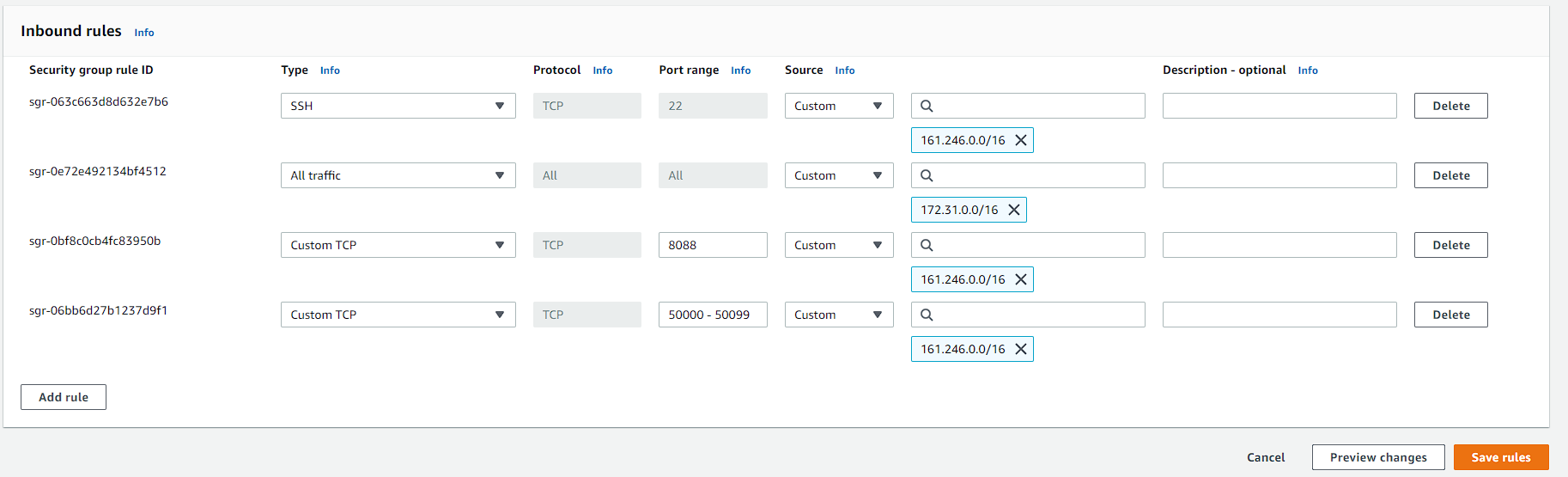
Hadoop Cheat Sheet

> Region -> Oregon

1. ตั้ง Security group

161.246.0.0/16



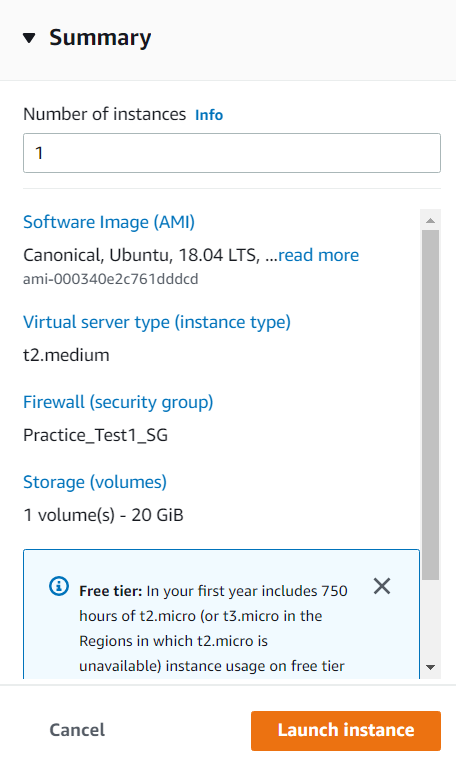
inbound security rule มี 2 กลุ่มหลักๆ

* ภายนอก คือ 22 สำหรับต่อ SSH, 8088, 50000-50099 สำหรับ Dashboard ของ Hadoop
* ภายในให้ allow all (ตัวที่ 2 ของ inbound)

