CLOUD APPLICATION DEVELOPMENT LABORATORY

LAB MANUAL

Course Code	**:**	**ACS110**
Regulations	**:**	**IARE –R18**
Semester	**:**	**VII**
Branch	**:**	**CSE | IT**

Prepared by

POLE ANAJAIAH

Assistant Professor

Department of Computer Science and Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad – 500043

COMPUTER SCIENCE AND ENGINEERING

Program Outcomes	
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and designsystem components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
Program Specific Outcomes (CSE)	
PSO1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer - based systems of varying complexity.
PSO2	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

INDEX

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

S. No	Experiment	Program Outcome Attained	Program Specific Outcomes Attained
1	VIRTUALIZATION : Install Oracle Virtual box and create two VMs on your laptop.	PO1, PO2, PO3	PSO1, PSO2
2	Install Turbo C in guest OS and execute C program.	PO1, PO2, PO3	PSO1, PSO2
3	Test ping command to test the communication between the guest OS and Host OS.	PO1, PO2, PO3	PSO1, PSO2
4	Install Hadoop single node setup.	PO2, PO3	PSO1, PSO2
5	Develop a simple hadoop application called Word Count. It counts the number of occurrences of each word in a given input set.	PO3, PO4	PSO1, PSO2
6	Develop hadoop application to count no of characters, no of words and each character frequency.	PO2, PO3	PSO1, PSO2
7	Develop hadoop application to process given data and produce results such as finding the year of maximum usage, year of minimum usage.	PO2, PO3	PSO1, PSO2
8	HADOOP Develop hadoop application to process given data and produce results such as how many female and male students in both schools the results should be in following format. GP-F #number GP-M #numbers MS-F #number MS-M #number	PO2, PO3	PSO1, PSO2
9	Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.	PO3, PO4	PSO1, PSO2
10	Design a protocol and use Simple Queue Service(SQS)to implement the barrier synchronization after the first phase.	PO3, PO4	PSO1, PSO2
11	Use the Zookeeper to implement the coordination model in Problem 10.	PO2, PO3	PSO1, PSO2
12	CLOUD PROGRAMMING :Develop a Hello World application using Google App Engine.	PO2, PO3	PSO1, PSO2
13	Develop a Guestbook Application using Google App Engine.	PO2, PO3	PSO1, PSO2
14	Develop a Windows Azure Hello World application using.	PO2, PO3	PSO1, PSO2
15	PIPES Create a Mashup using Yahoo! Pipes.	PO2, PO3	PSO1, PSO2

CLOUD APPLICATION DEVELOPMENT LABORATORY

OBJECTIVE:

The objective of cloud computing lab is to learn the cloud architecture and its efficiency, and tools to provide virtualization on cloud. The lab enables the study and implementation of infrastructure as a service, storage as a service, and user management on cloud. Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Cloud computing services usually provide common business applications on-line that are accessed from a web browser, while the software and data are stored on the servers. It is expected that Cloud Computing will help in pooling of computing resources of Government Departments into large clouds thereby increasing utilization of computing resources effectively. Besides, the self-service nature of cloud computing allows organizations to create elastic environments that expand and contract; based on the workload and target performance parameters.

OUTCOMES:

Students will be able to:

1. Define & implement Virtualization using different types of Hypervisors
2. Describe steps to perform on demand Application delivery using Ulteo .
3. Examine the installation and configuration of Open stack cloud
4. Analyze and understand the functioning of different components involved in Amazon Web services cloud platform.
5. Describe the functioning of Platform as a Service 6. Design & Synthesize Storage as a service using own Cloud

EXPERIMENT - 1

VIRTUALIZATION: Install Oracle Virtual box and create two VMs on your laptop

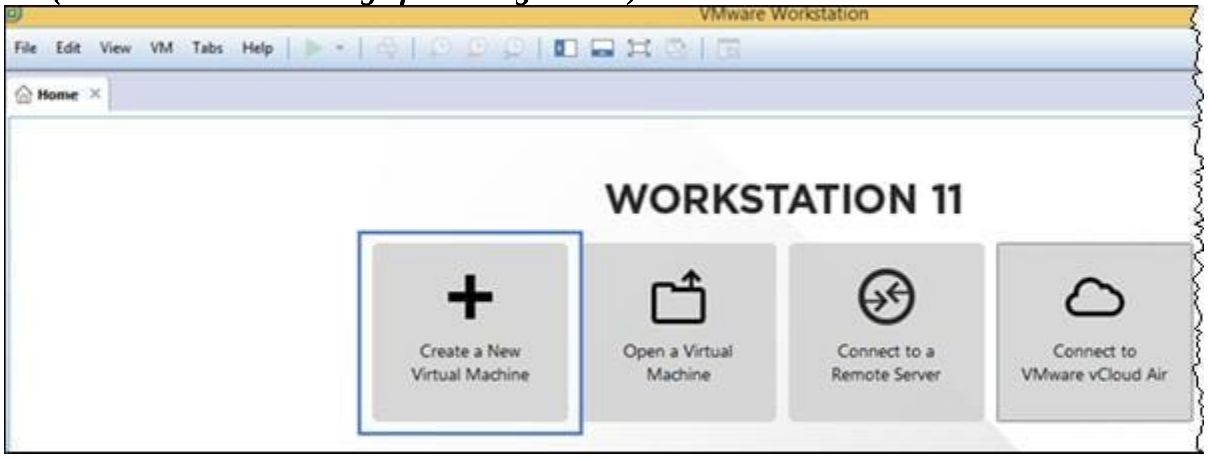
Points to Consider before installing Virtual Machine (VM):

1. Decide which applications you are going to install on your virtual machine. According to that install 32 or 64-bit Operating system in VM. Some applications are not compatible with old operating systems. E.g. If you are going to use UFT 12.01 it doesn't work with Windows XP. So you should install higher version of Windows to work with UFT. Check OS compatibility of your required application before proceeding with operating system installation.
2. Is your Processor supports Virtualization? Almost all of the new processors support virtualization but it is a good idea to check manufacturer's website to know the details. E.g. Intel Pentium Dual Core T2390 processor doesn't support virtualization for a 64-bit operating system. So, in that case, you should install 32-bit operating system compulsory.
3. VMWare also provides a trial period. So you should have a key or serial number to keep using it further.
4. We are going to use VMWare Virtualization software for demonstration.

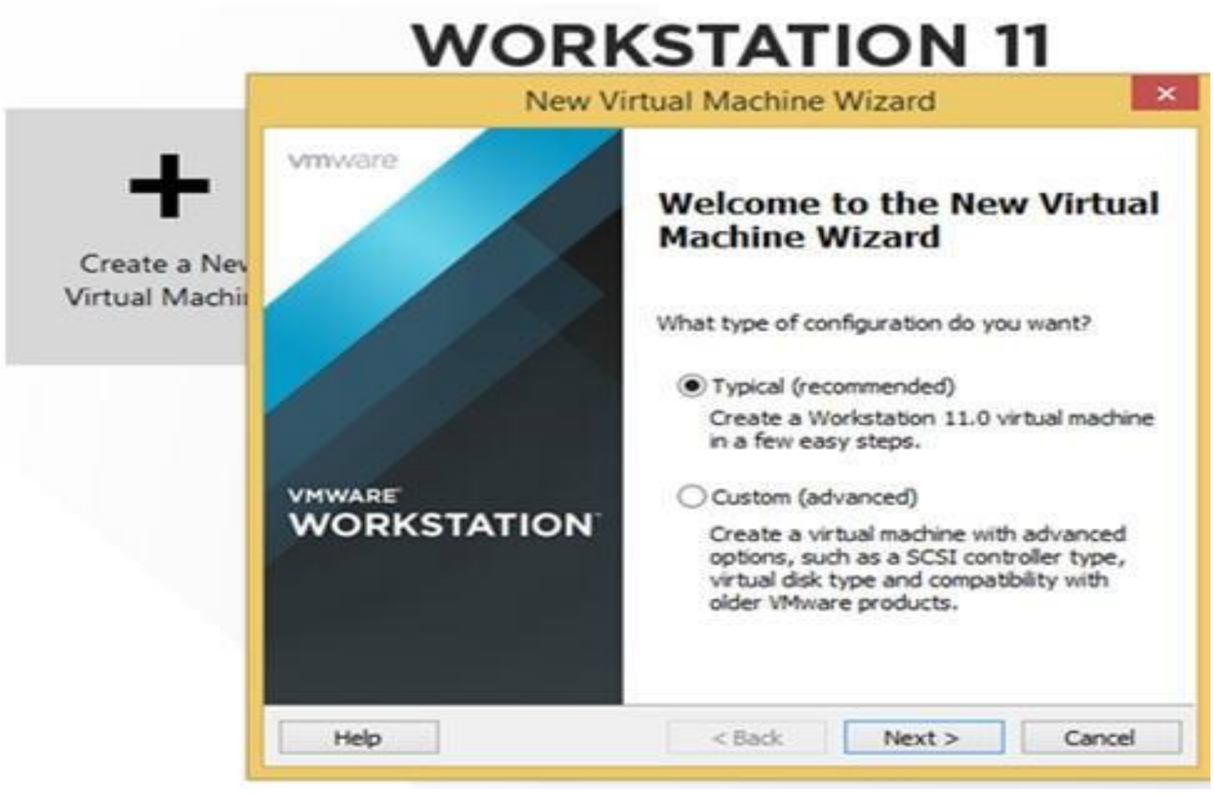
Steps to install and configure VMWare:

1. Download VMWare workstation trial version setup file *from here*. Set up is around 307 MB. Currently, version 12 is available. Please note we have set up screens on version 11.
2. Install VMWare on your machine. Setup is simple and requires to click Next button couple of times.
3. After installation open VMWare workstation by using either start menu or shortcut created on the desktop.
4. Click on “Create a New Virtual Machine”.

(Note: Click on the image for enlarged view)


VMware Workstation
File Edit View VM Tabs Help
Home
WORKSTATION 11
Create a New Virtual Machine
Open a Virtual Machine
Connect to a Remote Server
Connect to VMware vCloud Air

5. With default “Typical” selected click on Next button.



New Virtual Machine Wizard

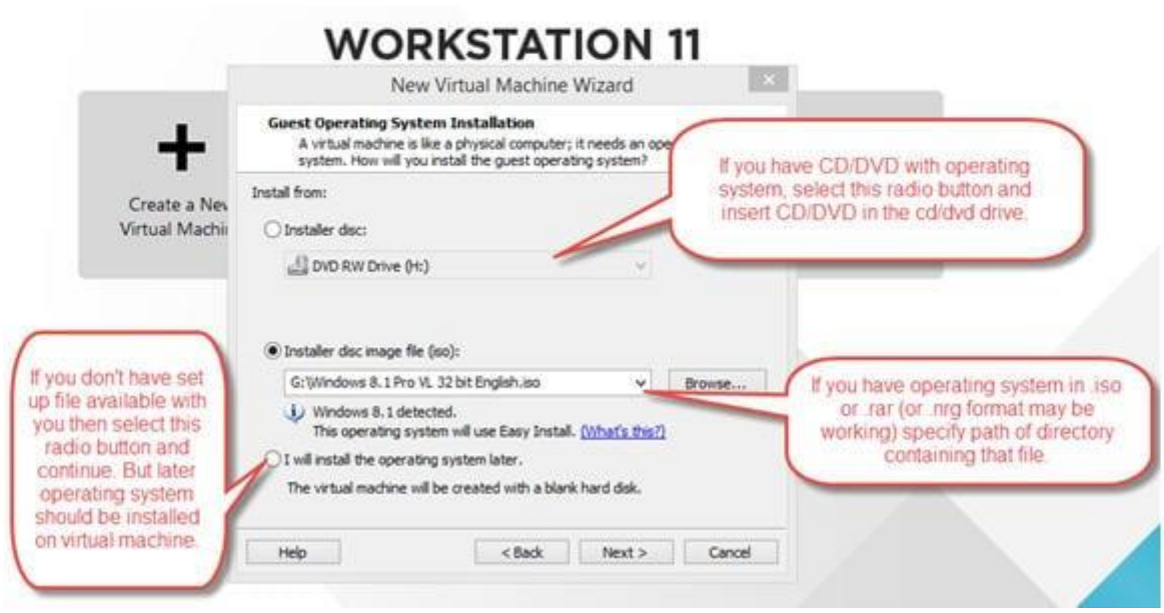
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

[✓ Typical (recommended)]
Create a Workstation 11.0 virtual machine in a few easy steps.

Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

6. Specify the path of the operating system set up file.


WORKSTATION 11

New Virtual Machine Wizard

Guest Operating System Installation
A virtual machine is like a physical computer; it needs an ope system. How will you install the guest operating system?

Install from:

○ Installer disc:
DVD RW Drive (H:)

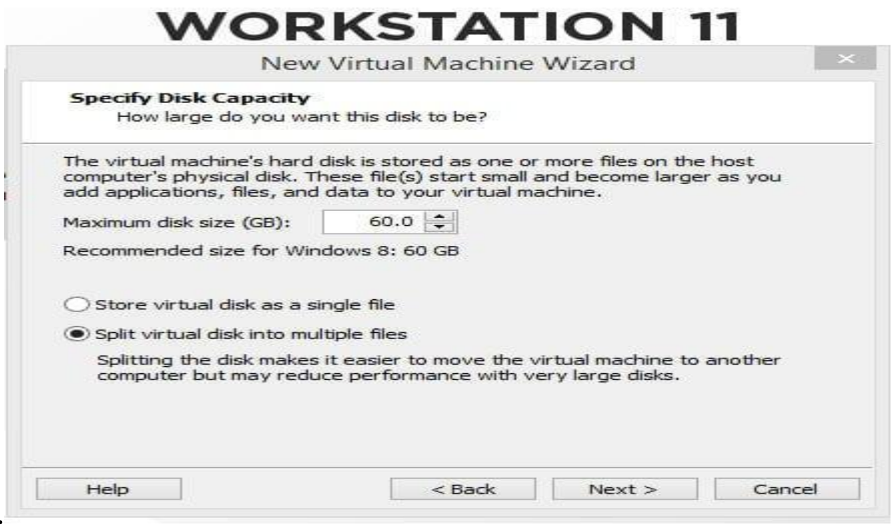
● Installer disc image file (iso):
G:\Windows 8.1 Pro VL 32 bit English.iso
Windows 8.1 detected.
This operating system will use Easy Install. (What's this?)

○ I will install the operating system later.
The virtual machine will be created with a blank hard disk.

Create a New Virtual Machine
Browse...
Help
< Back
Next >
Cancel
If you have CD/DVD with operating system, select this radio button and insert CD/DVD in the cd/dvd drive.
If you have operating system in .iso or .rar (or .nrg format may be working) specify path of directory containing that file
If you don't have set up file available with you then select this radio button and continue. But later operating system should be installed on virtual machine.

7. In the Next step you need to specify a Key or a serial number of operating system. If you are using trial version then that part can be skipped.

8. Enter the name for the virtual machine and specify a path to the directory where you want to create your virtual machine. It is recommended that the drive you're selecting to install virtual machine should have sufficient space.
9. Specify an amount of disk space you want to allocate for a virtual machine. Allocate disk space according to the size of software you are going to install on the virtual machine


WORKSTATION 11

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.

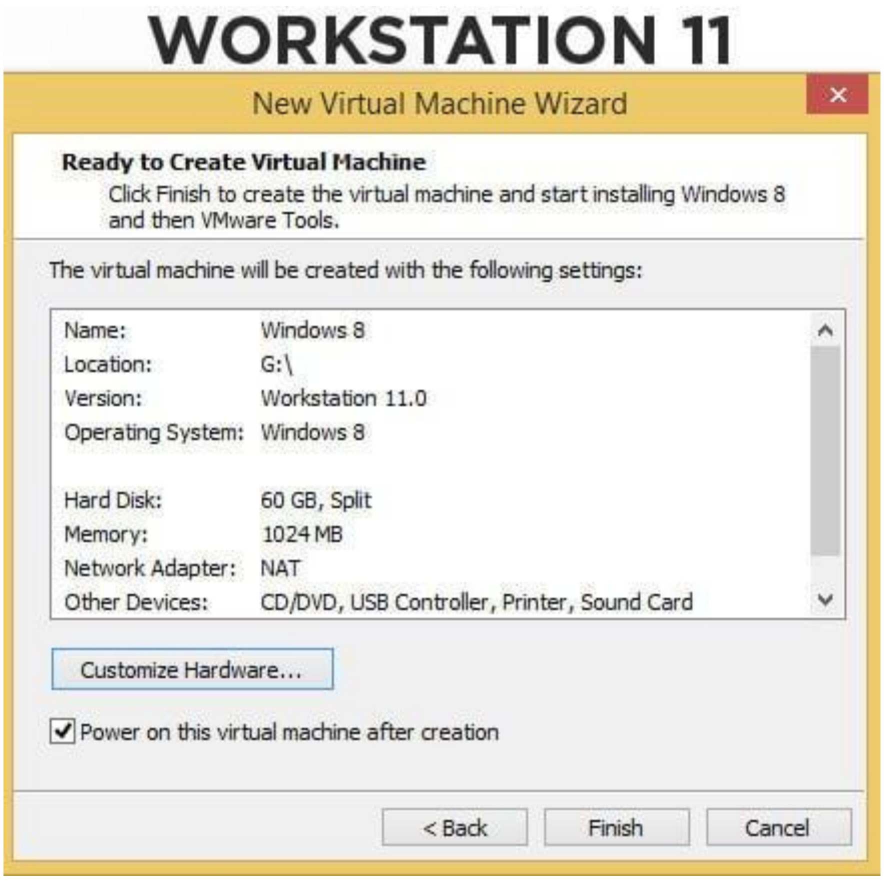
Maximum disk size (GB): 60.0

Recommended size for Windows 8: 60 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.

.

10. On the next screen it will show configuration you selected for a virtual machine.



WORKSTATION 11

New Virtual Machine Wizard

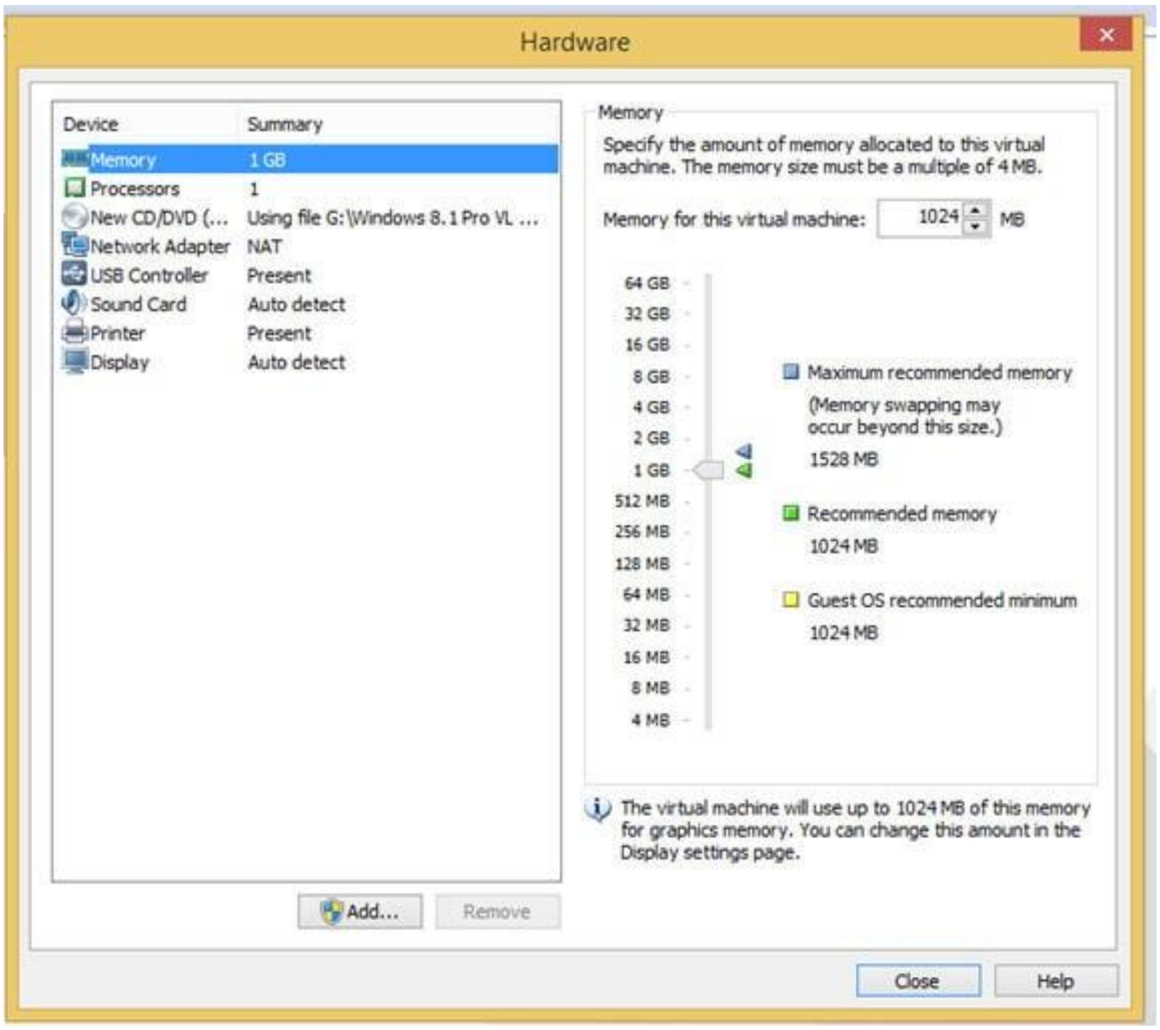
Ready to Create Virtual Machine

Click Finish to create the virtual machine and start installing Windows 8 and then VMware Tools.

