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
Edit: 2014/2/5

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

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.

In a simple figure, the brief process and the time taken of Setup S4 is below:

Fig. S4 Setup Process

Difficulties you might met:

At the first time setup S4, I met several problems. You might also met them when you do it as the official document. So in the later parts, I will write a "**Note**" if that part is difficult or I want to tell you my personal experience.

The **hardest part** in this process I think is "setup useful tools" and "setup S4". Setup software in Linux is not as easy as in Windows, because you need to type many command lines to configure the software.

Therefore I choose **Ubuntu** as the Linux system for S4, because setup software in Ubuntu is relatively easy, with only a "**sudo apt-get install <software>**" command.

Another difficulty you must keep in mind at any time is "**file authority**". Every folder in the Linux file system has a "authority". By default you login the Linux as a normal user such as "hl", but you need a "root user" authority to setup software, configure some system file, or create a new folder. Usually you need to add a "**sudo**" command before you execute a command which needs the "**root authority**". I will explain later in the places you need change authority.

Totally you will need about **4~5 hours** to setup S4 if you are not familiar with Linux.

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

- **Note:**

Minimum Hardware requirement

Later I will setup the Ubuntu Linux on this machine, because my computer is 64 bit system, I setup Ubuntu 64 bit. But if your computer system is 32 bit (e.g. your CPU is Pentumn 3,Pentumn 4), you should setup Ubuntu 32bit version. The minimum requirement for Ubuntu 12.04 is **512MB memory** and **5GB hard disk space**. If your computer does not satisfy this minimum requrement, you had better to setup a "light-weight" linux, such as Xubuntu and Lubuntu.

2.Setup Virtual Machine

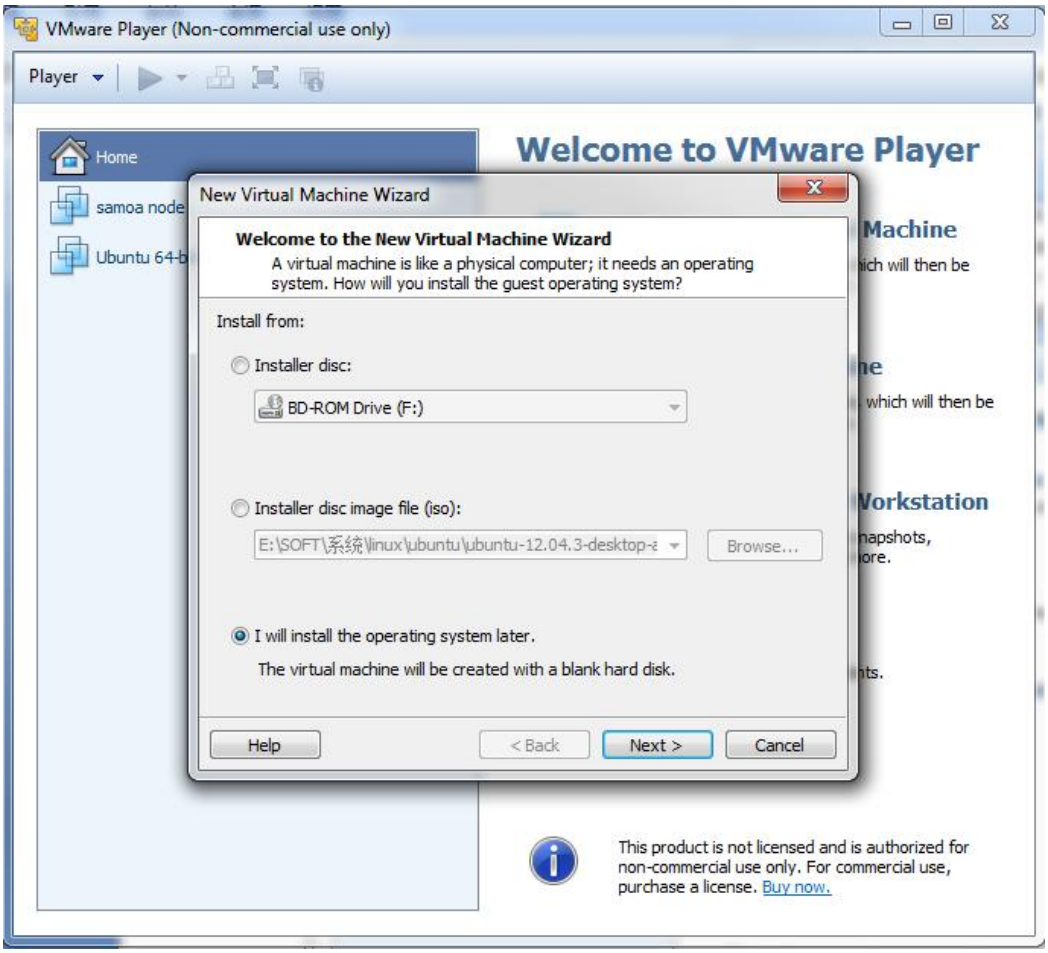
First I setup a virtual machine on Windows 7.

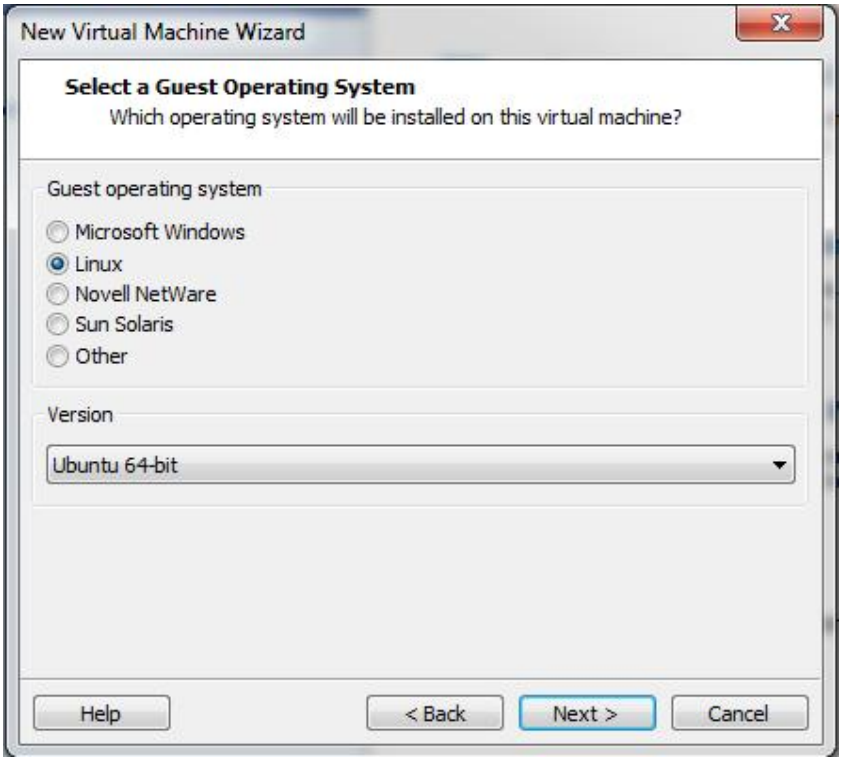
(1) Setup **VMware Player 5.0.2**.

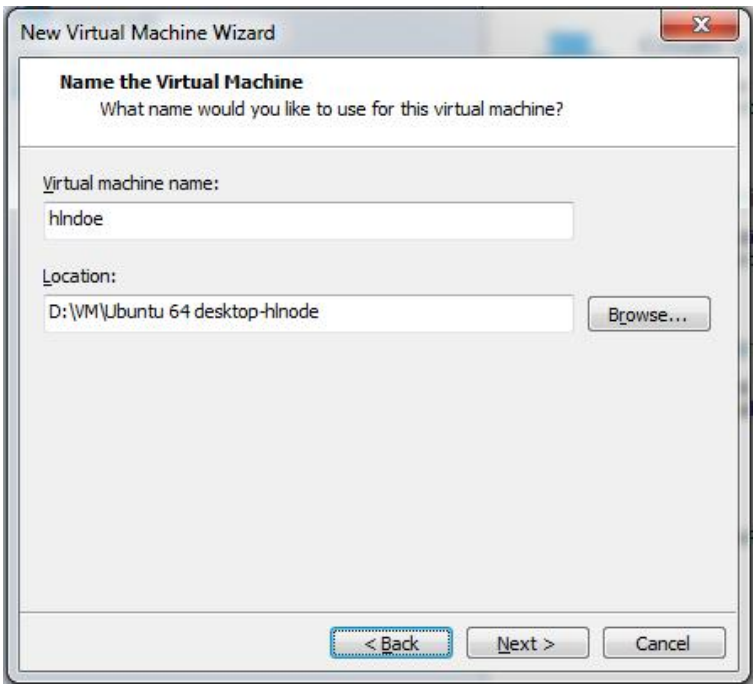
- **Note**: I also tried VMware player 10 but it is slower than VMware 5, so I recommend you to setup VMPlayer 5.

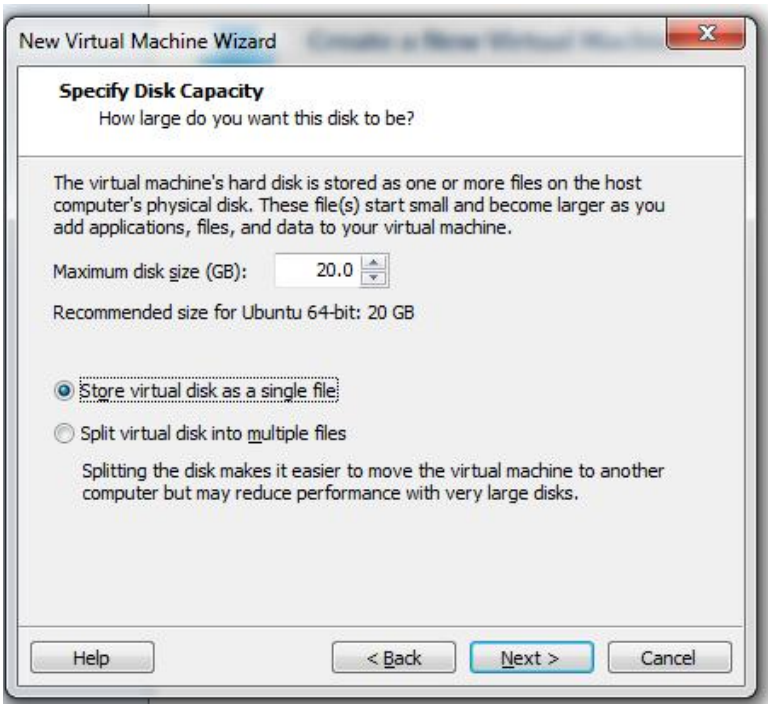
(2) Download **Ubuntu 12.04 Desktop**. Download it from the official website, and it should be a virtual CD image file "ubuntu-12.04.3-desktop-amd64.iso". This file is a virtual DVD file, so you can burn this file into a DVD to create an "Ubuntu setup disk".

