

Setup Yahoo S4

On Virtual Machine with Ubuntu Linux

Li Huang

Computer Engineering and Computer Science
J.B. Speed School of Engineering
University of Louisville
l0huan08@gmail.com

2014/1/25

Statement:

This article only reflects the author's personal opinion, and may contain some views not right. If you find something wrong in this article or have advices please contact the author. This article is for non-commercial purpose, which helps the people who are interested to SAMOA. Some figures are referenced from some other sources.

Contents

1.Setup Environment	2
2.Setup Virtual Machine	2
3.Setup Ubuntu Linux.....	5
4.Setup Services on Ubuntu	10
4.1 Setup file share service (FTP).	10
4.Setup Yahoo! S4	13
4.1 Setup JDK 1.7.....	13
4.2 Setup Gradle.....	13
4.3 Setup Git.....	14
4.3 Setup S4.....	15
4.4 Setup Zookeeper Client	17
4.5 Test S4	18
4.5.1 HelloApp.....	18
4.5.2 Twitter Trend App.....	21
4.6 Congratulations	24
5. Recommend Readings.....	24

SAMOA(Scalable Advanced Massive Online Analysis) is a new distributed streaming data mining

platform which is now being developing by Yahoo, and S4 is one of the support platform for running SAMOA(another is Twitter Storm). Since currently there is very few material can be find to introducing the Setup process of SAMOA and S4, in this article I will introduce my setup process of Yahoo! S4 platform (now S4 is also an Apache incubator project) in detail. I will continue to write how to setup SAMOA in another article.

1.Setup Environment

I use my daily-use laptop to setup SAMOA and S4. Its configuration is below:

Hardware:

Computer: Lenovo Laptop U310

CPU: Intel Core i3-2367M, with 2 cores at 1.40Ghz

Memory(RAM): 2.00 GB

System Type: 64 bit

Software:

Base operating system: Windows 7 Professional Service Pack 1 - 64 bit

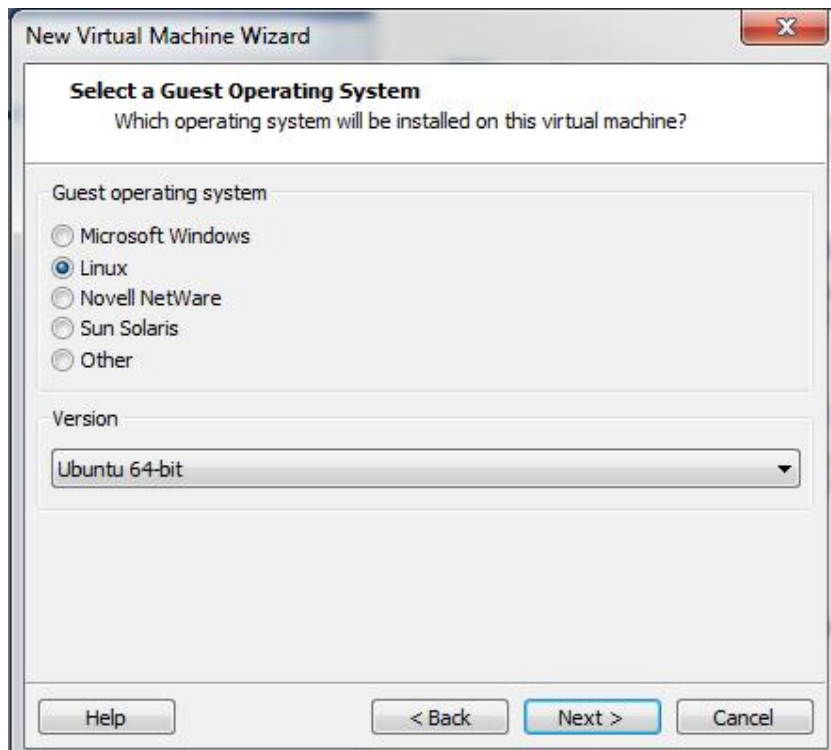
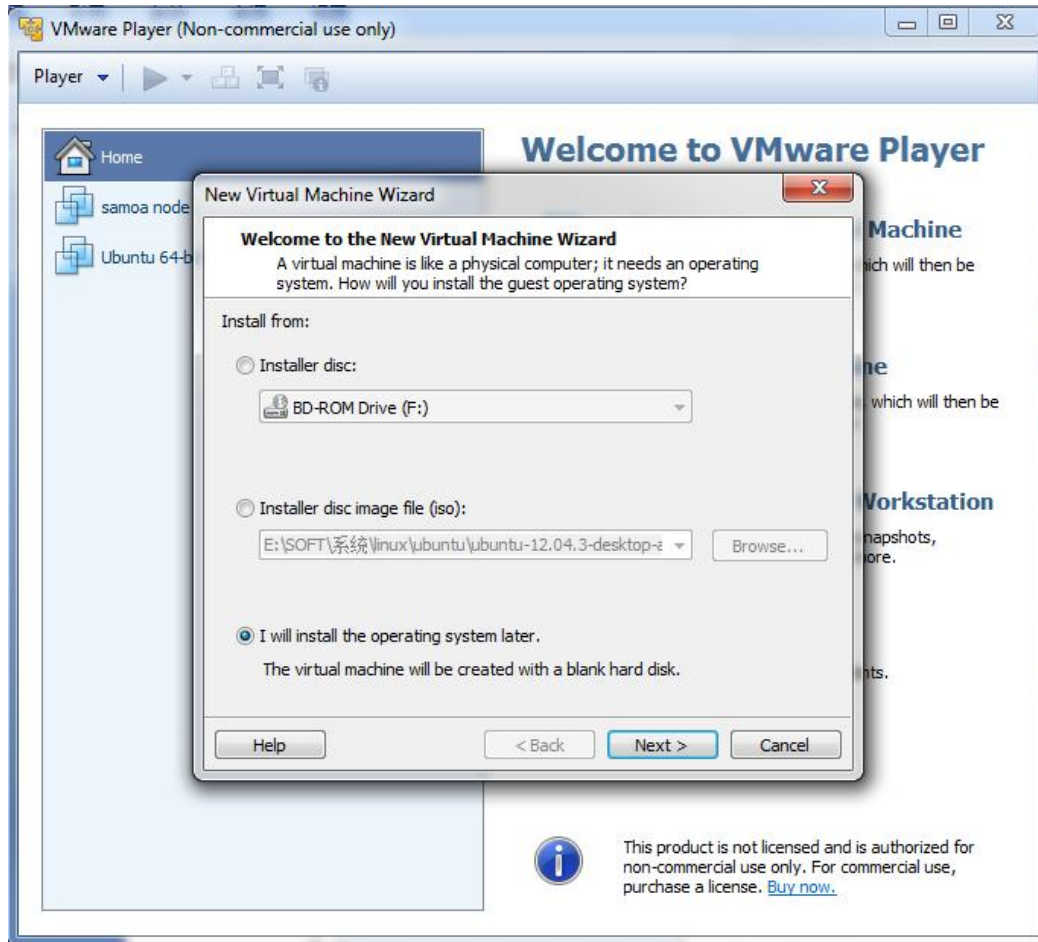
Virtual Machine to host S4: VMware Player 5.0.2