The virtual machine will be created with the following settings:

Name:	Windows 8
Location:	G:\
Version:	Workstation 11.0
Operating System:	Windows 8
Hard Disk:	60 GB, Split
Memory:	1024 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

11. It will allocate Hardware according to the default settings but you can change it by using Customize Hardware button in the above screen.
    You can specify what amount of RAM, a processor has to be allocated for a virtual machine. Do not allocate complete RAM or complete Processor for a virtual machine. Also, do not allocate very less RAM or processor. Leave default settings or allocate in such way that your application should be able to run on the virtual machine. Else it will result in a slow virtual machine.


Hardware

Device	Summary
Memory	1 GB
Processors	1
New CD/DVD (...	Using file G:\Windows 8.1 Pro VL ...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 1024 MB

Maximum recommended memory
(Memory swapping may occur beyond this size.)
1528 MB

Recommended memory
1024 MB

Guest OS recommended minimum
1024 MB

The virtual machine will use up to 1024 MB of this memory for graphics memory. You can change this amount in the Display settings page.

12. Click on the Finish button to create the virtual machine at the specified location and with specified resources.

If you have specified a valid file (.iso, .rar., .nrg) for the operating system it will take standard time to complete operating system set up on the virtual machine and then it will be ready to use your regular OS.

Notes:

- If you didn't specify any operating system while creating the virtual machine, later you can install it just like we do for your laptop or desktop machines. We can use CD/DVD or USB devices like Pen Drive or even set up a file on the disk to install the operating system in the VM.

- If your CD/DVD drive is not working then also it is very simple to install the operating system. Go to VM -> Settings – > select CD/DVD -> in the right half select radio button for ‘use ISO image from' and specify the path on your hard disk where the .iso file is placed. This location will be treated as CD/DVD drive of your machine.

- Make sure correct boot order is specified in BIOS so installation will start while getting VM power on (in this case guest OS is not installed).

Passing data between host and VM:
Generally, VM is having its own drive and it is not showing drives from host OS in the VM environment. Also, VM drive cannot be used from host OS

PRE LAB VIVA QUESTIONS ?

1. What is meant by virtualization?
2. What does virtual machine do?
3. What is the role of VM Ware in virtual machine?
4. Is 32 bit processor supports Virtual Machine?

POST LAB VIVA QUESTIONS:

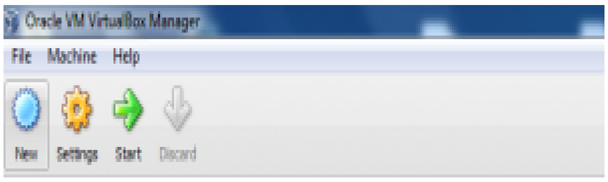
1. Explain the steps to install oracle virtual box?
2. How to create two VMs on your laptop?
3. How to specify an amount of disk space to allocate for a virtual machine?
4. Specify the full names of (.iso, .rar, .nrg) files?

EXPERIMENT – 2

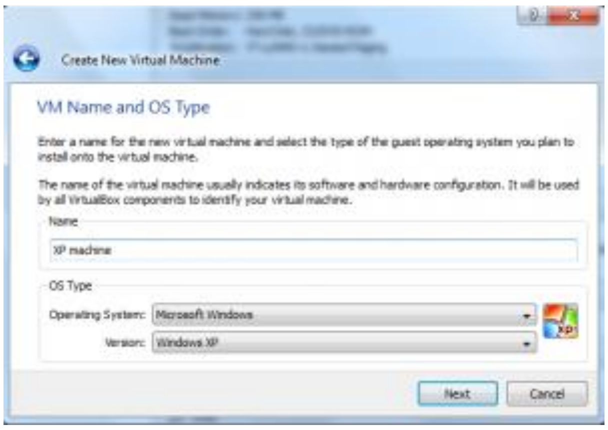
Install Turbo C in guest OS and execute C program

Step1: Create a Virtual Machine

First job is to create a virtual machine, to do so open VirtualBox and click "New" from the toolbar.



- ➢ Select "New" from the toolbar to create a new Virtual Machine
- ➢ Enter any name for your Virtual Machine and select the Operating System you're going to use as your GUEST OS.



Create New Virtual Machine

VM Name and OS Type

Enter a name for the new virtual machine and select the type of the guest operating system you plan to install onto the virtual machine.

The name of the virtual machine usually indicates its software and hardware configuration. It will be used by all VirtualBox components to identify your virtual machine.

Name

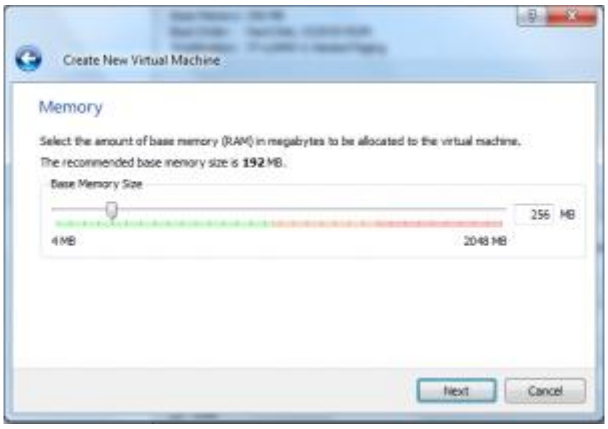
XP machine

OS Type

Operating System: Microsoft Windows

Version: Windows XP

- ➢ Select the OS you are going to use inside your Virtual Machine
- ➢ Select the RAM size to be allocated to your guest OS, choose this wisely it should be equal to or more than the minimum system requirement of your guest OS at the same time if it goes more than 50% of your physical machine's RAM it'll slow down your host OS.


Create New Virtual Machine

Memory

Select the amount of base memory (RAM) in megabytes to be allocated to the virtual machine.

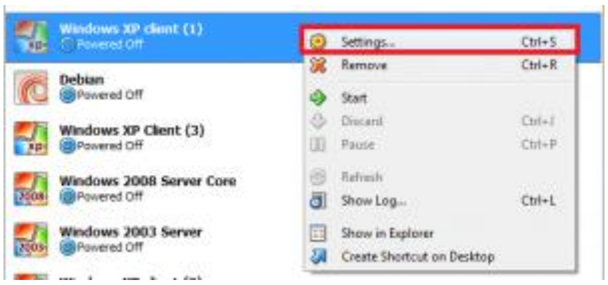
The recommended base memory size is **192** MB.

Base Memory Size: 256 MB

- Select the amount of RAM to be allocated to the Virtual Machine
  You need to create a Virtual Hard disk for your Virtual Machine, this is just a file with a .vdi extension which will contain all files stored inside that virtual machine. Choose a size suited for your guest OS and select the “Dynamically expanding disk” if you want to save disk creation time and file size. If you choose “Fixed size disk” it will take up the entire size specified but the performance of your Virtual Machine will be better. After clicking finish move on to the next step.

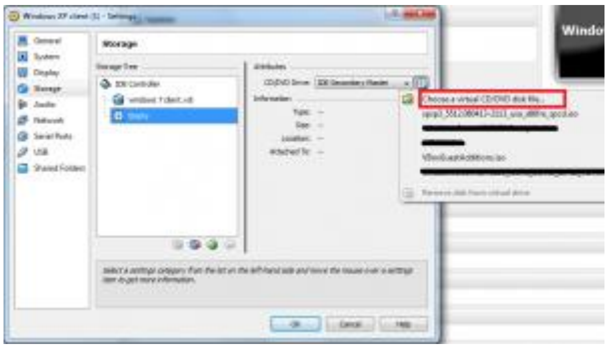
Step 2: Change the boot order of the VM

Right click the newly created Virtual machine and go to settings.

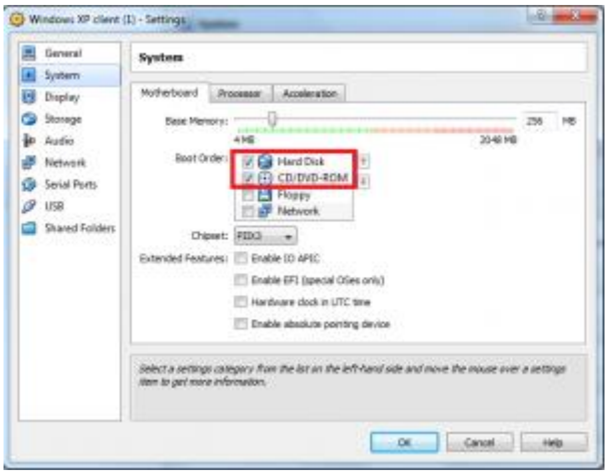


Go to the settings of your Virtual Machine

- Select storage from the left side list, select the CD icon and from the right select “Choose a virtual CD/DVD disk file”, navigate to the ISO image file of the OS on your computer. Now the OS image is mounted to your Virtual machine.



The ISO image selected will be mounted in the Optical drive of your Virtual Machine To change the boot device order of your virtual machine go to the “System” option from the left side list, select Hard Disk and click the up arrow to bring it to the top of the list. Make the CD/DVD-ROM the second device and uncheck the other devices.



Base Memory: 256 MB

Boot Order: ☑ Hard Disk, ☑ CD/DVD-ROM, ☐ Floppy, ☐ Network

Chipset: PIIX3

Extended Features: ☐ Enable IO APIC ☐ Enable EFI (special OSes only) ☐ Hardware clock in UTC time ☐ Enable absolute pointing device

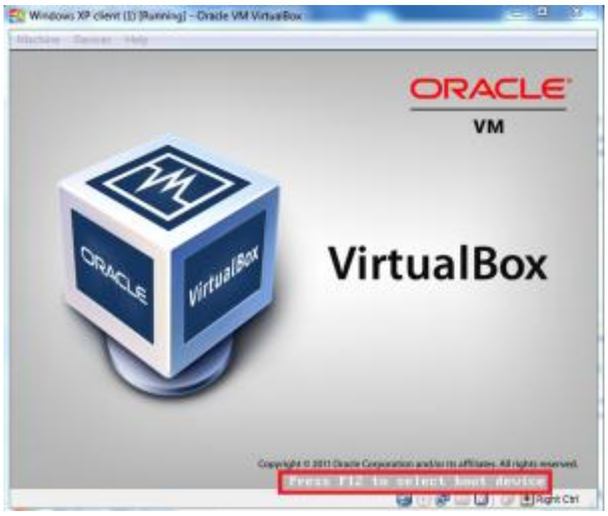
Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.

Set the boot device priority for the virtual machine

Step 3: Start the Virtual machine

To start the virtual machine double-click the VM, a window opens here you can

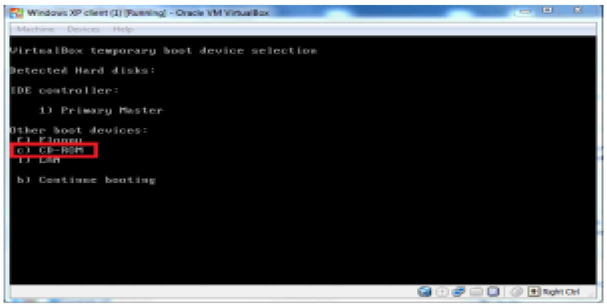
press F12 if you want to select a device of your choice to boot.


Windows XP client (1) [Running] - Oracle VM VirtualBox
Machine Devices Help
ORACLE VM
VirtualBox
Copyright © 2011 Oracle Corporation and/or its affiliates. All rights reserved.
Right Ctrl

```
Press F12 to select boot device
```

Press F12 to select the boot device of your choice

To boot from the CD-ROM press c.



```
VirtualBox temporary boot device selection

Detected Hard disks:

IDE controller:

    1) Primary Master

Other boot devices:
 f) Floppy
 c) CD-ROM
 l) LAN

 b) Continue booting
```

Press c to boot the VM from a CD image
Your VM will now boot from your Operating System image file and the OS installation will start as usual.Some important shortcuts
The following keyboard shortcuts can be used inside the VM to perform certain actions that might canflict with your physical machine

[Right Ctrl] + Del – Equivalent to pressing [ctrl] + [alt] + [delete] inside the Virtual Machine. If you press ctrl+alt+del the physical machine will also be affected.

[Right Ctrl] + H – (Halt) Equivalent to pressing the power button on the physical machine.

[Right Ctrl] + R – (Reset) Equivalent to pressing the reset button, will reset your virtual machine.

[Right CTRL] + F – (Fullscreen) Toggles fullscreen

PRE LAB VIVA QUESTIONS ?

1. What is meant guest OS?
2. What is meant by host OS?
3. Why the RAM size to be allocated to your guest OS?
4. Why to provide connectivity between guest OS and host OS?

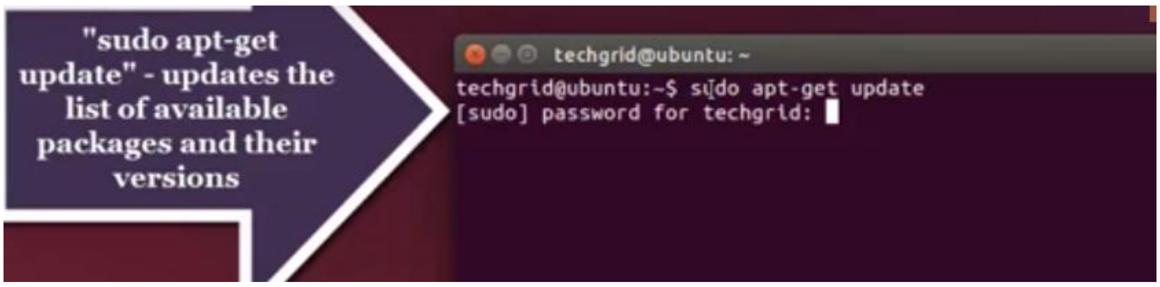
POST LAB VIVA QUESTIONS:

1. If guest OS takes the 50% of your physical machine’s RAM size what happens?
2. Why to create a Virtual Hard disk for your Virtual Machine?
3. What does .vdi extension file contains inside the virtual machine?
4. Define the “ Dynamically expanding disk”

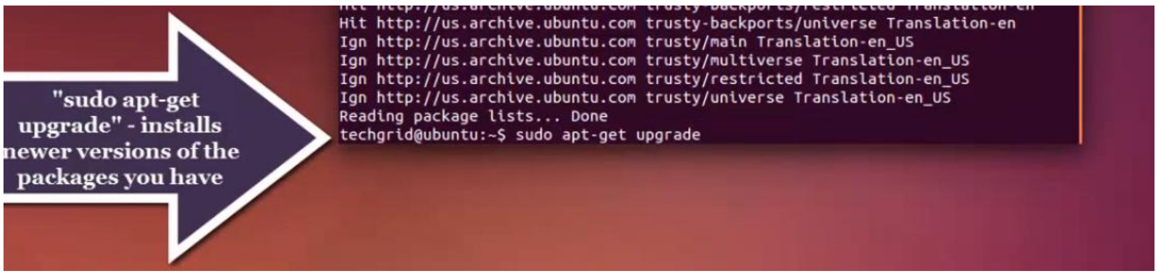
EXPERIMENT – 4

INSTALL HADOOP SINGLE NODE SETUP

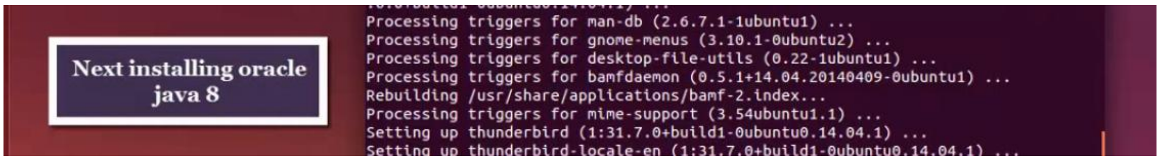
SPTEPS:


"sudo apt-get update" - updates the list of available packages and their versions
techgrid@ubuntu: ~

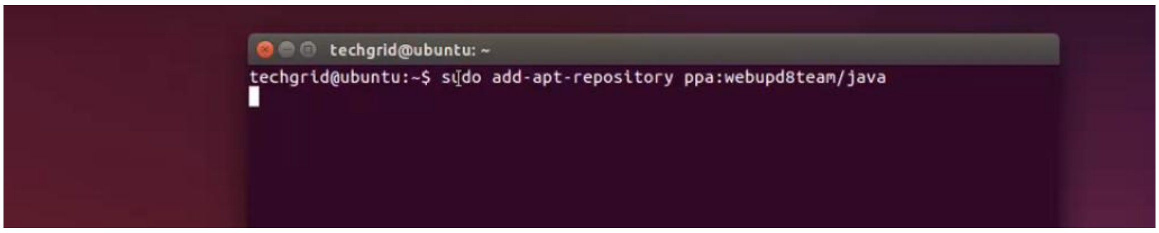
```
techgrid@ubuntu:~$ sudo apt-get update
[sudo] password for techgrid:
```


"sudo apt-get upgrade" - installs newer versions of the packages you have

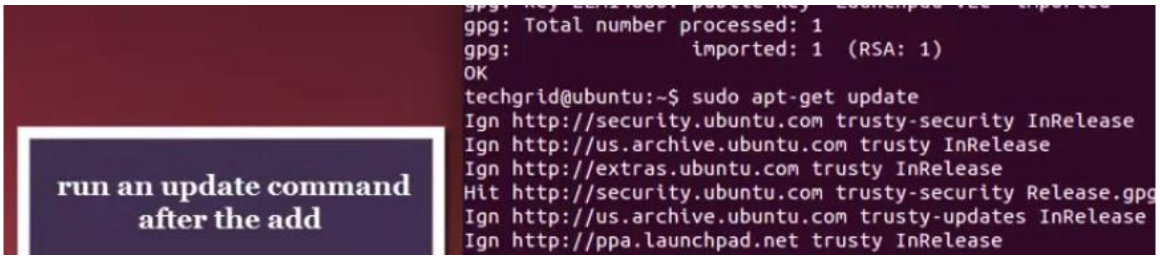
```
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Reading package lists... Done
techgrid@ubuntu:~$ sudo apt-get upgrade
```



