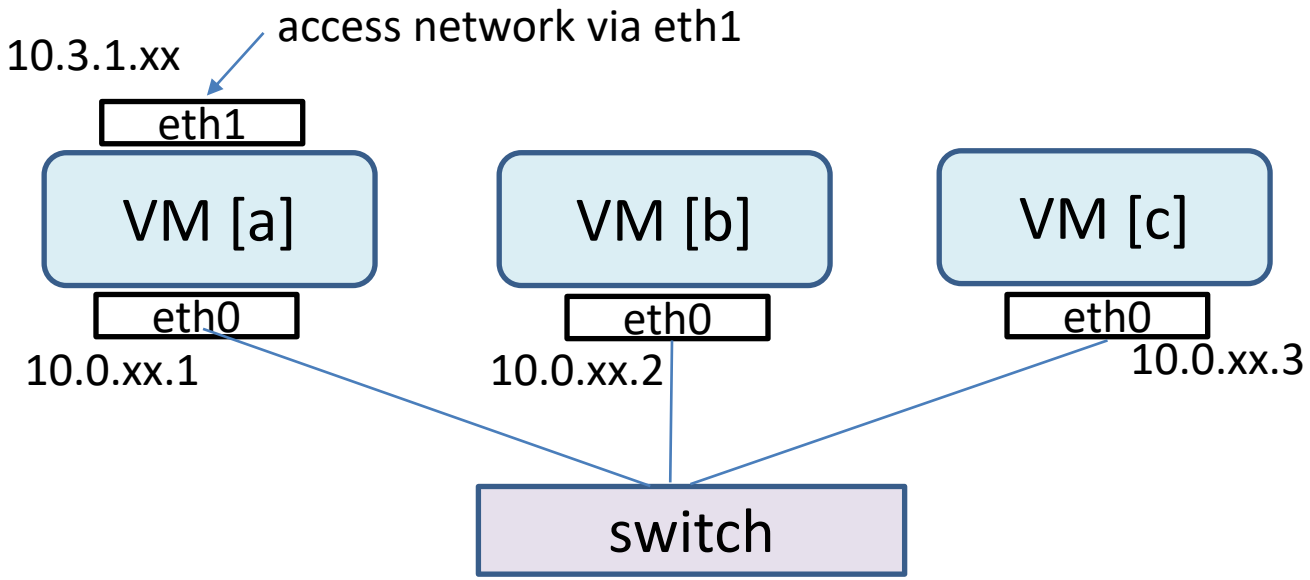
- VM Management
  - Connect VMs
  - Access VMs
- Hadoop Setup
  - Setup Hadoop Cluster (fully distributed mode)
  - WordCount Example

Outline

- VM Management
  - Overview
  - Access CSE network
  - Power on VM
  - Connect to VM
  - Add new users
  - VM configuration
  - Less password ssh

VM Overview

- Each group has three VMs
  - Check your emails for the VM information(may send next week)
- VM Overview
  - Only VM [a] can access network, can be SSH from external
  - VM [a], VM[b], VM[c] forms a small intra-net