ถ้าตั้งผิด

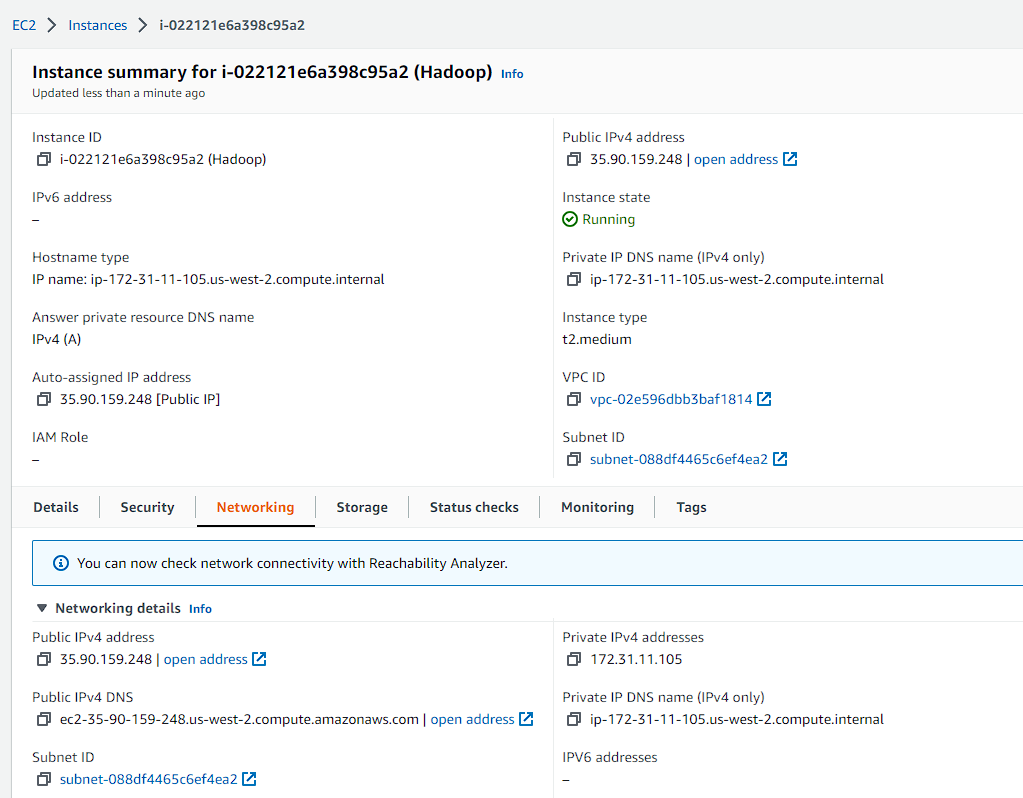
* ต่อ ssh ไม่ติด
* ต่อ dashboard ไม่ได้ทั้งที่เปิด hadoop แล้ว
* cluster ต่อกันไม่ติด

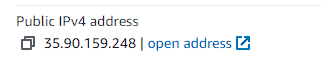
1. สร้าง instance



สร้างinstance

launch instance> ตั้งชื่อ > ใช้ubuntu > versionไหนก็ได้ อ.ใช้ 18.04 > instance type t2.medium > create new Key pair > ตั้งชื่อ > rsa > pem > network setting > เลือก security groupที่ทำไว้ > configure storage > 1\*20 GiB gp2 > create





# เข้าvscode

Shift+ctrl+p > ssh config เลือกอันปกติที่ใช้ > คำสั่ง

Host AWS-EC2

HostName public IP จาก instance

User ubuntu

IdentityFile "ที่อยู่key"

# Single Node Setup

sudo apt-get update

ถ้าลืมรัน จะทำให้ลง Java ไม่ได้ในบางครั้ง

ลง open-ssh

sudo apt-get install openssh-server

สร้าง key สำหรับ ssh

ssh-keygen

กำหนดให้ key นี้ไว้ใจได้

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

ถ้าลืมทำ จะทำให้ตอนทำ cluster เครื่องอื่นจะ ssh ไม่ติด หรือ error publickey ตอนรัน

test ssh

ssh localhost

exit

ลง JAVA

sudo apt-get install openjdk-8-jdk

ลง Hadoop

wget https://archive.apache.org/dist/hadoop/core/hadoop-2.6.5/hadoop-2.6.5.tar.gz && sudo tar -xvf hadoop-2.6.5.tar.gz && sudo mv ./hadoop-2.6.5 /usr/local/hadoop

Download + แตกไฟล์ + ย้ายไฟล์ที่โหลดมาไปไว้ที่ /usr/local/hadoop

ตั้งค่า environment

nano ~/.bashrc

แล้วเพิ่ม

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export PATH=$PATH:$JAVA\_HOME/bin

export HADOOP\_HOME=/usr/local/hadoop

export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

source ~/.bashrc

เพื่อให้ hadoop เรียกใช้ java ได้ และเรียกพวก start-dfs.sh ได้ ถ้าตั้งผิด

* hadoop จะเปิดไม่ได้
* รัน start-dfs.sh, start-yarn.sh ไม่ได้

สร้าง Folder สำหรับเก็บ Log

cd /usr/local && sudo mkdir /var/log/hadoop

ให้สิทธิ์แก้ไข

sudo chown -R ubuntu:ubuntu /var/log/hadoop

sudo chown จะเป็นการให้สิทธิ์การใช้งาน directory ถ้าตั้งผิดหรือลืม

* Permission denied ตอนรัน start-dfs.sh, start-yarn.sh

บอกที่เก็บ Log กับ Hadoop และ Yarn

cd /usr/local/hadoop/etc/hadoop

nano hadoop-env.sh

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP\_LOG\_DIR=/var/log/hadoop

nano yarn-env.sh

export YARN\_LOG\_DIR=/var/log/hadoop

ถ้าเปิด log ไม่ได้ก็อาจจะเป็นที่นี่ก็ได้

กำหนด core site ของ yarn

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://<private ip ของเครื่อง master>:9000</value>

