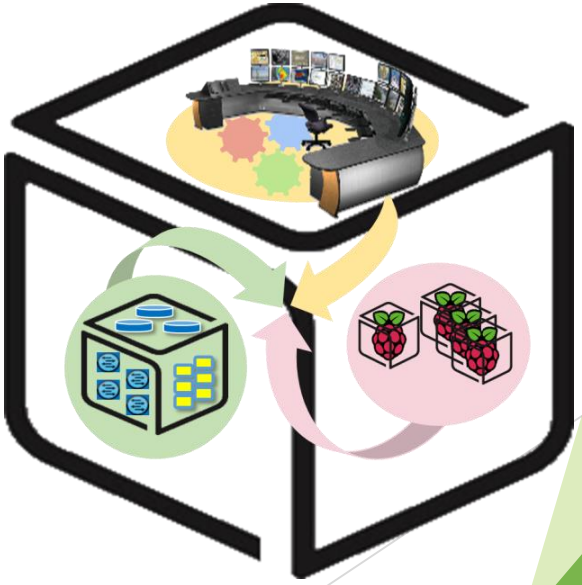
SmartX Labs
for Computer Systems

Cluster Lab
(2016, Spring)



NetCS Lab

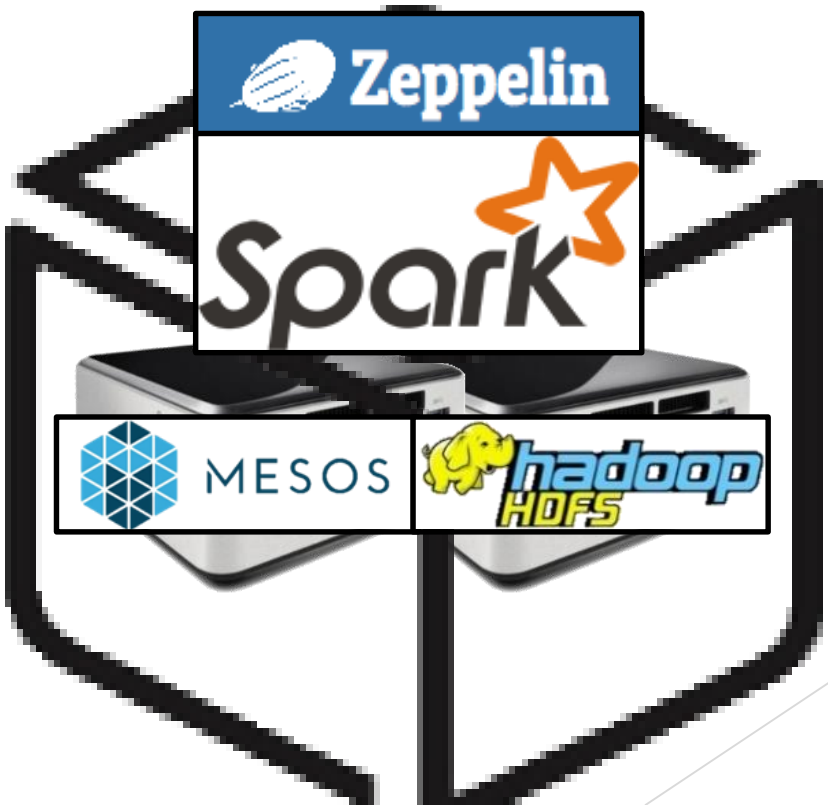
History and Contributor of Cluster Lab (2016. 05. 28.)

Version	Updated Date	Updated Contents	Contributor
-	2015/10	(구) Analytics Lab 작성	송지원
v1	2016/04	Cluster Lab 초안 작성	김승룡
v2	2016/05	Cluster Lab 수정	송지원
v3r3	2016/05/28	Cluster Lab 2차 수정 (내용 수정 및 추가)	송지원