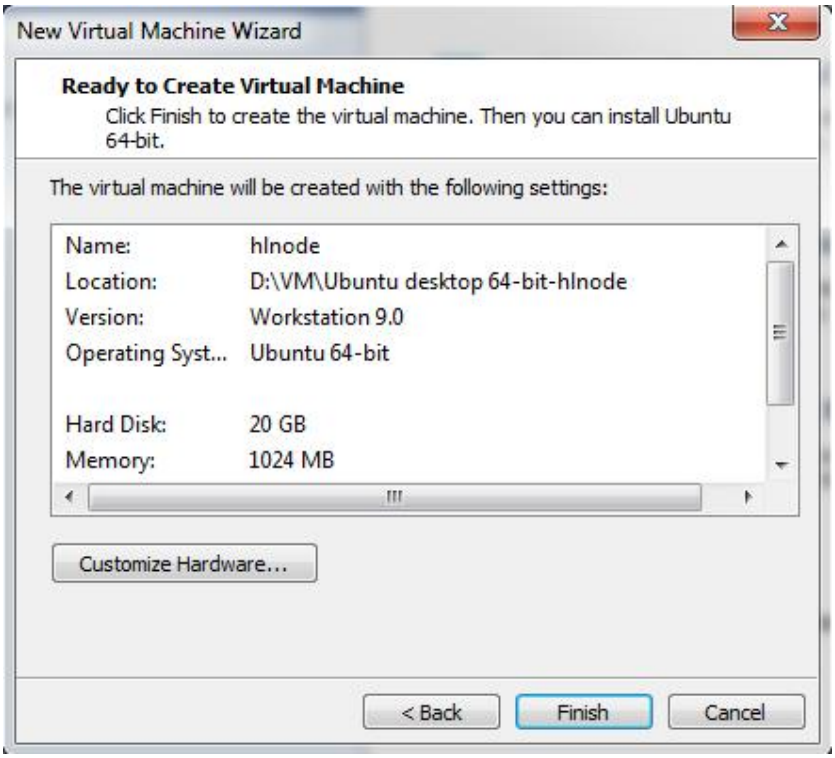
(3) Configure VM. Run the VMware Player, create a new virtual machine, and select "I will install the operating system later". Then select your system type. Then you can give a Maximum Disk Size for this virtual machine: I recommend 20GB, because Ubuntu system takes at least 5GB and you can keep 10GB 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".


VMware Player (Non-commercial use only)
Player
Home
samoa node
Ubuntu 64-bit
Welcome to VMware Player
New Virtual Machine Wizard
Welcome to the New Virtual Machine Wizard
A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?
Install from:
Installer disc:
BD-ROM Drive (F:)
Installer disc image file (iso):
E:\SOFT\系统\linux\ubuntu\ubuntu-12.04.3-desktop-a
Browse...
I will install the operating system later.
The virtual machine will be created with a blank hard disk.
Help
< Back
Next >
Cancel
This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.


New Virtual Machine Wizard
Select a Guest Operating System
Which operating system will be installed on this virtual machine?
Guest operating system
Microsoft Windows
Linux
Novell NetWare
Sun Solaris
Other
Version
Ubuntu 64-bit
Help
< Back
Next >
Cancel


New Virtual Machine Wizard
Name the Virtual Machine
What name would you like to use for this virtual machine?
Virtual machine name:
hlndoe
Location:
D:\VM\Ubuntu 64 desktop-hlnode
Browse...
< Back
Next >
Cancel


New Virtual Machine Wizard
Specify Disk Capacity
How large do you want this disk to be?
The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.
Maximum disk size (GB):
20.0
Recommended size for Ubuntu 64-bit: 20 GB
Store virtual disk as a single file
Split virtual disk into multiple files
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.
Help
< Back
Next >
Cancel



3.Setup Ubuntu Linux

After the initial configuration and start this Virtual Machine you just created. Now you need to setup Ubuntu. In my personal experience, it is very easy, that you just need to download the "setup iso" file and load it before you start the VM.

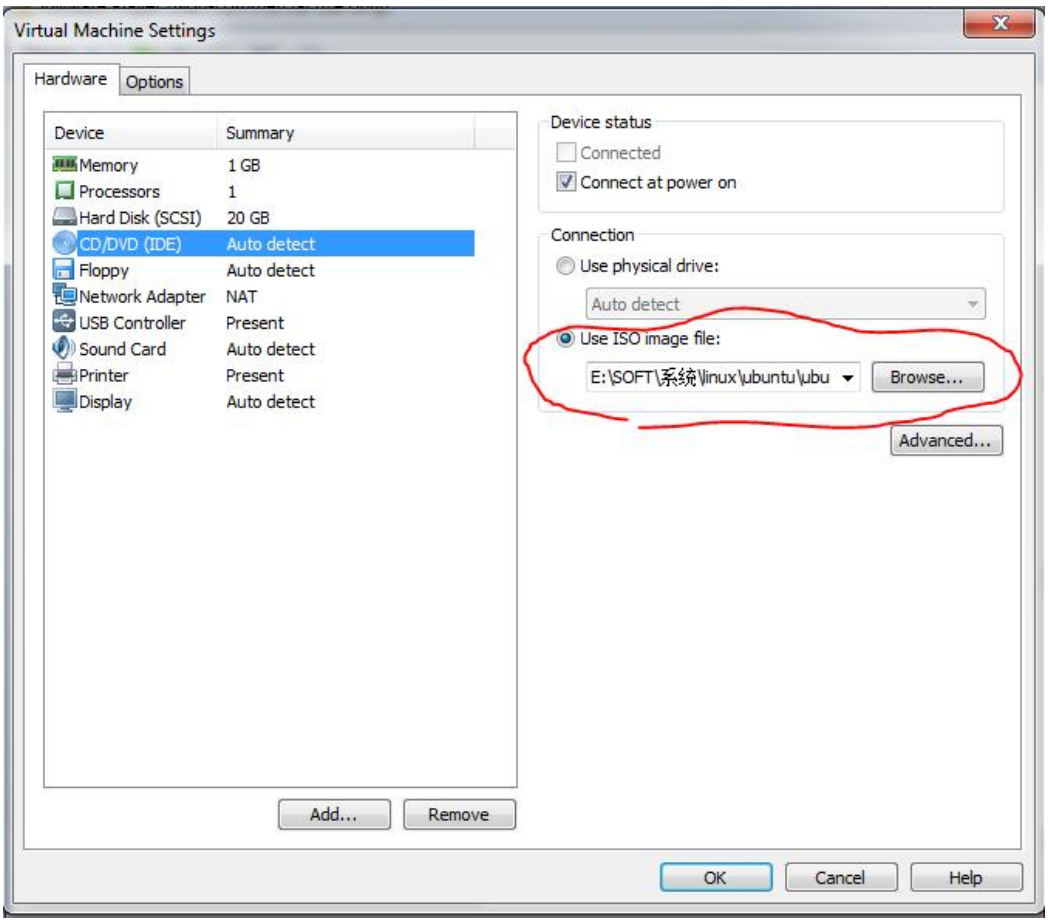
Process:

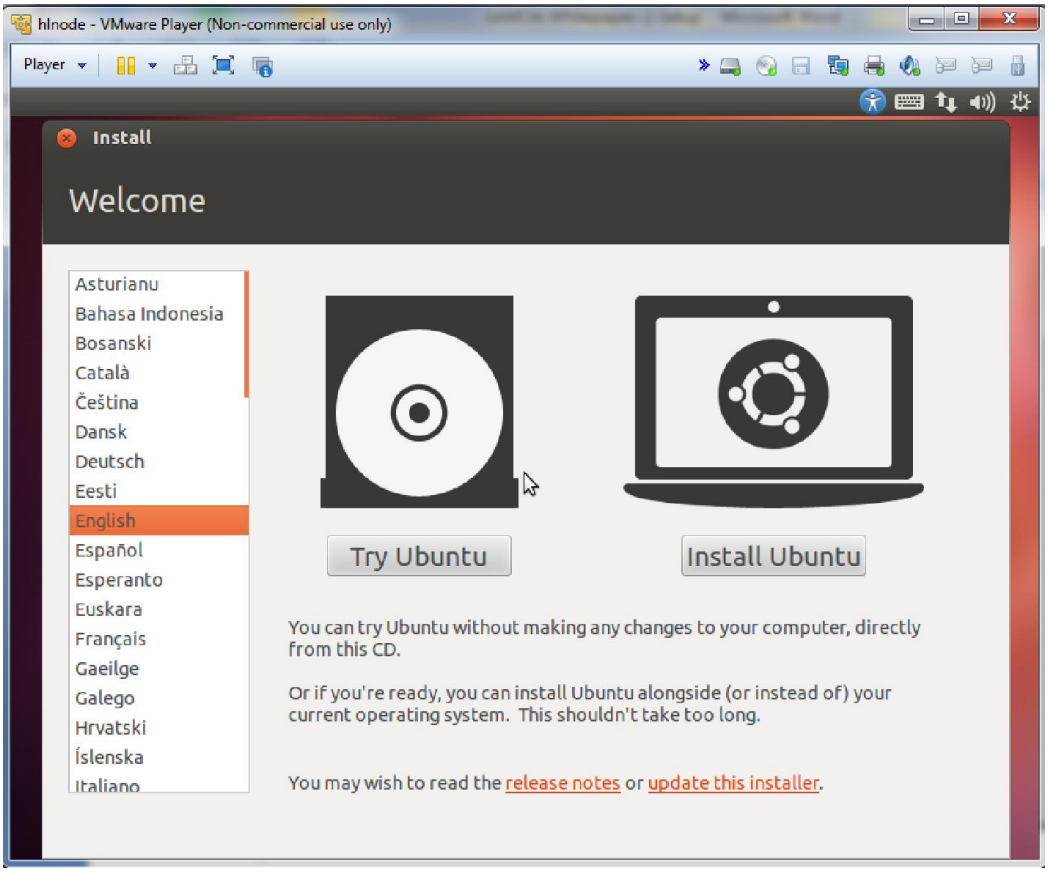
(1) Configure CD-rom. In VMplayer, choose your virtual machine node, and select "edit virtual machine settings". In the "Hardware" tab, select CD/DVD(IDE), choose "Use ISO Image file" and select the Ubuntu system's iso file.