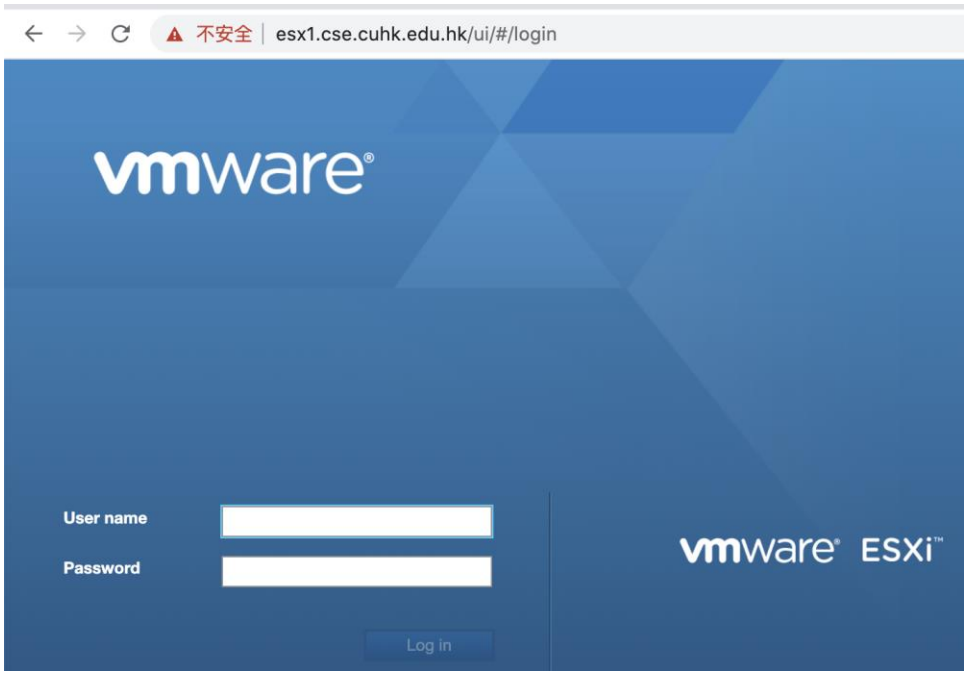


Access CSE network

- Inside CSE network
  - VM can only be accessed inside CSE network
  - CSE Lab (SHB 924), or via CSE VPN (Using OpenVPN), or CSE gateway
- In CSE lab or using CSE VPN
  - Support both HTTP and SSH
  - Can directly access CSE website or SSH to CSE machine
- Using CSE gateway
  - To access CSE website
    - Should setup the CSE gateway as http proxy
    - Check proxy setting here
  - Can directly support SSH

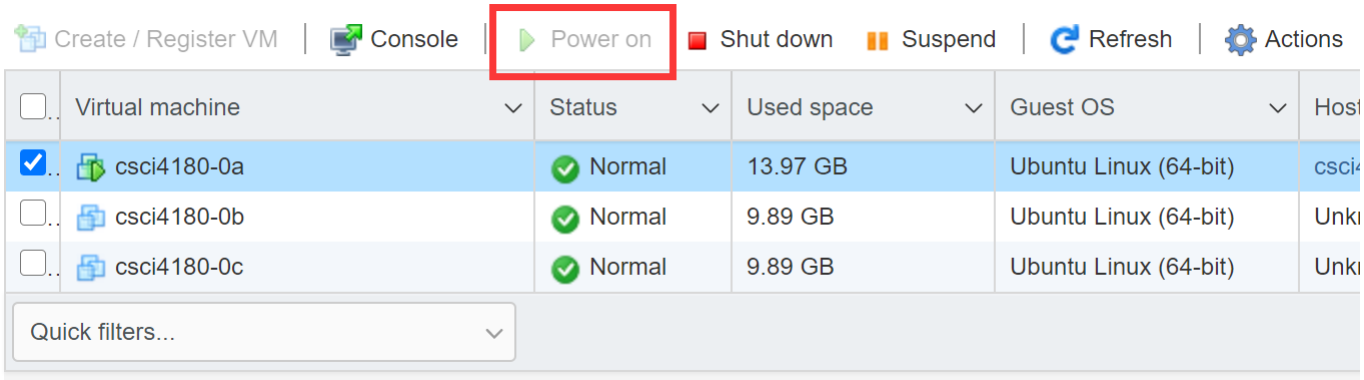
Power on the VM

- Access CSE network
- Web Interface
  - https://esx1.cse.cuhk.edu.hk/ui
    - Check the email for the link to login your group
  - Account
    - Cse unix name
  - Password
    - Cse unix password



Power on the VM

- Power on the VMs



- Actions in web UI
  - After power on the VM
    - Can close ui, and connect to VM via SSH
    - Just keep the VM power on, you don't need to access the web UI again
  - Or access VM in UI
    - Click the Console button in the left side of the power on button
    - Would be very slow, we suggest to access VM via SSH

Connect to VM

- Connect to VM
  - Only VM [a] can be accessed from external via SSH
    - Via the eth1 of VM [a]
    - VM [b] and VM [c] can be ssh from VM [a] by the intra-net
  - Default account and password for each VM
    - USER: csci4180, PWD: csci4180test
    - Each VM will require you to modify the password in your first login
      - **Make sure your VM doesn't be accessed by others**
  - SSH to VM [a] within CSE network
    - *$ssh -p 130xx csci4180@projgw.cse.cuhk.edu.hk*
    - *xx* is your VM group id, check the emails
  - SSH between VM [a|b|c]
    - *ssh 10.0.xx.1*, *ssh 10.0.xx.2*, *ssh 10.0.xx.3*

Transfer file among VMs

- Between external and VM [a]
  - Single file mycode.c
    - to VM [a]: $*scp -P 130xx mycode.c* csci4180@projgw.cse.cuhk.edu.hk:~/
    - to external: $*scp mycode.c [external user]@external.ip:~/*
  - Directory mydir
    - to VM [a]: $*scp -P 130xx –r mydir* csci4180@projgw.cse.cuhk.edu.hk:~/
    - to external: $*scp –r mydir [external user]@external.ip:~/*
- Between VM [a|b|c]
  - Single file mycode.c from VM [a] to VM [b]
    - *$scp mycode.c* csci4180@10.0.xx.2:~/
  - Directory mydir from VM [b] to VM [c]
    - *$scp –r mydir* csci4180@10.0.xx.3:~/