CSLab: Cluster LAB - Goal

SETUP to run data processing and visualization

- Install Mesos, HDFS, Spark, Zeppelin on NUC

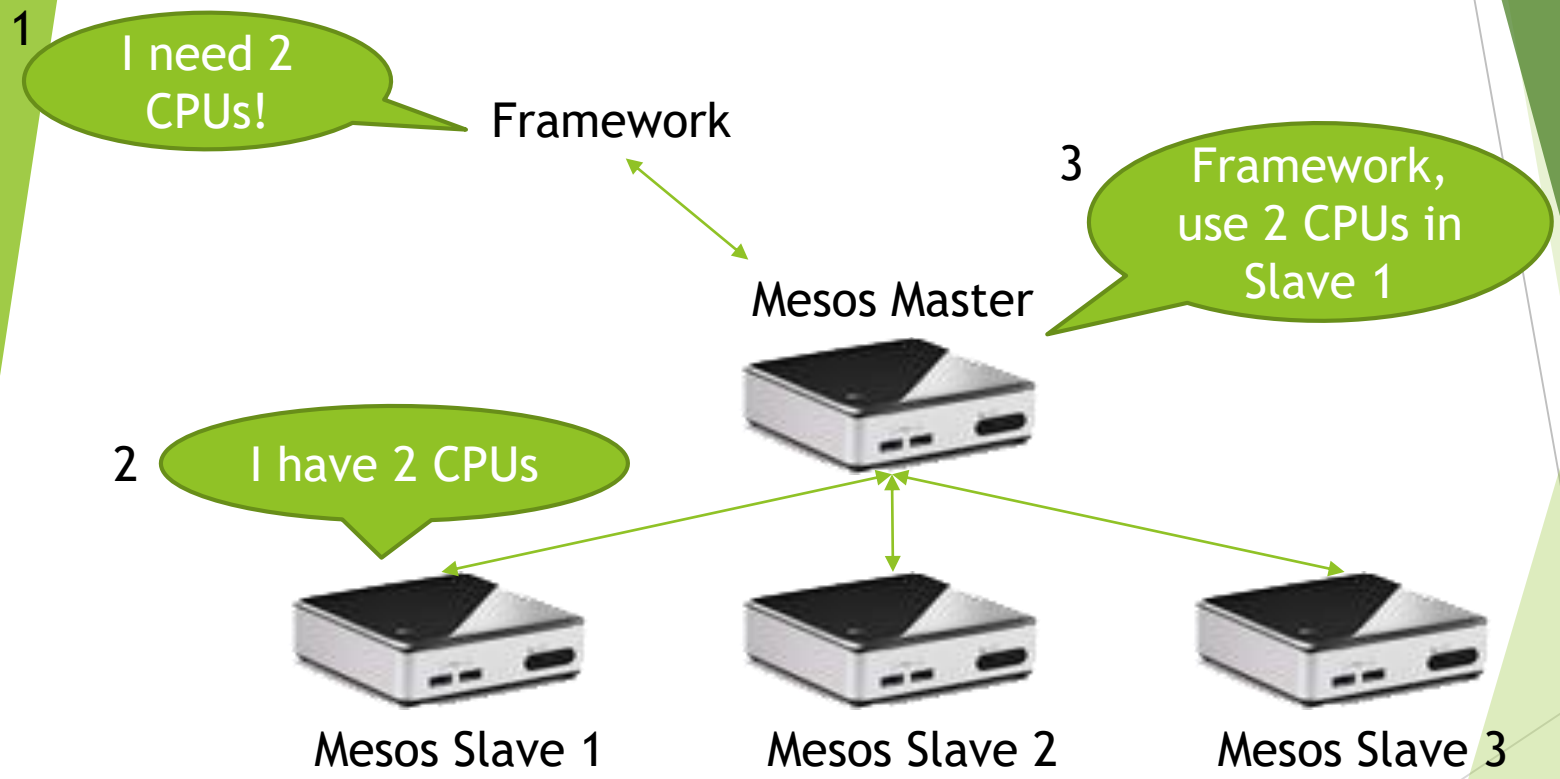


Apache Mesos
- Concept

What is Mesos?
A distributed systems kernel
Mesos is built using the same principles as the Linux kernel, only at a different level of abstraction. The Mesos kernel runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elastic Search) with API's for resource management and scheduling across entire datacenter and cloud environments.

- Cloud as a single computer
- Share resources across the machines

Apache Mesos
- Architecture



HDFS
- Concept

Hadoop Distributed FileSystem

- A distributed file system that provides high-throughput access to application data.