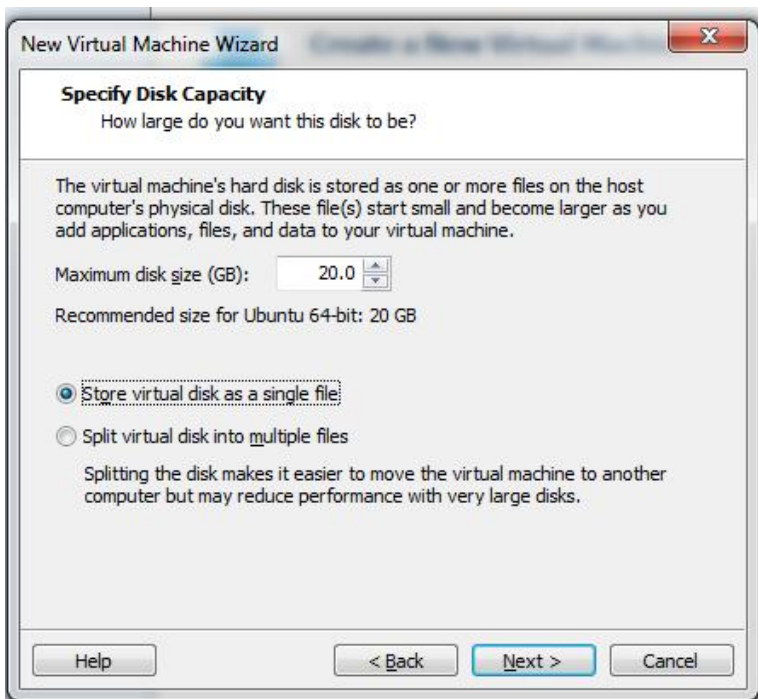
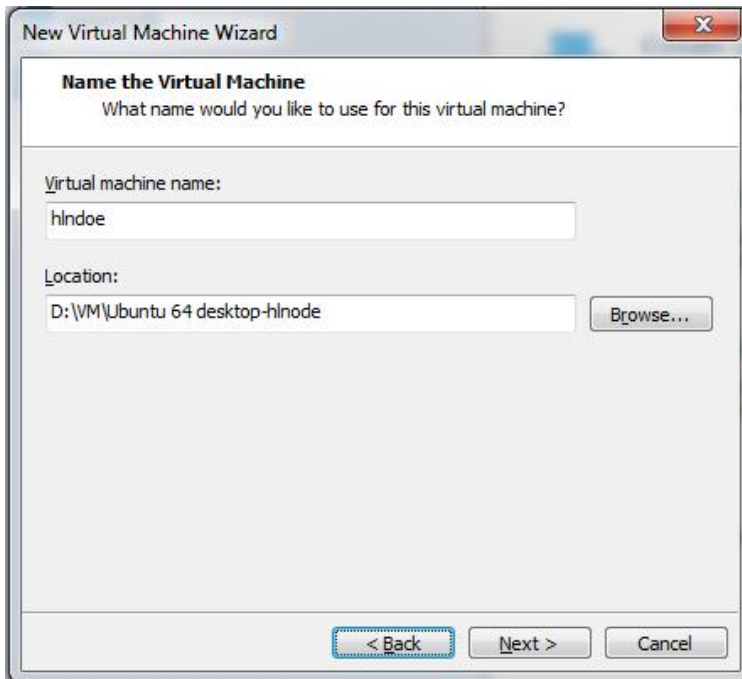
Operating system on Virtual Machine: Ubuntu Linux Desktop 12.04 – 64 bit

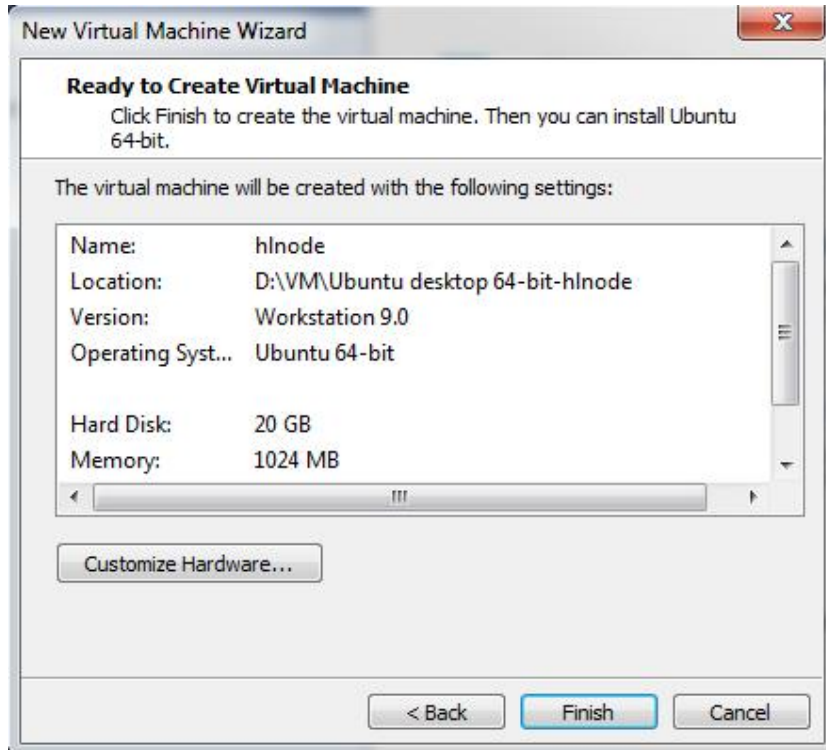
2.Setup Virtual Machine

First I setup a virtual machine on Windows 7.

- (1) Setup VMware Player 5.0.2. I also tried VMware player 10 but it is slower than VMware 5.
- (2) Download Ubuntu 12.04 Desktop. Download it from the official website, and it should be a virtual CD image file “ubuntu-12.04.3-desktop-amd64.iso”.
- (3) Configure VM. Then run the VMware Player, create a new virtual machine, and select “I will install the operating system later”. Then select your system type. Then you can give a Maximum Disk Size for this virtual machine: I recommend 20GB, because Ubuntu system takes at least 5GB and you can keep 10GB free space for your applications and data. To get better performance, I recommend select “Store virtual disk as a single file”. At last, you can custom your virtual machine hardware, such as memory. The most important things are: You should assign more than 512MB memory (If use 512MB it will be very slow, so for faster running speed I recommend 1024 MB), and set the Network Connection Mode as “NAT”.