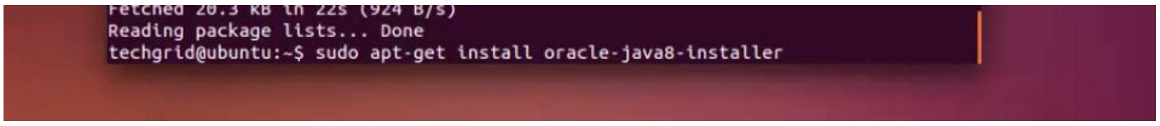
```
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for gnome-menus (3.10.1-0ubuntu2) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu1) ...
Processing triggers for bamfdaemon (0.5.1+14.04.20140409-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.54ubuntu1.1) ...
Setting up thunderbird (1:31.7.0+build1-0ubuntu0.14.04.1) ...
Setting up thunderbird-locale-en (1:31.7.0+build1-0ubuntu0.14.04.1) ...
```



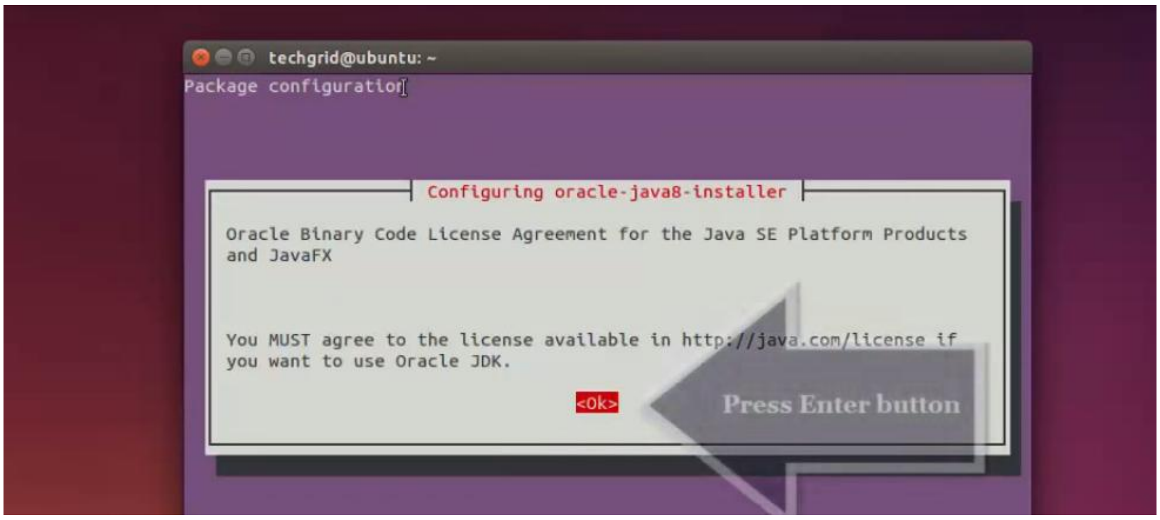
```
techgrid@ubuntu:~$ sudo add-apt-repository ppa:webupd8team/java
```



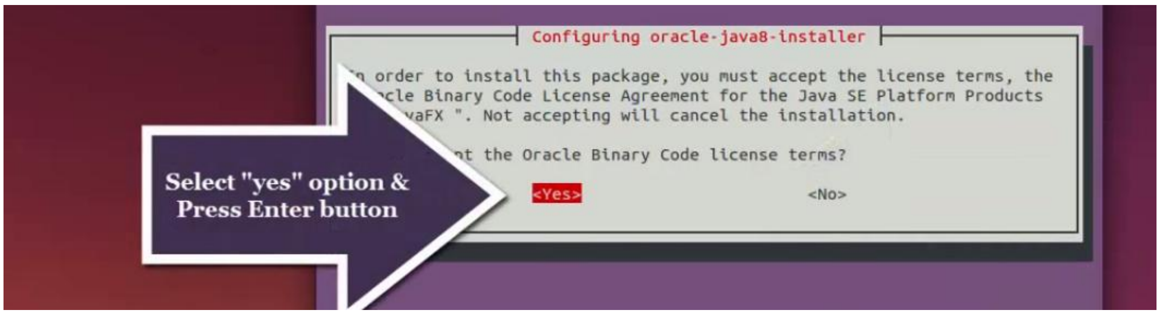
```
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
OK
techgrid@ubuntu:~$ sudo apt-get update
Ign http://security.ubuntu.com trusty-security InRelease
Ign http://us.archive.ubuntu.com trusty InRelease
Ign http://extras.ubuntu.com trusty InRelease
Hit http://security.ubuntu.com trusty-security Release.gpg
Ign http://us.archive.ubuntu.com trusty-updates InRelease
Ign http://ppa.launchpad.net trusty InRelease
```



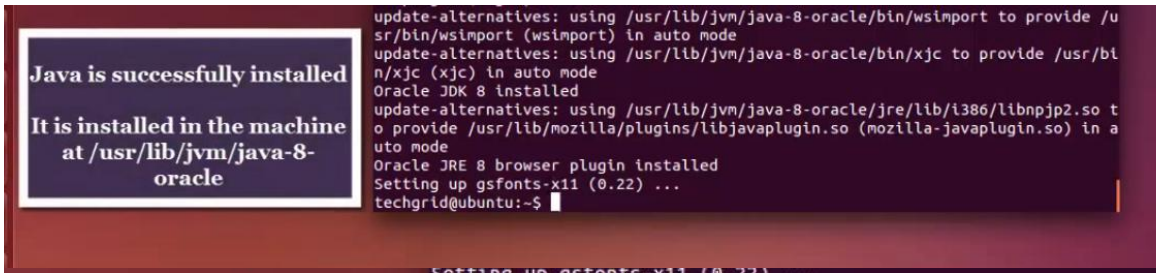
```
Fetched 20.3 kB in 22s (924 B/s)
Reading package lists... Done
techgrid@ubuntu:~$ sudo apt-get install oracle-java8-installer
```



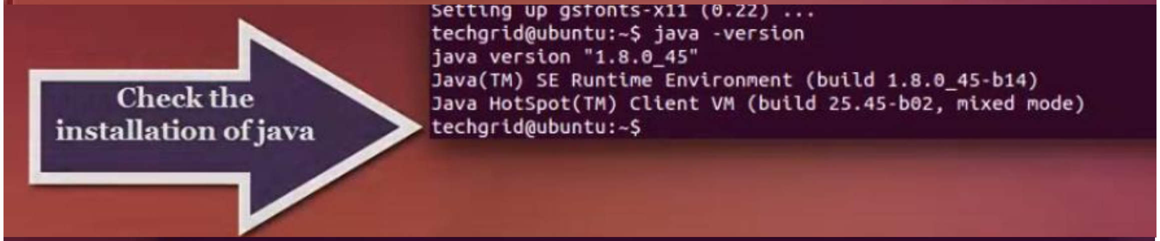
```
Package configuration

Configuring oracle-java8-installer

Oracle Binary Code License Agreement for the Java SE Platform Products
and JavaFX

You MUST agree to the license available in http://java.com/license if
you want to use Oracle JDK.

<Ok>
```



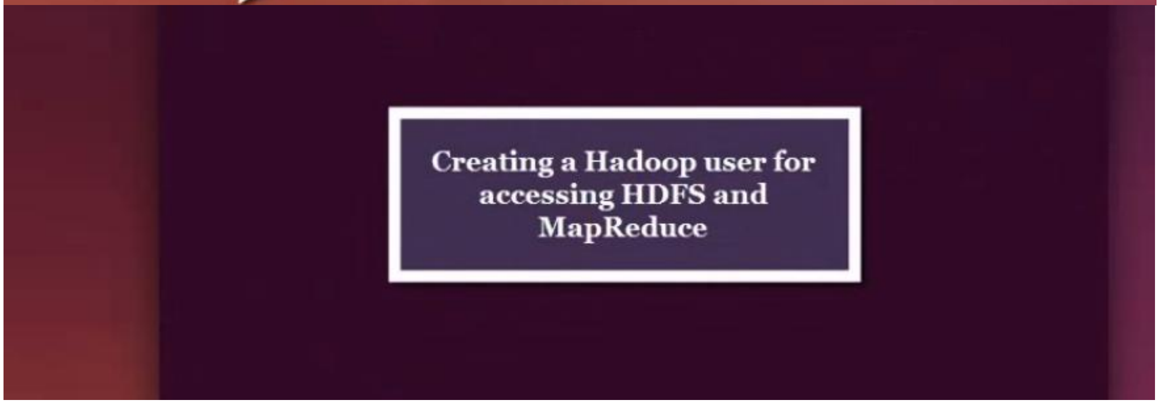
```
Configuring oracle-java8-installer

n order to install this package, you must accept the license terms, the
cle Binary Code License Agreement for the Java SE Platform Products
vaFX ". Not accepting will cancel the installation.

nt the Oracle Binary Code license terms?

<Yes>                              <No>
```

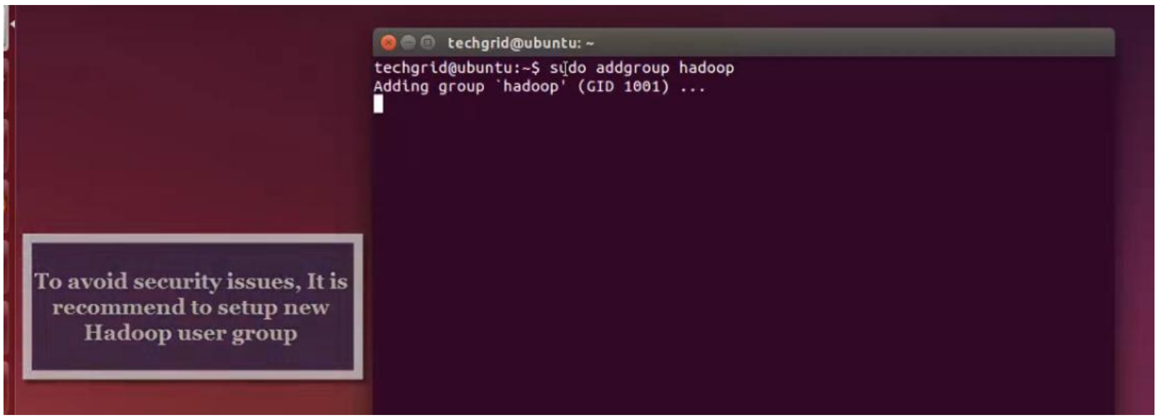

Java is successfully installed
It is installed in the machine at /usr/lib/jvm/java-8-oracle

```
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/wsimport to provide /u
sr/bin/wsimport (wsimport) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/xjc to provide /usr/bi
n/xjc (xjc) in auto mode
Oracle JDK 8 installed
update-alternatives: using /usr/lib/jvm/java-8-oracle/jre/lib/i386/libnpjp2.so t
o provide /usr/lib/mozilla/plugins/libjavaplugin.so (mozilla-javaplugin.so) in a
uto mode
Oracle JRE 8 browser plugin installed
Setting up gsfonts-x11 (0.22) ...
techgrid@ubuntu:~$
```

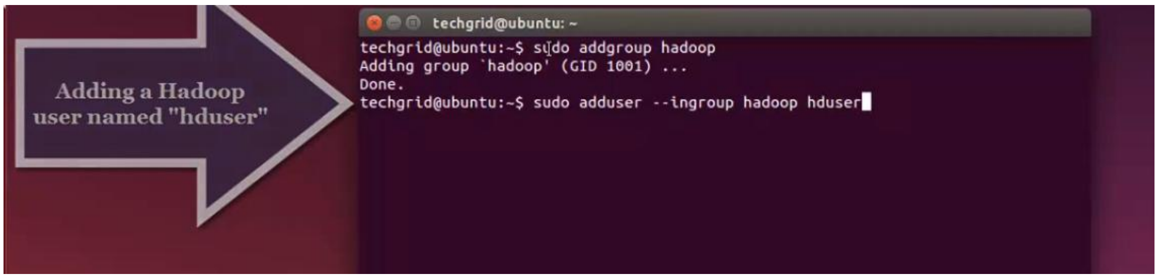


```
Setting up gsfonts-x11 (0.22) ...
techgrid@ubuntu:~$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) Client VM (build 25.45-b02, mixed mode)
techgrid@ubuntu:~$
```



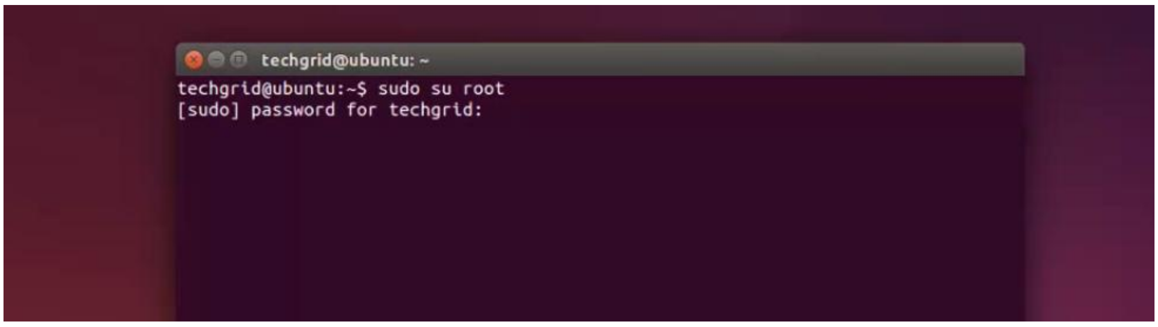

To avoid security issues, It is recommend to setup new Hadoop user group

```
techgrid@ubuntu:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
```

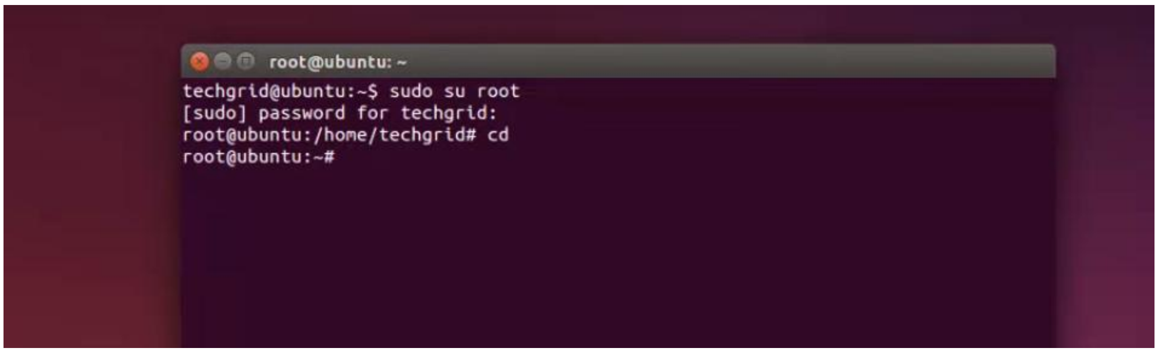


```
techgrid@ubuntu:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
techgrid@ubuntu:~$ sudo adduser --ingroup hadoop hduser
```

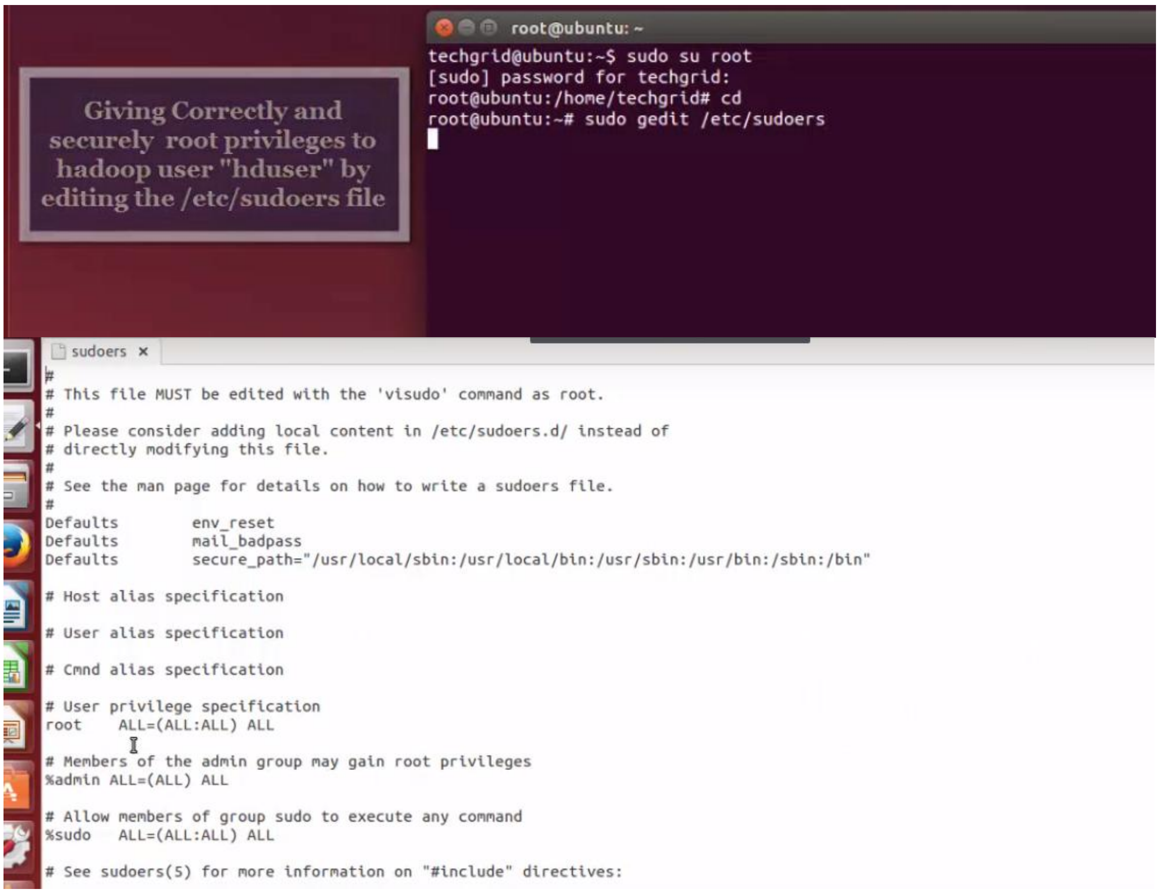
In new Terminal:



```
techgrid@ubuntu:~$ sudo su root
[sudo] password for techgrid:
```



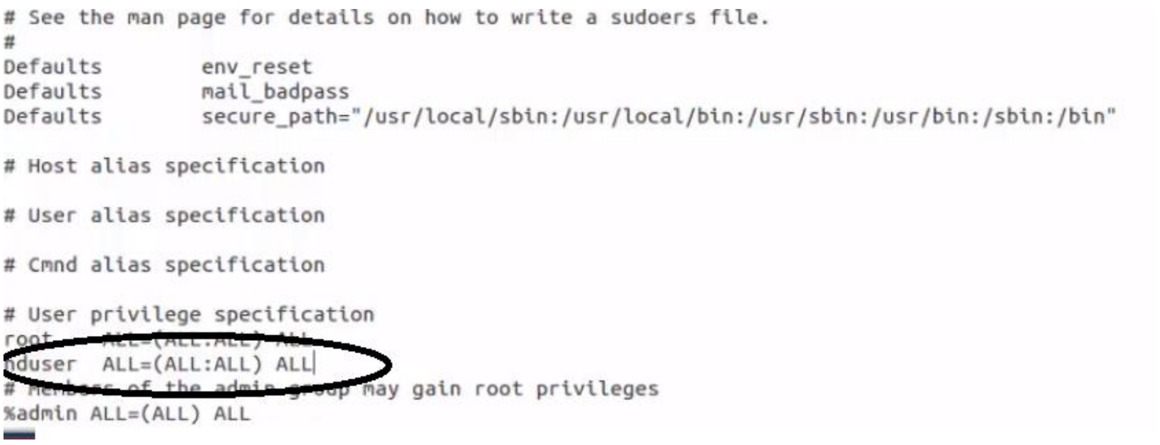
```
techgrid@ubuntu:~$ sudo su root
[sudo] password for techgrid:
root@ubuntu:/home/techgrid# cd
root@ubuntu:~#
```