</property>

</configuration>

ตรง highlight ให้แทนด้วย ip หมดเลย เช่น

<value>hdfs://<private ip ของเครื่อง master>:9000</value>

เป็น

<value>hdfs://172.31.11.234:9000</value>

ถ้าตั้งผิดจะตั้ง cluster ไม่ได้ slave node ไม่ขึ้น หรือ service ล่มไปเลย ทุกเครื่องใน cluster จะมี ip ตรงนี้เป็นอันเดียวกัน

ตั้ง Folder สำหรับ Data

sudo mkdir -p /var/hadoop\_data/namenode && sudo mkdir -p /var/hadoop\_data/datanode && sudo chown ubuntu:ubuntu -R /var/hadoop\_data

เอาไว้เก็บ Data ของ hadoop ถ้าไม่ได้ตั้งหรือ chown จะทำให้ hadoop เขียนข้อมูลไม่ได้ หรือเปิดไม่ติด

ตั้งค่า Hadoop

nano hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/var/hadoop\_data/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/var/hadoop\_data/datanode</value>

</property>

</configuration>

nano yarn-site.xml

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value><private ip ของเครื่อง master></value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value><private ip ของเครื่อง master>:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value><private ip ของเครื่อง master>:8031</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value><private ip ของเครื่อง master>:8032</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value><private ip ของเครื่อง master>:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value><private ip ของเครื่อง master>:8088</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property> <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

ตรง highlight ให้แทนด้วย ip หมดเลย เช่น

<value><private ip ของเครื่อง master>:8088</value>

เป็น

<value>172.31.11.234:8088</value>

ถ้าตั้งผิดจะตั้ง cluster ไม่ได้ slave node ไม่ขึ้น หรือ service ล่มไปเลย ทุกเครื่องใน cluster จะมี ip ตรงนี้เป็นอันเดียวกัน

ตั้งค่า Map-reduce framework

cp mapred-site.xml.template mapred-site.xml

nano mapred-site.xml

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

format data

hdfs namenode -format && hdfs namenode -format

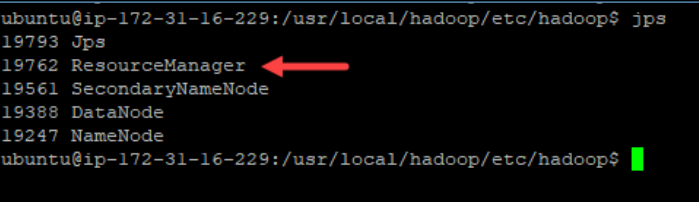
ถ้า hadoop หลอนๆ ก็ลองรันซักที แต่ระวังข้อมูลใน hdfs หาย