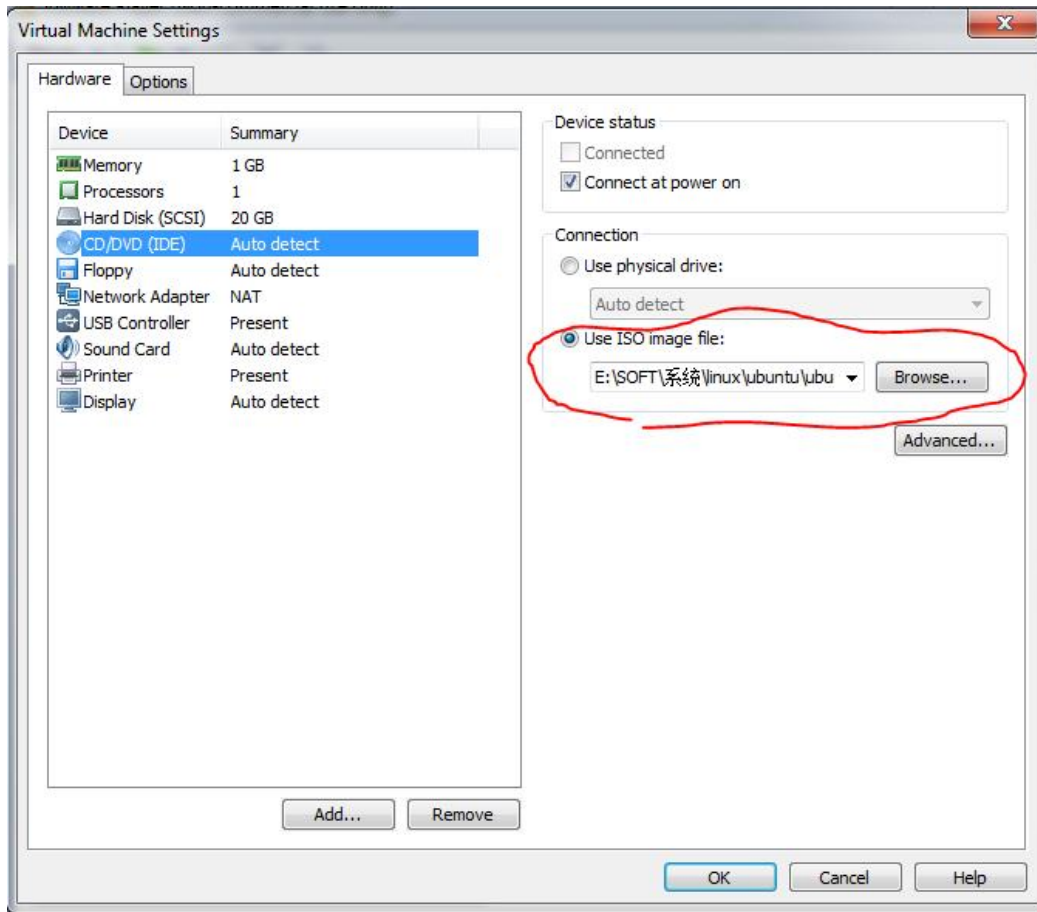


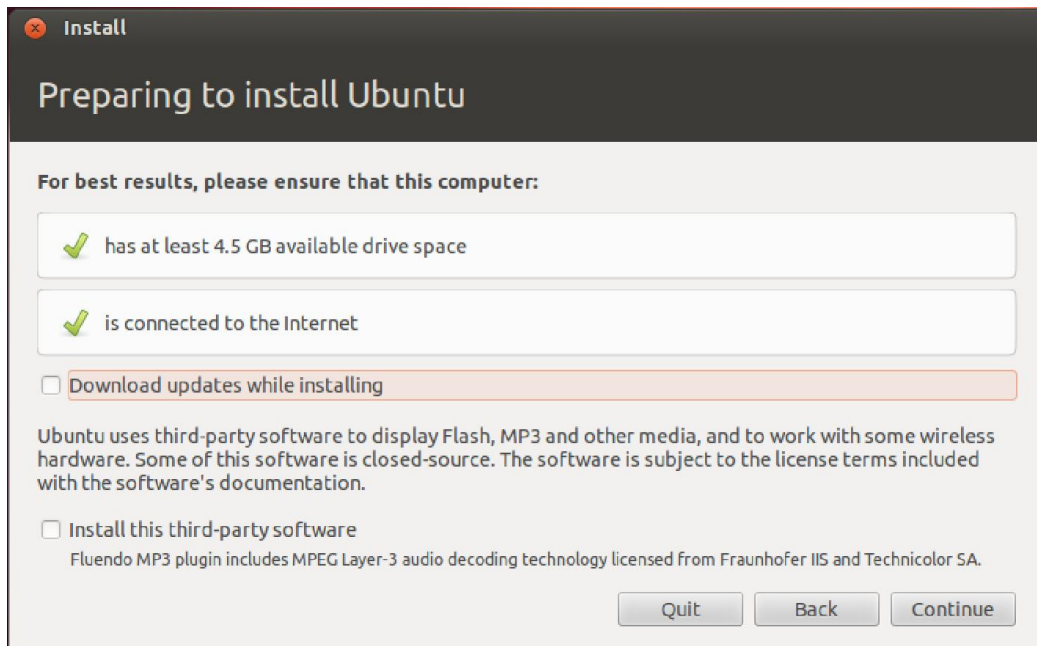
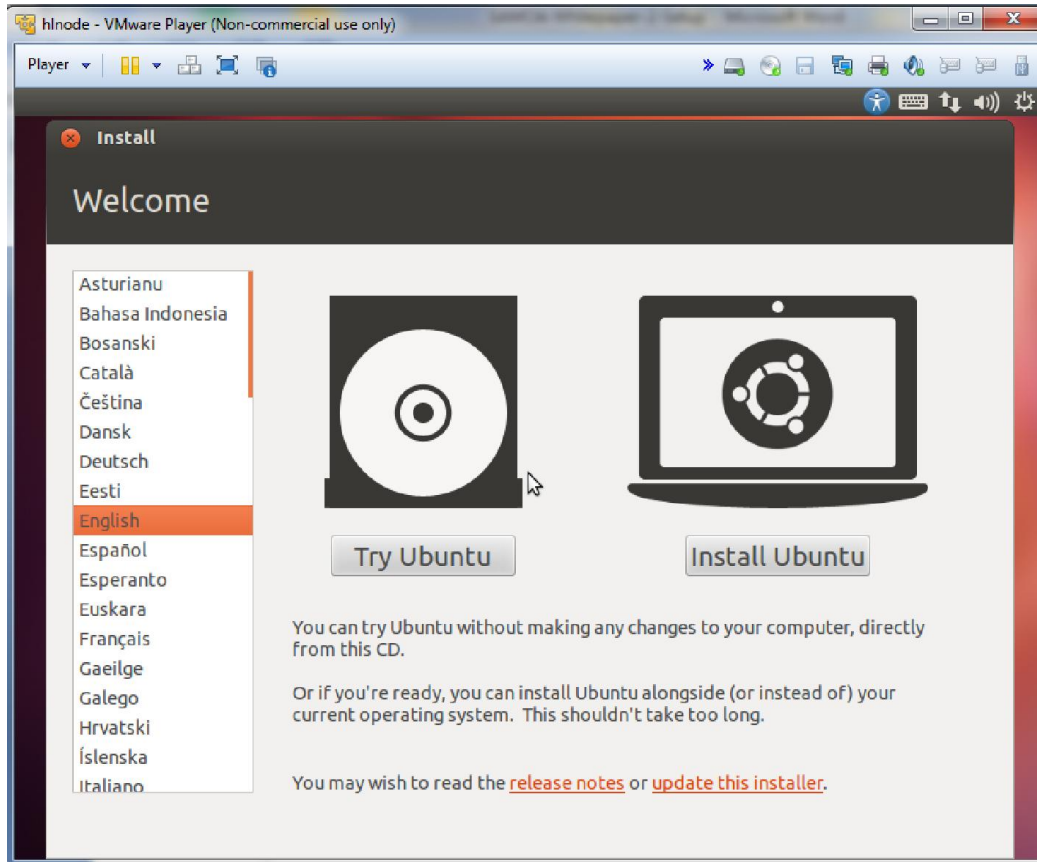
3.Setup Ubuntu Linux

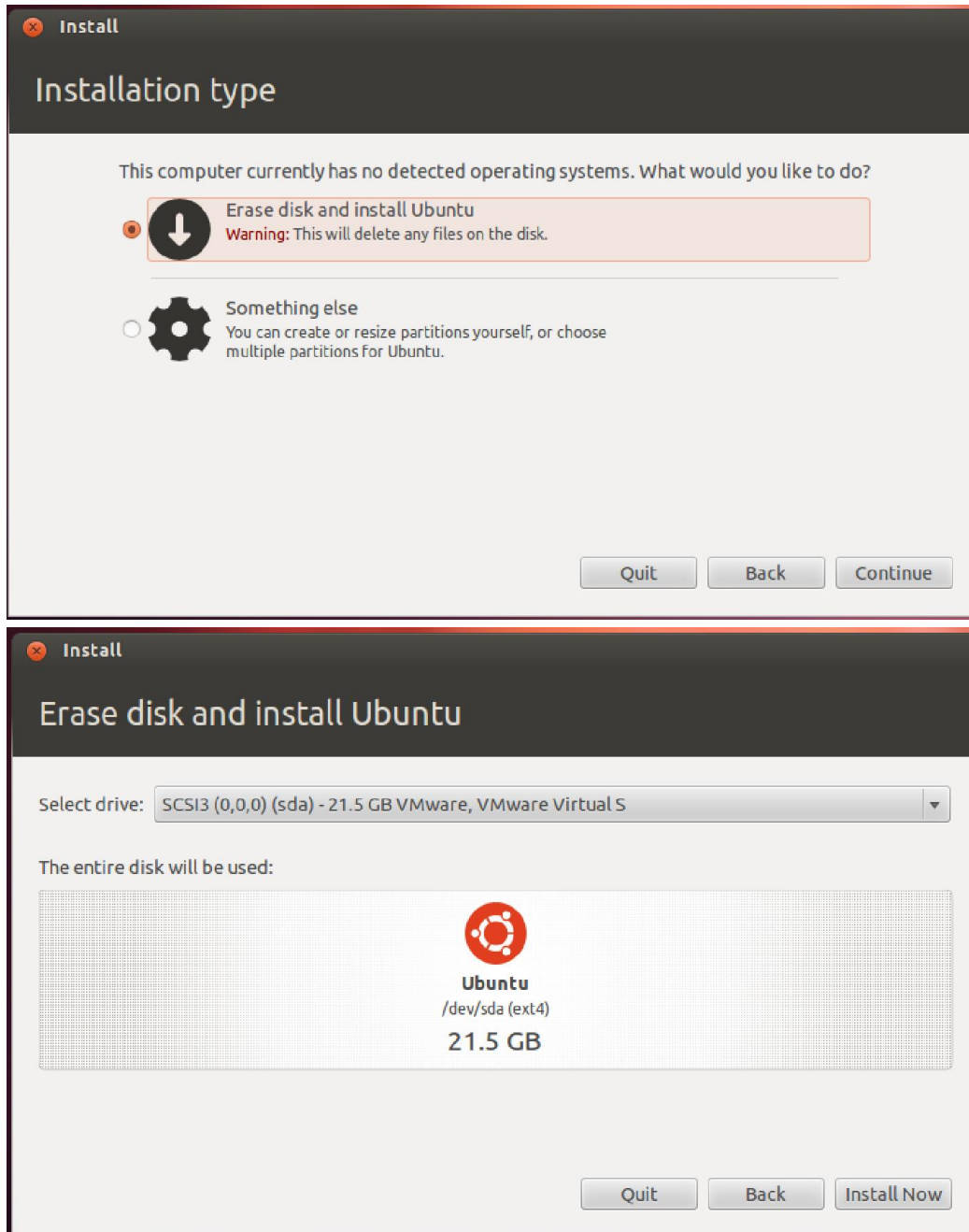
After the initial configuration and start this Virtual Machine you just created. Now you need to setup Ubuntu.