Giving Correctly and securely root privileges to hadoop user "hduser" by editing the /etc/sudoers file
root@ubuntu: ~
sudoers

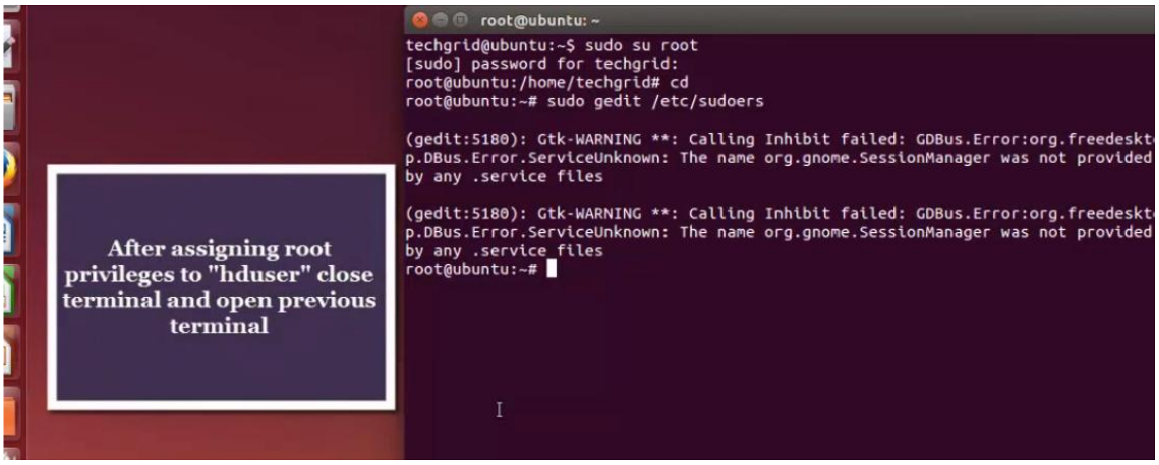
```
techgrid@ubuntu:~$ sudo su root
[sudo] password for techgrid:
root@ubuntu:/home/techgrid# cd
root@ubuntu:~# sudo gedit /etc/sudoers
```

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
```

You can add the user above

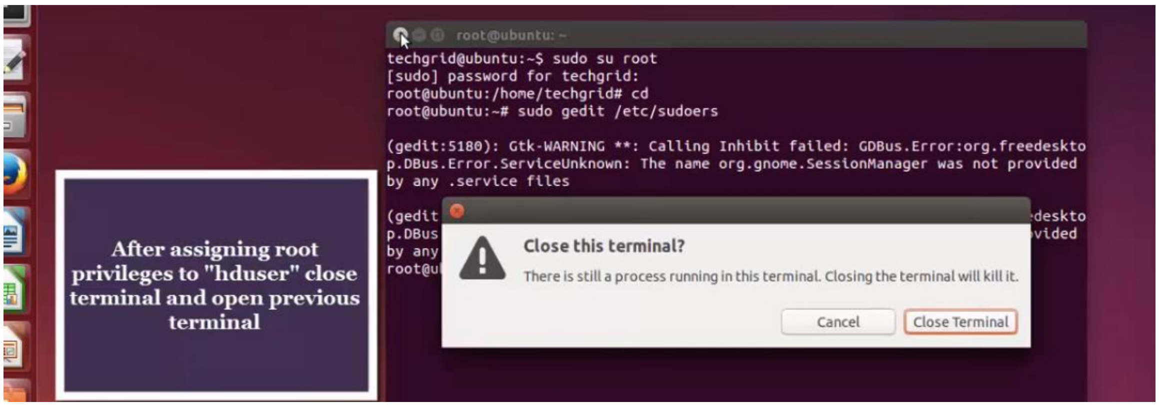


```
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
hduser  ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```



After assigning root privileges to "hduser" close terminal and open previous terminal

```
techgrid@ubuntu:~$ sudo su root
[sudo] password for techgrid:
root@ubuntu:/home/techgrid# cd
root@ubuntu:~# sudo gedit /etc/sudoers

(gedit:5180): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskt
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

(gedit:5180): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskt
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files
root@ubuntu:~# 
```



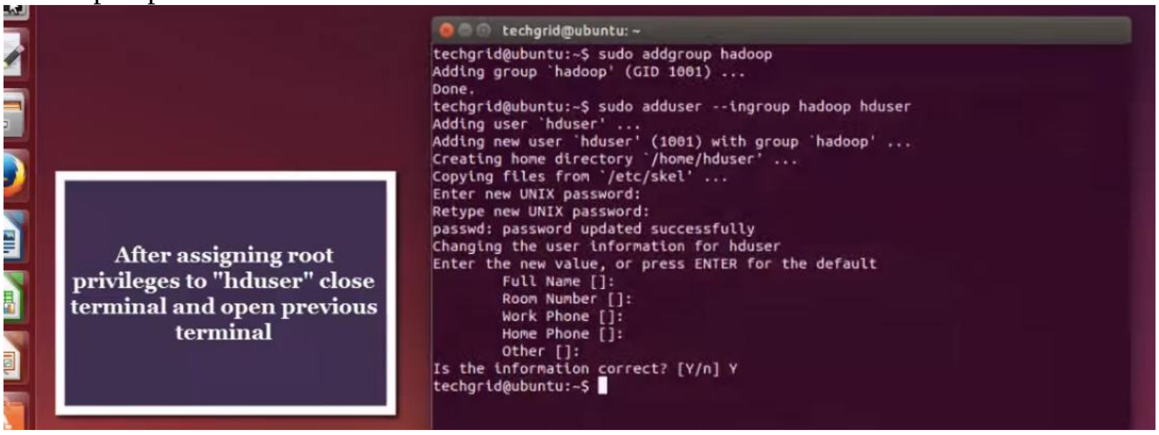
After assigning root privileges to "hduser" close terminal and open previous terminal

```
techgrid@ubuntu:~$ sudo su root
[sudo] password for techgrid:
root@ubuntu:/home/techgrid# cd
root@ubuntu:~# sudo gedit /etc/sudoers

(gedit:5180): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files
```

Close this terminal?

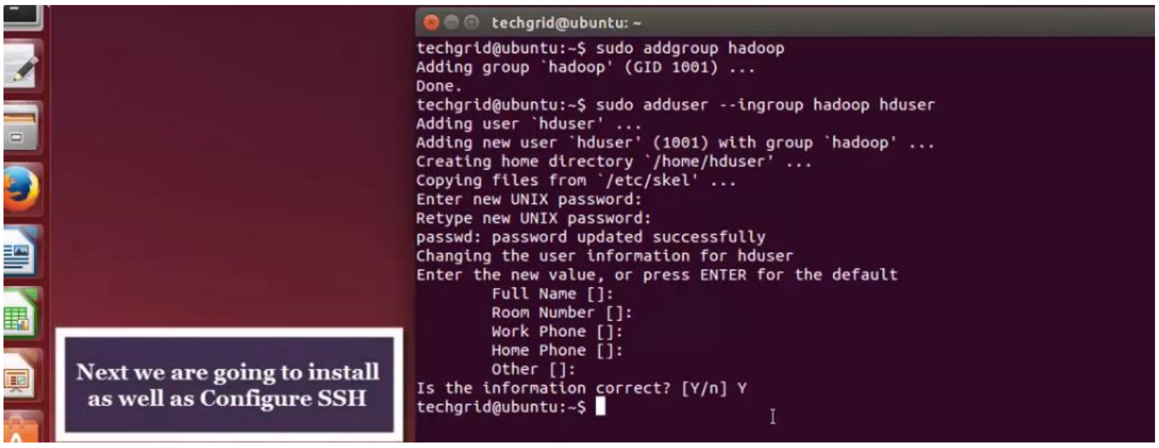
There is still a process running in this terminal. Closing the terminal will kill it.

Then open previous terminal:



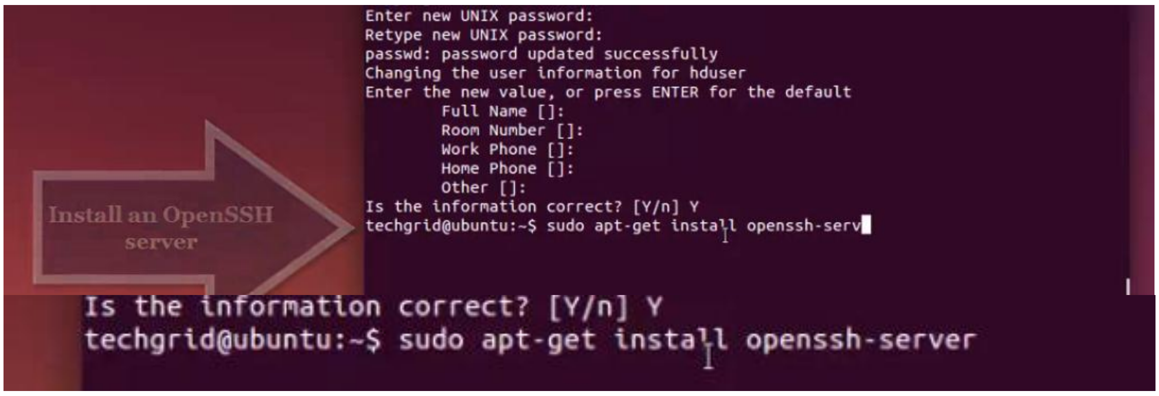
After assigning root privileges to "hduser" close terminal and open previous terminal

```
techgrid@ubuntu:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
techgrid@ubuntu:~$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
techgrid@ubuntu:~$ 
```



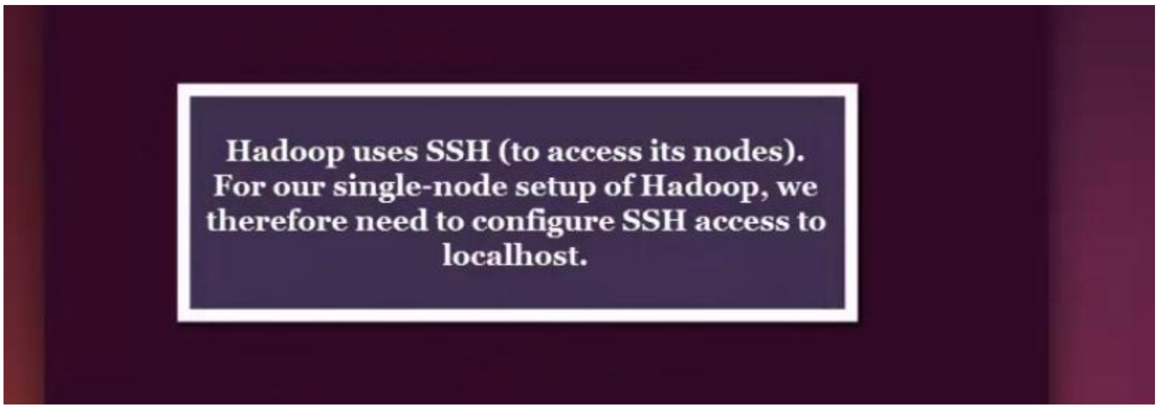
Next we are going to install as well as Configure SSH

```
techgrid@ubuntu:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
techgrid@ubuntu:~$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
techgrid@ubuntu:~$ 
```

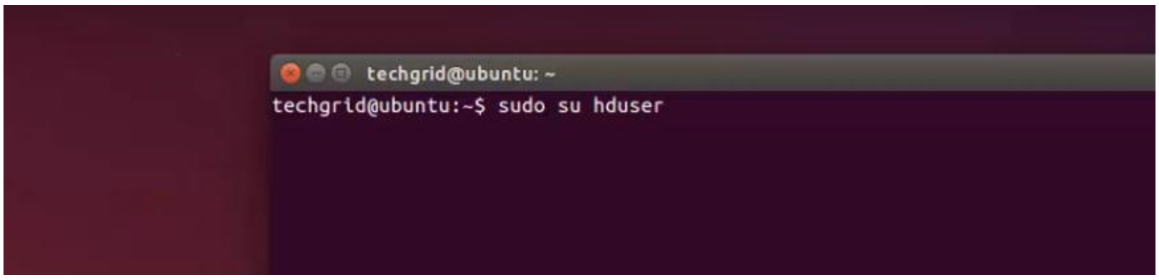


```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
techgrid@ubuntu:~$ sudo apt-get install openssh-serv
```

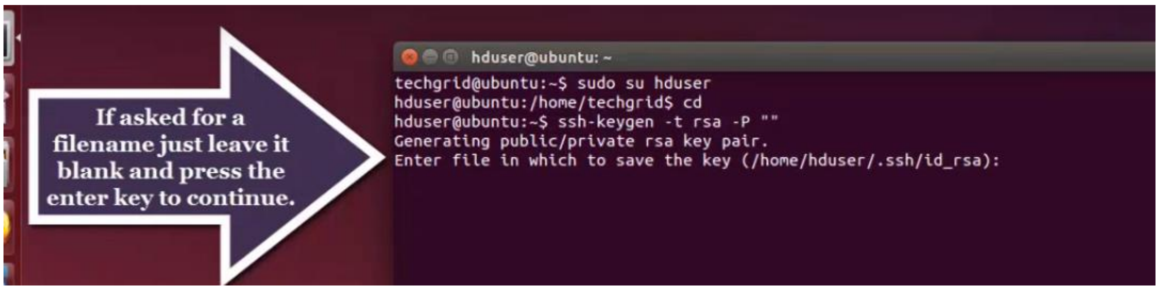
```
Is the information correct? [Y/n] Y
techgrid@ubuntu:~$ sudo apt-get install openssh-server
```



Hadoop uses SSH (to access its nodes). For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost.

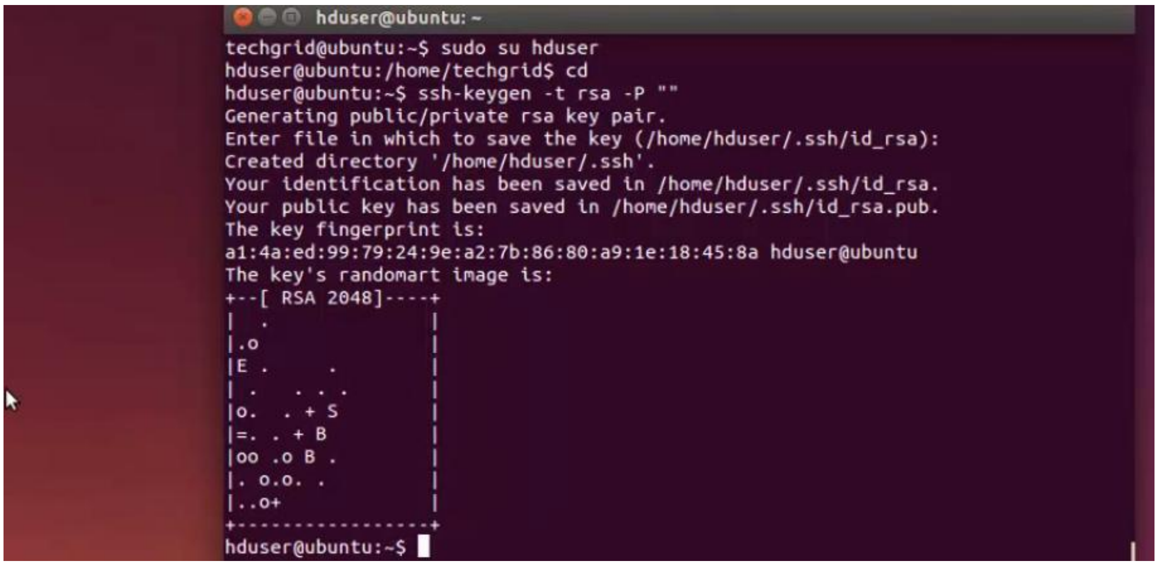


```
techgrid@ubuntu:~$ sudo su hduser
```

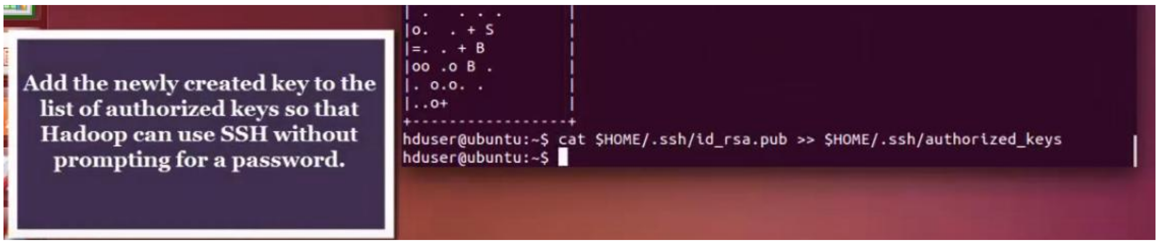


If asked for a filename just leave it blank and press the enter key to continue.