เปิด

start-dfs.sh && start-yarn.sh

เทส service หลังจาก start-dfs.sh เเละ start-yarn.sh เเล้ว

jps



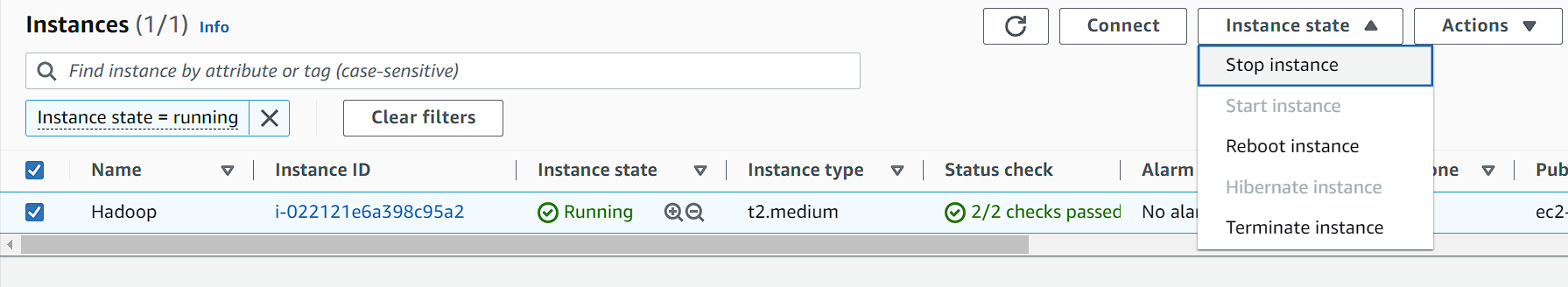
ถ้าเซ็ตถูกหมดจะได้แบบนี้

เปิด http://<public ip ของ master>:50070 จะต้องได้หน้า dashboard

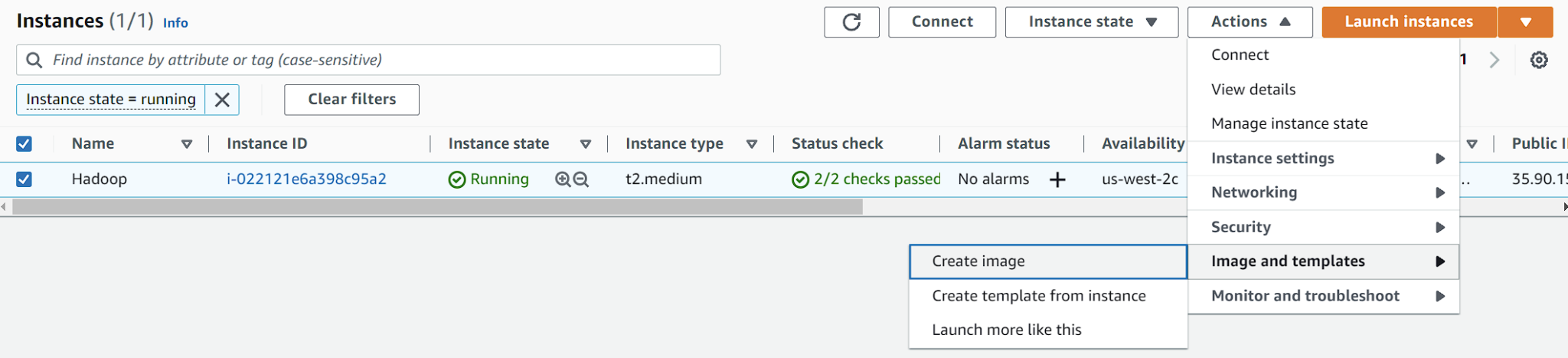
ถ้าเปิดไม่ติด

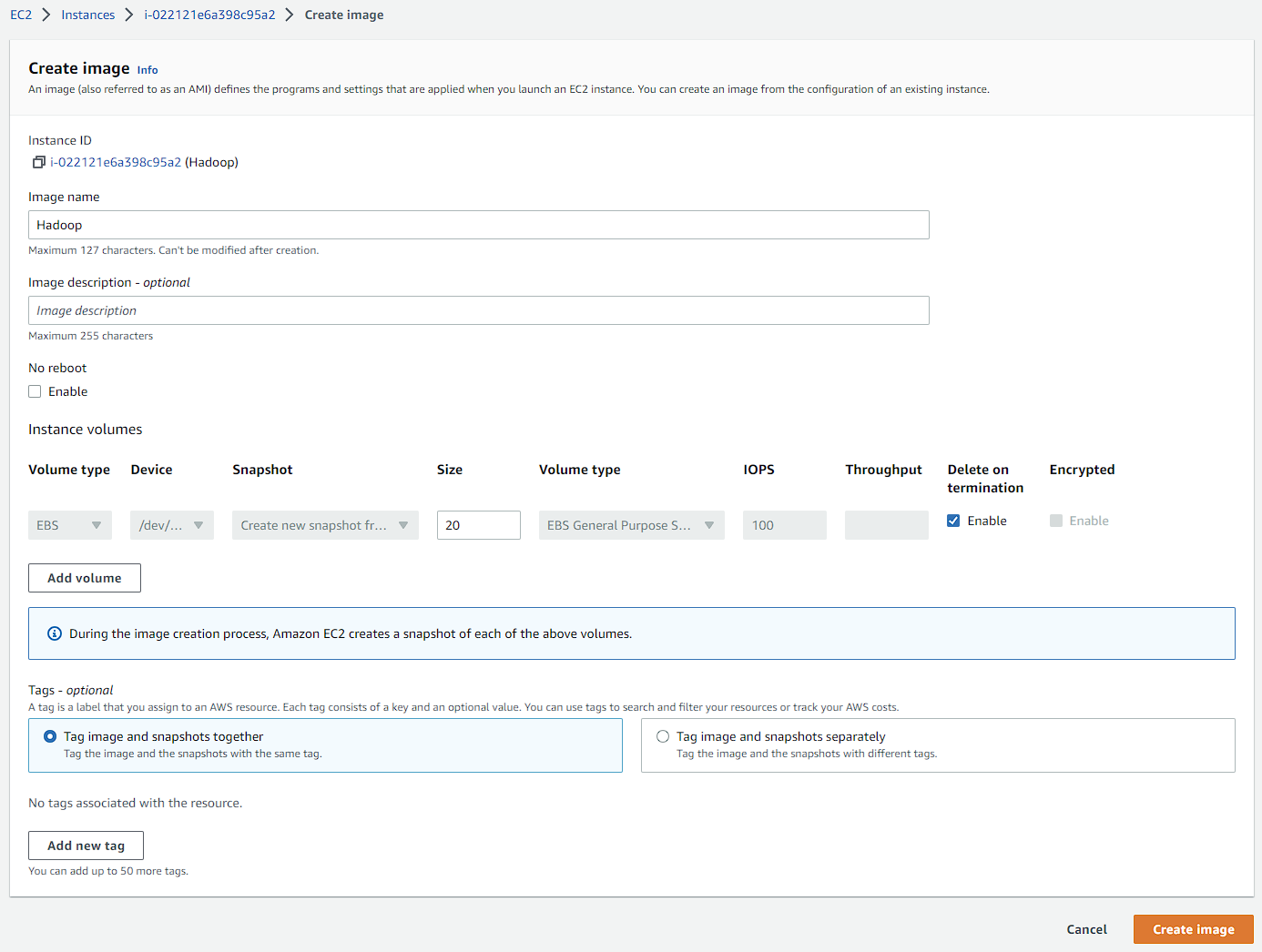
* ลองกับไปดู security group ตรง inbound ว่าได้เพิ่ม 50000-50099 จากคอมที่เราใช้ (161.246.0.0/16 ถ้าอยู่ที่ สจล.)

