

# Hadoop Cheat Sheet

> Region -> Oregon

1. ตั้ง Security group

161.246.0.0/16

The screenshot shows the 'Inbound rules' configuration page in the AWS Management Console. It displays a table of four inbound rules for a security group. Each rule has a 'Security group rule ID', 'Type', 'Protocol', 'Port range', 'Source', 'Description - optional', and a 'Delete' button. The rules are:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Delete
sgr-063c663d8d632e7b6	SSH	TCP	22	Custom	161.246.0.0/16	Delete
sgr-0e72e492134bf4512	All traffic	All	All	Custom	172.31.0.0/16	Delete
sgr-0bf8c0cb4fc83950b	Custom TCP	TCP	8088	Custom	161.246.0.0/16	Delete
sgr-0cbb6d27b1237d9f1	Custom TCP	TCP	50000 - 50099	Custom	161.246.0.0/16	Delete

At the bottom, there is an 'Add rule' button and a 'Cancel' button. On the right, there are 'Preview changes' and 'Save rules' buttons.

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

2. สร้าง instance

The screenshot shows the 'Summary' page of the AWS EC2 instance creation wizard. It displays the following configuration:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 18.04 LTS, ...read more (ami-000340e2c761ddcd)
- Virtual server type (instance type):** t2.medium
- Firewall (security group):** Practice\_Test1\_SG
- Storage (volumes):** 1 volume(s) - 20 GiB

At the bottom, there is a 'Cancel' button and a 'Launch instance' button. A blue box at the bottom left contains a message: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier'.

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

EC2 > Instances > i-022121e6a398c95a2

### Instance summary for i-022121e6a398c95a2 (Hadoop) [Info](#)

Updated less than a minute ago

Instance ID i-022121e6a398c95a2 (Hadoop)	Public IPv4 address 35.90.159.248   <a href="#">open address</a>
IPv6 address -	Instance state <span>Running</span>
Hostname type IP name: ip-172-31-11-105.us-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-11-105.us-west-2.compute.internal
Answer private resource DNS name IPv4 (A)	Instance type t2.medium
Auto-assigned IP address 35.90.159.248 [Public IP]	VPC ID vpc-02e596dbb3baf1814
IAM Role -	Subnet ID subnet-088df4465c6ef4ea2

Details | Security | **Networking** | Storage | Status checks | Monitoring | Tags

[i](#) You can now check network connectivity with Reachability Analyzer.

#### ▼ Networking details [Info](#)

Public IPv4 address 35.90.159.248   <a href="#">open address</a>	Private IPv4 addresses 172.31.11.105
Public IPv4 address 35.90.159.248   <a href="#">open address</a>	Private IP DNS name (IPv4 only) ip-172-31-11-105.us-west-2.compute.internal
Subnet ID subnet-088df4465c6ef4ea2	IPv6 addresses -

## 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 ไม่ได้ 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-2.6.0/hadoop-2.6.0.tar.gz && sudo tar -xvf hadoop-2.6.0.tar.gz && sudo mv  
./hadoop-2.6.0 /usr/local/hadoop
```

Download + แยกไฟล์ + ย้ายไฟล์ที่โหลดมาไว้ที่ /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 ถ้าติดตั้งผิด

- 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 ไม่ได้ก็อาจจะเป็นที่นี้ก็ได้

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 ก็ลองรันซ้ำที แต่ระวังข้อมูลใน hdfs หาย

เปิด

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

ทดสอบ service หลังจาก start-dfs.sh และ start-yarn.sh แล้ว