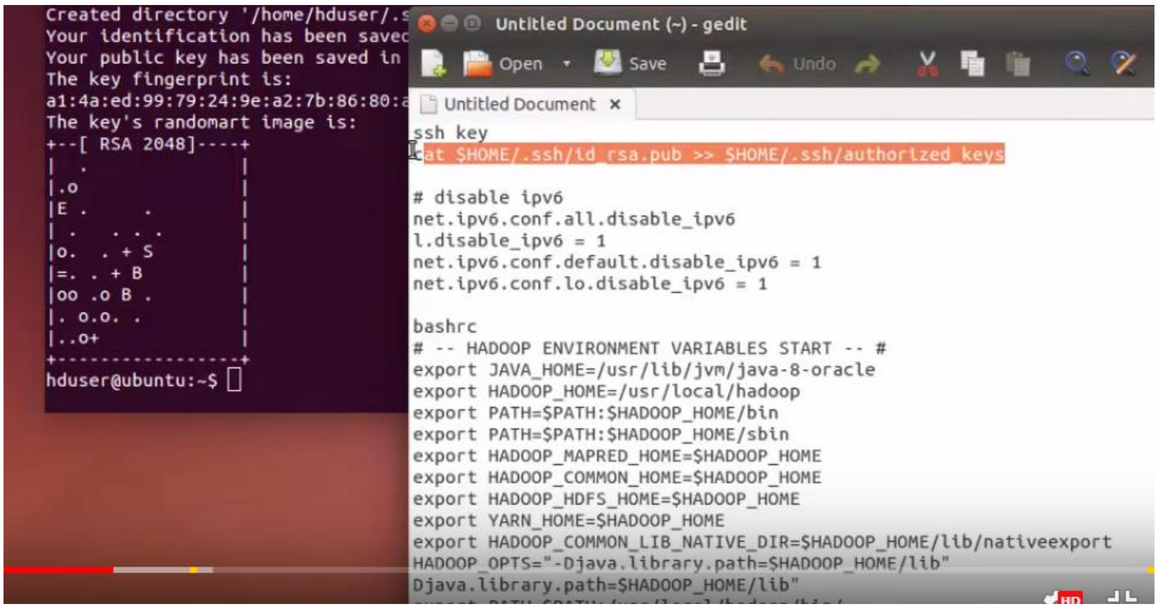
```
techgrid@ubuntu:~$ sudo su hduser
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
```



```
techgrid@ubuntu:~$ sudo su hduser
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
a1:4a:ed:99:79:24:9e:a2:7b:86:80:a9:1e:18:45:8a hduser@ubuntu
The key's randomart image is:
+--[ RSA 2048]----+
|  .              |
|.o               |
|E .      .       |
| .    . . .      |
|o.   . + S       |
|=.  . + B        |
|oo .o B .        |
|. o.o. .         |
|..o+             |
+-----------------+
hduser@ubuntu:~$
```

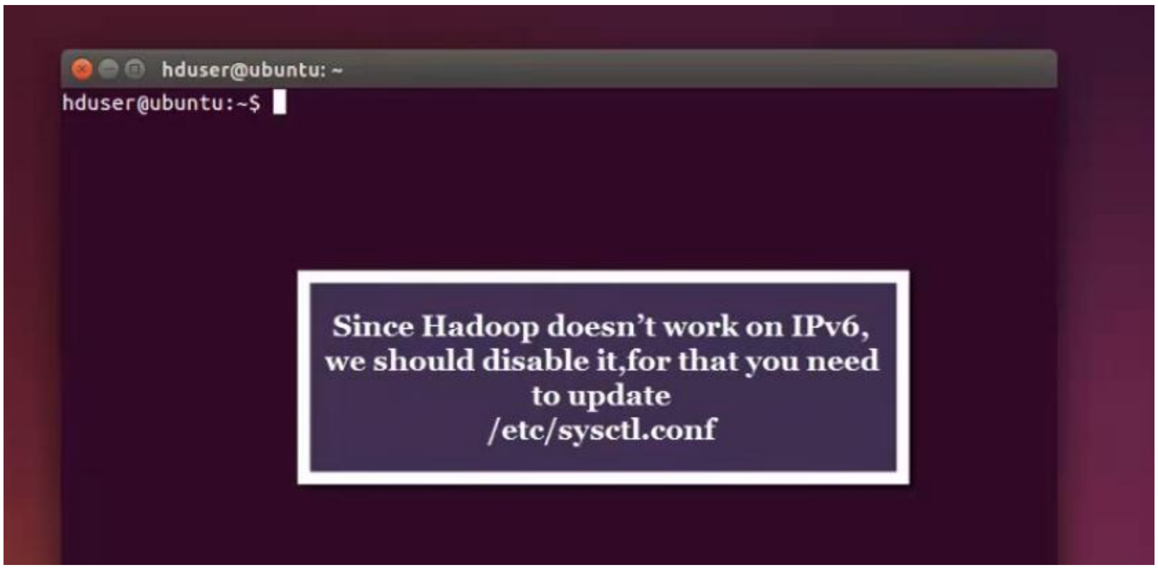

Add the newly created key to the list of authorized keys so that Hadoop can use SSH without prompting for a password.

```
|o.   . + S       |
|=.  . + B        |
|oo .o B .        |
|. o.o. .         |
|..o+             |
+-----------------+
hduser@ubuntu:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
hduser@ubuntu:~$
```



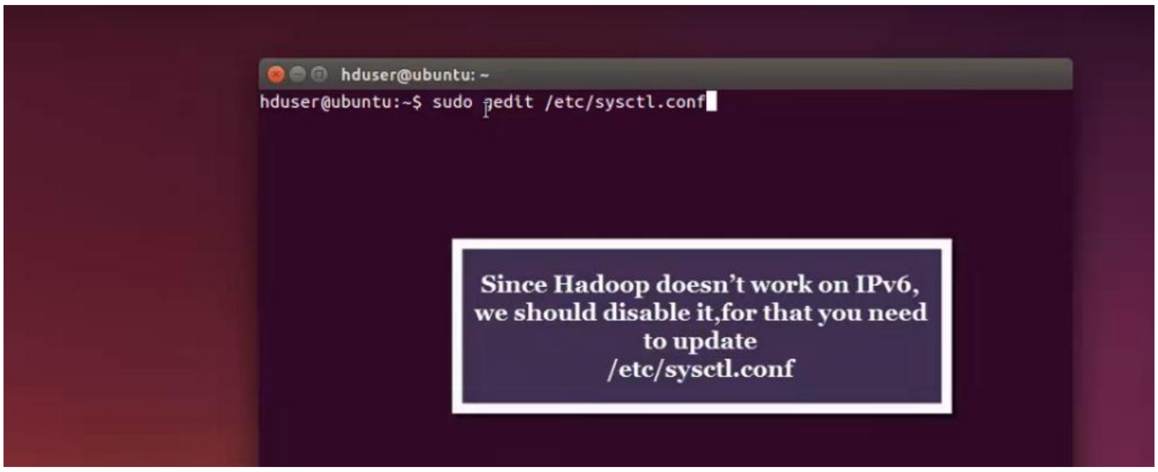
```
Created directory '/home/hduser/.s
Your identification has been savec
Your public key has been saved in
The key fingerprint is:
a1:4a:ed:99:79:24:9e:a2:7b:86:80:a
The key's randomart image is:
+--[ RSA 2048]----+
|  .              |
|.o               |
|E .      .       |
| .    . . .      |
|o.   . + S       |
|=.  . + B        |
|oo .o B .        |
|. o.o. .         |
|..o+             |
+-----------------+
hduser@ubuntu:~$
```

```
ssh key
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

# disable ipv6
net.ipv6.conf.all.disable_ipv6
l.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

bashrc
# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/nativeexport
HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
Djava.library.path=$HADOOP_HOME/lib"
```



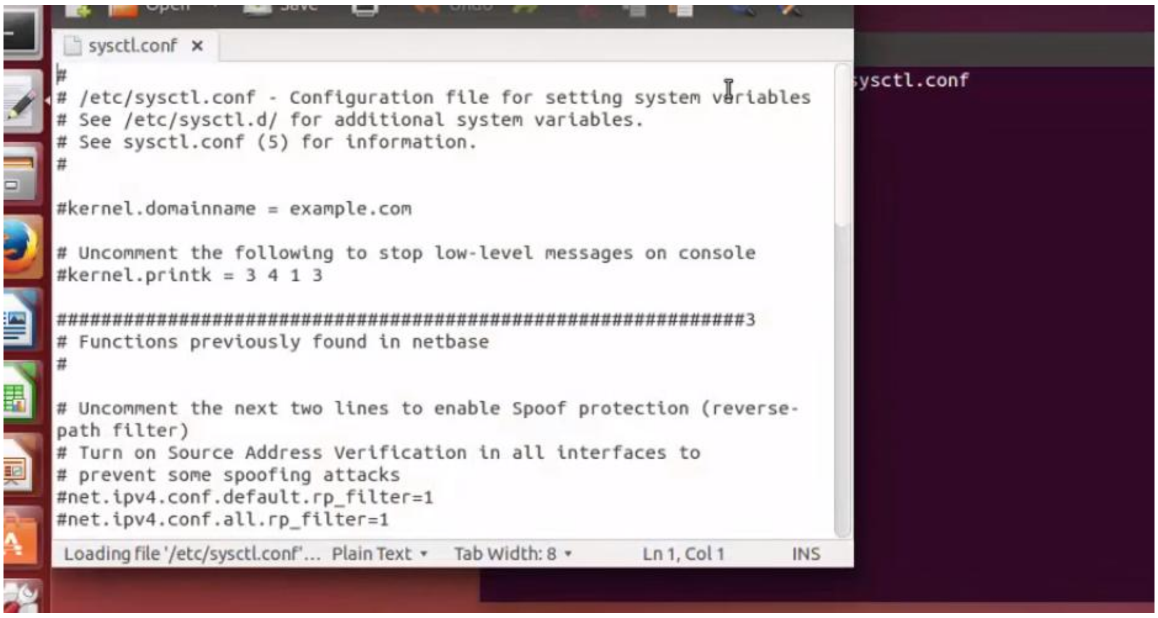
```
hduser@ubuntu:~$
```

Since Hadoop doesn't work on IPv6, we should disable it,for that you need to update /etc/sysctl.conf



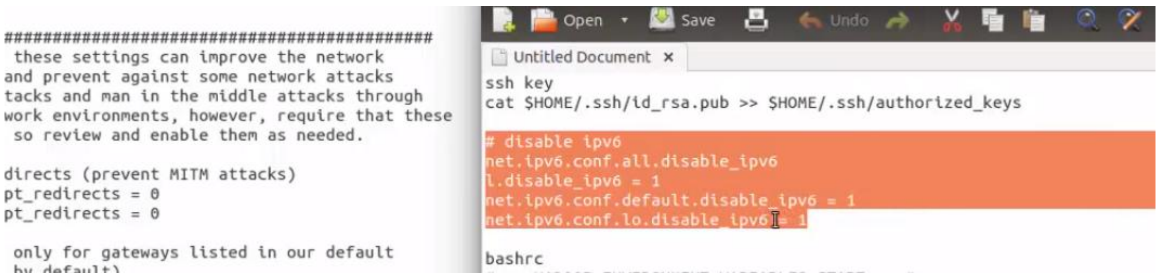
```
hduser@ubuntu:~$ sudo gedit /etc/sysctl.conf
```

Since Hadoop doesn't work on IPv6, we should disable it,for that you need to update /etc/sysctl.conf



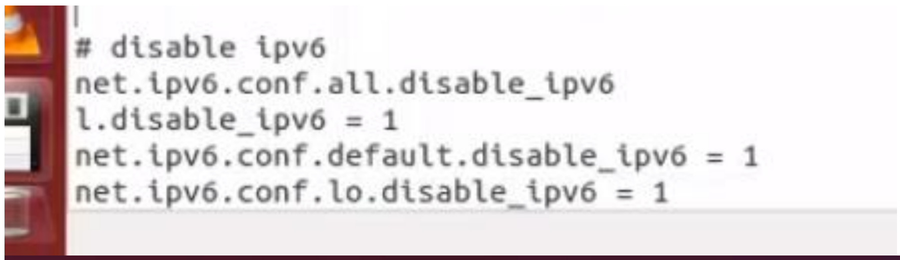
```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-
path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
```

sysctl.conf
Loading file '/etc/sysctl.conf'... Plain Text Tab Width: 8 Ln 1, Col 1 INS

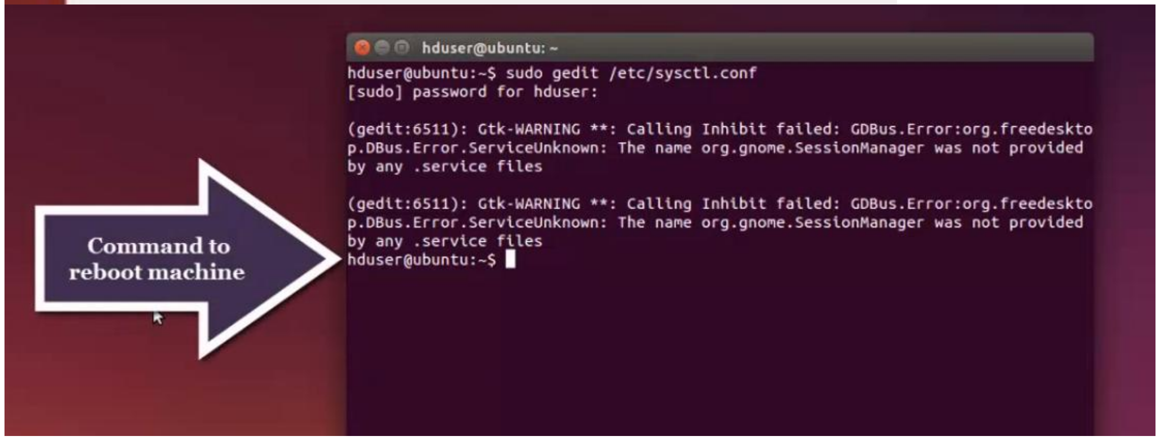


```
################################################
 these settings can improve the network
and prevent against some network attacks
tacks and man in the middle attacks through
work environments, however, require that these
 so review and enable them as needed.

directs (prevent MITM attacks)
pt_redirects = 0
pt_redirects = 0

 only for gateways listed in our default
by default)
```

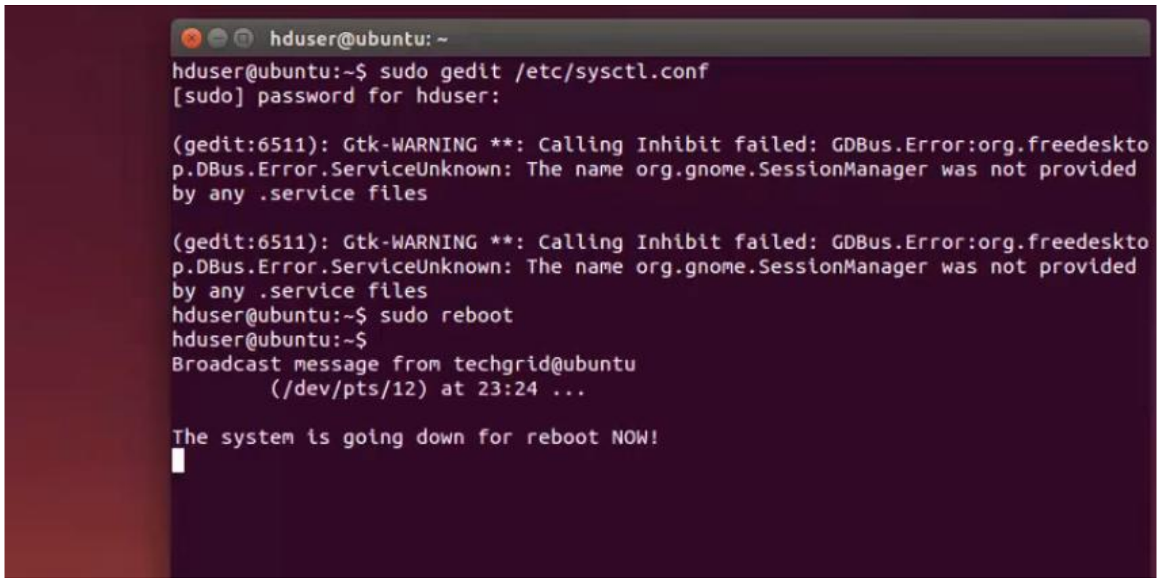
```
ssh key
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

# disable ipv6
net.ipv6.conf.all.disable_ipv6
l.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

bashrc
```



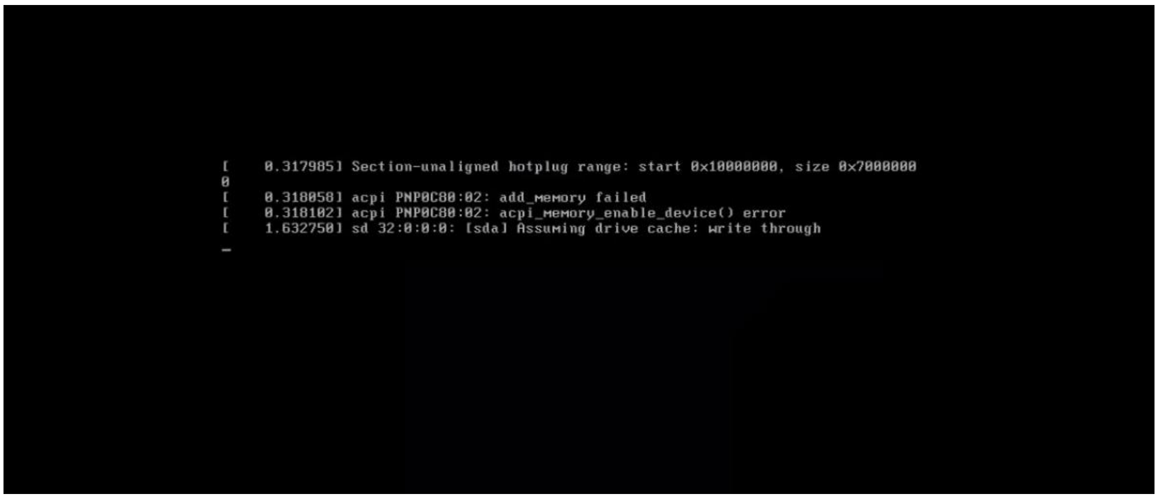
```
# disable ipv6
net.ipv6.conf.all.disable_ipv6
l.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```



```
hduser@ubuntu:~$ sudo gedit /etc/sysctl.conf
[sudo] password for hduser:

(gedit:6511): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

(gedit:6511): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files
hduser@ubuntu:~$
```

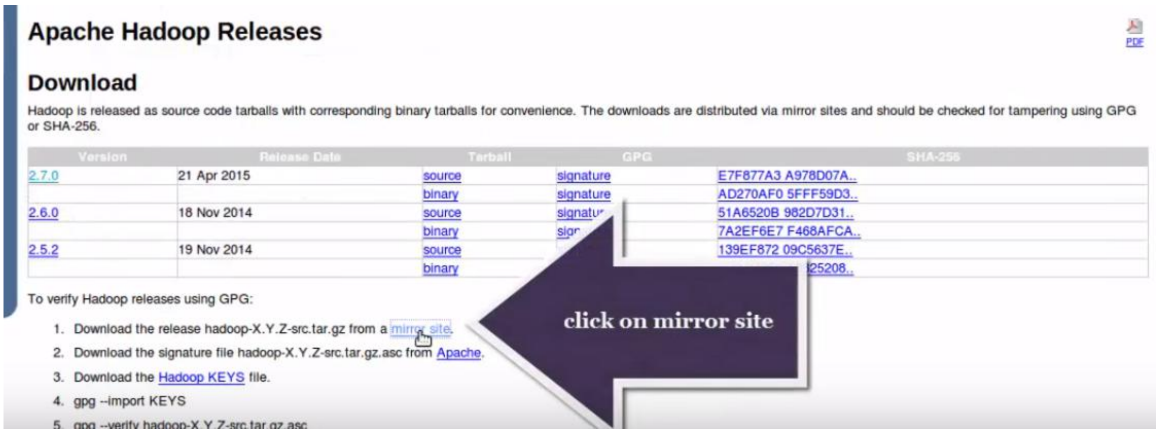


```
hduser@ubuntu:~$ sudo gedit /etc/sysctl.conf
[sudo] password for hduser:

(gedit:6511): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

(gedit:6511): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files
hduser@ubuntu:~$ sudo reboot
hduser@ubuntu:~$
Broadcast message from techgrid@ubuntu
        (/dev/pts/12) at 23:24 ...