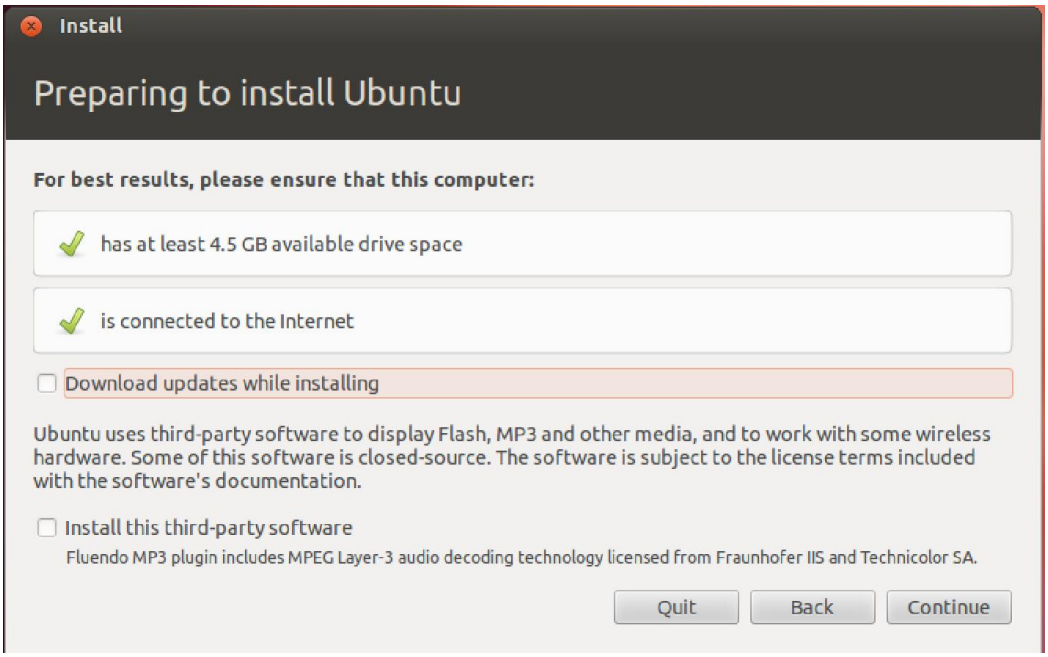
(2) Start virtual machine. Now you can see the picture of Ubuntu Setup. Choose "English" and click the "Install Ubuntu" button. Then you can just **use the default settings** and click "continue".

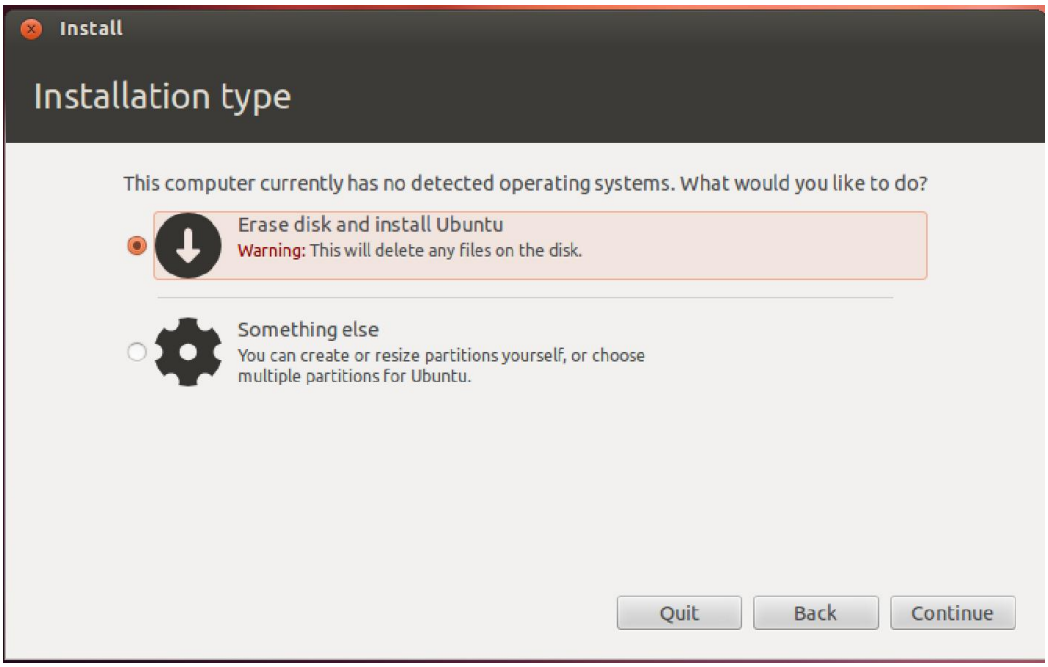
- **Note**: To speed up the installation, please **DO NOT** check the "**download update while install**" or "**install the third-party software**". In the "**Installation type**" please choose "**Erase disk and install Ubuntu**" so that to make a clear install on the VM.

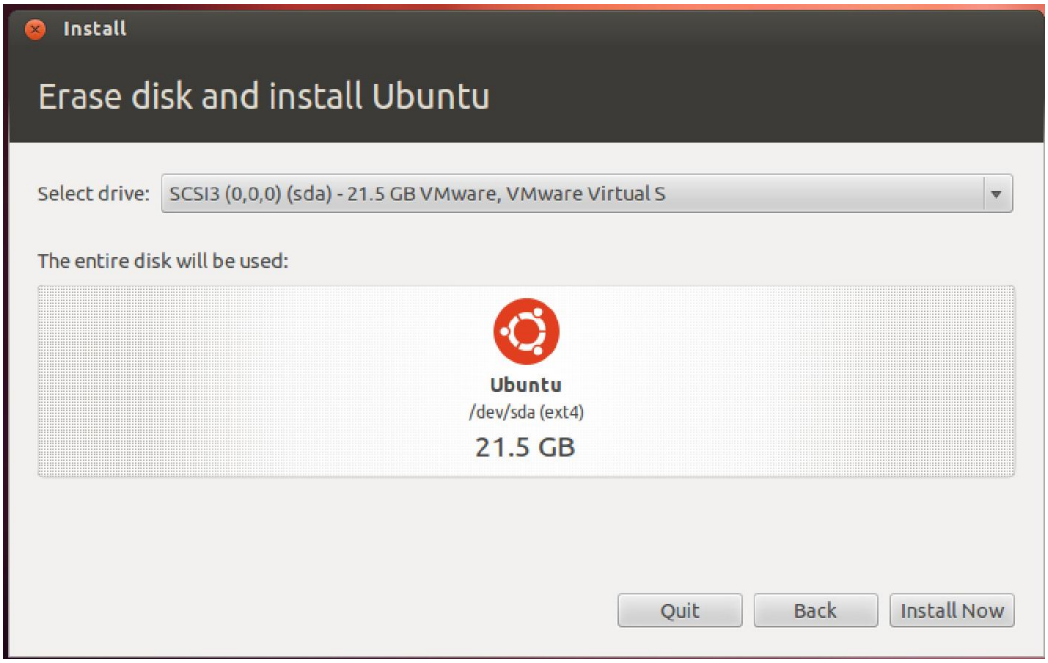
(3) Setup username and machine name. There is a step called "Who are you", in this step you need to specify your computer name and username.