- (1) Configure CD-rom. In VMplayer, choose your virtual machine node, and select “edit virtual machine settings”. In the “Hardware” tab, select CD/DVD(IDE), choose “Use ISO Image file” and select the Ubuntu system’s iso file.
- (2) Start virtual machine. Now you can see the picture of Ubuntu Setup. Choose “English” and click the “Install Ubuntu” button. Then you can use the default settings and just click “continue”.
- (3) Setup username and machine name. There is a step called “Who are you”, in this step you need to specify your computer name and username.

(1)







Install

Who are you?

Your name: ✓

Your computer's name: ✓
The name it uses when it talks to other computers.

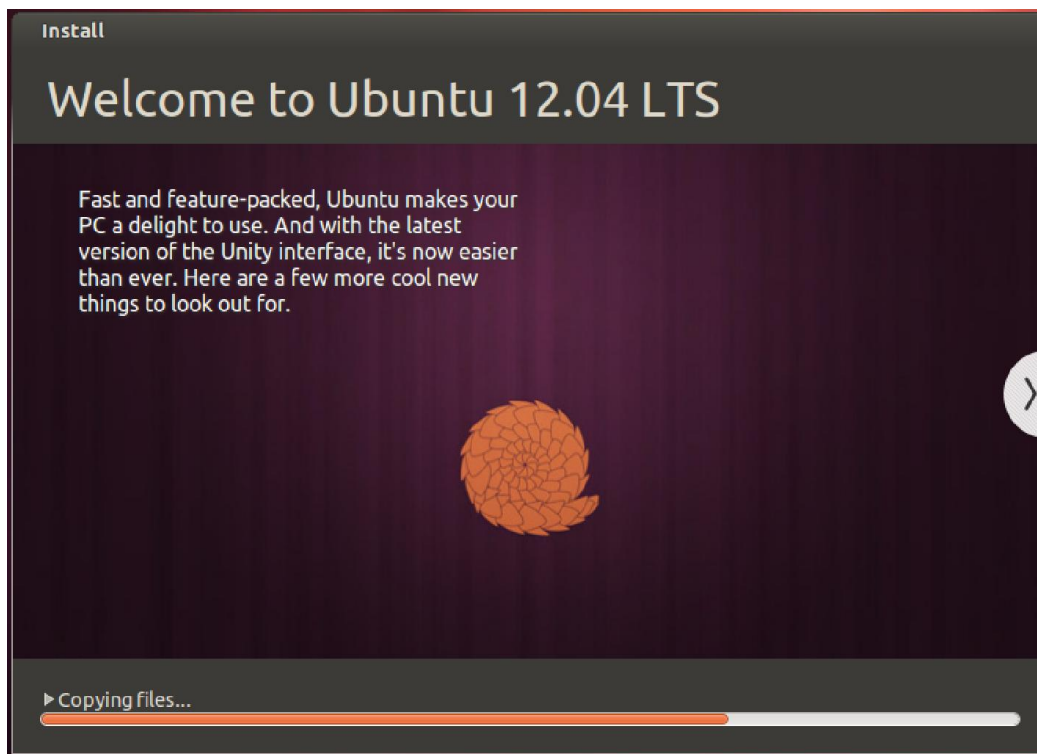
Pick a username: ✓

Choose a password: **Short password**

Confirm your password: ✓

☐ Log in automatically
☒ Require my password to log in
☐ Encrypt my home folder

Back Continue



4.Setup Services on Ubuntu

4.1 Setup file share service (FTP).

To make file can be shared between virtual machine system(Ubuntu) and the host system(Windows 7), you need to setup a file share service. There are usually two options: FTP and SAMBA. I setup FTP as file share service, because its easier than SAMBA; and I use FileZilla as the FTP tool for Windows. The detail process of setup FTP please see:

<http://www.wikihow.com/Set-up-an-FTP-Server-in-Ubuntu-Linux>

http://www.ehow.com/how_6867026_run-ftp-server-linux.html

http://wiki.ubuntu.com.cn/Vsftpd#stand_alone.E5.92.8Csuper_daemon (Chinese)

Generally, the steps are:

(1) Setup FTP. Download and install “vsftpd” service from Internet.

Type: `sudo apt-get install vsftpd`

In Ubuntu, you can install software online by Ubuntu Software Center, or from command line like “`sudo apt-get install <software name>`”

(2) Configure FTP. Use the root user to edit the “/etc/vsftpd.conf”. And enable the options below:

***Allow anonymous FTP:**

`anonymous_enable=YES`

***Allow anonymous upload:**

`write_enable=YES`

`anon_mkdir_write_enable=YES`

`anon_upload_enable=YES`

`change "ftpd_banner"`

(3) In terminal, enter the FTP folder “/srv/ftp”, create a new folder “upload” for upload, and change this folder can be read/wrote by anonymous users:

`cd /srv/ftp`

`sudo mkdir upload`

`sudo chmod -R 777 upload`

(4) Restart FTP service. Enter “`sudo service vsftpd restart`” in terminal.

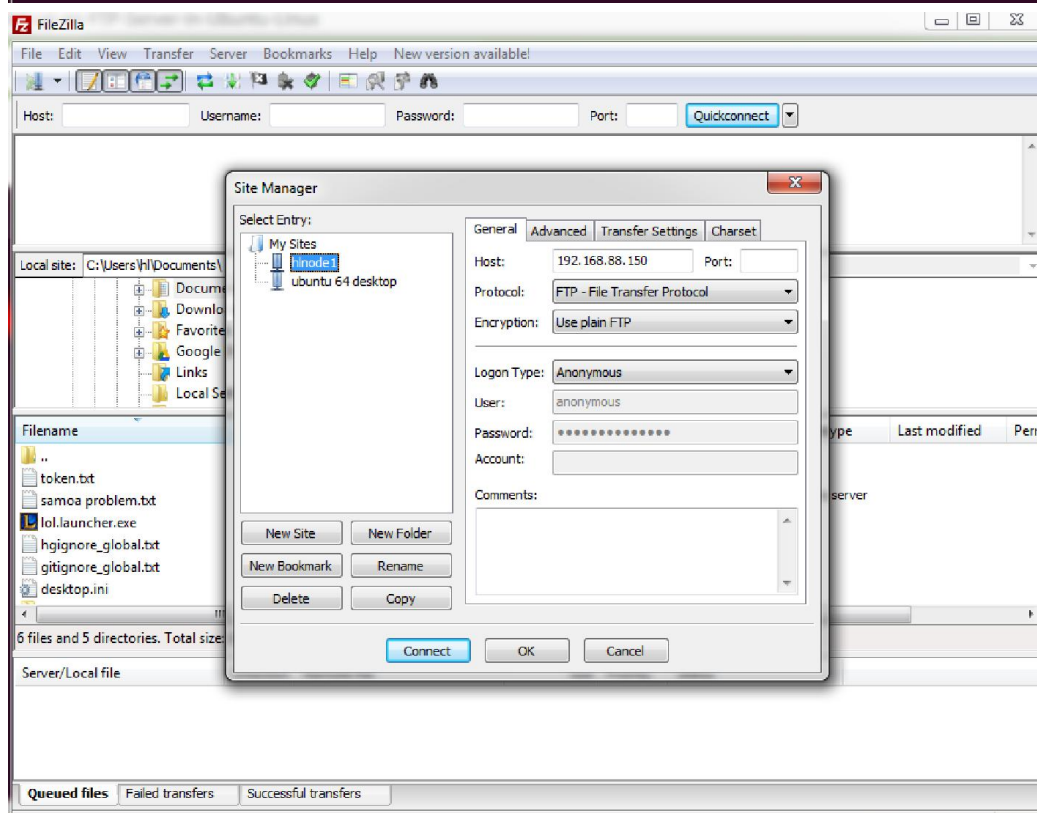
(5) Test your FPT service. Start a terminal, test your VM’s ip address by enter “ifconfig”. Now you can see the ip adress after “inet addr:”, such as “192.168.88.150”. Then start FileZilla in Windows, create a new Site, and enter this IP address in “Host” entree. Keep login mode as “anonymous mode” and “plane FTP”. Try to upload a file into “upload” folder. It it success, you can find this file in Ubuntu’s “/srv/ftp/upload” folder. In the future you can upload the file through FileZilla and get it in the “/srv/ftp/upload” folder.