The system is going down for reboot NOW!
```



```
[    0.317985] Section-unaligned hotplug range: start 0x10000000, size 0x7000000
0
[    0.318058] acpi PNP0C80:02: add_memory failed
[    0.318102] acpi PNP0C80:02: acpi_memory_enable_device() error
[    1.632750] sd 32:0:0:0: [sda] Assuming drive cache: write through
_
```

After reboot system now we are going to Download latest Apache Hadoop source from Apache mirrors



Apache Hadoop Releases

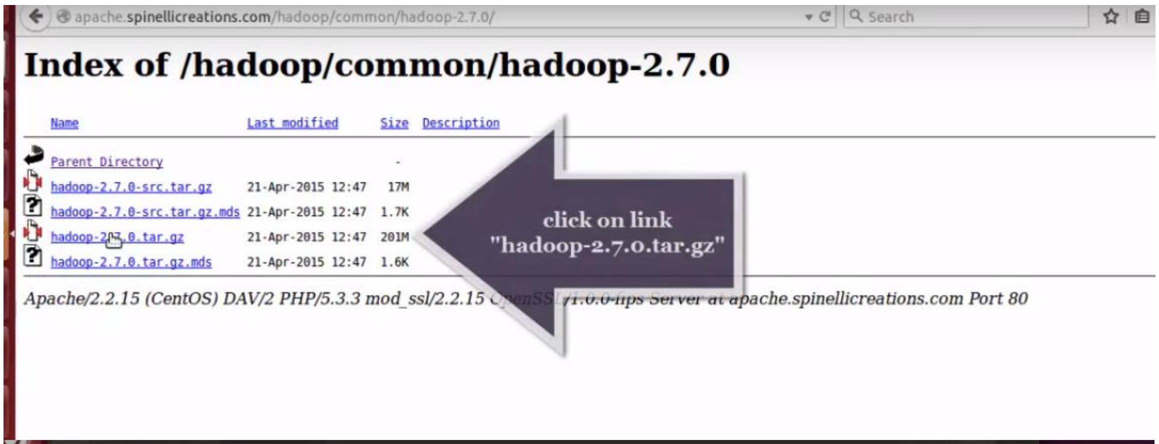
Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

Version	Release Date	Tarball	GPG	SHA-256
2.7.0	21 Apr 2015	source	signature	E7F877A3 A978D07A..
		binary	signature	AD270AF0 5FFF59D3..
2.6.0	18 Nov 2014	source	signatur	51A6520B 982D7D31..
		binary	sigr	7A2EF6E7 F468AFCA..
2.5.2	19 Nov 2014	source	[illegible]	139EF872 09C5637E..
		binary	[illegible]	25208..

To verify Hadoop releases using GPG:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from Apache.
3. Download the Hadoop KEYS file.
4. gpg --import KEYS
5. gpg --verify hadoop-X.Y.Z-src.tar.gz.asc

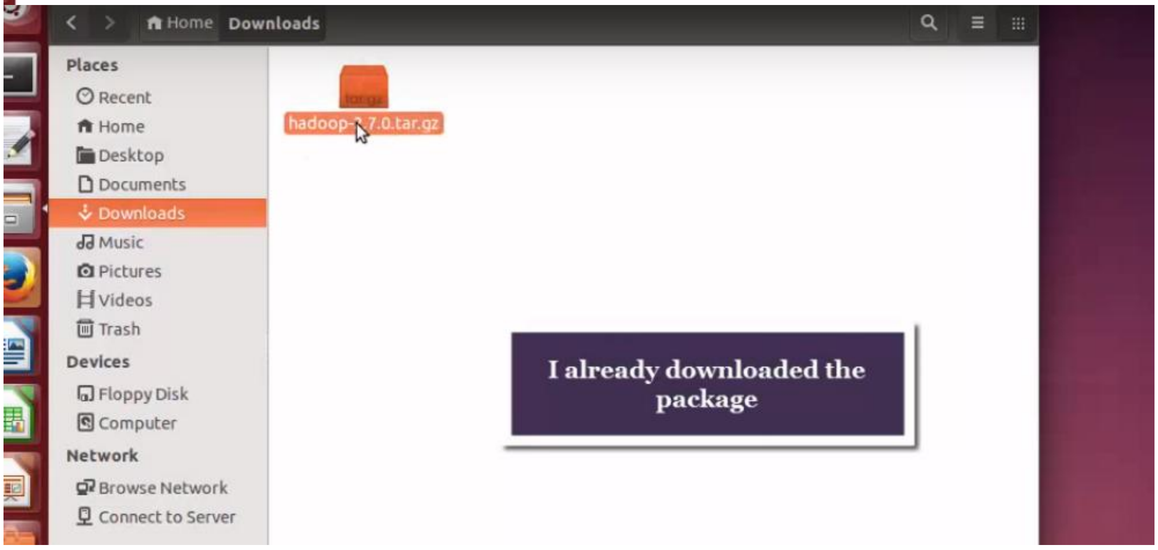

apache.spinellicreations.com/hadoop/common/hadoop-2.7.0/

Index of /hadoop/common/hadoop-2.7.0

```
Name                          Last modified      Size  Description
Parent Directory                                   -
hadoop-2.7.0-src.tar.gz       21-Apr-2015 12:47   17M
hadoop-2.7.0-src.tar.gz.mds   21-Apr-2015 12:47  1.7K
hadoop-2.7.0.tar.gz           21-Apr-2015 12:47  201M
hadoop-2.7.0.tar.gz.mds       21-Apr-2015 12:47  1.6K
```

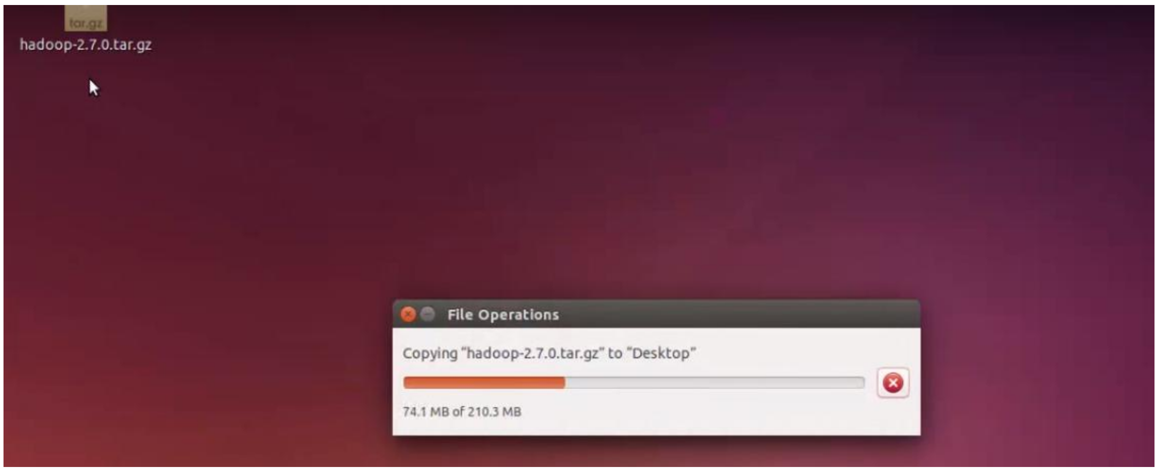
Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3 mod_ssl/2.2.15 OpenSSL/1.0.0-fips Server at apache.spinellicreations.com Port 80

click on link "hadoop-2.7.0.tar.gz"



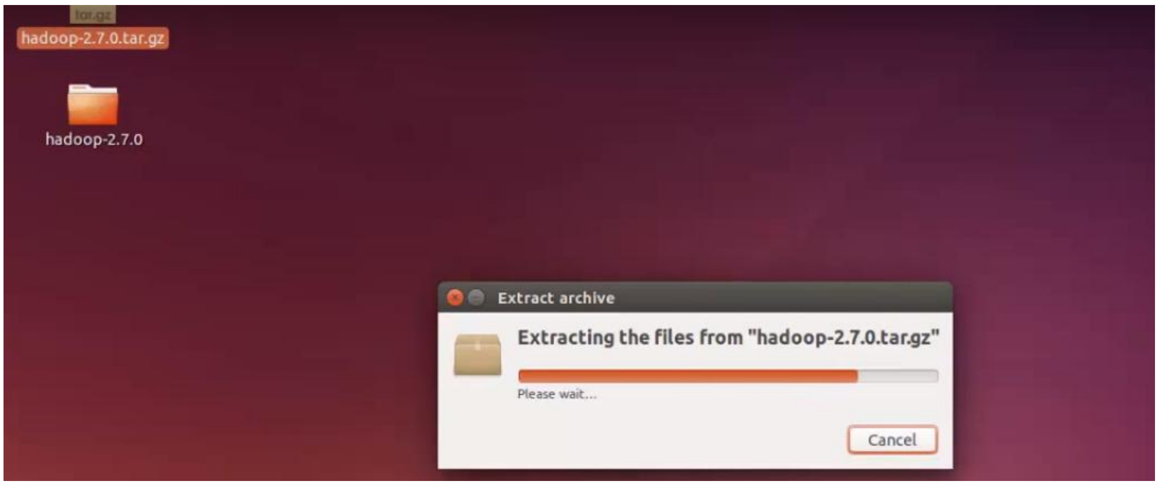
I already downloaded the package

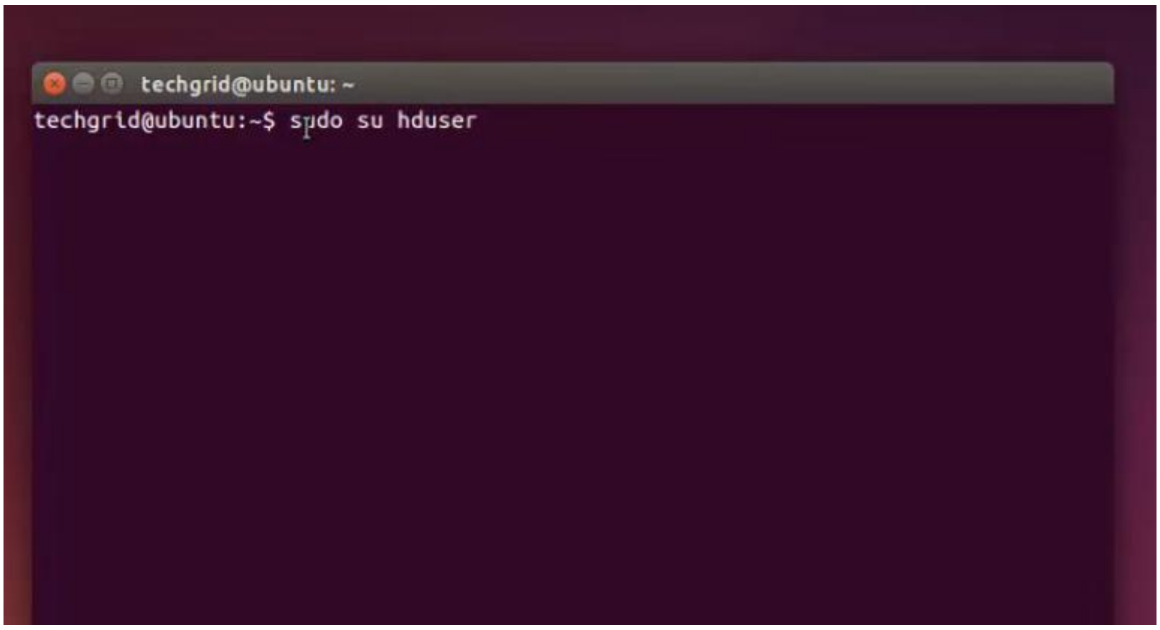
Move to desktop



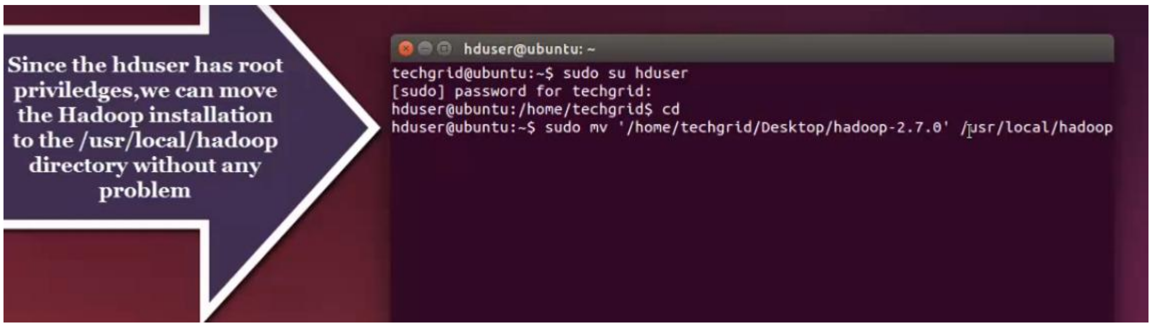
Copying "hadoop-2.7.0.tar.gz" to "Desktop"

74.1 MB of 210.3 MB



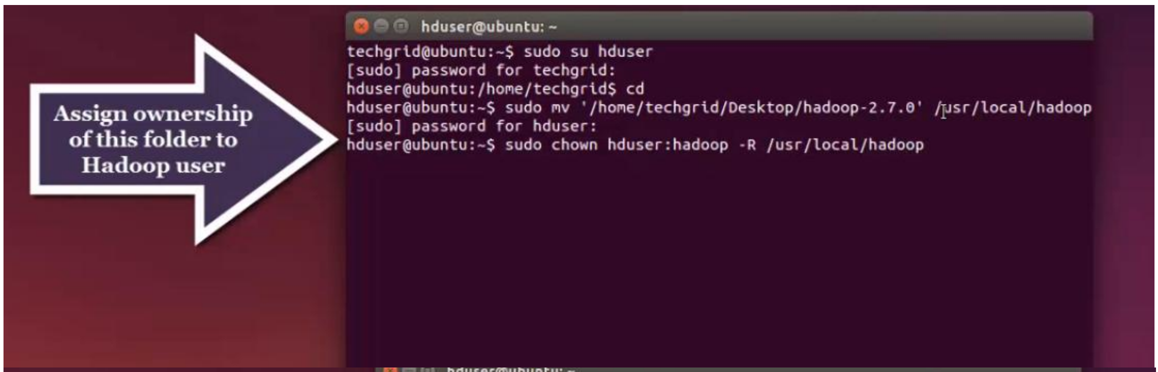


```
techgrid@ubuntu:~$ sudo su hduser
```

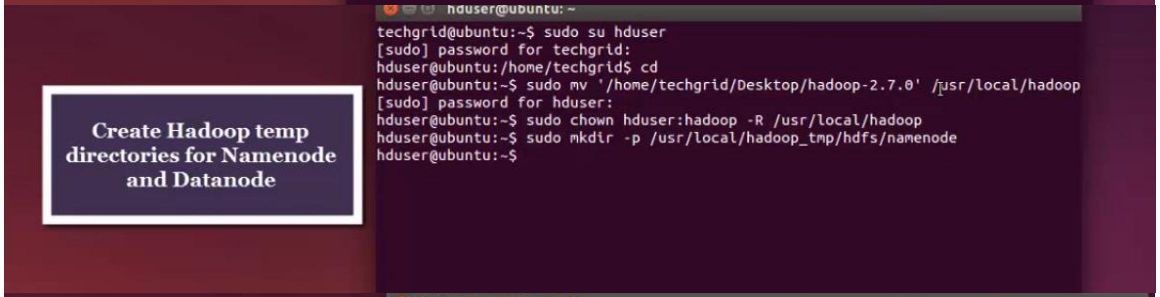


Since the hduser has root priviledges,we can move the Hadoop installation to the /usr/local/hadoop directory without any problem

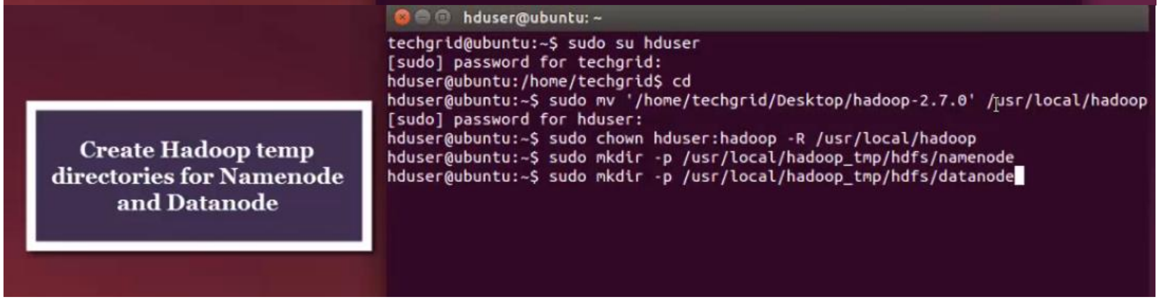
```
techgrid@ubuntu:~$ sudo su hduser
[sudo] password for techgrid:
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ sudo mv '/home/techgrid/Desktop/hadoop-2.7.0' /usr/local/hadoop
```



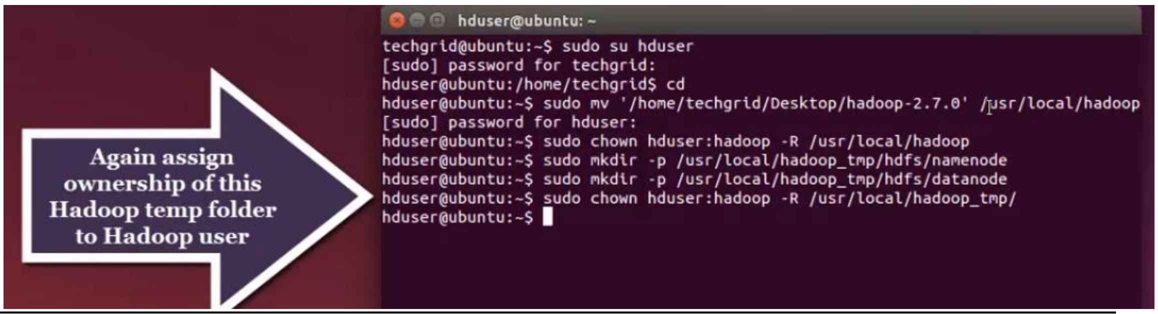
```
techgrid@ubuntu:~$ sudo su hduser
[sudo] password for techgrid:
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ sudo mv '/home/techgrid/Desktop/hadoop-2.7.0' /usr/local/hadoop
[sudo] password for hduser:
hduser@ubuntu:~$ sudo chown hduser:hadoop -R /usr/local/hadoop
```



```
techgrid@ubuntu:~$ sudo su hduser
[sudo] password for techgrid:
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ sudo mv '/home/techgrid/Desktop/hadoop-2.7.0' /usr/local/hadoop
[sudo] password for hduser:
hduser@ubuntu:~$ sudo chown hduser:hadoop -R /usr/local/hadoop
hduser@ubuntu:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
hduser@ubuntu:~$
```

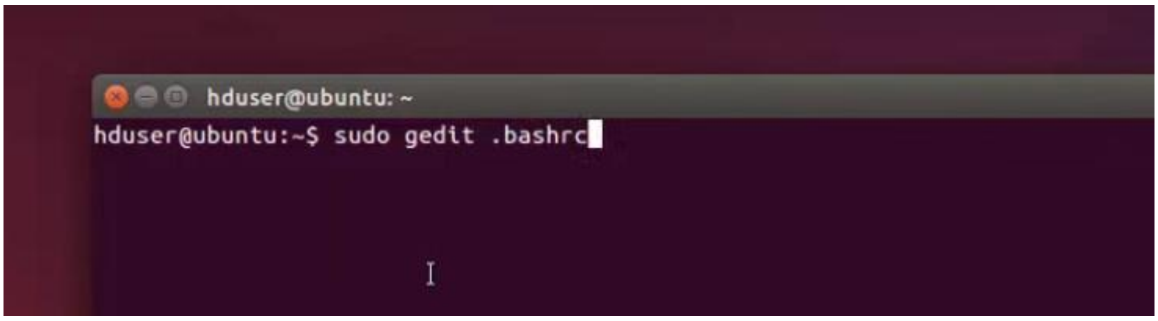


```
techgrid@ubuntu:~$ sudo su hduser
[sudo] password for techgrid:
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ sudo mv '/home/techgrid/Desktop/hadoop-2.7.0' /usr/local/hadoop
[sudo] password for hduser:
hduser@ubuntu:~$ sudo chown hduser:hadoop -R /usr/local/hadoop
hduser@ubuntu:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
hduser@ubuntu:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