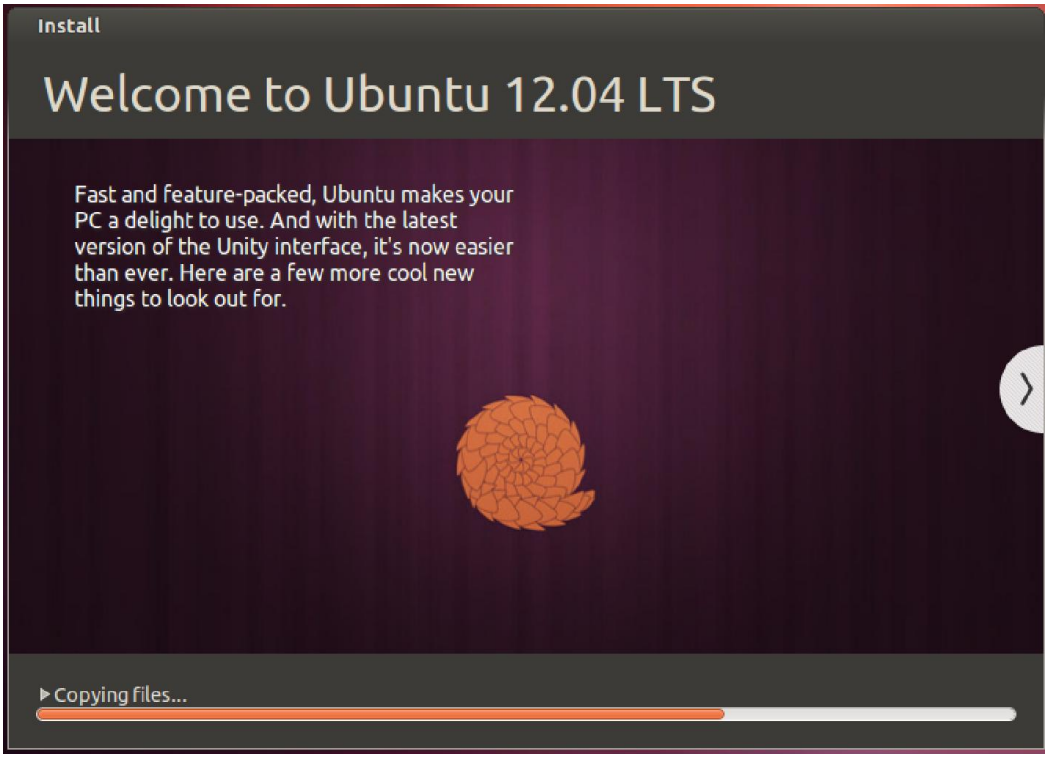

Virtual Machine Settings
Hardware
Options
Device
Summary
Memory
1 GB
Processors
1
Hard Disk (SCSI)
20 GB
CD/DVD (IDE)
Auto detect
Floppy
Auto detect
Network Adapter
NAT
USB Controller
Present
Sound Card
Auto detect
Printer
Present
Display
Auto detect
Add...
Remove
Device status
Connected
Connect at power on
Connection
Use physical drive:
Auto detect
Use ISO image file:
E:\SOFT\系统\linux\ubuntu\ubu
Browse...
Advanced...
OK
Cancel
Help


hlnode - VMware Player (Non-commercial use only)
Player
Install
Welcome
Asturianu
Bahasa Indonesia
Bosanski
Català
Čeština
Dansk
Deutsch
Eesti
English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Try Ubuntu
Install Ubuntu
You can try Ubuntu without making any changes to your computer, directly from this CD.
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.
You may wish to read the release notes or update this installer.


Install
Preparing to install Ubuntu
For best results, please ensure that this computer:
has at least 4.5 GB available drive space
is connected to the Internet
Download updates while installing
Ubuntu uses third-party software to display Flash, MP3 and other media, and to work with some wireless hardware. Some of this software is closed-source. The software is subject to the license terms included with the software's documentation.
Install this third-party software
Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Technicolor SA.
Quit
Back
Continue


Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete any files on the disk.
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.
Quit
Back
Continue


Install
Erase disk and install Ubuntu
Select drive:
SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S
The entire disk will be used:
Ubuntu
/dev/sda (ext4)
21.5 GB
Quit
Back
Install Now



4.Setup Necessary Tools

Before we setup S4, we need to setup some necessary tools. They are:

(1) **FPT service**. To make you share files between your Windows and VM.

(2) **JDK 1.7**. Java Develop Kit, necessary basic platform to compile and run the Java codes. S4 and SAMOA and many other software are written by JAVA and require this tool.
(3) **Gradle**. A tool to build S4.
(4) **Git**. The software to download and update S4 and SAMOA source code, because they are managed in Git server.
(5) **Zookeeper Client**. This software allow you to control the Zookeeper service, which coordinate the nodes in cluster.

Note:

In the official document, some of them are not mentioned, but in actual setup, they are necessary!

Usually you can setup them by type "**sudo apt-get install <software>**". But after setup each of them, you must configure the "environment variable file"--"**/etc/profile**". Moreover, if a software is not setup by "apt-get" but by download and extract to some folder--usually "**/opt/<software>**", you need to change the "file authority" of this folder to allow all users can access it.

In my first installation, I just setup **JDK 1.6** and **Gradle**. They are enough for S4, but when I want to setup SAMOA, **JDK 1.6 is NOT supported but JDK 1.7 is needed**. When you want to update to the newest version of SAMOA and S4, **Git** is more efficient than just download the "package" of this version. And when you need to stop a running applications(**App**) in a cluster, you need Zookeeper Client.

Please also note I use **Gradle 1.4** instead of Gradle 1.6 in the official document, because **Gradle 1.4 is the correct supported tool for S4 0.6.0**.

Therefore, **BE CAREFUL** to the procedures below, especially the **configuration** parts! The whole time to setup these tools is about **1 hour**.

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.

- **Note**: There are usually two options -- **FTP** and **SAMBA**. I recommend you setup FTP as file share service, because **FTP is easier to use and has better compatibility 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)

In detail, 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>**"

(1) Configure FTP. Use the root user to edit the configuration file of FTP, "**/etc/vsftpd.conf**". And enable the options below:

 # Allow anonymous FTP:

anonymous_enable=YES

 # Allow anoymous upload:

write_enable=YES

anon_mkdir_write_enable=YES

anon_other_write_enable=YES

anon_upload_enable=YES

 # Security

anon_world_readable_only=NO

***change** "ftpd_banner"*

(2) Create upload folder. 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

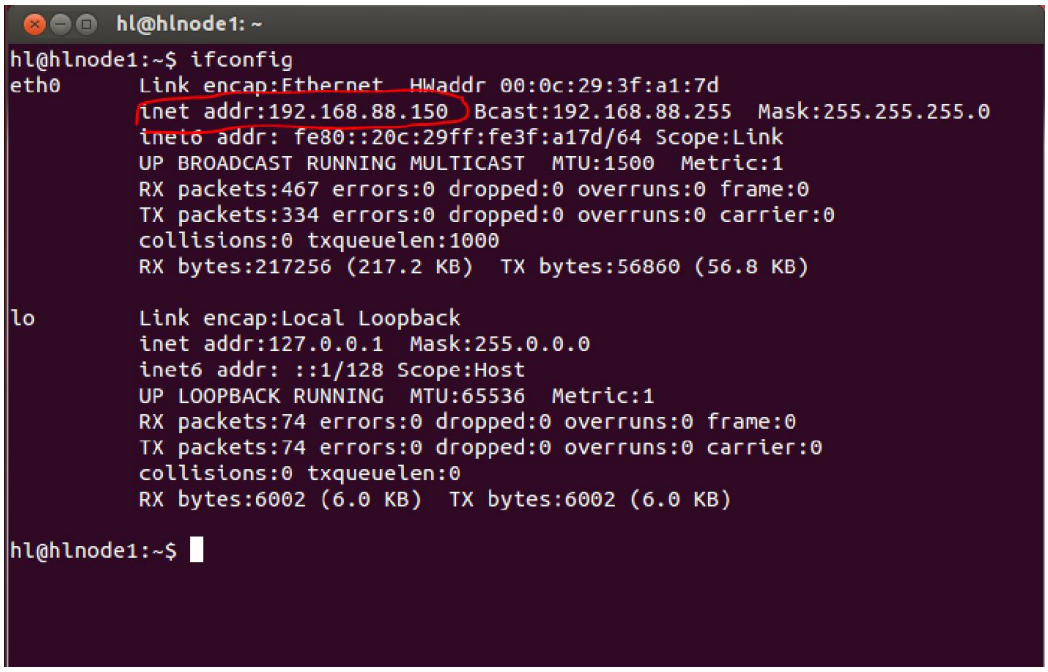
sudo chmod –R 777 upload

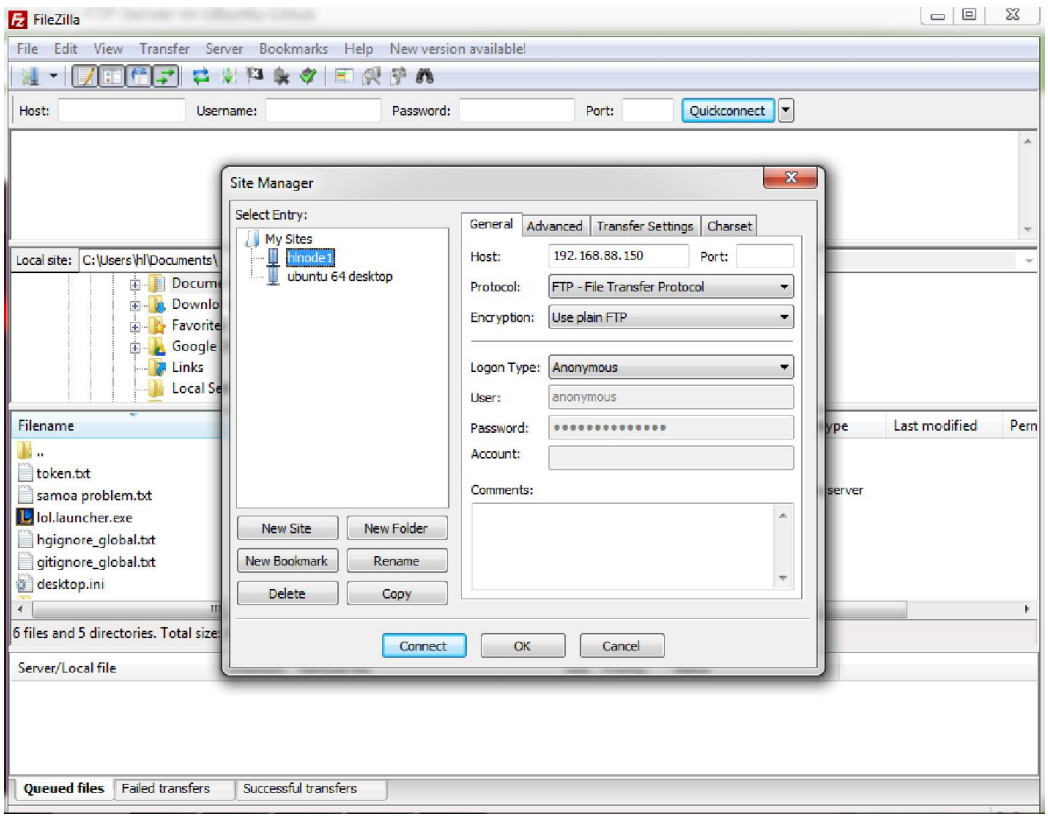
- **Note:** the command "sudo chmod -R 777 <folder name>" is to change the authority of this folder can be WRITE/READ/EXECUTE by all user.