เข้า aws dashboard >> Stop instance รอให้เครื่อปิด



>> Actions >> Image and templates >> Create image ตั้ง Image name แล้ว create + รอยาวๆ





Q&A

ตอน sudo apt-get update ขึ้น connection timeout

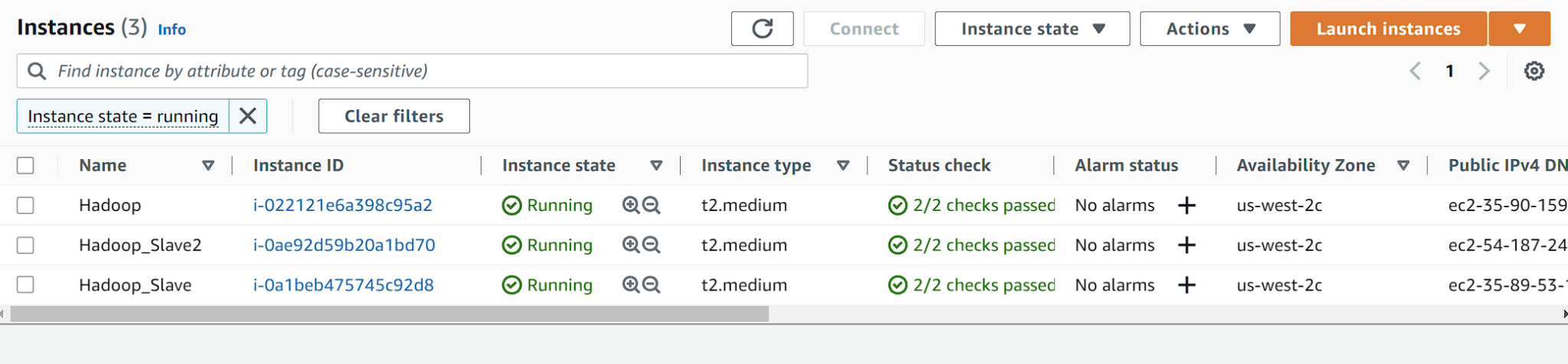
* ดูที่ security group ว่าตั้ง outbound เป็น all traffic from 0.0.0.0/0 รึป่าว ถ้าไม่มีเครื่องจะรันพวก apt-get, wget ไม่ได้ รวมถึง hadoop ด้วย

ตั้งค่าถูกหมดแล้ว แต่ jps แล้ว service บางตัวไม่ขึ้น, เข้า :50070 ไม่ได้

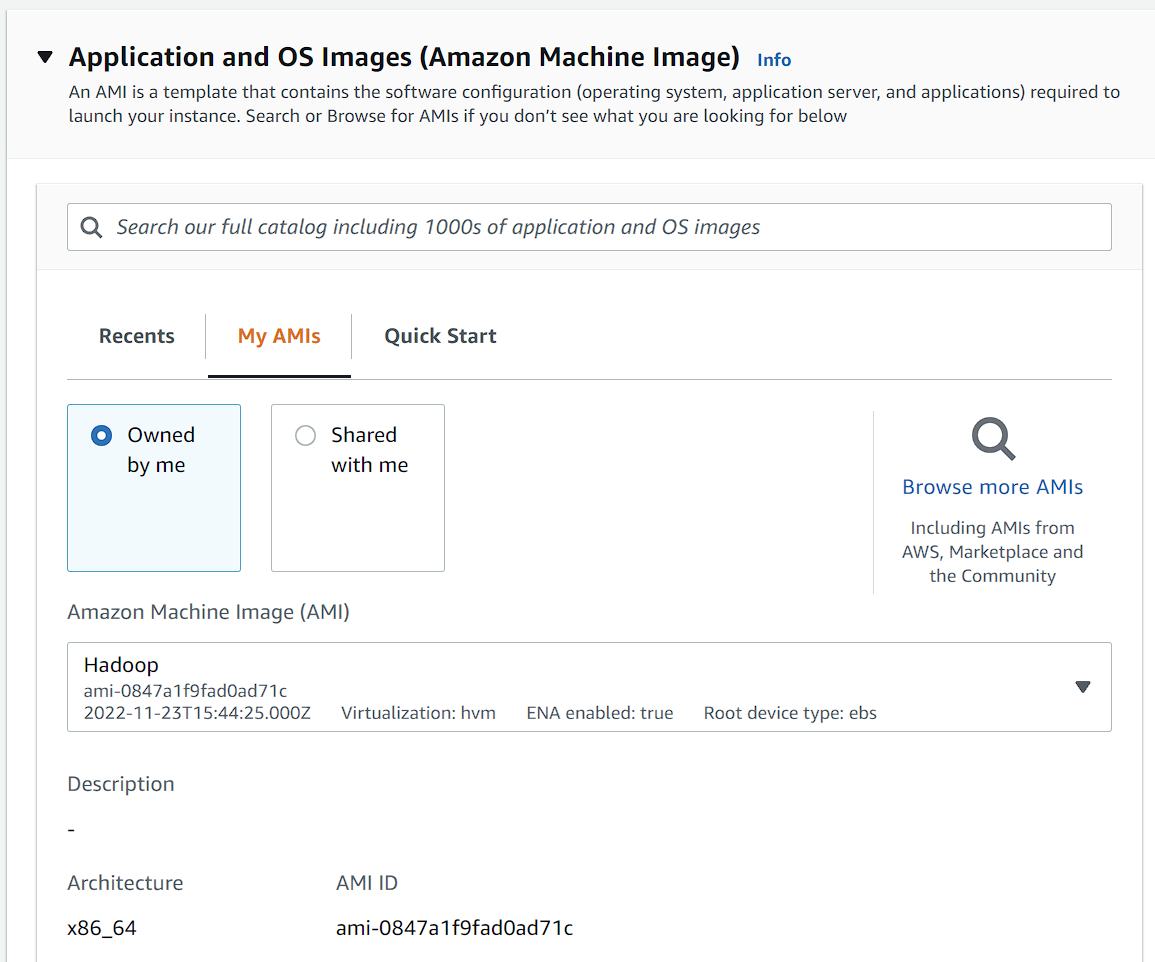
* ลองรัน stop-all.sh && hdfs namenode -format && hdfs namenode -format แล้ว start ใหม่
* อาจจะเป็นที่ตอนรันครั้งแรกแล้วแตก แล้วแก้โดยไม่ format