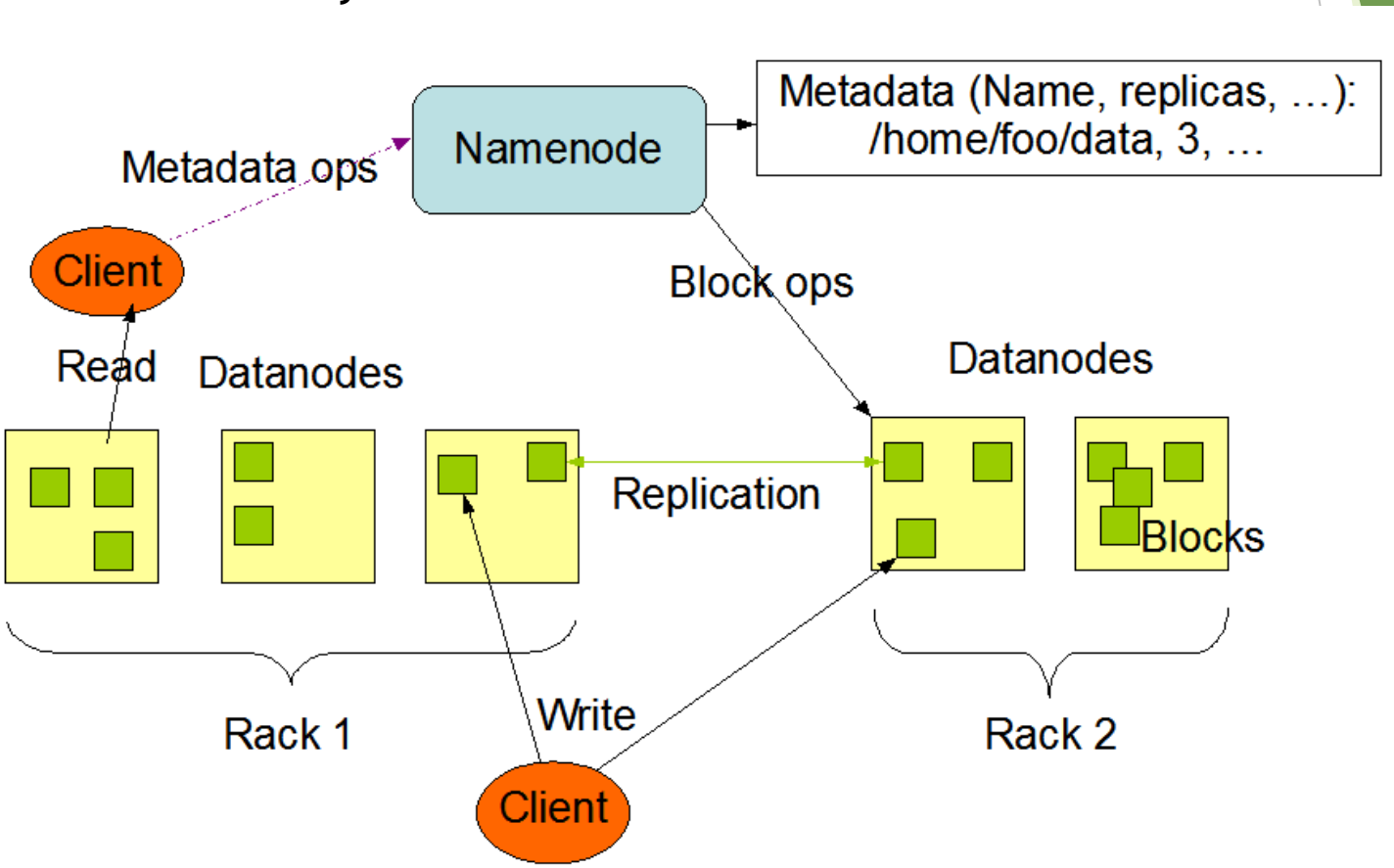
Features

- Fault tolerance by detecting faults and applying quick, automatic recovery
- Portability across heterogeneous commodity hardware and operating systems
- Scalability to reliably store and process large amounts of data
- Economy by distributing data and processing across clusters of commodity personal computers
- Efficiency by distributing data and logic to process it in parallel on nodes where data is located
- Reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures

HDFS
- Architecture

<Master/Slave architecture>

- NameNode: A single node which manages the file system namespace and regulates access to files by clients.
- DataNode: DataNodes manage storage attached to the nodes that they run on.

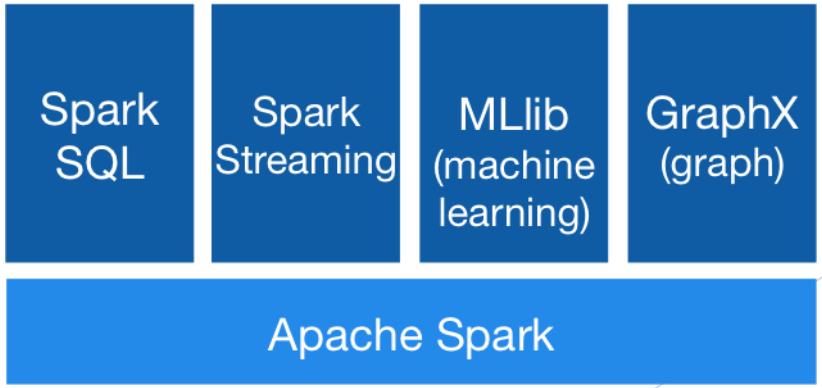


Apache Spark
- Concept

Apache Spark™ is a fast and general engine for large-scale data processing.

- In-memory data processing framework: Fast!
- Easy to use, community fastly growing
- Libraries: SQL and DataFrame, Streaming, MLlib, GraphX
- Run on standalone or Mesos, Yarn, etc