Access VMs

- Access VMs
  - Changed VM hostname from csci4180 to vm1, vm2, vm3
    - So we can better identify different VMs
  - On **each** VM
    - *$sudo vim /etc/hostname*
    - Changed the hostname
      - E.g., changed hostname form csci4180 to vm1 on VM [a]
      - similarly, to vm2 on VM [b], to vm3 on VM [c]
    - Also change the hosts *$sudo vim /etc/hosts*
      - E.g., Change line "127.0.1.1 csci4430" to ”10.0.0.x vm[x]"
    - $*sudo reboot*
      - Make the change takes effect, it takes about 1~2 minutes to reboot
  - Login VMs after reboot
    - Now the prompt of VM [a] should be csci4180@vm1

Access VMs

- Access VMs
  - Create a new user hadoop for each VM
    - Later we will use this user for our hadoop cluster
  - On **each** VM
    - $*sudo adduser hadoop*
      - Just press enter for all requirements
      - This creates a new user hadoop
    - $*sudo usermod -aG sudo hadoop*
      - Add the user hadoop to sudo group, so we will have root privileges
    - $*su hadoop*
      - Change to hadoop user
      - Now the prompt should be hadoop@vm1

Access VMs

- Configure hosts
  - On **each** VM
    - $*sudo vim /etc/hosts*
    - Mapping the ip with host name, append the following at the end

```
10.0.0.1        vm1
10.0.0.2        vm2
10.0.0.3        vm3
```

  - Now you can ssh using the host name rather than using ip
    - e.g., $*ssh hadoop@vm2*, $*ssh vm2*
    - Also works when you transfer file

Access VMs

- Configure ssh without entering password
  - On **each** VM as hadoop user
    - *$ssh-keygen*
      - Always press enters for any requirements
    - *$ssh-copy-id hadoop@vm1*
    - *$ssh-copy-id hadoop@vm2*
    - *$ssh-copy-id hadoop@vm3*
  - Now to verify
    - On vm1 you can ssh to vm2 and vm3 without entering password
      - –$*ssh hadoop@vm2*
      - –$*ssh hadoop@vm3*
    - Similar on vm2 and vm3

Outline

- Hadoop Setup (Fully distributed mode)
  - Install java
  - Download hadoop
  - Setup hadoop configuration
  - Test hadoop
  - Run the wordcount example

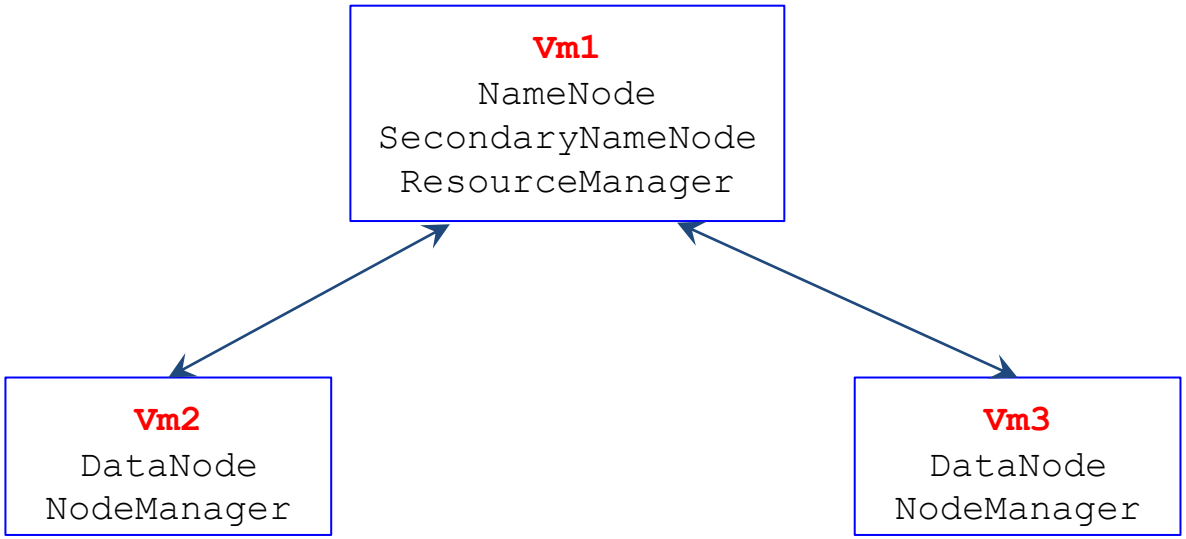
  - In assignment-1 1-4 parts, you only need to configure Hadoop in pseudo-distributed mode. For the bonus part, you will need to configure Hadoop in fully distributed mode. (See details in specification)
  - In this tutorial, we focus on fully distributed mode