# Cluster Setup

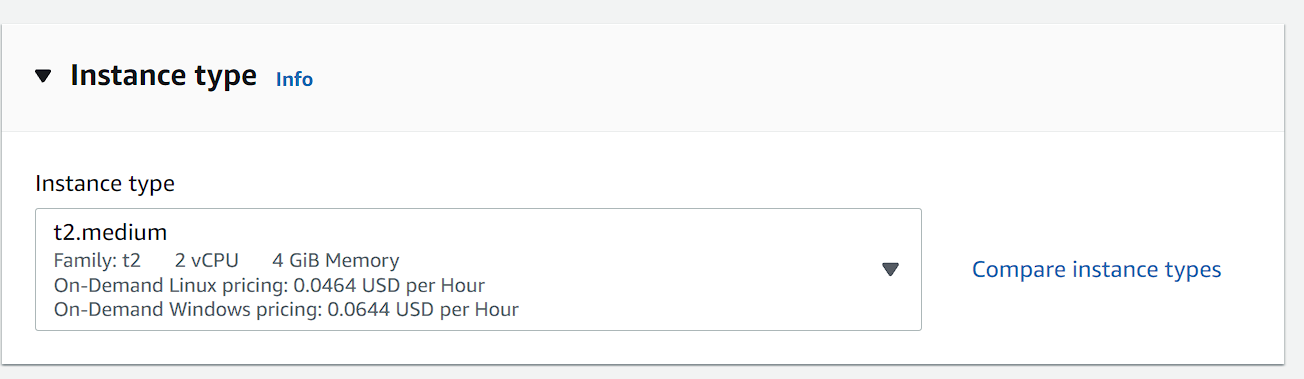
ไปที่ Launch instances

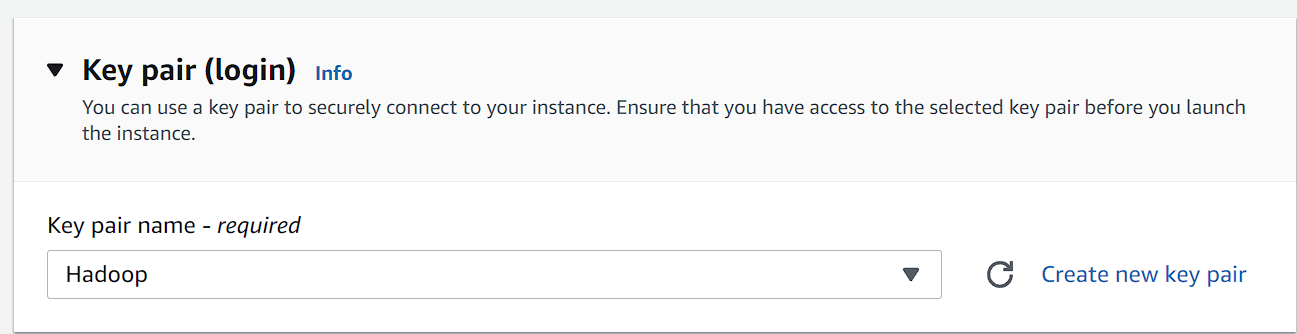


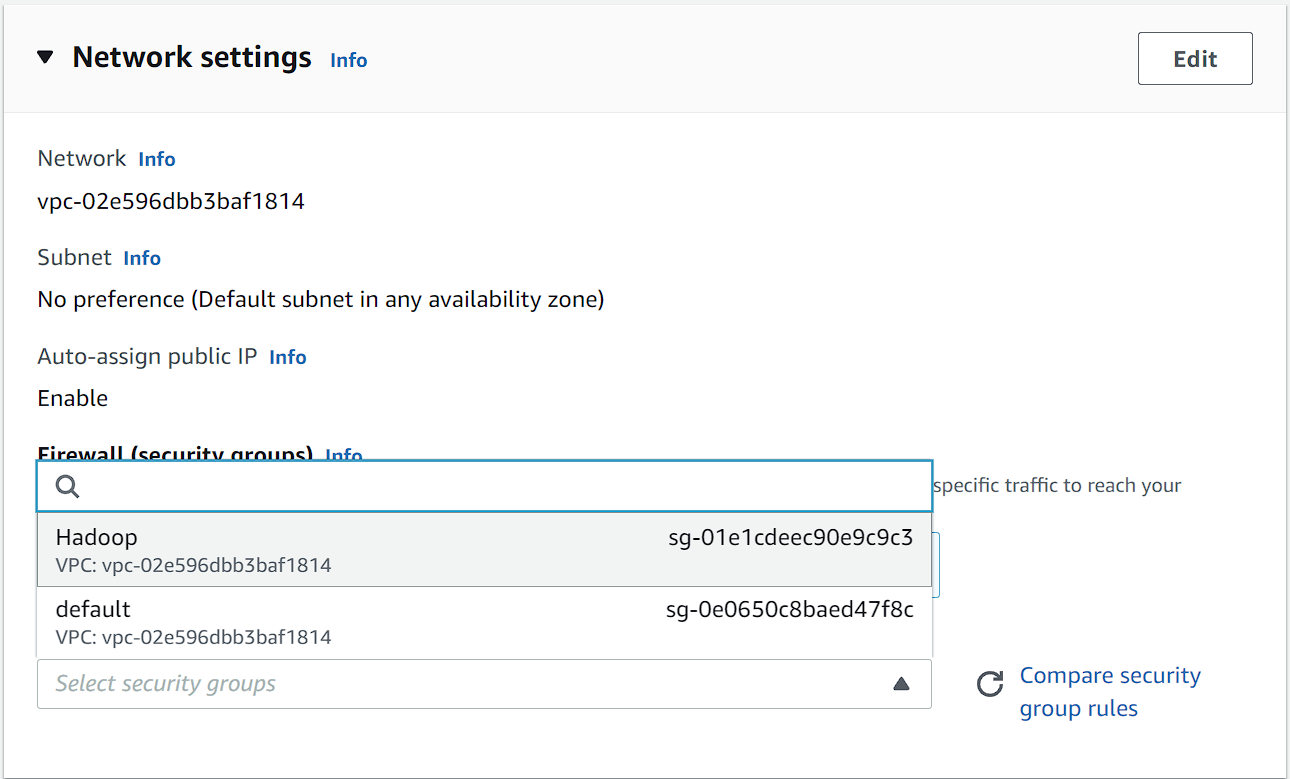
เลือก image ที่สร้างไว้



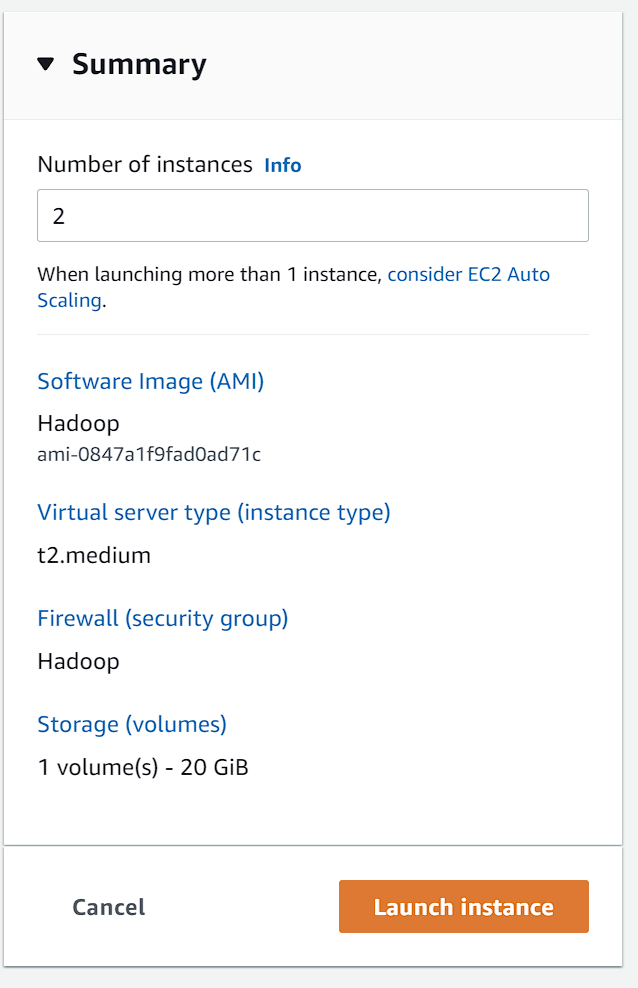
ตั้งค่าต่างๆ







Launch 2 instances



ที่ทุกเครื่อง

sudo rm ~/.ssh/known\_hosts

แก้ fingerprint error ในเครื่องที่สร้างใหม่ตอน start-dfs.sh, start-yarn.sh

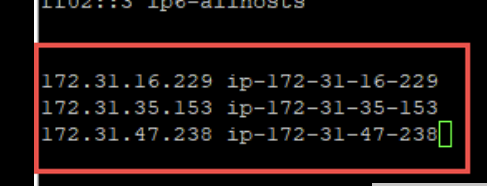
172.31.11.48 ip-172-31-11-48

172.31.0.65 ip-172-31-0-65

172.31.3.214 ip-172-31-3-214

กำหนด hostname ของ slave และ master

sudo nano /etc/hosts



ใส่ private ip ของทุกเครื่อง กับ hostname

Tips: hostname จริงๆ ตั้งได้อิสระ จะเป็น 172.31.16.229 slave ก็ได้ ทีนี้เวลาจะเอาไปใส่ใน /usr/local/hadoop/etc/hadoop/slaves ก็จะใช้ slave แทน ip-172-31-16-229