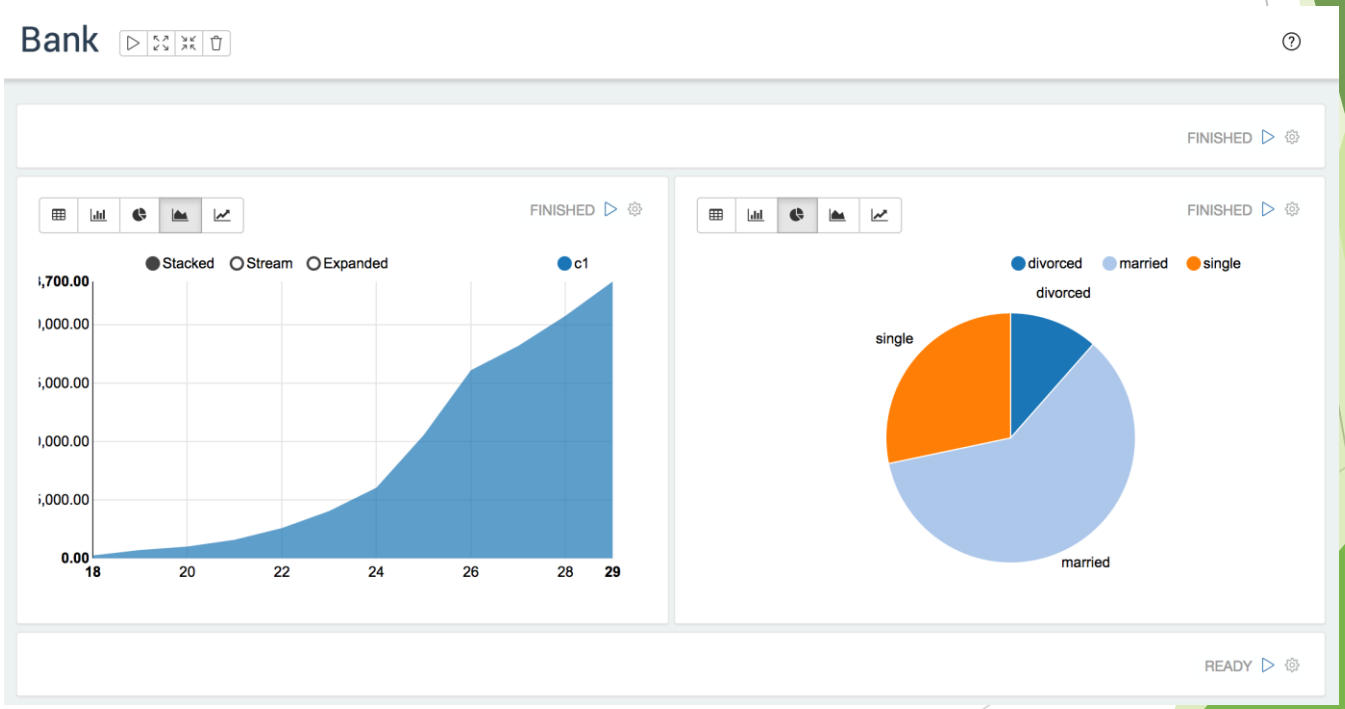
- Scala, Java, Python



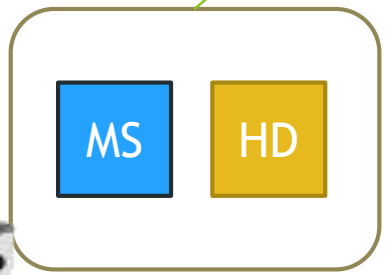
Apache Zeppelin
-Concept

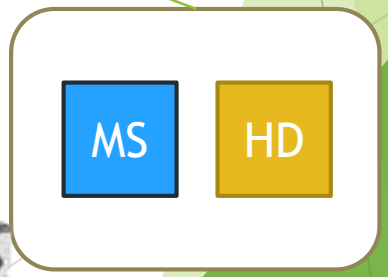
A web-based notebook that enables interactive data analytics.