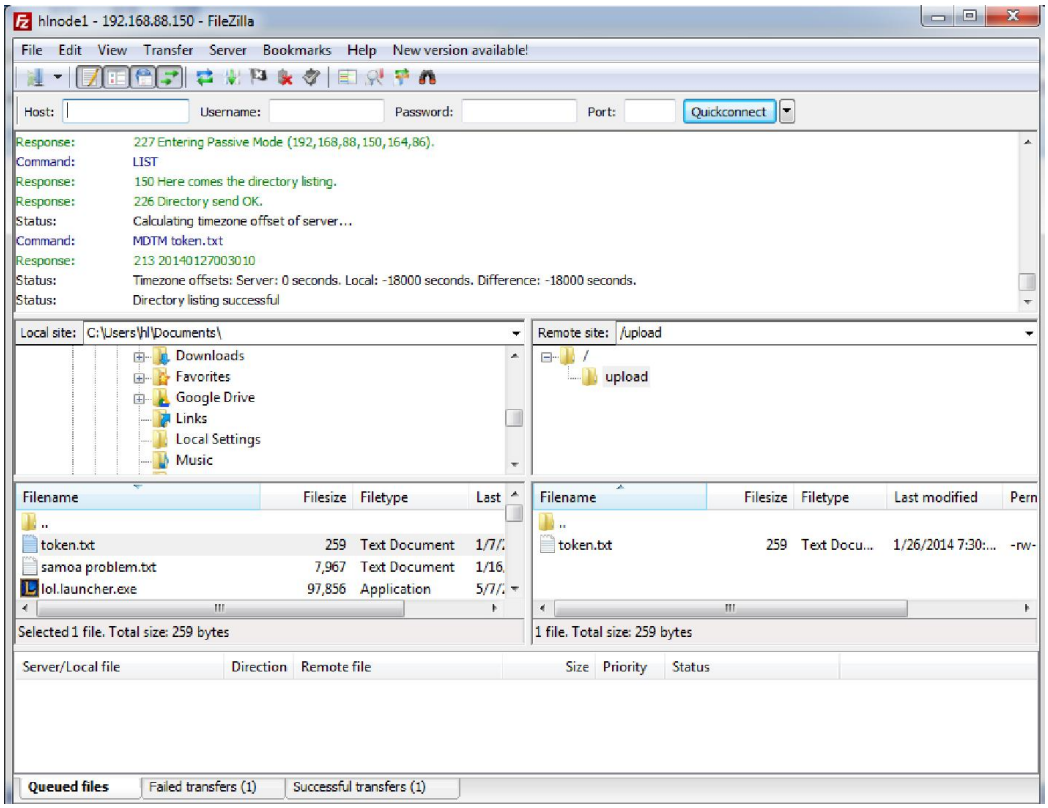
(3) Restart FTP service. Enter "**sudo service vsftpd restart**" in terminal.

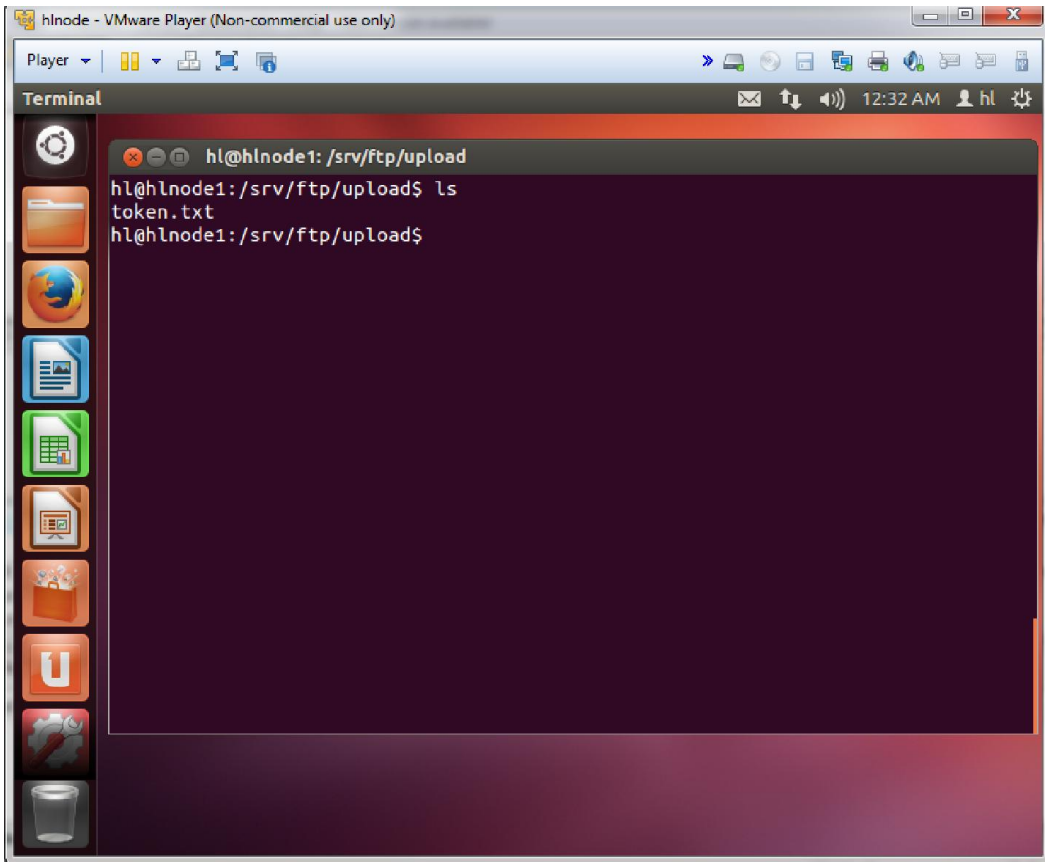
(4) Test your FPT service. Start a terminal, test your VM's ip address by enter "**ifconfig**". Now you can see the ip adress after "inet addr:", such as "192.168.88.150". Then start FileZilla in Windows, create a new Site, and enter this IP address in "Host" entree. Keep login mode as "anonymous mode" 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 when you cannot connect to the FTP server, you need to type "ifconfig" to see the newest IP address, and 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:~$


FileZilla
File Edit View Transfer Server Bookmarks Help New version available!
Host: Username: Password: Port: Quickconnect
Site Manager
Select Entry:
My Sites
hlnode1
ubuntu 64 desktop
General Advanced Transfer Settings Charset
Host: 192.168.88.150 Port:
Protocol: FTP - File Transfer Protocol
Encryption: Use plain FTP
Logon Type: Anonymous
User: anonymous
Password:
Account:
Comments:
New Site New Folder
New Bookmark Rename
Delete Copy
Connect OK Cancel
Local site: C:\Users\hl\Documents\
Filename
token.txt
samoa problem.txt
lol.launcher.exe
hgignore_global.txt
gitignore_global.txt
desktop.ini
Last modified
6 files and 5 directories. Total size:
Server/Local file
Queued files Failed transfers Successful transfers


hlnode1 - 192.168.88.150 - FileZilla
File Edit View Transfer Server Bookmarks Help New version available!
Host: Username: Password: Port: Quickconnect
Response: 227 Entering Passive Mode (192,168,88,150,164,86).
Command: LIST
Response: 150 Here comes the directory listing.
Response: 226 Directory send OK.
Status: Calculating timezone offset of server...
Command: MDTM token.txt
Response: 213 20140127003010
Status: Timezone offsets: Server: 0 seconds. Local: -18000 seconds. Difference: -18000 seconds.
Status: Directory listing successful
Local site: C:\Users\hl\Documents\
Downloads
Favorites
Google Drive
Links
Local Settings
Music
Remote site: /upload
/
upload
Filename Filesize Filetype Last
token.txt 259 Text Document
samoa problem.txt 7,967 Text Document
lol.launcher.exe 97,856 Application
Filename Filesize Filetype Last modified
token.txt 259 Text Docu... 1/26/2014 7:30:...
Selected 1 file. Total size: 259 bytes
1 file. Total size: 259 bytes
Server/Local file Direction Remote file Size Priority Status
Queued files Failed transfers (1) Successful transfers (1)


hlnode - VMware Player (Non-commercial use only)
Player
Terminal
12:32 AM hl
hl@hlnode1: /srv/ftp/upload
hl@hlnode1:/srv/ftp/upload$ ls
token.txt
hl@hlnode1:/srv/ftp/upload$

4.2 Setup JDK 1.7

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. Instead of "**Oracle/Sun JDK**", Ubuntu integrated "**OpenJDK**" runtime inside, so for better compatibility, I setup **OpenJDK 1.7**.

(1) Uninstall Java old version. At first, you should uninstall current version of your JAVA. (Usually it is OpenJDK 1.6). Just go to the "**Ubuntu Software Center**", then find "java"

(2) Setup JDK 7. 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".

(3) Edit system environment file.

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 should apply this change in environment file by type:

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

to check whether JDK has been configured right.

- **Note**: the command "**source**", also called ".", "dot" command is the command to execute a **shell script** or a executable program. "**/etc/profile**" is the script running on each reboot of Linux.

(4) Replace system default JAVA. For some system that need old version JAVA which we cannot uninstall, please replace the default JAVA runtime by tell the Ubuntu:

```
$sudo update-alternatives --install /usr/bin/java java
  /usr/lib/jvm/java-7-openjdk-amd64/bin/java 300
$sudo update-alternatives --install /usr/bin/javac javac
  /usr/lib/jvm/ java-7-openjdk-amd64/bin/javac 300
$ sudo update-alternatives --config java
```

We can see there are 2 options to replace java (support /usr/bin/java):

Item	Path	priority	mode
0	/usr/lib/jvm/java-6-openjdk/jre/bin/java	1061	automode
1	/usr/lib/jvm/java-6-openjdk/jre/bin/java	1061	manual
* 2	/usr/lib/jvm/java-6-sun/bin/java	300	manual

Enter the item number we want use as Java path: **2**

At last, we need to verify, by typing "`java -version`"

```
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)
```

Then we have replaced system's default JAVA to the new version.