ก็อปคีย์ของ master ไปให้ slave ทุกเครื่อง

scp /home/ubuntu/.ssh/id\_rsa.pub <slave hostname>:/home/ubuntu/.ssh/master.pub

รันที่ master ตามจำนวน slave ที่มี

กำหนดให้คีย์ที่ก็อปมาเป็นคีย์ที่ไว้ใจได้

cat /home/ubuntu/.ssh/master.pub >> /home/ubuntu/.ssh/authorized\_keys

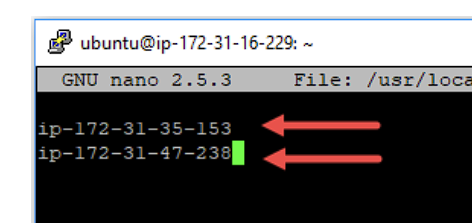
รันที่ slave ทุกเครื่อง

บอก master ว่า slave คือเครื่องไหนบ้าง

nano /usr/local/hadoop/etc/hadoop/slaves

รันที่ master

แล้วใส่ hostname ของ slave ทุกตัวลงไป

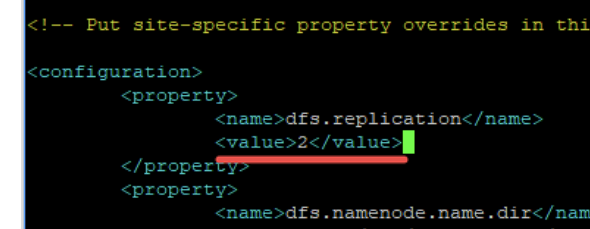


แก้ replication ของการเก็บไฟล์ใน hdfs

nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml

รันที่ master

แก้ dfs.replication เป็น 2



ล้างข้อมูลใหม่หมด

rm -rf /var/hadoop\_data/namenode/\* && rm -rf /var/hadoop\_data/datanode/\*

รันทุกเครื่อง

ล้าง namenode

hdfs namenode -format

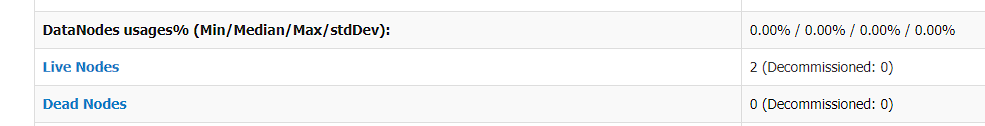
รันที่ master

เปิดระบบ

start-dfs.sh && start-yarn.sh

รันที่ master

เปิด http:// 44.211.119.243:50070 จะต้องได้หน้า dashboard และมี live node เป็น 2



สร้าง folder ใน hdfs

hdfs dfs -mkdir /inputs && hdfs dfs -mkdir /outputs

เครื่องไหนก็ได้

เอา data เข้า hdfs

hdfs dfs -copyFromLocal ./input\_data.txt /inputs/input\_data.txt

เครื่องไหนก็ได้

input\_data.txt จะไป wget มาจากไหนก็ได้ แล้วแต่งาน เช่น

* wget https://www.gutenberg.org/files/1342/1342-0.txt
* mv 1342-0.txt input\_data.txt

# Map-Reduce

เอา Code มาใส่เครื่อง master

จะสร้างไฟล์ใหม่บน server เลยก็ได้ แต่ถ้า code อยู่ใน notebook ต้องก็อปไปใส่ master ก่อน

ใน window terminal/cmd ใช้คำสั่ง

scp -i "<ที่อยู่ของ key ที่ได้จาก aws>" -r <ชื่อ folder ที่จะย้าย>ubuntu@<public ip ของ master>:/home/ubuntu/<ชื่อ folder ที่ปลายทาง>

เช่น

scp -i "../../.ssh/Hadoop.pem" -r Hadoop-MapReduce-master ubuntu@23.212.53.248:/home/ubuntu/Hadoop-MapReduce-master

สร้าง directory สำหรับ class ต่างๆ

mkdir classes

โหลด class

javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d classes WordCount.java

Note: ./\*.java รันjavaทุกตัว (แทนWordCount.java)

สร้าง .jar

jar -cvf ./wordcount.jar -C wordcount\_classes/ .

submit .jar ให้ yarn

yarn jar ./wordcount.jar WordCount /inputs/\* /outputs/wordcount\_output\_dir01

ดู output ทั้งหมดที่มี

hdfs dfs -ls /outputs/wordcount\_output\_dir01

อ่านไฟล์ output

hdfs dfs -cat /outputs/wordcount\_output\_dir01/part-r-00000