Support Spark



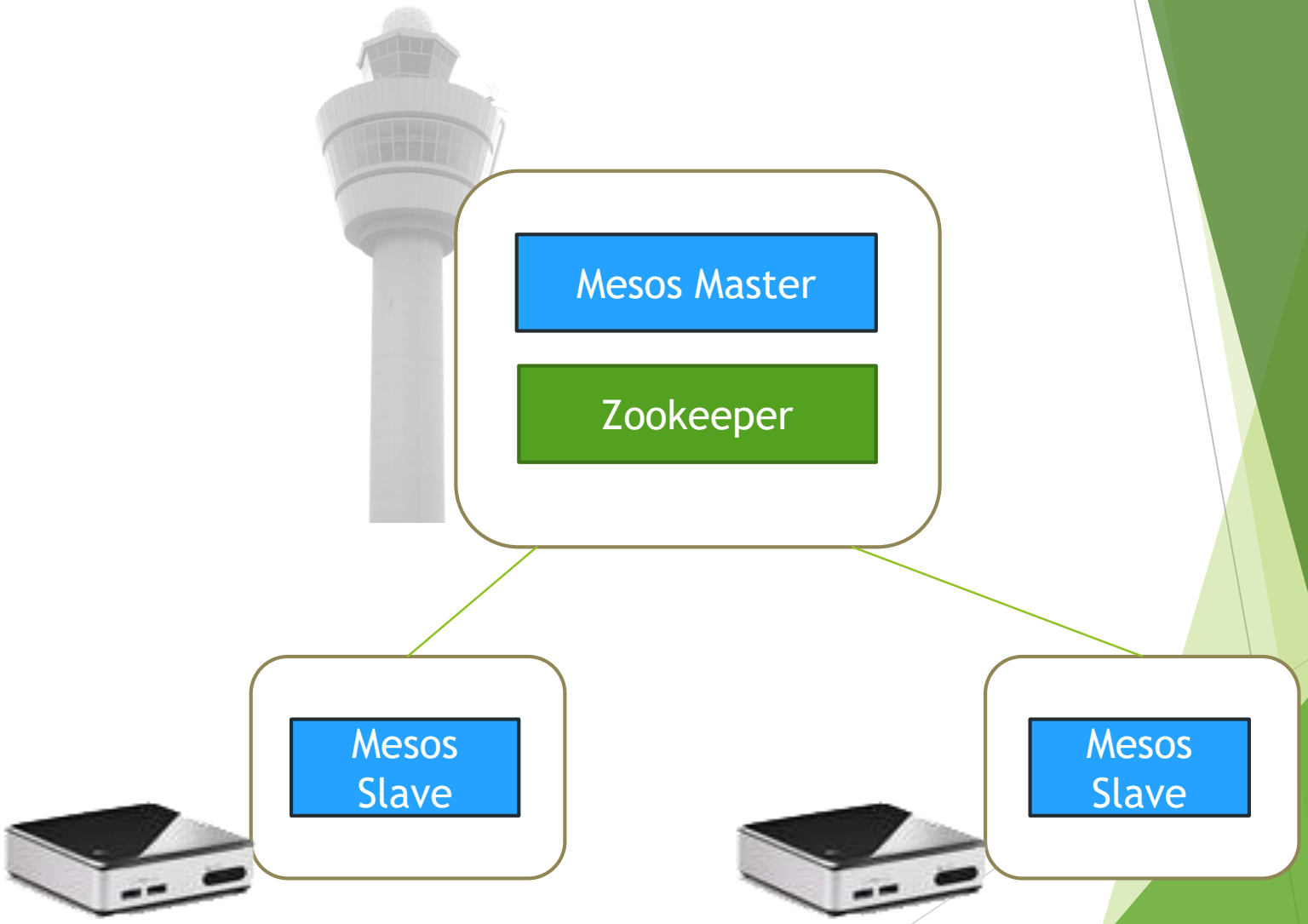
0. Cluster Overview





- MM Mesos Master
- MS Mesos Slave
- HN HDFS NameNode
- HD HDFS DataNode
- Z Zookeeper
- SP Spark
- ZP Zeppelin

1. Apache Mesos
- Install



1. Apache Mesos
- Installation Procedure

Prerequisite: Ubuntu must be 64bit

1. Add Mesosphere repository
2. Install Mesos Master
3. Install Mesos Slave
4. Check on the Web UI

1. Apache Mesos
- Install: Add Mesosphere repository

Add the repository to Tower and NUCs.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv E56151BF

DISTRO=$(lsb_release -is | tr '[:upper:]' '[:lower:]')

CODENAME=$(lsb_release -cs)

echo "deb http://repos.mesosphere.io/${DISTRO} ${CODENAME} main" | sudo
tee /etc/apt/sources.list.d/mesosphere.list

sudo apt-get -y update
```

1. Apache Mesos
- Install: Mesos Master

```
sudo apt-get -y install mesos marathon
sudo reboot

sudo service mesos-slave stop
echo manual | sudo tee /etc/init/mesos-slave.override
echo <TOWER_IP_ADDR> | sudo tee /etc/mesos-master/ip
echo <TOWER_IP_ADDR> | sudo tee /etc/mesos-master/hostname
echo zk://<TOWER_IP_ADDR>:2181/mesos | sudo tee /etc/mesos/zk
echo <YOUR_NAME> | sudo tee /etc/mesos-master/cluster
sudo service zookeeper restart
sudo service mesos-master restart
sudo service marathon restart