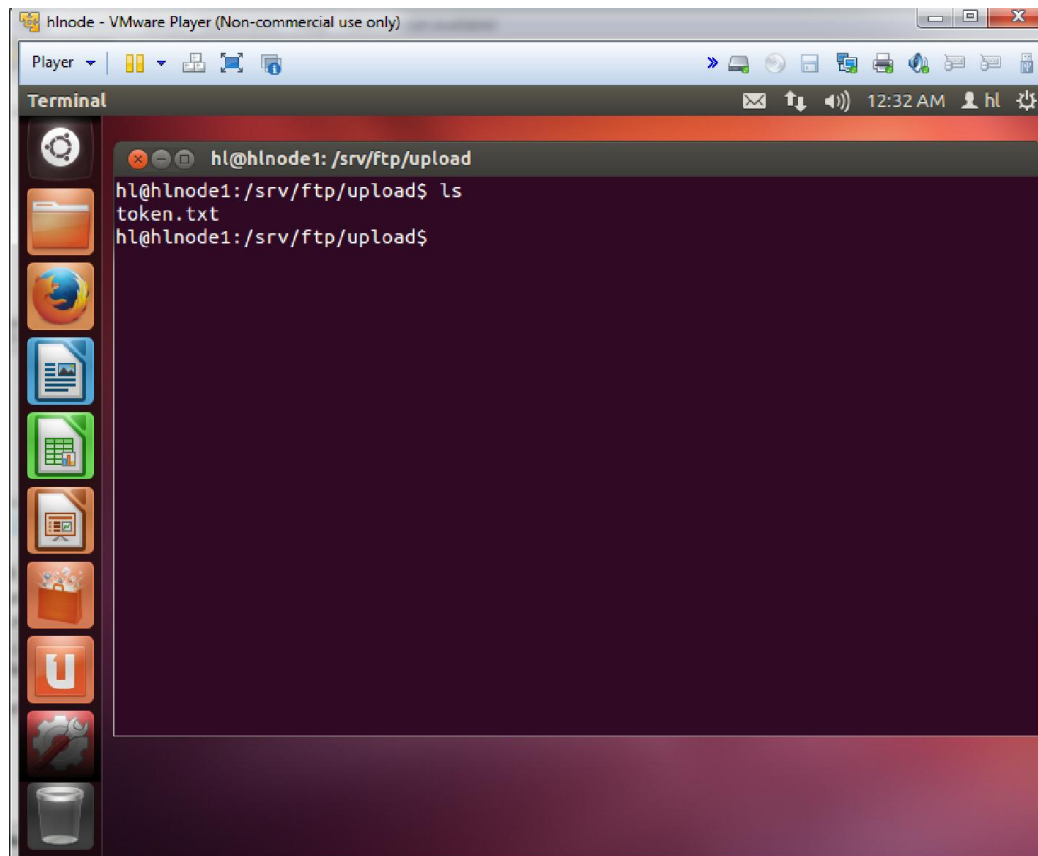
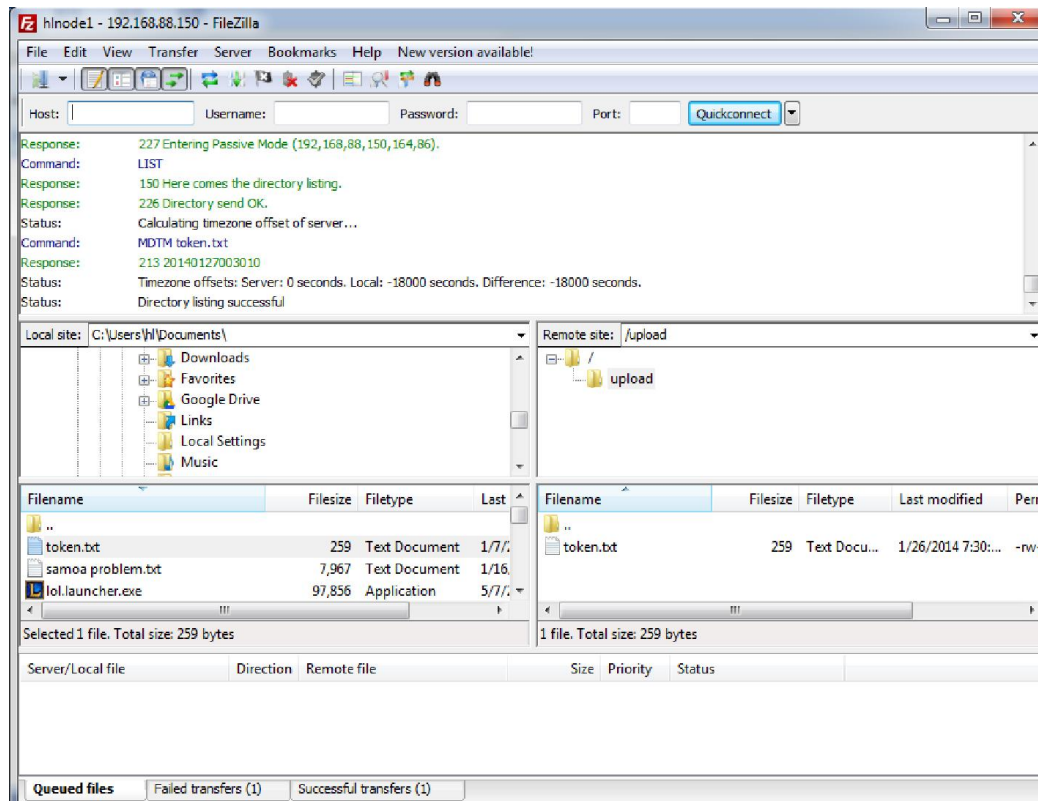
Note: Virtual Machine’s IP address usually changes, so you need to reconfigure the “host address” in FileZilla to make sure to connect to the virtual machine.

```
hl@hlnode1: ~
hl@hlnode1:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:3f:a1:7d
          inet addr:192.168.88.150 Bcast:192.168.88.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3f:a17d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:467 errors:0 dropped:0 overruns:0 frame:0
          TX packets:334 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:217256 (217.2 KB)  TX bytes:56860 (56.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6002 (6.0 KB)  TX bytes:6002 (6.0 KB)

hl@hlnode1:~$
```





4.Setup Yahoo! S4

SAMOA supported two Stream Process Engines (SPE): Yahoo S4! and Twitter Storm. At first I tried to setup Storm, but I failed. Because Storm need more or equal than 3 virtual machine running simultaneously, and each one need at least 512M memory. So totally I need 1500 MB free memory. However I do not have so much memory in the laptop, so later I give up Storm and try to setup S4.

You can find the **official document of how to setup S4** at:

<https://github.com/yahoo/samoa/wiki/Executing-SAMOA-with-Apache-S4>

I also do the process as this document, but still met some problems. So I will show my steps and how to solve the problems below.

4.1 Setup JDK 1.7

At first you should setup JDK 1.7, which is the basic environment for running all the software we setup later. You can find a lot of articles about how to setup JDK 1.7, so I will not talk this in detail. I just setup OpenJdk by entering the command below in terminal:

```
sudo apt-get install openjdk-7-jdk
```

After install, you can type “java -version” to check the jdk version, and you should see:

```
hl@hlnode1:/opt/gradle-1.4$ java -version
java version "1.7.0_51"
OpenJDK Runtime Environment (IcedTea 2.4.4) (7u51-2.4.4-0ubuntu0.12.04.2)
OpenJDK 64-Bit Server VM (build 24.45-b08, mixed mode)
```

JDK is installed in “/usr/lib/jvm” folder, you can find it such as “java-7-openjdk-amd64”.

Then we need to edit the environment file “/etc/profile” by adding:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$JAVA_HOME/db/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:$JAVA_HOME/db/bin
```

Then you can apply the change by type:

```
source /etc/profile
javac
```

to check whether jdk has been configured right.

4.2 Setup Gradle

Setup Gradle 1.4. Gradle is a tool to build S4 on your Linux system. So you need to setup it before you set S4.

Note: The official document write “Gradle 1.6”, but I tried and found when use Gradle 1.6, S4 0.6

cannot be successful build. At last I found Gradle 1.4 should be the right version.

(1) Download Gradle 1.4 from official website or

<http://services.gradle.org/distributions/gradle-1.4-bin.zip>

Upload it to your Ubuntu system. And Unzip it to “/opt/gradle-1.4” folder:

```
cd /opt
sudo unzip gradle-1.4-bin.zip -d /opt/
sudo chmod -R 777 /opt/gradle-1.4 (to make gradle runnable by all users)
```

Note: I setup S4, SAMOA and other software which I downloaded myself to “/opt” folder, which is the recommended folder to setup your own software.

(2) Edit your system environment file by enter “sudo gedit /etc/profile”. Add these lines in the end this file:

```
export GRADLE_HOME=/opt/gradle-1.4
export PATH=$PATH:$GRADLE_HOME/bin
```

(4) After you change the environment file, you need to apply it by type

```
source /etc/profile
```

(5) Install Gradle by type:

```
cd $GRADLE_HOME
gradle
```

Note: you must cd into your \$GRADLE_HOME folder, such as /opt/gradle-1.4, then type gradle to install Gradle.