4.3 Setup Gradle

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.*

Process:

(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
```

- ***Note**: **gedit** is a GUI text editor software such as "notepad" in Windows.*

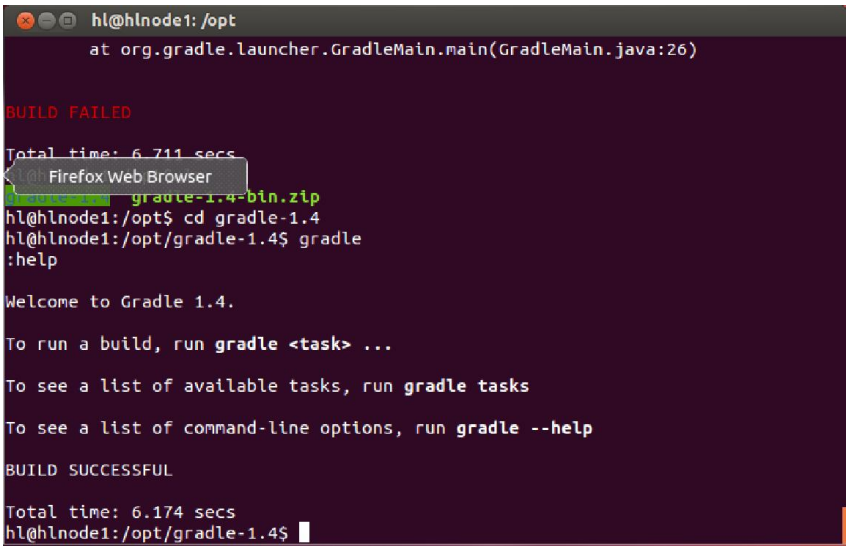
(4) After you change the environment file, you need to apply it by type

```
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.*



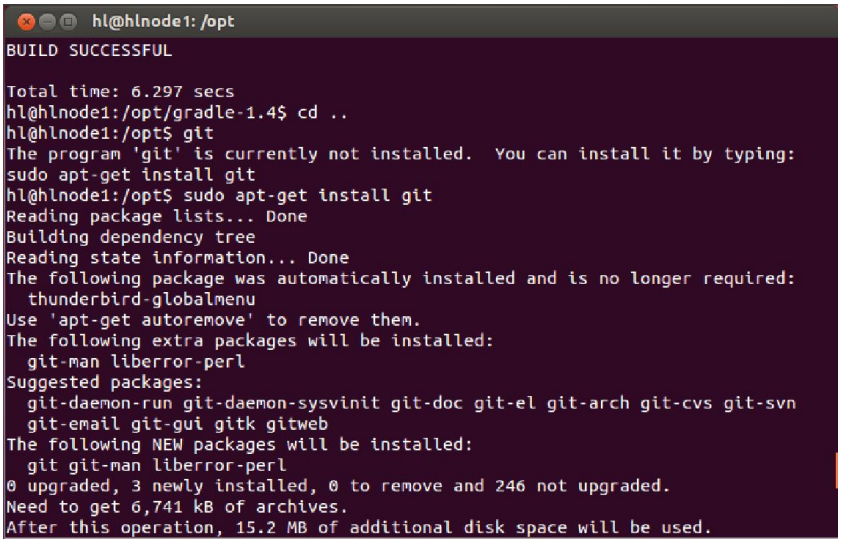
4.4 Setup Git

Git is the tool to manage the source code. Many open-source projects(including S4 and SAMOA) can be found in "GitHub" website. Git is similar as "SVN" and "CVS". 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)
```



Note: There are also some **GUI clients for Git**, I recommend "**Smart Git**" which is a very good and has both Windows and Linux version.

5.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 my laptop. Moreover, Storm is a little bit hard to configure, so at last I gave 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 below, which has a little bit difference from the official document.

5.1 S4 Setup Process

(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)

(2) Then type in terminal to create the S4 folder:

```
sudo chmod -R 777 /opt
cd /opt
sudo mkdir s4        (create a folder)
sudo chmod -R 777 s4   (let this folder be readable/writable/excutable by all the users)
```

- **Note**: "**sudo chmod -R 777 <folder>**" is necessary to change the "**/opt**" and "**/s4**" folder to WRITE/READ/EXECUTE by all users. If you do not do this chmod, you cannot execute S4 normally.

(3) Unzip S4 into this folder:

```
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
```

- **Note:**

Please do not use Git to clone s4, because we need the 0.6.0 final version, not the other versions to get full compatibility of SAMOA. Once I tried to use Git to clone S4, later I found SAMOA cannot build s4 package, because I got S4 ver0.5.

(4) 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
```

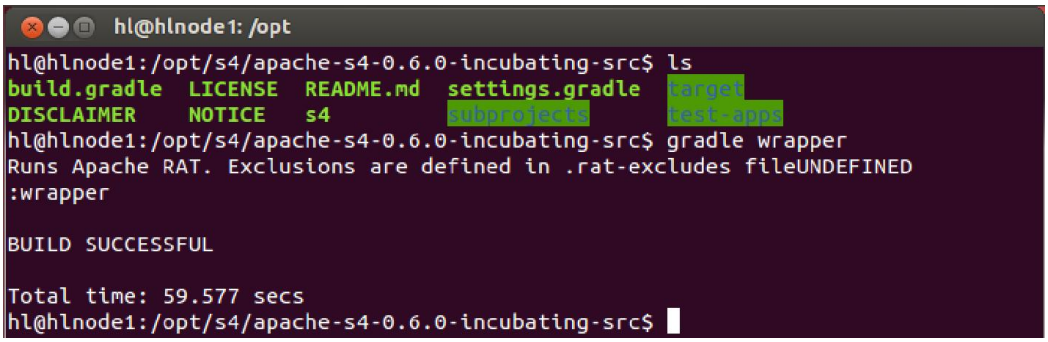
(5) There are some dependencies issues, therefore you should run the wrapper task first by typing:

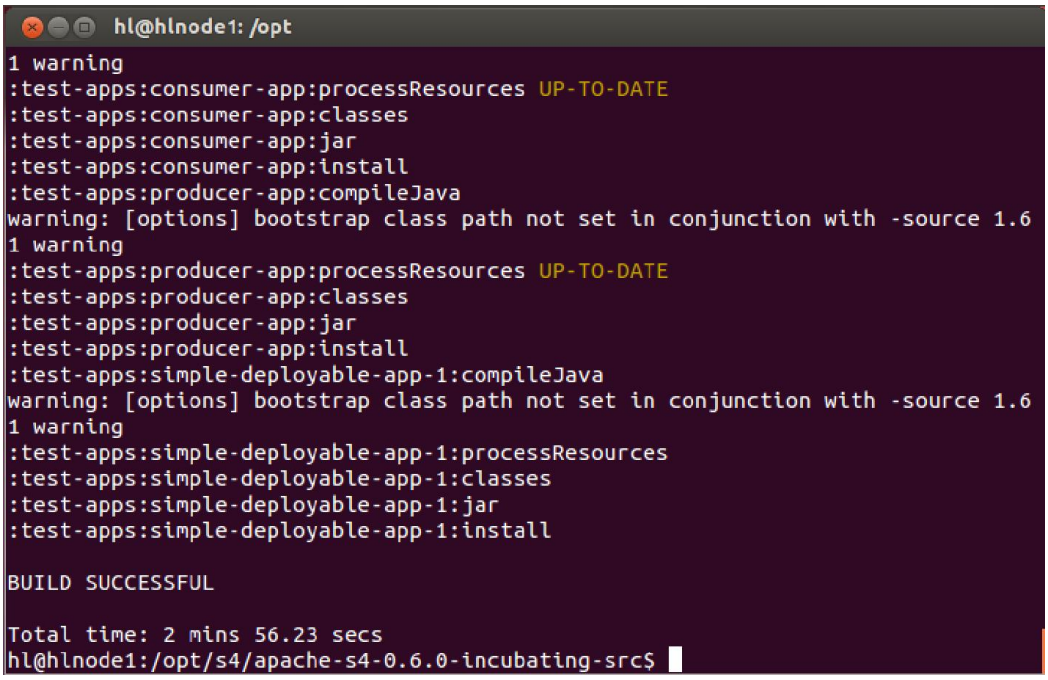
```
cd $S4_HOME
gradle wrapper
```

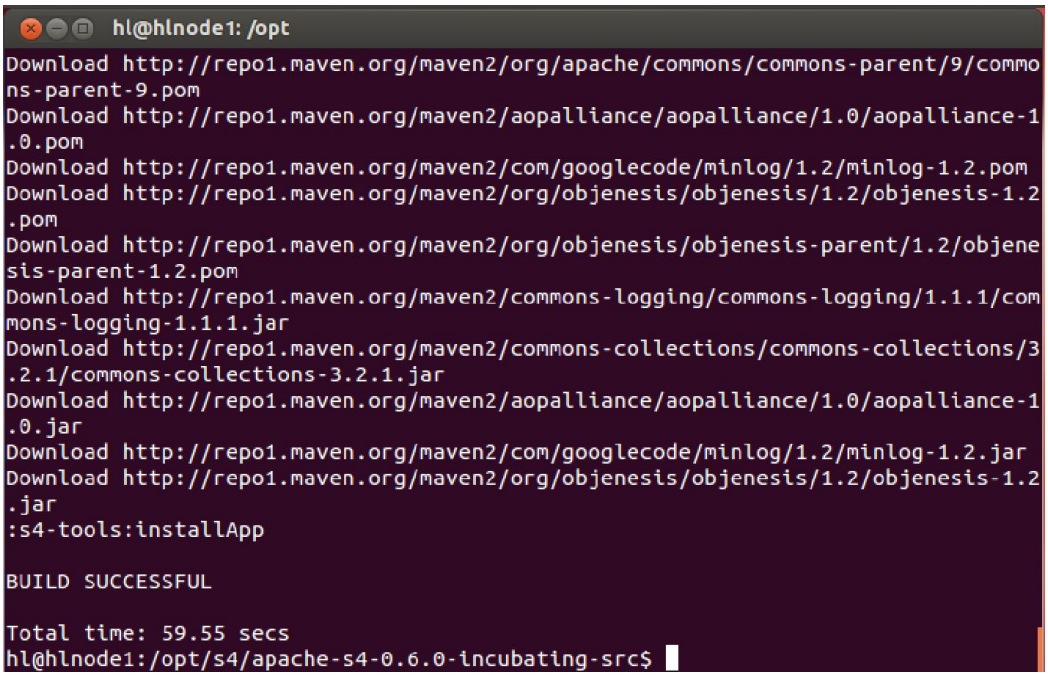
(6) Install the artifacts for Apache S4 by running "`gradle install`" in the S4_HOME directory.