echo 1 | sudo tee /etc/zookeeper/conf/myid
```

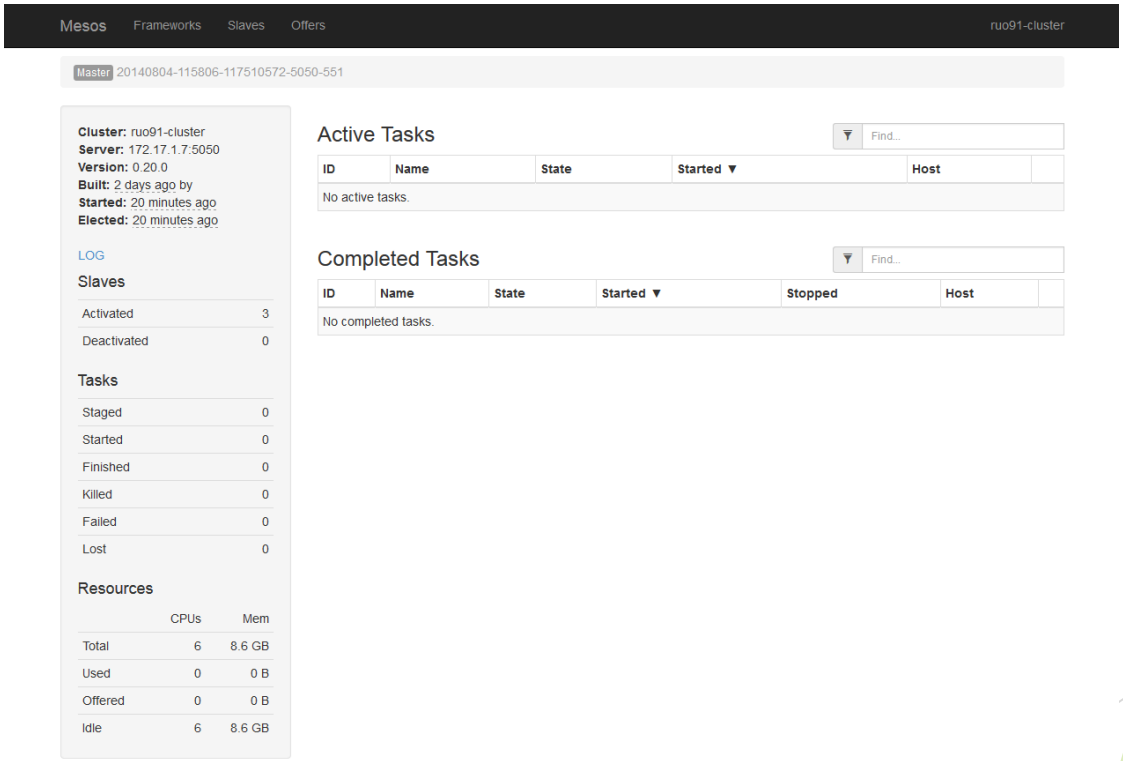
1. Apache Mesos
- Install: Mesos Slave

```
sudo apt-get -y install mesos
sudo reboot

sudo service mesos-master stop
echo manual | sudo tee /etc/init/mesos-master.override
sudo service zookeeper stop
echo manual | sudo tee /etc/init/zookeeper.override
sudo apt-get -y remove --purge zookeeper

echo <NUC_IP_ADDR> | sudo tee /etc/mesos-slave/ip

echo <NUC_IP_ADDR> | sudo tee /etc/mesos-slave/hostname

echo zk://<MASTER_IP_ADDR>:2181/mesos | sudo tee /etc/mesos/zk
sudo reboot
```

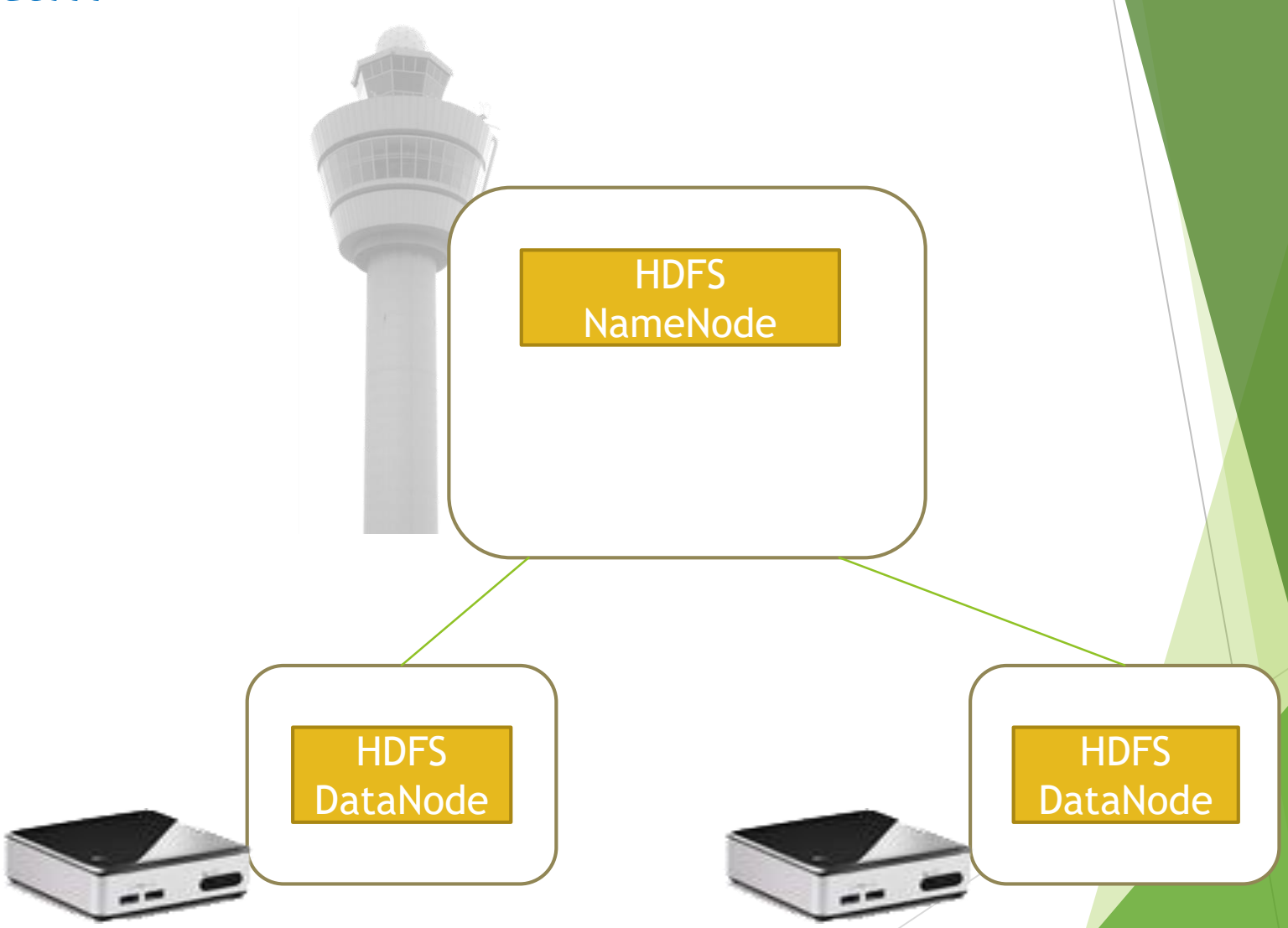
We installed Mesos Master on Tower, so
Tower IP address will be Master IP address.

1. Apache Mesos
- Check on the Web UI

In your web browser, go to
`http://<MASTER-IP-ADDR>:5050`



1. HDFS
- Install



2. HDFS
- Installation Procedure

1. Set hostnames
2. Configure accounts and SSH settings
3. Download and Unzip Hadoop
4. Configure HDFS
5. Start and test

2. HDFS
- Set Hostname

1. Registration host names