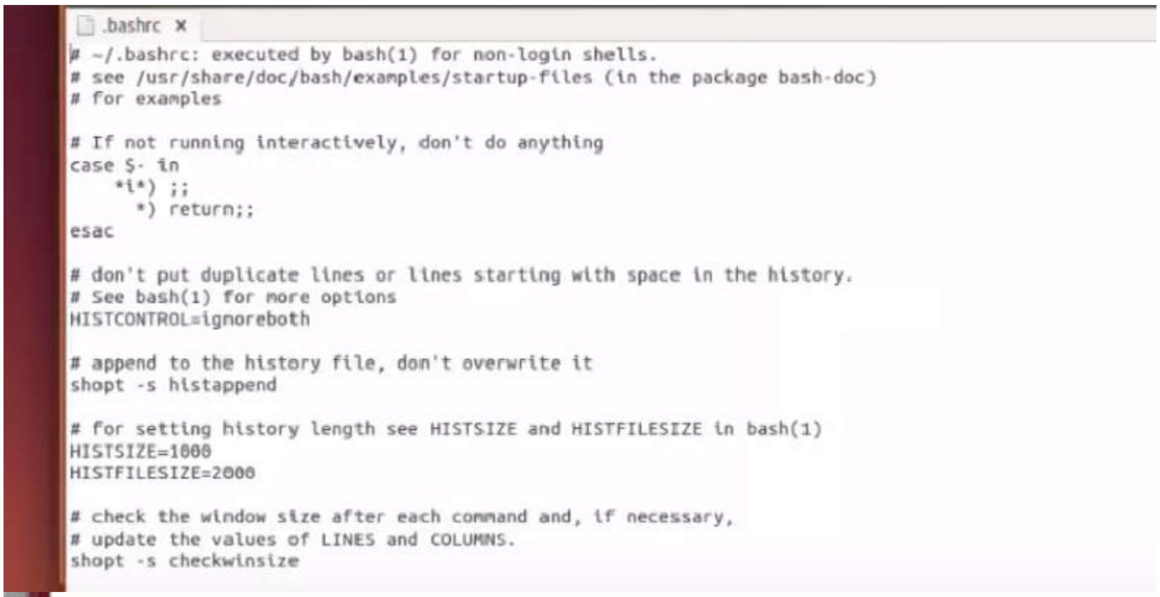

Again assign ownership of this Hadoop temp folder to Hadoop user
hduser@ubuntu: ~

```
techgrid@ubuntu:~$ sudo su hduser
[sudo] password for techgrid:
hduser@ubuntu:/home/techgrid$ cd
hduser@ubuntu:~$ sudo mv '/home/techgrid/Desktop/hadoop-2.7.0' /usr/local/hadoop
[sudo] password for hduser:
hduser@ubuntu:~$ sudo chown hduser:hadoop -R /usr/local/hadoop
hduser@ubuntu:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
hduser@ubuntu:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
hduser@ubuntu:~$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
hduser@ubuntu:~$
```

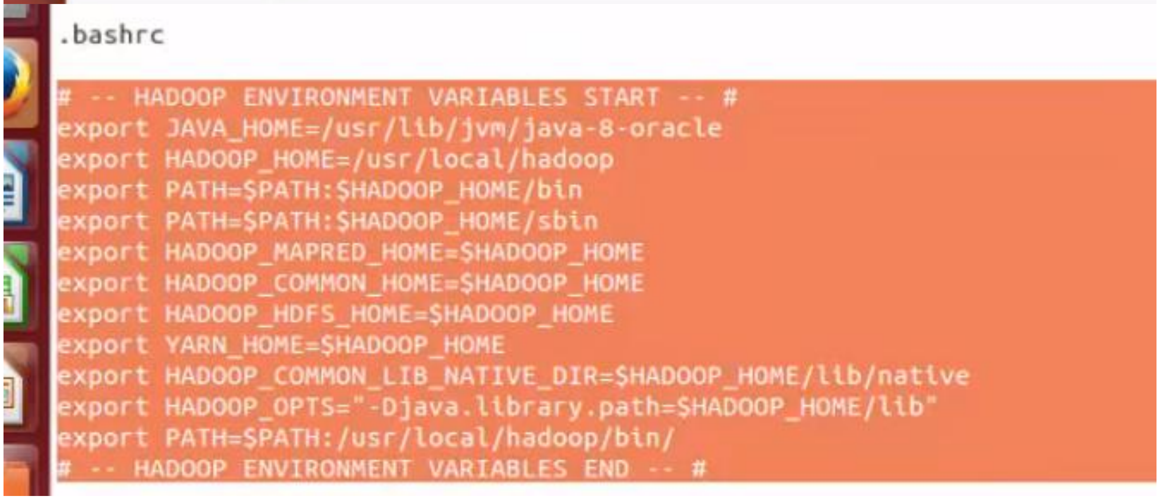
Today end class here



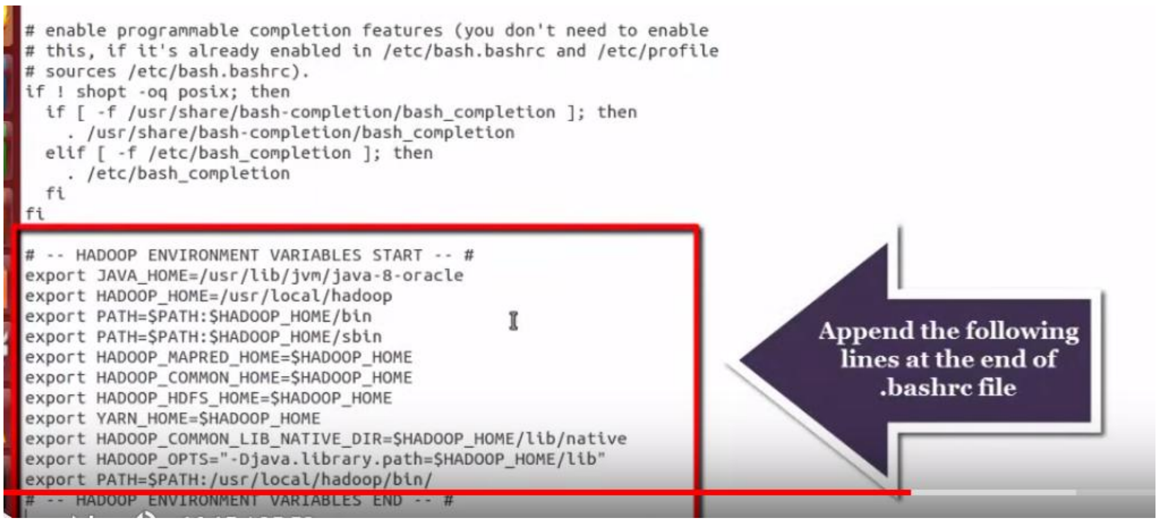
```
hduser@ubuntu:~$ sudo gedit .bashrc
```



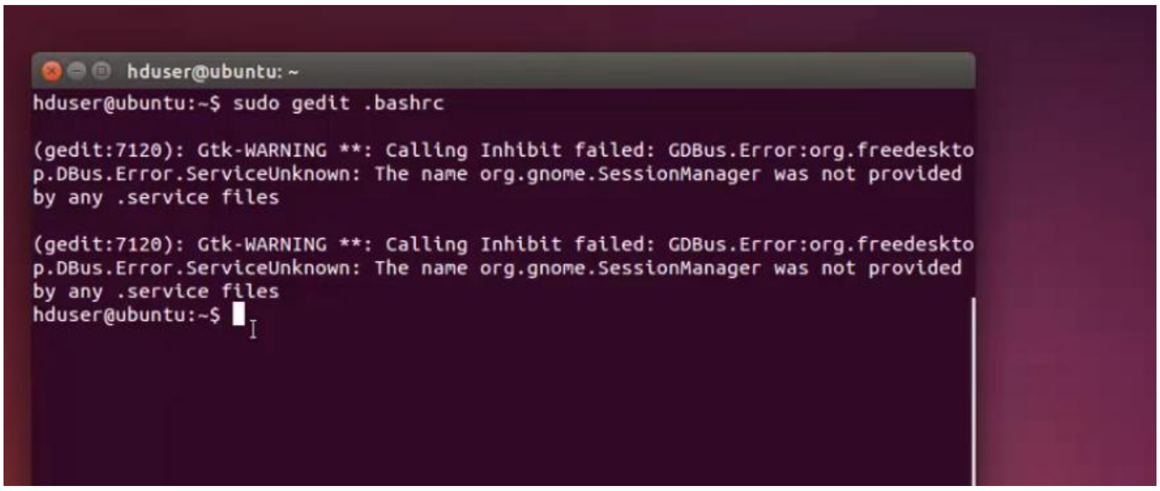
```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize
```



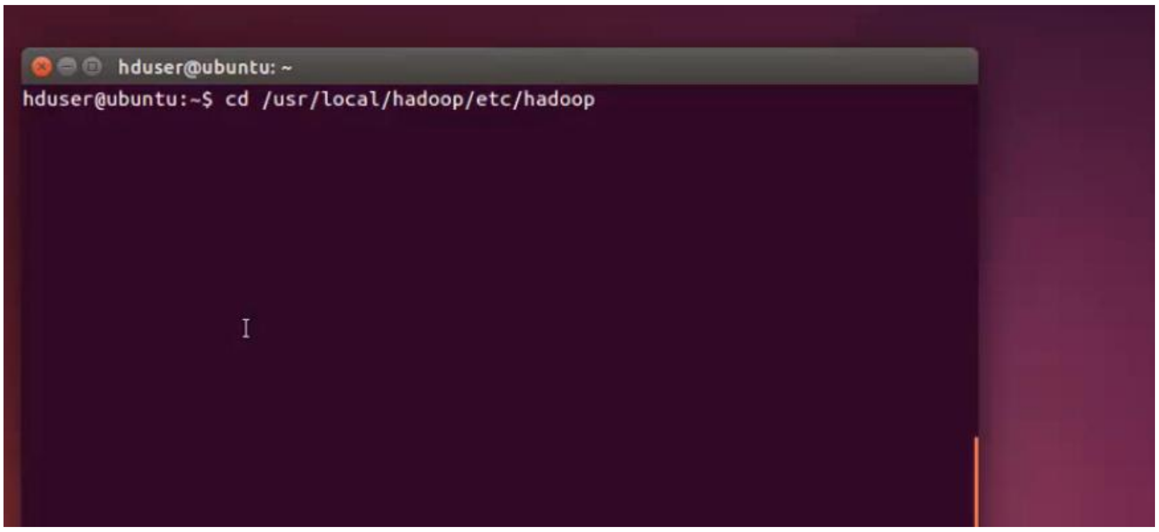
```
# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export PATH=$PATH:/usr/local/hadoop/bin/
# -- HADOOP ENVIRONMENT VARIABLES END -- #
```



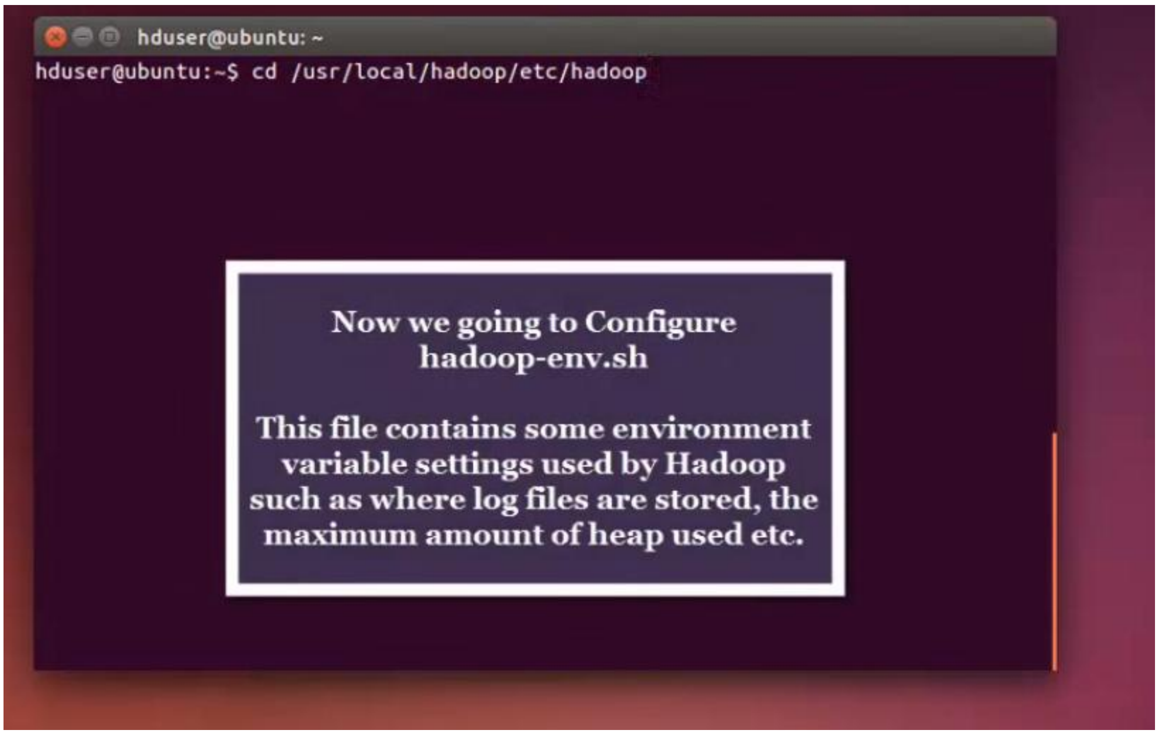
```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export PATH=$PATH:/usr/local/hadoop/bin/
# -- HADOOP ENVIRONMENT VARIABLES END -- #
```



```
hduser@ubuntu: ~
hduser@ubuntu:~$ sudo gedit .bashrc

(gedit:7120): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

(gedit:7120): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files
hduser@ubuntu:~$
```



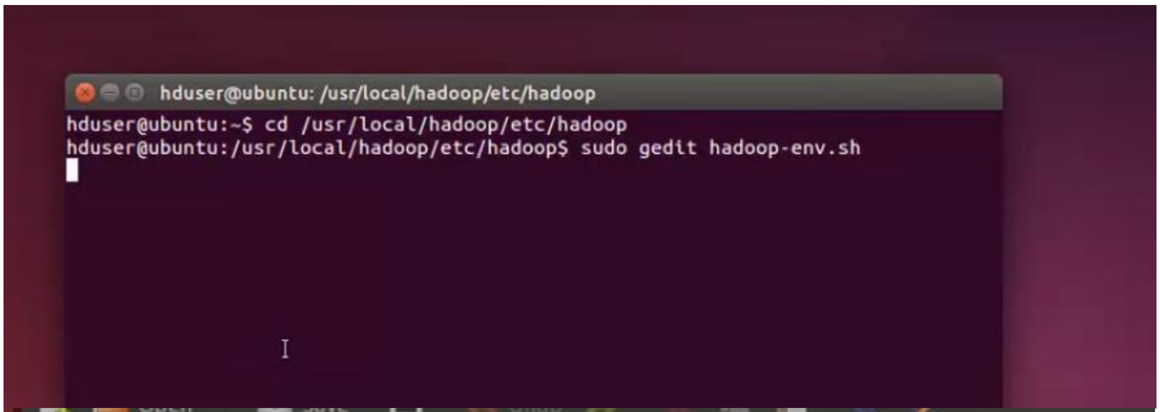
```
hduser@ubuntu: ~
hduser@ubuntu:~$ cd /usr/local/hadoop/etc/hadoop
```



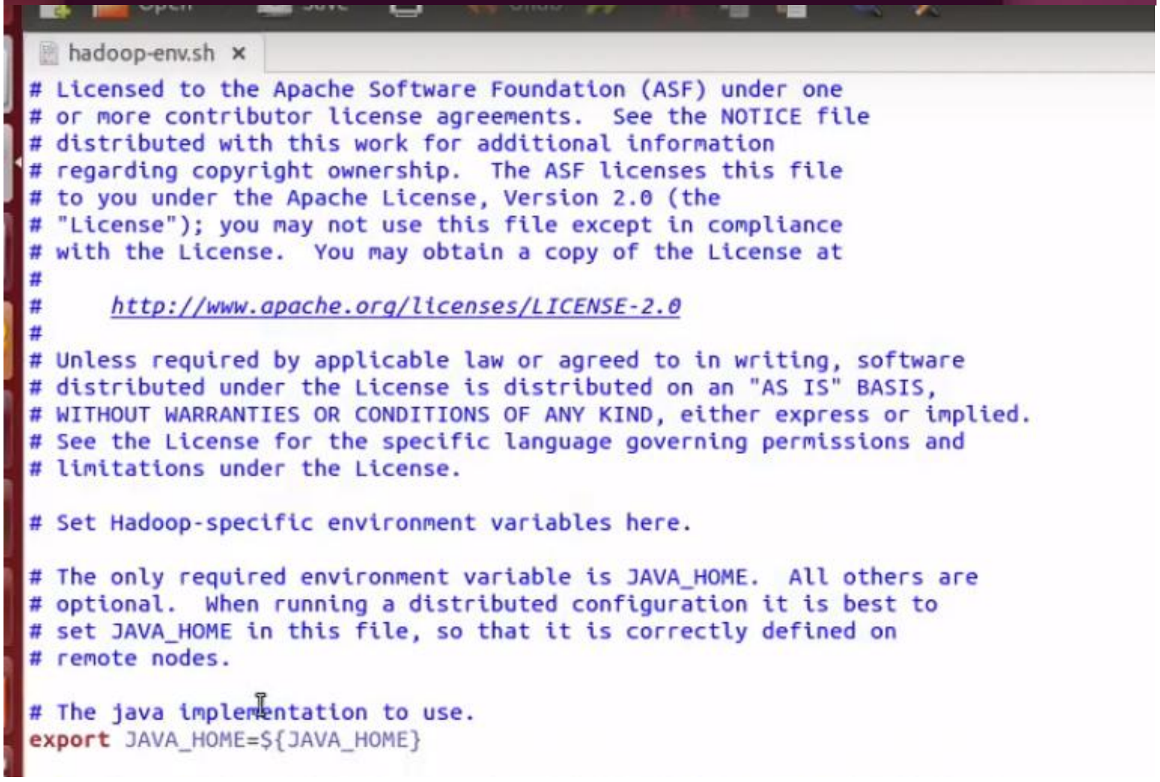
```
hduser@ubuntu: ~
hduser@ubuntu:~$ cd /usr/local/hadoop/etc/hadoop
```

Now we going to Configure hadoop-env.sh

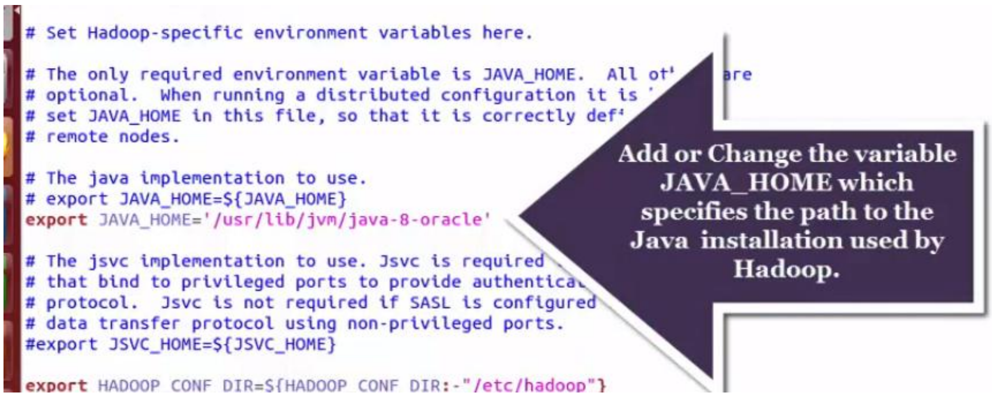
This file contains some environment variable settings used by Hadoop such as where log files are stored, the maximum amount of heap used etc.



```
hduser@ubuntu:~$ cd /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo gedit hadoop-env.sh
```