Install the S4-TOOLS, "`gradle s4-tools::installApp`".

(7) Done. Now you can configure and run your Apache S4 cluster.







5.2 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 I use S4 at the first time, I do not know how to stop the app on the cluster. After many searches on Internet, I found the way to do this job.

- **Note**: When you run the wrong app on a cluster, and later you want to stop this app, or re-create the cluster, you need to **delete the corresponding node** in Zookeeper by use the **Zookeeper Client**.

Ubuntu has a **GUI Zookeeper Client** called "**zooinspector**" and a **terminal client** called "zkCli.sh". 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 "`zooinspector`" to enter the zookeeper manager.

(3) When you start the **zookinspector**, you can click the "connect" button in the menu. Then you will see the nodes managed by **zookeeper service**.

(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.

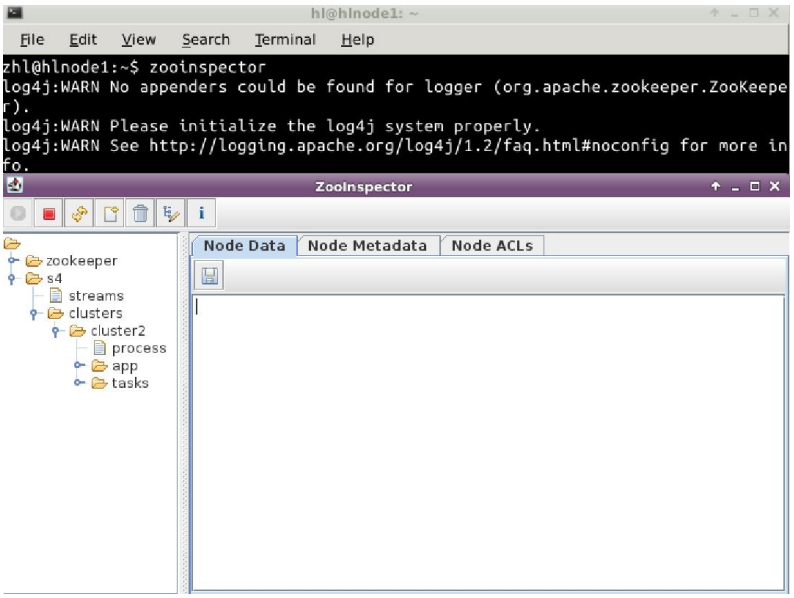
(5) The zookeeper is installed in "**/usr/share/zookeeper**" folder, you can run the "`/usr/share/zookeeper/zkCli.sh`" to manage the nodes in terminal. Please find the commands help of **zkCli** on the Internet. The commonly used ones are:

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



6. Test S4

6.1 HelloApp

The first sample app can be found at:
http://incubator.apache.org/s4/doc/0.6.0/walkthrough/
HelloApp is a very simple example, but I still met some problems when I first time to run it. This example can help you to know the basic procedure of running an app on S4.

Brief steps:

- 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.

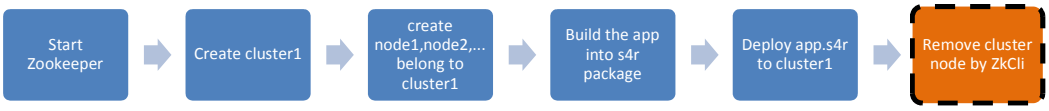


Fig. Process to deploy app into cluster by S4.

Detail process:

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

(1) Create a new project:

```
cd $S4_HOME
./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

======Create 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: (see official document, but it is wrong)

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

- ***Note:*** *In my test, the above way does not work, and an error shows. S4 0.6.0 does not support the "**adapter**" command; You must package the adapter app and deploy it by normal way.*

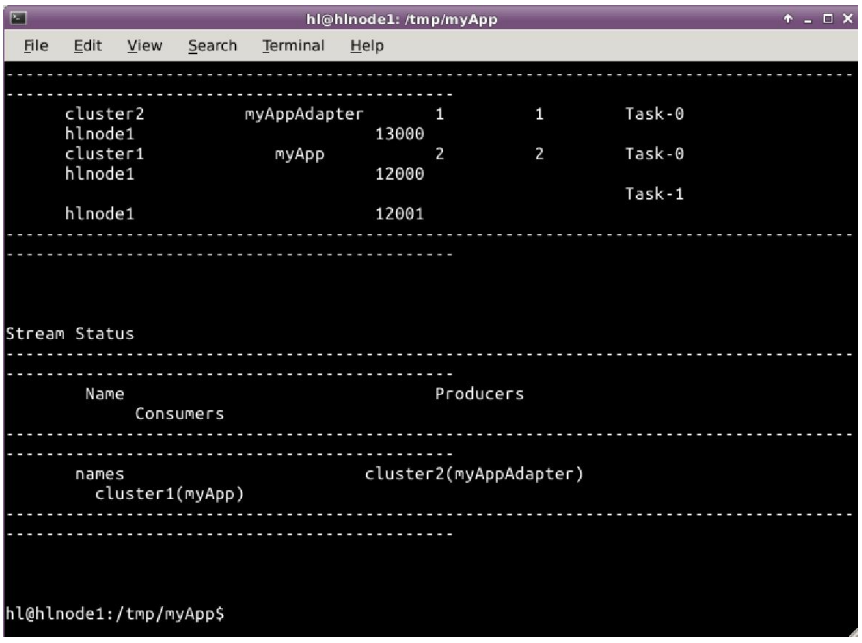
The correct way:

```
./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
```

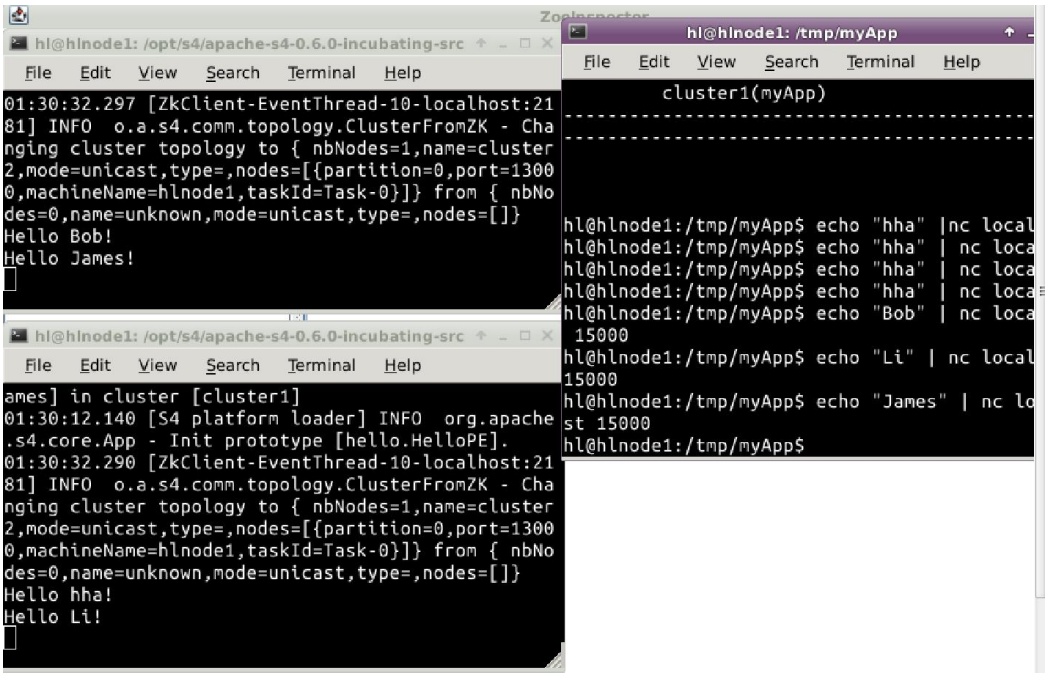


(8) Provide some data to external stream(our adapter is listening to port 15000)

echo "Bob" | nc localhost 15000

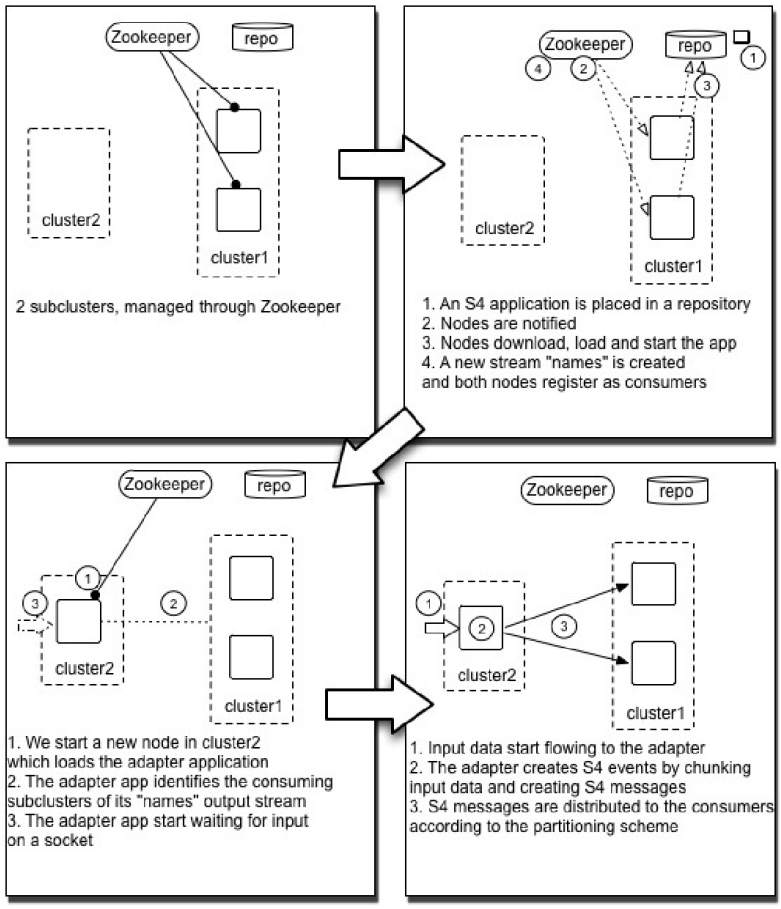
(9) One of the nodes should output in its console:

Hello Bob!



Summary:

The figure below shows what have we done in this example.



6.2 Twitter Trend App

Another example is twitter trend app. It is a little bit complicated. You can skip this example, or try it if you want to know more about S4. This example shows how to collect information from Internet (Twitter website) and process the information in cluster by S4.

The official document can be found at:

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

- ***Note: S4 0.6.0 now has a problem to run this example. To solve this problem, you must apply a patch.***

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:

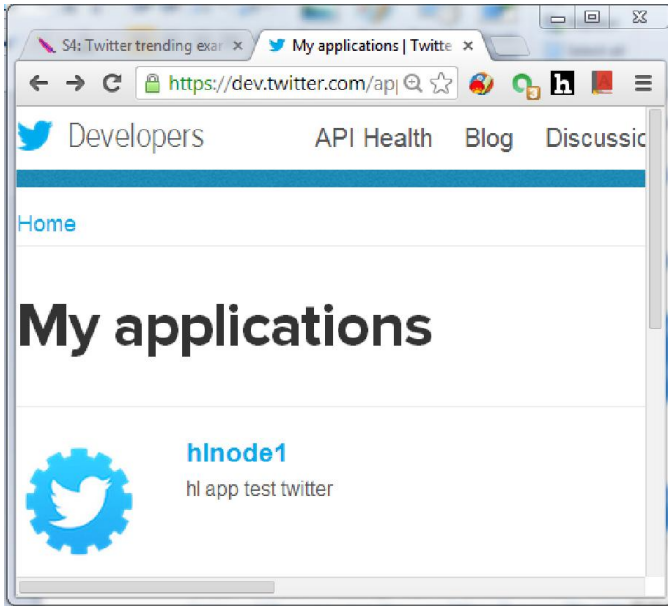
Access token: 2281204758-mncegkx6ovQouC7Kc7knvFhE4TKuJcKlfGeW9QF

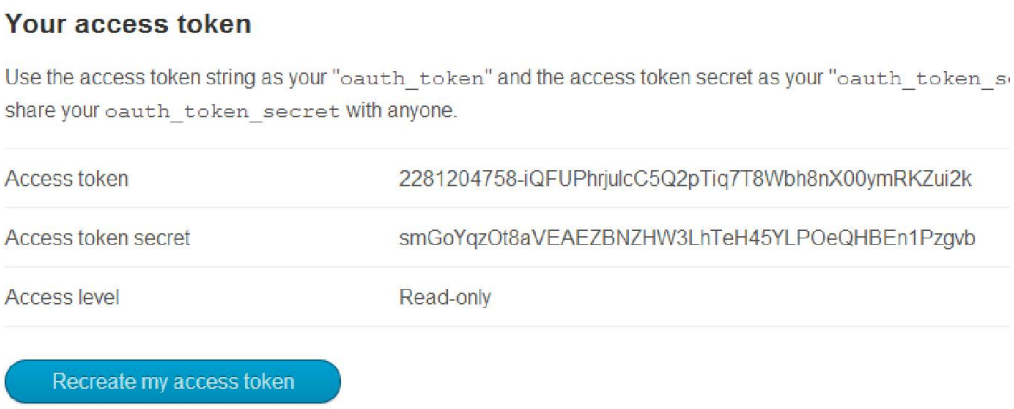
Access token secret: ywh0uHKyPCdNMNBcwlos2xSV3tQLbhI23HcpkwctjKSf0

Access level Read-only

Consumer key: poKIpBh7mYyz4GYohsRRHA

Consumer secret: lPewsgVCvP8kIDZC5wESq8S9RWhHb6FRz6iu5134Stc



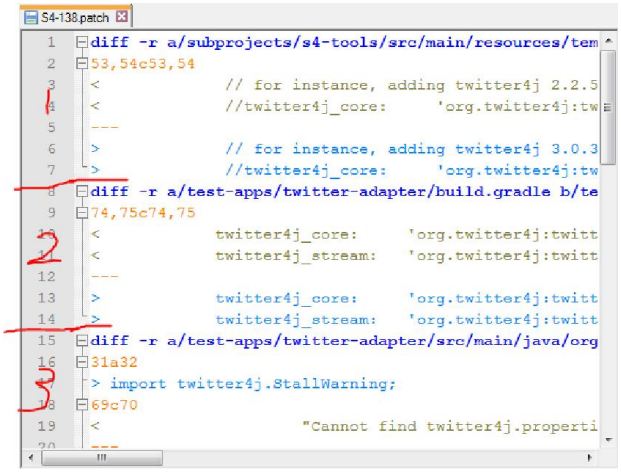


(3) Then update the codes in folders "**twitter-counter**" and **"twitter-adapter**" from your "**$S4_HOME/test-apps**" folder:

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 patch file)
cd $S4_HOME
gedit S4-138.patch          (edit this patch)
```

(4) Then split this file into 4 small patch files--"patch1,patch2,patch3,patch4", each "diff" command a file. Please see the figure below, I use the red line to identify the place to split the file.



Note: Normally you can apply a patch by enter "`patch S4-138.patch`", but I found I found this way cannot work here, 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 patches, 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.

(5) After patch, you need to rebuild S4:

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

(6) 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).

(7) 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-co
 unter.s4r
```

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

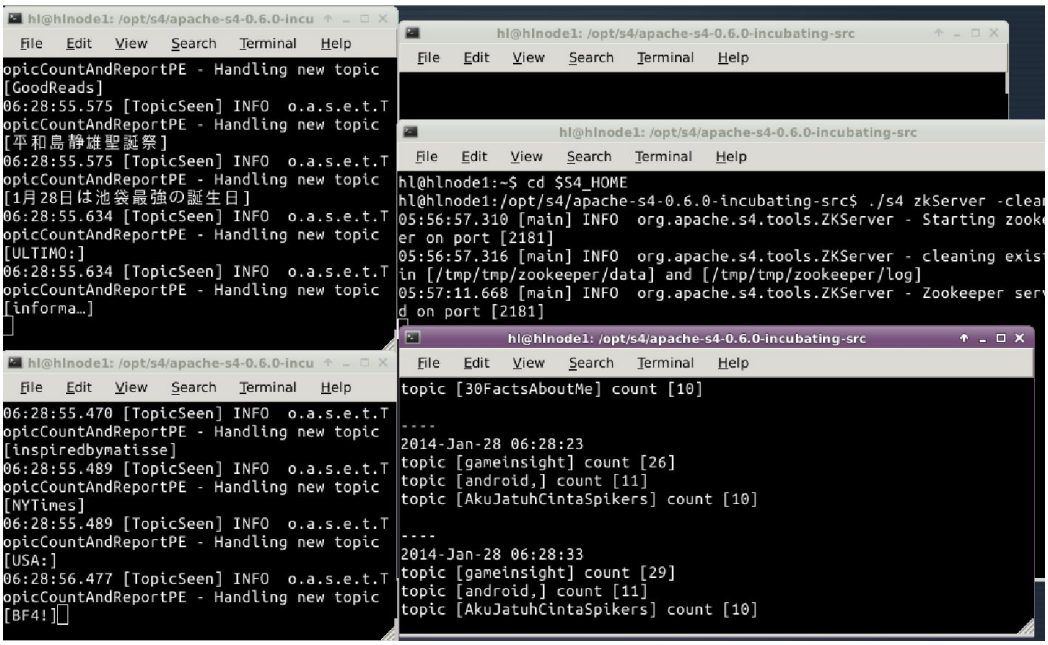
```
./s4 s4r -b=`pwd`/test-apps/twitter-adapter/build.gradle
 -appClass=org.apache.s4.example.twitter.TwitterInputAdapte
 r twitter-adapter

./s4 deploy -appName=twitter-adapter -c=cluster2
-s4r=`pwd`/test-apps/twitter-adapter/build/libs/twitter-ada
pter.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:



Summary

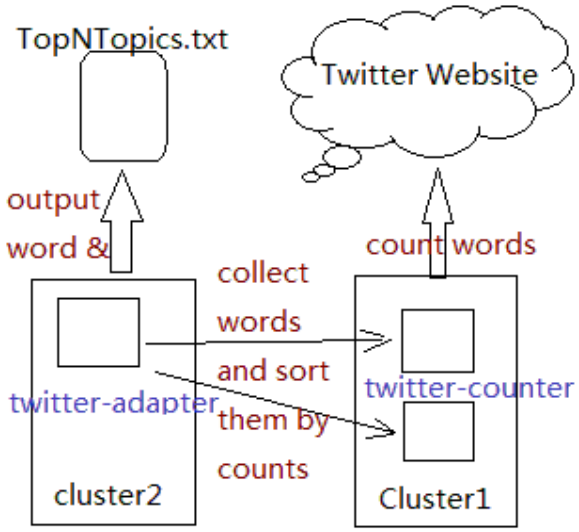


Fig. Twitter-Trend example diagram

6.3 Congratulations

So far, the hardest part of SAMOA ---- S4, has been finished. You 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.

7. 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

- Git GUI clients

Gives a few Git GUI tools, help you to manage the codes clone by Git. I recommend “Smart Git”.
http://git-scm.com/downloads/guis