```
vi /etc/hosts
```

Edit: Type IP addresses and hostnames of all nodes.

Ex)
tower 192.168.0.1
nuc1 192.168.0.2
nuc2 192.168.0.3

Do this for all tower and NUCs.

2. HDFS
- Configure accounts and SSH settings

1. Set root password
   - `sudo passwd`

2. Create Hadoop account and login
   - `sudo -s`
   - `adduser hadoop`
   - `adduser hadoop sudo`
   - `su hadoop`

Do this for all tower and NUCs.

2. HDFS

- Configure accounts and SSH settings

3. Generate key (just press enter x 3) in tower and NUCs
   - `ssh-keygen -t rsa`

4. Modify key permission
   - `cd .ssh`
   - `chmod 700 ./`
   - `chmod 755 ../`
   - `chmod 600 id_rsa`
   - `chmod 644 id_rsa.pub`
   - `chmod 644 authorized_keys`
   - `chmod 644 known_hosts`

5. Copy key from Tower to NUCs
   - `scp id_rsa.pub hadoop@nuc_hostname:id_rsa.pub`

6. Registration (for each NUC)
   - `cat ~/id_rsa.pub >> ~/.ssh/authorized_keys`

7. Login via SSH to check if you can login to NUC without password.

2. HDFS
- Download and Unzip Hadoop

1. Download and Unzip
   - `wget http://mirror.apache-kr.org/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz`
   - `tar -xvzf hadoop-2.7.2.tar.gz`
   - `mv hadoop-2.7.2 hadoop`

2. HDFS
- Configuration

1. Go to the directory which contains configuration files.
   - cd hadoop/etc/hadoop
   - hadoop-env.sh, core-site.xml, hdfs-site.xml

1. <Hadoop-env.sh>

Edit: `export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64`

2. <core-site.xml>

Edit:
```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://hostname:9000/</value>
    </property>
</configuration>
```

2. HDFS
- Configuration

4. <hdfs-site.xml>

Edit:
```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file://home/hadoop/hadoop/namenode</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file://home/hadoop/hadoop/datanode</value>
    </property>
</configuration>
```

5. Deploy configuration files to NUCs.

```
cd ..
scp -r hadoop hadoop@hostname:
```

2. HDFS
- Start and Test

1. Format NameNode.

```
bin/hdfs namenode -format
```

2. Start HDFS.

```
sbin/start-dfs.sh
```

3. Make a directory and upload a file to HDFS to check if it is working.

```
bin/hadoop fs -mkdir /user
bin/hadoop fs -put any_file /user/
bin/hadoop fs -ls /user/
```

You can also see on the web:
http://<NameNode_IP>:50070

3. Apache Spark
- Install

```
wget http://mirror.apache-kr.org/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz
hadoop fs –put /user/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz

tar xzf spark-1.6.1-bin-hadoop2.6.tgz

cd spark-1.6.1-bin-hadoop2.6/conf
cp spark-env.sh.template spark-env.sh
vi spark-env.sh
```

Edit:
```
export MESOS_NATIVE_JAVA_LIBRARY=/usr/local/lib/libmesos.so
export MASTER=mesos://<MASTER_IP_ADDR>:5050
export SPARK_EXECUTOR_URI=hdfs://<TOWER_IP_ADDR>/user/spark-1.6.1-bin-hadoop2.6.tgz
```

Test Spark
```
cd ..
bin/pyspark

data = range(1, 10001)
distData = sc.parallelize(data)
distData.filter(lambda x: x < 10).collect()
```

Go to Mesos web UI and see Spark framework running.

4. Apache Zeppelin
- Install (on Mesos)