Install Java

- Install java
  - On **each** VM as hadoop user
    - Download openjdk-8u342:here
      - vm1 can download java by wget, or download in your local and scp to vm1
      - vm2 and vm3 should be copy from vm1 by scp
    - *$sudo mkdir /usr/lib/jvm*
      - Or any directory you like
    - *$tar xvf openlogic-openjdk-8u342-b07-linux-x64.tar*
    - *$sudo mv openlogic-openjdk-8u342-b07-linux-x64/ /usr/lib/jvm/jdk8u342*
    - Update environment variable, *$vim ~/.bashrc*
      - export JAVA_HOME=/usr/lib/jvm/jdk8u342
      - export PATH=$PATH:$JAVA_HOME/bin
    - *$source ~/.bashrc*
    - *$java -version*
    - You can check the value of JAVA_HOME and PATH by
      - echo $JAVA_HOME
      - echo $PATH

Hadoop (Fully distributed mode)

- Overview of our hadoop cluster
  - Architecture



Download Hadoop

- Download Hadoop
  - Download Hadoop on vm1
    - Set http&https proxy(Add to ~/.bashrc)
      - export http_proxy="http://proxy.cse.cuhk.edu.hk:8000/"
      - export https_proxy="http://proxy.cse.cuhk.edu.hk:8000/"
    - wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz
    - Decompress Hadoop
      - *$tar zxvf hadoop-2.7.3.tar.gz*
  - Download Hadoop on vm2 and vm3
    - On vm1, scp the binary file to vm2 and vm3
      - *scp hadoop-2.7.3.tar.gz hadoop@vm2:~/*
      - *scp hadoop-2.7.3.tar.gz hadoop@vm3:~/*
    - Then do the same thing as on vm1
  - Note: It is more convenient that you configure everything in vm1 and directly copy the whole hadoop directory on vm1 to vm2 and vm3

Setup Hadoop (Fully distributed mode)

- Setup environment
  - Add following $*vim ~/.bashrc*
    - `export HADOOP_HOME=/home/hadoop/hadoop-2.7.3`
    - `export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin`
    - `export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar`
  - $*source ~/.bashrc*
  - Similarly, you can finish all configuration in vm1, and directly copy the configuration file *~/.bashrc* to vm2 and vm3, source it

Setup Hadoop (Fully distributed mode)

- Configuration file of Hadoop
  - We have provided these file in course website
  - Download the **csci4180_tuto2.tar.gz** in tutorial page
  - On EACH node
    - *$tar zxvf csci4180_tuto2.tar.gz*
    - *$cp csci4180_tuto2/csci4180_hadoop_conf/* hadoop-2.7.3/etc/hadoop*
  - Again: It is more convenient that you configure everything in vm1 and directly copy the whole hadoop directory on vm1 to vm2 and vm3

Setup Hadoop (Fully distributed mode)

- Format namenode on vm1
  - Check your configuration
    - $hadoop version
    - You should see the hadoop information
  - $*hadoop namenode -format*
  - Start hadoop cluster on namenode
    - $*start-dfs.sh*
    - $*start-yarn.sh*

Setup Hadoop (Fully distributed mode)

- Setup hadoop cluster
  - Operations related to HDFS
    - List files in hdfs
      - $hadoop fs –ls <hdfs URI>
    - Make directory
      - $hadoop fs -mkdir -p <hdfs URI>
    - Remove file from hdfs
      - $hadoop fs -rm <hdfs URI>
    - Write into hdfs
      - $hadoop fs -put <local flie> <hdfs URI>
    - Read from hdfs
      - $hadoop fs -get <hdfs URI> <local file>
    - Show the content of the file in hdfs
      - $hadoop fs -cat <hdfs URI>

Setup Hadoop (Fully distributed mode)

- WordCount Example
  - Find the source code **WordCount.java** in csci4180_tuto2.tar.gz
  - Compile the source code
    - $mkdir ~/wordcount
    - $cp ~/csci4180_tuto2/WordCount.java ~/wordcount
    - $cd ~/wordcount
    - $hadoop com.sun.tools.javac.Main WordCount.java
    - $jar cf wc.jar WordCount*.class

Setup Hadoop (Fully distributed mode)

- WordCount Example
  - Make the HDFS directories
    - /user/hadoop/wordcount/input
      - The input directory in HDFS, which contains all the input file for MapReduce
      - $hdfs dfs -mkdir /user && hdfs dfs -mkdir /user/hadoop && hdfs dfs -mkdir /user/hadoop/input
    - /user/hadoop/wordcount/output
      - The output directory in HDFS
      - Make sure this doesn't exist before running, otherwise will casue error
  - Run the example
    - $hadoop jar wc.jar WordCount /user/hadoop/input /user/hadoop/output
  - Then check the result in output directory

Simple Demo

Target of Assignment-1

- Pseudo-distributed mode
  - https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

CSCI 4180 – Tutorial 2

VM Management and Hadoop Setup

– End –

References:

- MapReduce Tutorial
- Parameters of MapReduce