```
jobs
```

```
ubuntu@ip-172-31-16-229:/usr/local/hadoop/etc/hadoop$ jps
19793 Jps
19762 ResourceManager
19561 SecondaryNameNode
19388 DataNode
19247 NameNode
ubuntu@ip-172-31-16-229:/usr/local/hadoop/etc/hadoop$
```

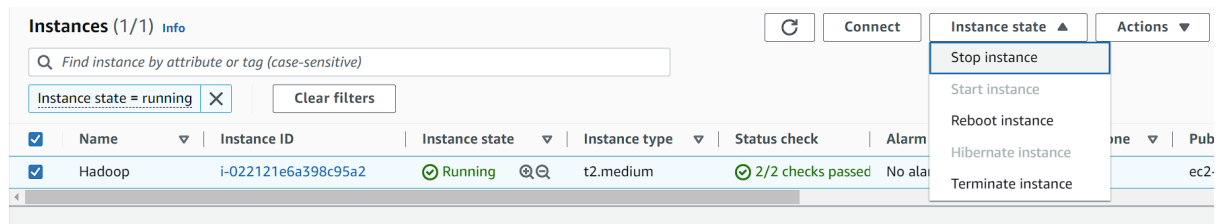
ถ้าเซตถูกหมดจะได้แบบนี้

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

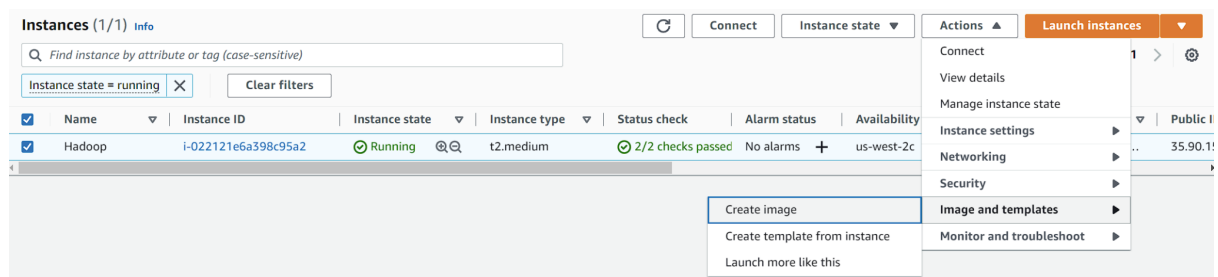
ถ้าเปิดไม่ติด

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

เข้า aws dashboard >> Stop instance รอให้เครื่องปิด



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





EC2 > Instances > i-022121e6a398c95a2 > Create image

### Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID  
i-022121e6a398c95a2 (Hadoop)

Image name  
Hadoop  
Maximum 127 characters. Can't be modified after creation.

Image description - optional  
Image description  
Maximum 255 characters

No reboot  
☐ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	20	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Add new tag  
You can add up to 50 more tags.

Cancel Create image

## Q&A

ตอน sudo apt-get update ขึ้น connection timeout

- ดูที่ security group ว่าตั้ง outbound เป็น all traffic from 0.0.0.0/0 รีปาว 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
- อาจจะเป็นที่ตอนรันครั้งแรกแล้วตก แล้วแก้โดยไม่ format

# Cluster Setup

## ไปที่ Launch instances

Instances (3) Info

Find instance by attribute or tag (case-sensitive)


Instance state = running X Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
<input type="checkbox"/>	Hadoop	i-022121e6a398c95a2	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c	ec2-35-90-159
<input type="checkbox"/>	Hadoop_Slave2	i-0ae92d59b20a1bd70	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c	ec2-54-187-24
<input type="checkbox"/>	Hadoop_Slave	i-0a1beb475745c92d8	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c	ec2-35-89-53-

เลือก image ที่สร้างไว้

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 Search our full catalog including 1000s of application and OS images

Recents

**My AMIs**

Quick Start

☒ Owned  
by me

☐ Shared  
with me



[Browse more AMIs](#)

Including AMIs from  
AWS, Marketplace and  
the Community

Amazon Machine Image (AMI)

Hadoop

ami-0847a1f9fad0ad71c

2022-11-23T15:44:25.000Z

Virtualization: hvm

ENA enabled: true

Root device type: ebs



Description

-

Architecture

x86\_64

AMI ID

ami-0847a1f9fad0ad71c

ตั้งค่าต่างๆ

## ▼ Instance type [Info](#)

Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory

On-Demand Linux pricing: 0.0464 USD per Hour

On-Demand Windows pricing: 0.0644 USD per Hour



[Compare instance types](#)

## ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Hadoop



[Create new key pair](#)

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-02e596dbb3baf1814

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

Q

Hadoop

VPC: vpc-02e596dbb3baf1814

sg-01e1cdeec90e9c9c3

default

VPC: vpc-02e596dbb3baf1814

sg-0e0650c8baed47f8c

Select security groups ▲

specific traffic to reach your

Compare security group rules

Launch 2 instances

▼ Summary

Number of instances [Info](#)

2

When launching more than 1 instance, [consider EC2 Auto Scaling](#).

Software Image (AMI)

Hadoop

ami-0847a1f9fad0ad71c

Virtual server type (instance type)

t2.medium

Firewall (security group)

Hadoop

Storage (volumes)

1 volume(s) - 20 GiB

Cancel

Launch instance

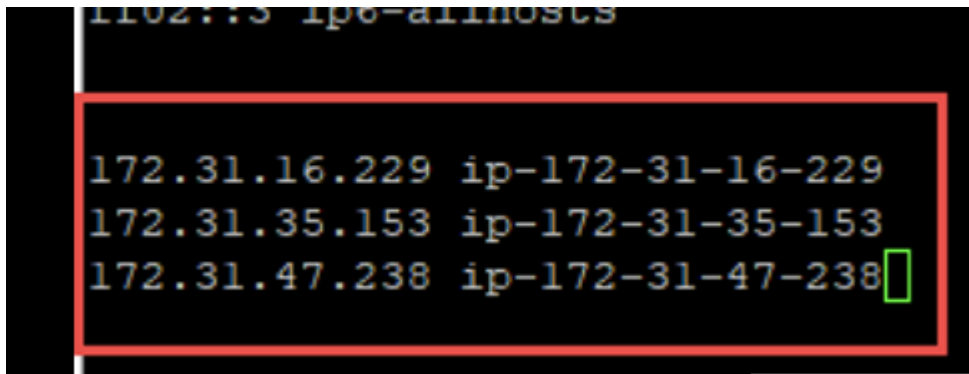
ที่ทุกเครื่อง