```
wget http://mirror.apache-
kr.org/incubator/zeppelin/0.6.0-incubating/zeppelin-
0.6.0-incubating-bin-all.tgz

tar xzf zeppelin-0.6.0-incubating-bin-all.tgz

cd zeppelin-0.6.0-incubating-bin-all/conf
cp zeppelin-env.sh.template zeppelin-env.sh
vi zeppelin-env.sh
```

Edit:

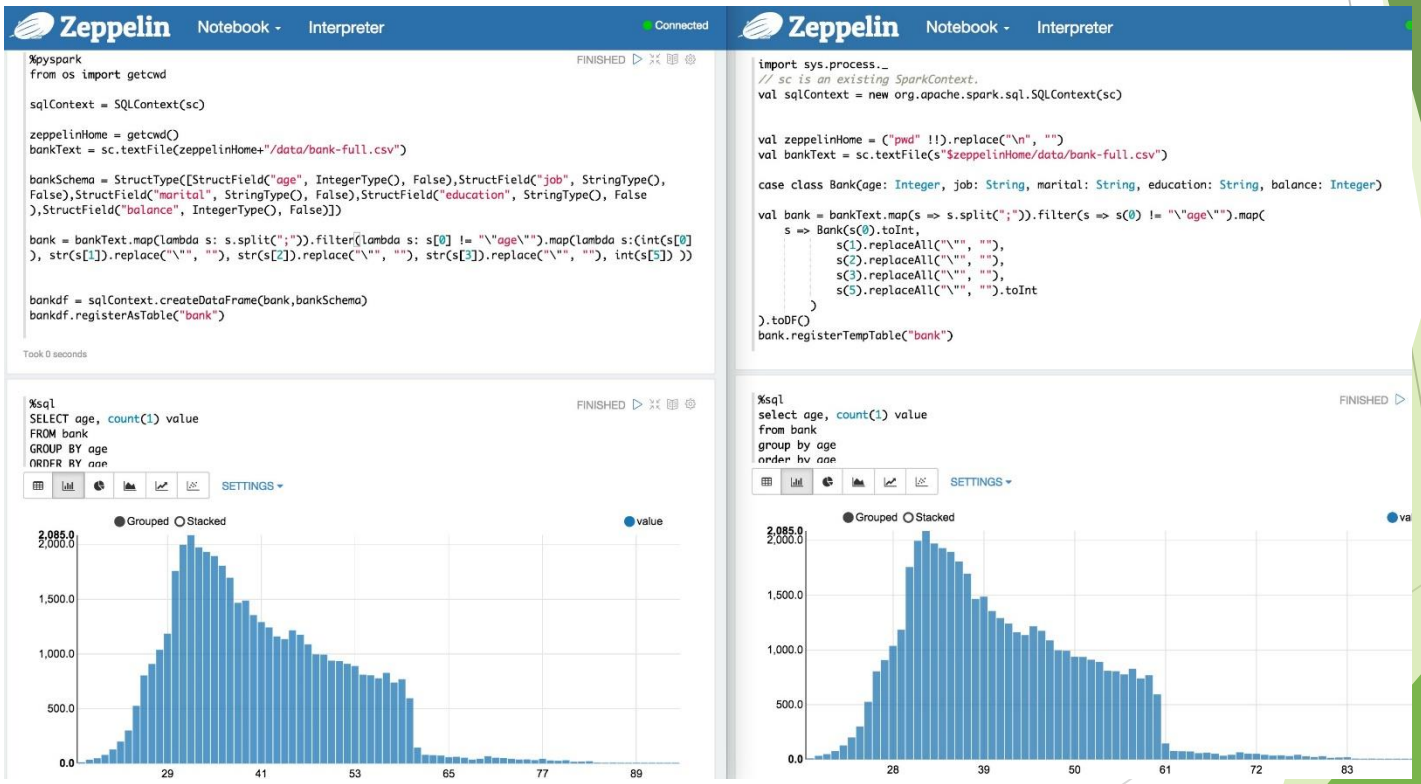
```
export MESOS_NATIVE_JAVA_LIBRARY=/usr/local/lib/libmesos.so
export MASTER=mesos://<MASTER_IP_ADDR>:5050
export SPARK_EXECUTOR_URI=hdfs://<HDFS_IP_ADDR>/user/spark-
1.6.1-bin-hadoop2.6.tgz
```

```
cd ..
bin/zeppelin-daemon.sh start
```

http://<Tower IP-ADDR>:8080

4. Apache Zeppelin
- Run Example

Press ‘Run’ button to test the sample codes.



4. Apache Zeppelin
- Tip: Zeppelin Standalone mode

If you have trouble running Zeppelin on Mesos, you can run Zeppelin in standalone mode.

```
rm conf/zeppelin-env.sh
bin/zeppelin-daemon.sh start
#(or if daemon is already running, use ‘restart’ instead of ‘start.’)
```

http://<IP-ADDR>:8080

Thank You for Your Attention Any Questions?