```
hl@hlnode1: /opt
at org.gradle.launcher.GradleMain.main(GradleMain.java:26)

BUILD FAILED
Total time: 6.711 secs
hl@hlnode1: /opt$ cd gradle-1.4
hl@hlnode1: /opt/gradle-1.4$ gradle
:help

Welcome to Gradle 1.4.

To run a build, run gradle <task> ...

To see a list of available tasks, run gradle tasks

To see a list of command-line options, run gradle --help

BUILD SUCCESSFUL
Total time: 6.174 secs
hl@hlnode1: /opt/gradle-1.4$
```

4.3 Setup Git

Git is the tool to manage the source code, many open-source programs uses Git. Many programs can be found in “GitHub” website. You can setup Git by type:

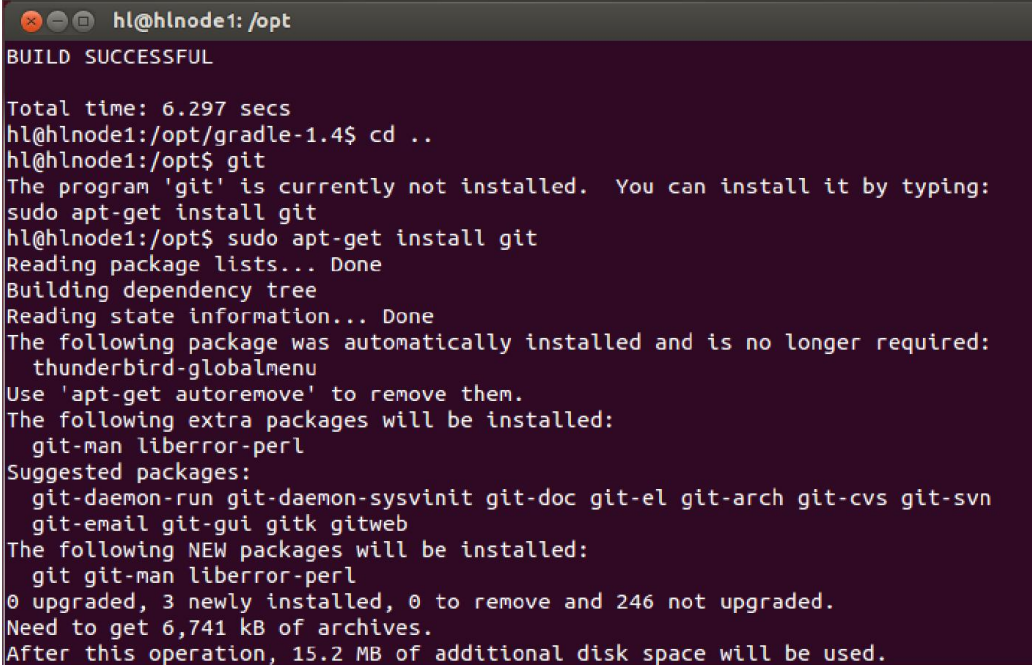

```
sudo apt-get install git
```

The usage of git can be found online. The most common commands are:

```
git clone <source-code-url> (Download a repository to your current folder).
```

```
git fetch origin <branch> (Update your local repository to a branch)
```

```
git pull origin <branch> (Update and merge your local repository to a branch)
```



```
hl@hlnode1: /opt
BUILD SUCCESSFUL

Total time: 6.297 secs
hl@hlnode1:/opt/gradle-1.4$ cd ..
hl@hlnode1:/opt$ git
The program 'git' is currently not installed. You can install it by typing:
sudo apt-get install git
hl@hlnode1:/opt$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  thunderbird-globalmenu
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-arch git-cvs git-svn
  git-email git-gui gitk gitweb
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 246 not upgraded.
Need to get 6,741 kB of archives.
After this operation, 15.2 MB of additional disk space will be used.
```

4.3 Setup S4

(1) Download the latest Apache S4 0.6.0 release:

<http://www.apache.org/dist/incubator/s4/s4-0.6.0-incubating/apache-s4-0.6.0-incubating-src.zip> (I use this way)

Then type in terminal:

```
cd /opt
sudo mkdir s4 (create a folder)
sudo chmod -R 777 s4 (let this folder be
readable/writable/executable by all the users)
sudo unzip /srv/ftp/upload/apache-s4-0.6.0-incubating-src.zip -d
/opt/s4
sudo chmod -R 777 apache-s4-0.6.0-incubating-src
```

Or you can clone S4 project from git:

```
git clone
```

<https://git-wip-us.apache.org/repos/asf/incubator-s4.git>.

- (2) Setup environment by add lines at the end of “/etc/profile”:

```
export S4_HOME=/opt/s4/apache-s4-0.6.0-incubating-src
export PATH=$S4_HOME:$PATH
```
- (3) There are some dependencies issues, therefore you should run the wrapper task first by typing:

```
cd $S4_HOME
gradle wrapper
```
- (4) Install the artifacts for Apache S4 by running “`gradle install`” in the S4_HOME directory.
 Install the S4-TOOLS, “`gradle s4-tools::installApp`”.
- (5) Done. Now you can configure and run your Apache S4 cluster.

```
hl@hlnode1: /opt
hl@hlnode1:/opt/s4/apache-s4-0.6.0-incubating-src$ ls
build.gradle  LICENSE  README.md  settings.gradle  target
DISCLAIMER   NOTICE  s4         subprojects      test-apps
hl@hlnode1:/opt/s4/apache-s4-0.6.0-incubating-src$ gradle wrapper
Runs Apache RAT. Exclusions are defined in .rat-excludes file
:wrapper

BUILD SUCCESSFUL

Total time: 59.577 secs
hl@hlnode1:/opt/s4/apache-s4-0.6.0-incubating-src$
```

```
hl@hlnode1: /opt
1 warning
:test-apps:consumer-app:processResources UP-TO-DATE
:test-apps:consumer-app:classes
:test-apps:consumer-app:jar
:test-apps:consumer-app:install
:test-apps:producer-app:compileJava
warning: [options] bootstrap class path not set in conjunction with -source 1.6
1 warning
:test-apps:producer-app:processResources UP-TO-DATE
:test-apps:producer-app:classes
:test-apps:producer-app:jar
:test-apps:producer-app:install
:test-apps:simple-deployable-app-1:compileJava
warning: [options] bootstrap class path not set in conjunction with -source 1.6
1 warning
:test-apps:simple-deployable-app-1:processResources
:test-apps:simple-deployable-app-1:classes
:test-apps:simple-deployable-app-1:jar
:test-apps:simple-deployable-app-1:install

BUILD SUCCESSFUL

Total time: 2 mins 56.23 secs
hl@hlnode1:/opt/s4/apache-s4-0.6.0-incubating-src$
```



```

hl@hlnode1: /opt
Download http://repo1.maven.org/maven2/org/apache/commons/commons-parent/9/commons-parent-9.pom
Download http://repo1.maven.org/maven2/aopalliance/aopalliance/1.0/aopalliance-1.0.pom
Download http://repo1.maven.org/maven2/com/googlecode/minlog/1.2/minlog-1.2.pom
Download http://repo1.maven.org/maven2/org/objenesis/objenesis/1.2/objenesis-1.2.pom
Download http://repo1.maven.org/maven2/org/objenesis/objenesis-parent/1.2/objenesis-parent-1.2.pom
Download http://repo1.maven.org/maven2/commons-logging/commons-logging/1.1.1/commons-logging-1.1.1.jar
Download http://repo1.maven.org/maven2/commons-collections/commons-collections/3.2.1/commons-collections-3.2.1.jar
Download http://repo1.maven.org/maven2/aopalliance/aopalliance/1.0/aopalliance-1.0.jar
Download http://repo1.maven.org/maven2/com/googlecode/minlog/1.2/minlog-1.2.jar
Download http://repo1.maven.org/maven2/org/objenesis/objenesis/1.2/objenesis-1.2.jar
:s4-tools:installApp

BUILD SUCCESSFUL

Total time: 59.55 secs
hl@hlnode1: /opt/s4/apache-s4-0.6.0-incubating-src$

```

4.4 Setup Zookeeper Client

S4 uses Zookeeper to manage the network communication of the cluster, we need to setup a Zookeeper Client for easier control the cluster of S4. Now S4 does not support delete a cluster or remove an app with a simple way. When you run the wrong app on a cluster, and later you want to stop this app, or re-setup the cluster, you need to delete the corresponding node in Zookeeper by use the Zookeeper Client. The setup steps of Zookeeper Client is below:

- (1) Open “**Ubuntu Software Center**”, search the software “**zookeeper**”, then setup it.
- (2) Then you can start a terminal, and type “**zooinspect**” to enter the zookeeper manager.
- (3) When you start the zooinspect, you can click the “connect” button in the menu. Then you will see the nodes.
- (4) You can see “**/s4/clusters**” node in the left panel. This is the root node for all clusters running on S4. You can delete a child node of “**/s4/clusters**”, so that you deleted the cluster from zookeeper.
- (5) The zookeeper is installed in “**/usr/share/zookeeper**” folder, you can also run the “**/usr/share/zookeeper/zkCli.sh**” to manage the nodes in text-terminal. Please find the commands help of **zkCli** on the Internet. The commonly used ones are:

```
ls / (show root nodes)
```

```
ls /s4 (show the nodes)
```

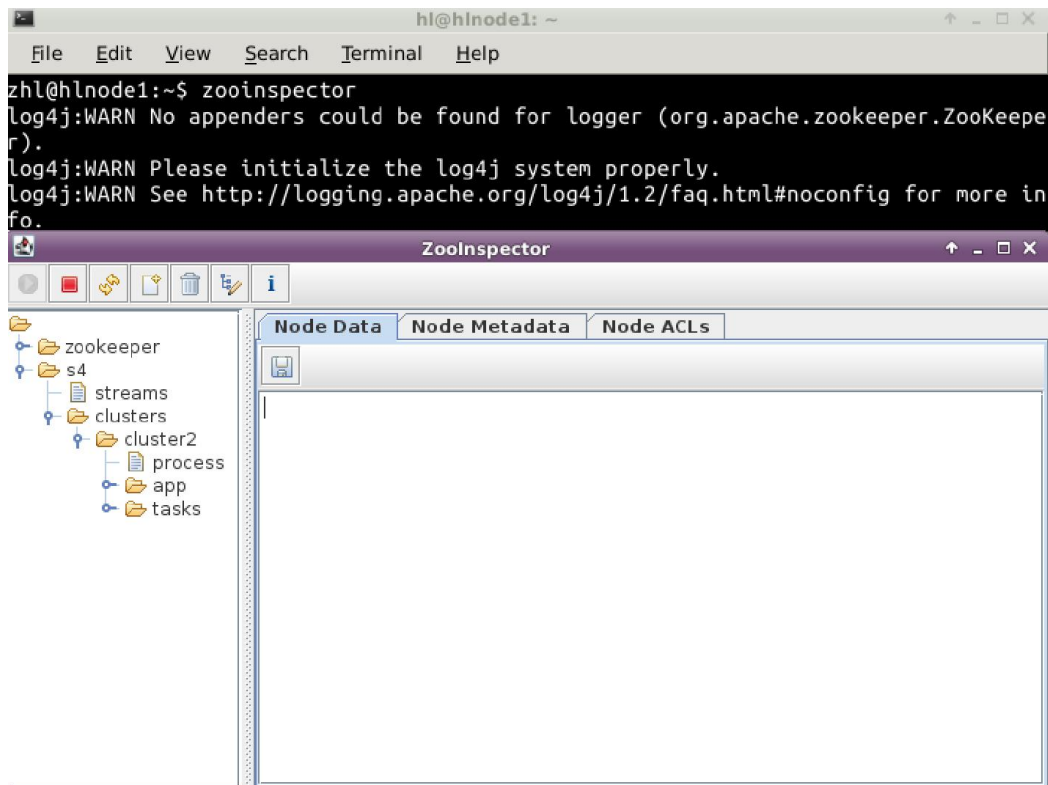
```
rmdir /s4/clusters/cluster (delete a node)
```

```
quit (quit zkCli)
```

- (6) Edit the environment file “**/etc/profile**”, add these lines:

```
export ZOOKEEPER_HOME=/usr/share/zookeeper
```

```
export PATH=$PATH:$ZOOKEEPER_HOME/bin
```



4.5 Test S4

4.5.1 HelloApp

The first sample app can be found at:

<http://incubator.apache.org/s4/doc/0.6.0/walkthrough/>

However, it has some problem in this original document for the first example, HelloApp. So I list my steps to test HelloApp here:

===== Create Cluster 1 =====

(1) Create a new project:

```
./s4 newApp myApp -parentDir=/tmp
```

(you just create a new app called myApp, and put it into /tmp folder)

```
cd /tmp/myApp (goto the myApp folder)
```

```
gradlew (build this new app)
```

(create a myApp.s4r package in build/libs folder)

```
./s4 s4r -a=hello.HelloApp -b=/tmp/myApp/build.gradle myApp
```

(2) Start Zookeeper

```
cd $S4_HOME
```

```
./s4 zkServer -clean
```

(don't close this terminal)

(3) Create cluster1 for running myApp

(You must open an new terminal)

```
./s4 newCluster -c=cluster1 -flp=12000 -nbTasks=2
```

(4) Start two nodes for this cluster

Open 2 terminals, enter in each terminal:

```
cd $S4_HOME  
./s4 node -c=cluster1
```

(5) Deploy myApp to cluster 1

```
./s4 deploy -s4r=/tmp/myApp/build/libs/myApp.s4r -c=cluster1  
-appName=myApp
```

You can enter "`./s4 status`" to check the status

===== Cluster 2 =====

(6) Create cluster2 for running an adapter for myApp

```
./s4 newCluster -c=cluster2 -nbTasks=1 -flp=13000
```

(7) Create and Deploy a adapter app to cluster 2

```
cd /tmp/myApp
```

Then, type

```
./s4 adapter -appClass=hello.HelloInputAdapter -c=cluster2  
-p=s4.adapter.output.stream=names
```

(Note: In my test, this way does not work, and an error shows.

Or:

```
./s4 node -c=cluster2 -p=s4.adapter.output.stream=names  
./s4 s4r -a=hello.HelloInputAdapter -b=/tmp/myApp/build.gradle  
myAppAdapter  
./s4 deploy -s4r=/tmp/myApp/build/libs/myAppAdapter.s4r -c=cluster2  
-appName=myAppAdapter
```

Then you can enter "`./s4 status`" to check the status

Note: In this step, the command from the official document cannot work:

```
./s4 deploy -appClass=hello.HelloInputAdapter  
-p=s4.adapter.output.stream=names -c=cluster2 -appName=adapter
```

```
hl@hlnode1: /tmp/myApp
File Edit View Search Terminal Help

-----
cluster2      myAppAdapter    1      1      Task-0
hlnode1      13000
cluster1      myApp                    2      2      Task-0
hlnode1      12000
hlnode1      12001                    Task-1
-----

Stream Status
-----
Name                Consumers                Producers
-----
names
cluster1(myApp)      cluster2(myAppAdapter)
-----

hl@hlnode1: /tmp/myApp$
```

(8) Provide some data to external stream(our adapter is listening to port 15000)

echo "Bob" | nc localhost 15000

(9) One of the nodes should output in its console:

Hello Bob!

```
hl@hlnode1: /opt/s4/apache-s4-0.6.0-incubating-src
File Edit View Search Terminal Help

01:30:32.297 [ZkClient-EventThread-10-localhost:2181] INFO o.a.s4.comm.topology.ClusterFromZK - Changing cluster topology to { nbNodes=1,name=cluster2,mode=unicast,type=,nodes=[{partition=0,port=13000,machineName=hlnode1,taskId=Task-0}]} from { nbNodes=0,name=unknown,mode=unicast,type=,nodes=[]}
Hello Bob!
Hello James!

hl@hlnode1: /opt/s4/apache-s4-0.6.0-incubating-src
File Edit View Search Terminal Help

ames] in cluster [cluster1]
01:30:12.140 [S4 platform loader] INFO org.apache.s4.core.App - Init prototype [hello.HelloPE].
01:30:32.290 [ZkClient-EventThread-10-localhost:2181] INFO o.a.s4.comm.topology.ClusterFromZK - Changing cluster topology to { nbNodes=1,name=cluster2,mode=unicast,type=,nodes=[{partition=0,port=13000,machineName=hlnode1,taskId=Task-0}]} from { nbNodes=0,name=unknown,mode=unicast,type=,nodes=[]}
Hello hha!
Hello Li!
```