```
sudo rm ~/.ssh/known_hosts
```

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

กำหนด hostname ของ slave และ master

```
sudo nano /etc/hosts
```



```
172.31.16.229 ip-172-31-16-229
172.31.35.153 ip-172-31-35-153
172.31.47.238 ip-172-31-47-238
```

ใส่ private ip ของทุกเครื่อง กับ hostname

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

ก๊อปปี้ของ master ไปให้ slave ทุกเครื่อง

```
scp /home/ubuntu/.ssh/id_rsa.pub <slave hostname>:/home/ubuntu/.ssh/master.pub
```

รันที่ master ตามจำนวน slave ที่มี

กำหนดให้คีย์ที่ก๊อปปี้มาเป็นคีย์ที่ไว้วางใจได้

```
cat /home/ubuntu/.ssh/master.pub >> /home/ubuntu/.ssh/authorized_keys
```

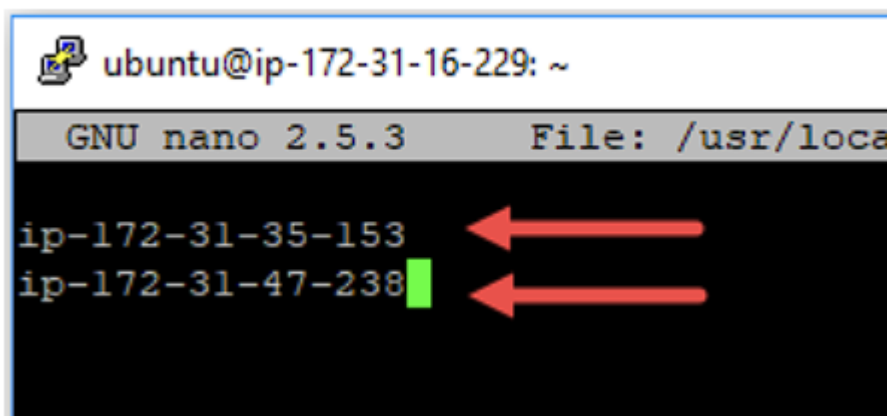
รันที่ slave ทุกเครื่อง

บอก master ว่า slave คือเครื่องไหนบ้าง

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

รันที่ master

แล้วใส่ hostname ของ slave ทุกตัวลงไป



```
ubuntu@ip-172-31-16-229: ~
GNU nano 2.5.3 File: /usr/local/hadoop/etc/hadoop/slaves
ip-172-31-35-153
ip-172-31-47-238
```

แก้ replication ของการเก็บไฟล์ใน hdfs

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

รันที่ master

แก้ dfs.replication เป็น 2

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

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
```

ล้างข้อมูลทั้งหมด

```
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://<public ip ของ master>:50070> จะต้องได้หน้า dashboard และมี live node เป็น 2

DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)

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 แล้วแต่งงาน 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 ต้องก๊อปปี้ไปใส่ 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 รัน java ทุกตัว (แทน WordCount.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-000000
```