```
hl@hlnode1: /tmp/myApp
File Edit View Search Terminal Help

cluster1(myApp)
-----
hl@hlnode1: /tmp/myApp$ echo "hha" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "hha" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "hha" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "hha" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "Bob" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "Li" | nc localhost 15000
hl@hlnode1: /tmp/myApp$ echo "James" | nc localhost 15000
hl@hlnode1: /tmp/myApp$
```

Summary:

The steps to create and run cluster on S4 are:

- Start **Zookeeper**, zookeeper occupies a terminal.

- Create a new **cluster**.
- Create N **nodes** for cluster, each node occupies a terminal.
- Create your App.
- Build your App into **s4r** package.
- Deploy your App's **s4r** package to cluster. It will automatically run on the cluster.
- If you want to **stop** or **remove** cluster1, delete the node `"/s4/clusters/cluster1"` from a zookeeper client (**zkCli** or **zooinspector**).
- If you want to stop **zookeeper**, or a **node**, just close the **terminal** it belongs.

4.5.2 Twitter Trend App

Another example is twitter trend app. It is a little bit complicated. You can find the official document at:

http://incubator.apache.org/s4/doc/0.6.0/twitter_trending_example/

Note: This original document has some problems with the S4 0.6.0.

After reference from the Internet, I solve the problems at last. The steps are:

- (1) Register a Twitter account, such as `<username1, password1>`
- (2) Then go to <https://dev.twitter.com/docs/auth/tokens-devtwittercom> to create a **developer app**, such as `<hlnode1>`, and get an "access token". Such as:

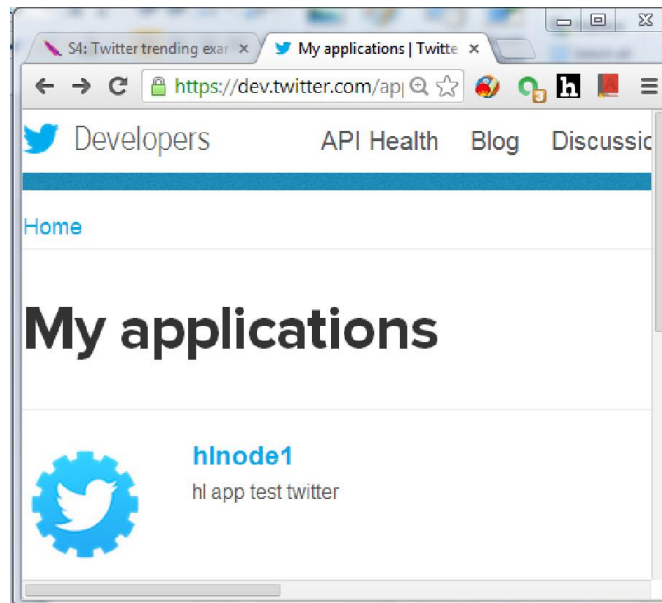
Access token: 2281204758-mncegkx6ovQouC7Kc7knvFhE4TKuJcKlfGeW9QF

Access token secret: ywh0uHKyPCdNMNBcwlos2xSV3tQLbhI23HcpkwctjKSf0

Access level Read-only

Consumer key: poKlpBh7mYyz4GYohsRRHA

Consumer secret: lPewsgVCvP8kIDZC5wESq8S9RWhHb6FRz6iu5134Stc



Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret".
Share your oauth_token_secret with anyone.

Access token	2281204758-iQFUPhrjlcC5Q2pTiq7T8Wbh8nX00ymRKZui2k
Access token secret	smGoYqzOt8aVEAEZBNZHW3LhTeH45YLPoeQHBE1Pzgvb
Access level	Read-only

Recreate my access token

- (3) Then update the codes in folders **"twitter-counter"** and **"twitter-adapter"** from your **"\$S4_HOME/test-apps"** folder:

Go to <https://issues.apache.org/jira/browse/S4-138>, and download the patch file **"S4-138.patch"**. Then transform this file from FileZilla. Then in your Ubuntu, enter commands in terminal:

```
cd /srv/ftp/upload      (goto FTP folder)
chmod -R 777 S4-138.patch
cp S4-138.patch $S4_HOME (copy file)
cd $S4_HOME
patch S4-138.patch      (apply a patch)
```

Note: here I found I found this way cannot work, because this patch file is a "normal" diff file, which only support patch one file at a time. So I try to split this file into four small patch, each of which only apply one file. Then use

```
patch -n patch1
patch -n patch2
patch -n patch3
patch -n patch4
```

And enter the corresponding file to patch when system asked.
After patch, you need to rebuild S4:

```
gradle wrapper
gradle install
gradle s4-tools::installApp
```

- (4) Create a **"twitter4j.properties"** file in the folder **"/home/hl"** (hl is my username). The content of this file please see <http://twitter4j.org/en/configuration.html>, such as:

```
oauth.consumerKey=<consumer key>
oauth.consumerSecret=<consumer secret>
oauth.accessToken=<access token>
oauth.accessTokenSecret=<access token secret>
```

You can find the right value in step (2).

- (5) According to the official document from http://incubator.apache.org/s4/doc/0.6.0/twitter_trending_example/, create 2 clusters, and start the "twitter trend" app.

===Start Zookeeper and Create 2 clusters =====

```
cd $S4_HOME
./s4 zkServer -clean
(start a new terminal and goto $S4_HOME)
./s4 newCluster -c=cluster1 -nbTasks=2 -flp=12000
./s4 newCluster -c=cluster2 -nbTasks=1 -flp=13000
```

=== Start 3 nodes in 3 terminals =====

```
./s4 node -c=cluster1
(start a new terminal and goto $S4_HOME)
./s4 node -c=cluster1
(start a new terminal and goto $S4_HOME)
./s4 node -c=cluster2
```

=== Packaging and Deploy **twitter-counter** app =====

```
(start a new terminal and goto $S4_HOME)
./s4 s4r -b=`pwd`/test-apps/twitter-counter/build.gradle
-appClass=org.apache.s4.example.twitter.TwitterCounterApp
twitter-counter

./s4 deploy -appName=twitter-counter -c=cluster1
-s4r=`pwd`/test-apps/twitter-counter/build/libs/twitter-counter.s4r
```

=== Packaging and Deploy **twitter-adapter** app =====

```
./s4 s4r -b=`pwd`/test-apps/twitter-adapter/build.gradle
-appClass=org.apache.s4.example.twitter.TwitterInputAdapter
twitter-adapter

./s4 deploy -appName=twitter-adapter -c=cluster2
-s4r=`pwd`/test-apps/twitter-adapter/build/libs/twitter-adapter.s4r -p=s4.adapter.output.stream=RawStatus
```

=== See the output result file, it shows **Top 10 Frequent Words in the tweets** right now

=====

```
tail -f TopNTopics.txt
```

The final result should be like that:


```

hl@hlnode1: /opt/s4/apache-s4-0.6.0-incu
File Edit View Search Terminal Help
oplicCountAndReportPE - Handling new topic
[GoodReads]
06:28:55.575 [TopicSeen] INFO o.a.s.e.t.T
oplicCountAndReportPE - Handling new topic
[平和島静雄聖誕祭]
06:28:55.575 [TopicSeen] INFO o.a.s.e.t.T
oplicCountAndReportPE - Handling new topic
[1月28日は池袋最強の誕生日]
06:28:55.634 [TopicSeen] INFO o.a.s.e.t.T
oplicCountAndReportPE - Handling new topic
[ULTIMO:]
06:28:55.634 [TopicSeen] INFO o.a.s.e.t.T
oplicCountAndReportPE - Handling new topic
[informa...]

hl@hlnode1: /opt/s4/apache-s4-0.6.0-incubating-src
File Edit View Search Terminal Help
hl@hlnode1:~$ cd $S4_HOME
hl@hlnode1:/opt/s4/apache-s4-0.6.0-incubating-src$ ./s4 zkServer -clean
05:56:57.310 [main] INFO org.apache.s4.tools.ZKServer - Starting zooke
er on port [2181]
05:56:57.316 [main] INFO org.apache.s4.tools.ZKServer - cleaning exist
in [/tmp/tmp/zookeeper/data] and [/tmp/tmp/zookeeper/log]
05:57:11.668 [main] INFO org.apache.s4.tools.ZKServer - Zookeeper ser
d on port [2181]

hl@hlnode1: /opt/s4/apache-s4-0.6.0-incubating-src
File Edit View Search Terminal Help
topic [30FactsAboutMe] count [10]
----
2014-Jan-28 06:28:23
topic [gameinsight] count [26]
topic [android_] count [11]
topic [AkuJatuhCintaSpikers] count [10]
----
2014-Jan-28 06:28:33
topic [gameinsight] count [29]
topic [android_] count [11]
topic [AkuJatuhCintaSpikers] count [10]

```

4.6 Congratulations

So far, the hardest part of SAMOA ---- S4, has been finished. Your future step should be setup SAMOA, which is much easier.

S4 itself is a distributed streaming processing platform, so you can create your own app based on it, or you can try more examples.

5. Recommend Readings

- Official website of Yahoo S4:
<http://incubator.apache.org/s4/>
- Official website of Yahoo SAMOA, including a document tell how to setup S4:
<http://yahoo.github.io/samoa/>
<https://github.com/yahoo/samoa/wiki/Executing-SAMOA-with-Apache-S4>
- How to setup XFCE on Ubuntu
--xfce is a lightweight disktop manager that can make Ubuntu run faster with less resources.
It is very useful for virtual machines and old computers.
<http://www.psychocats.net/ubuntu/xfce>