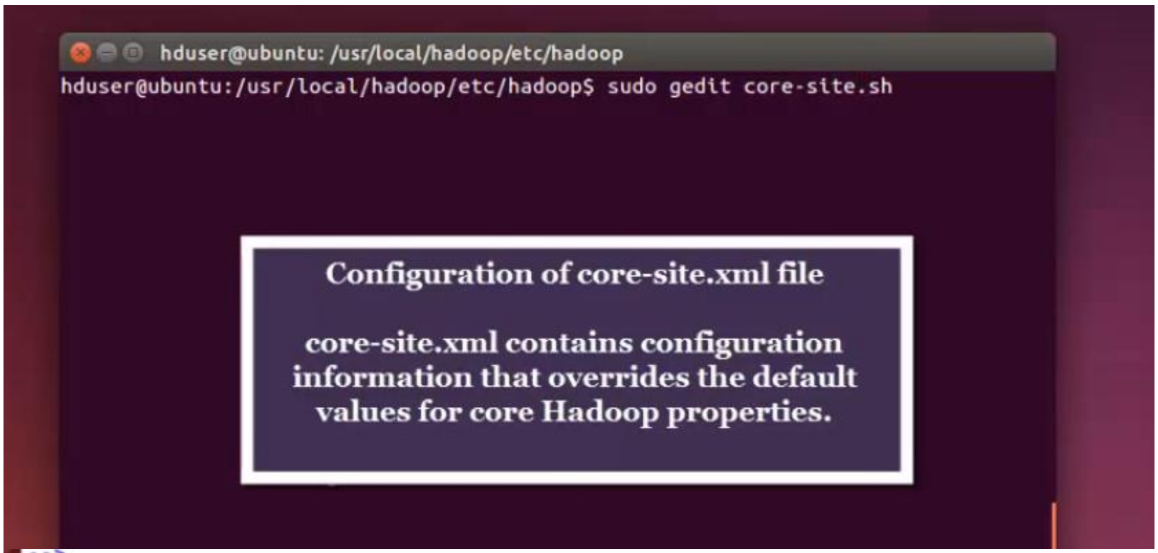


```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
```



```
# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All ot    are
# optional.  When running a distributed configuration it is
# set JAVA_HOME in this file, so that it is correctly def
# remote nodes.

# The java implementation to use.
# export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME='/usr/lib/jvm/java-8-oracle'

# The jsvc implementation to use. Jsvc is required
# that bind to privileged ports to provide authentica
# protocol.  Jsvc is not required if SASL is configured
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
```

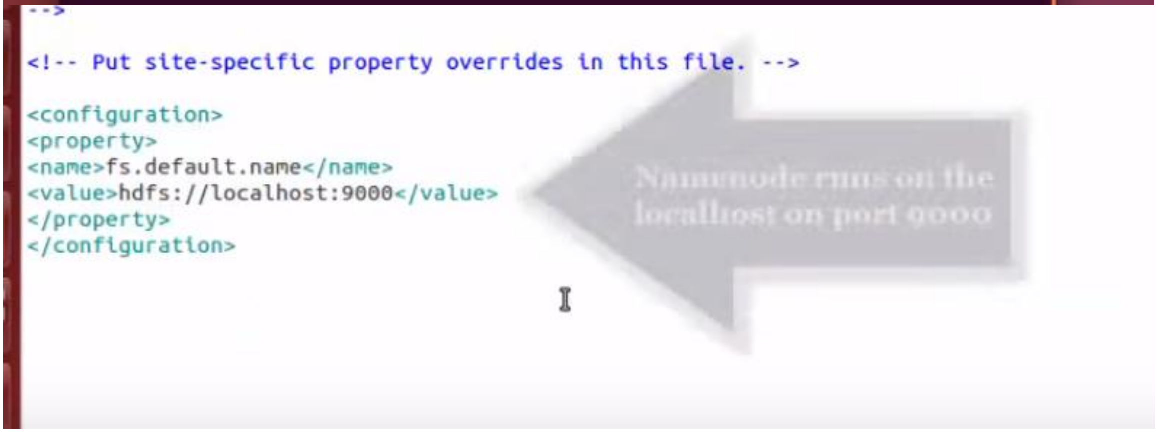
Add or Change the variable JAVA_HOME which specifies the path to the Java installation used by Hadoop.



```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.sh
```

Configuration of core-site.xml file

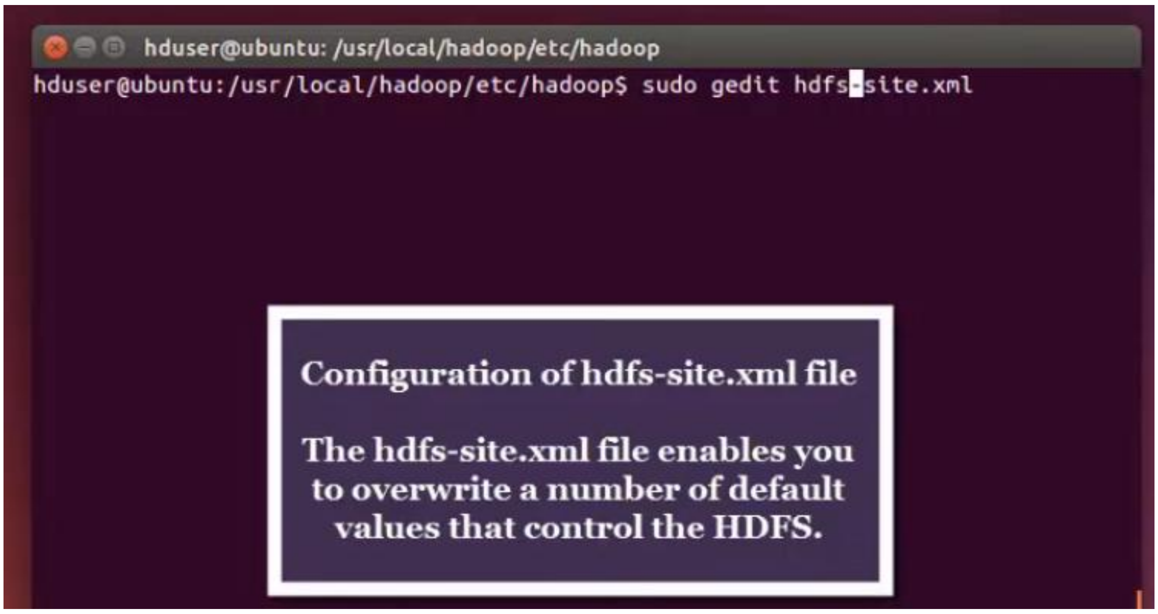
core-site.xml contains configuration information that overrides the default values for core Hadoop properties.



```
-->

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

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Namenode runs on the localhost on port 9000

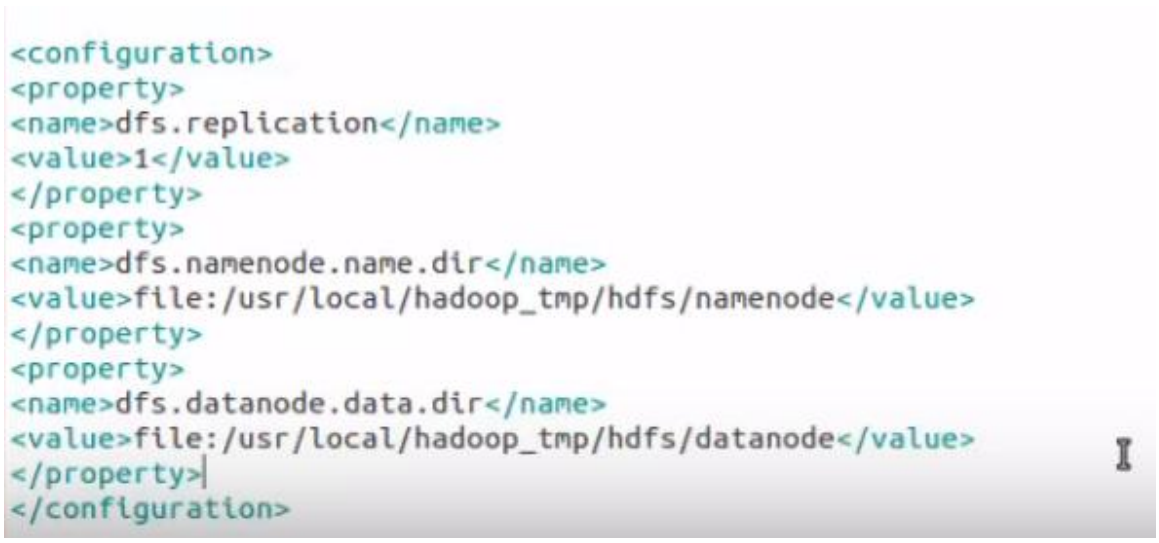
5



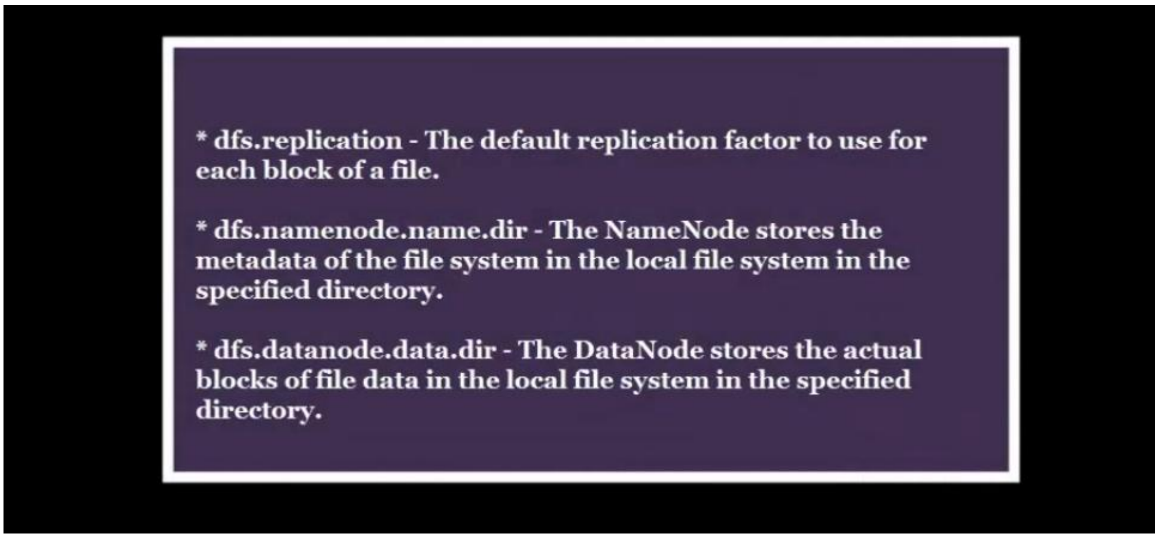
```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml
```

Configuration of hdfs-site.xml file

The hdfs-site.xml file enables you to overwrite a number of default values that control the HDFS.



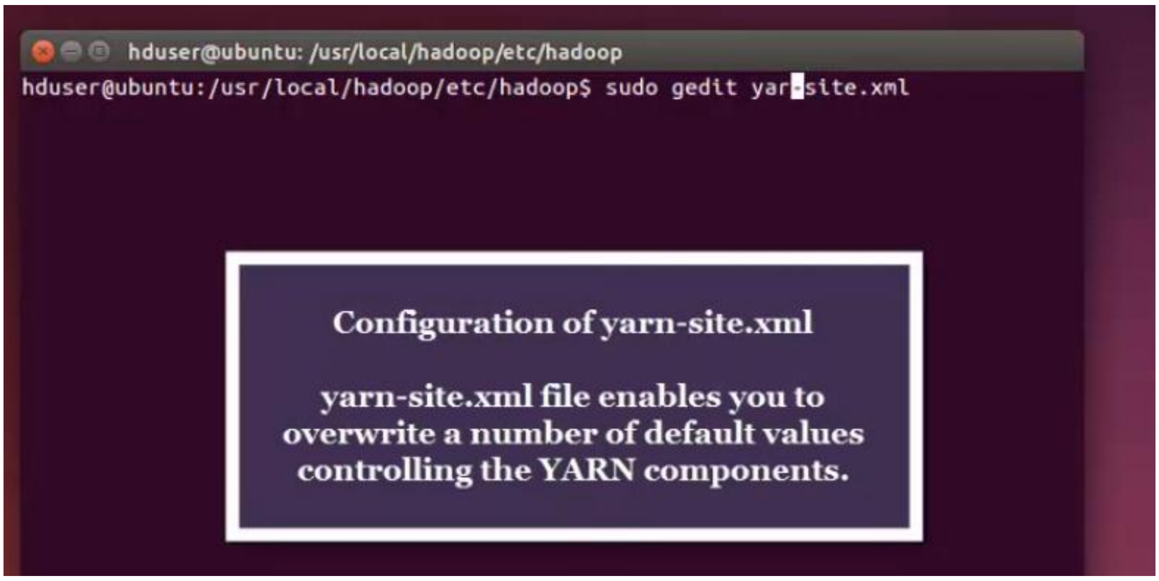
```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>
```



* dfs.replication - The default replication factor to use for each block of a file.

* dfs.namenode.name.dir - The NameNode stores the metadata of the file system in the local file system in the specified directory.

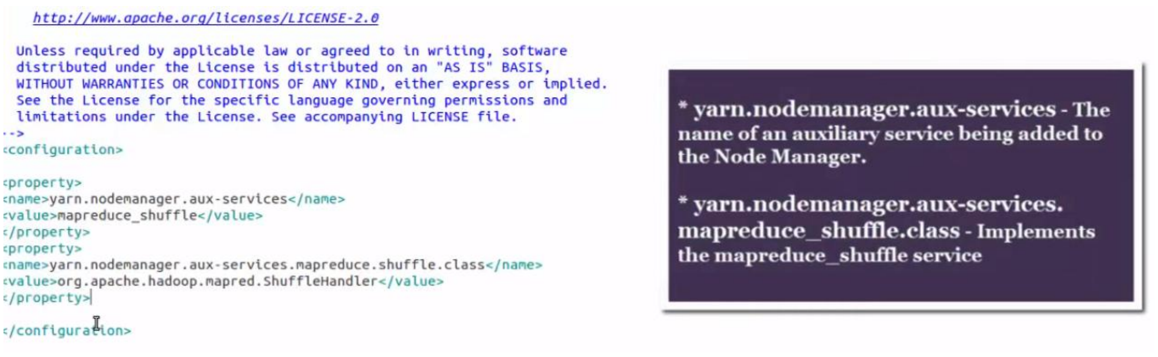
* dfs.datanode.data.dir - The DataNode stores the actual blocks of file data in the local file system in the specified directory.



```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo gedit yarn-site.xml
```

Configuration of yarn-site.xml

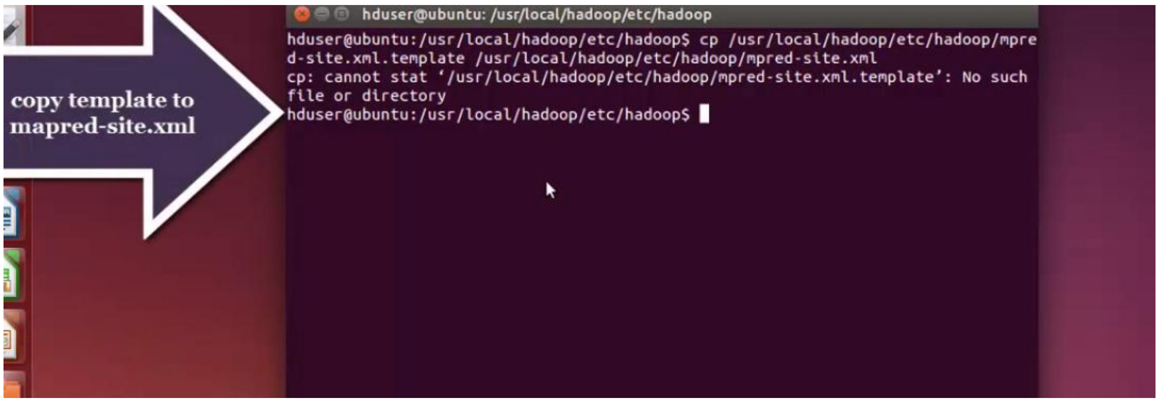
yarn-site.xml file enables you to overwrite a number of default values controlling the YARN components.



```
   http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<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>
```

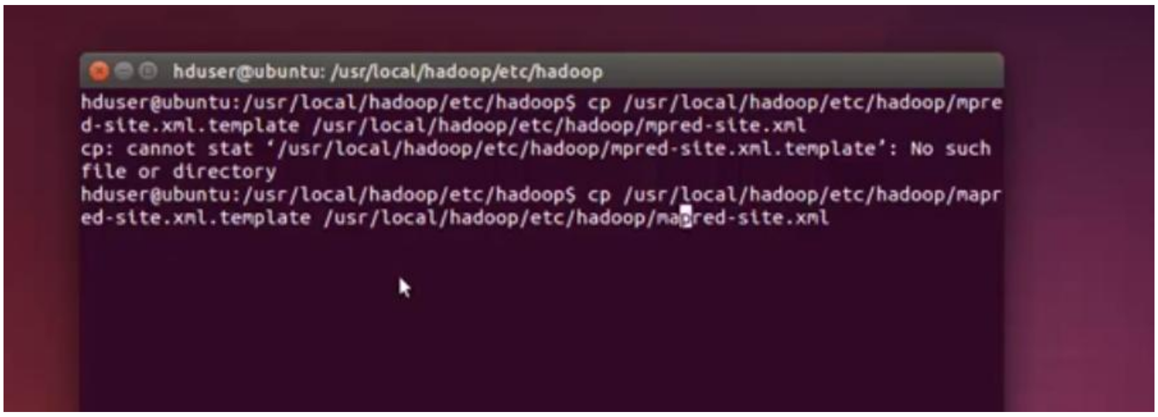
* yarn.nodemanager.aux-services - The name of an auxiliary service being added to the Node Manager.

* yarn.nodemanager.aux-services. mapreduce_shuffle.class - Implements the mapreduce_shuffle service

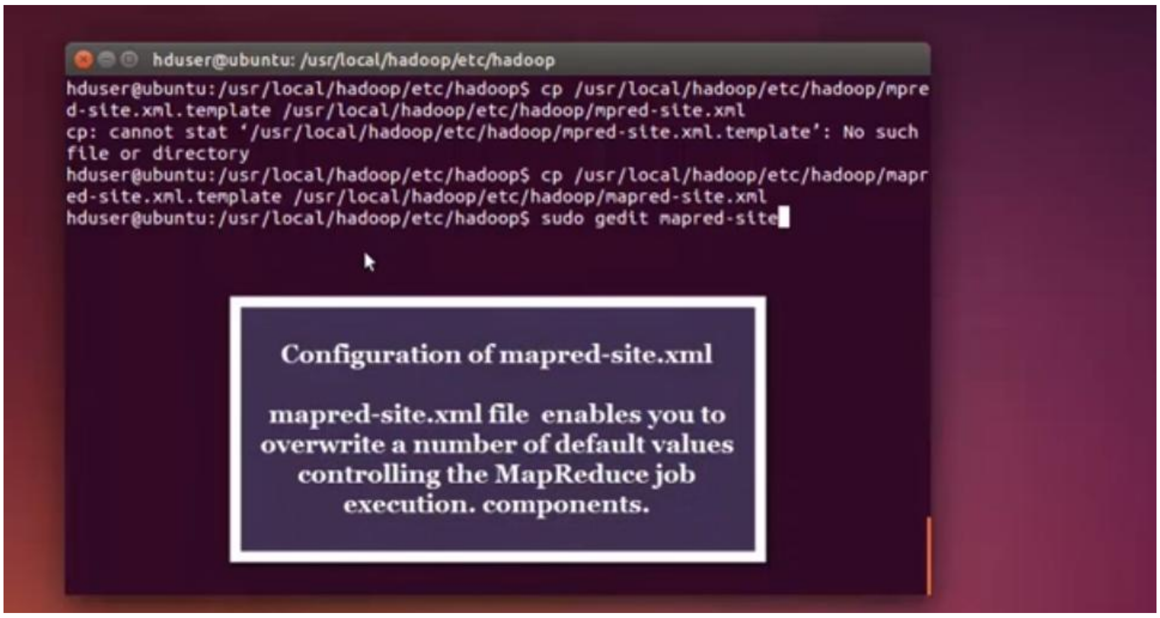


copy template to mapred-site.xml

```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cp /usr/local/hadoop/etc/hadoop/mpre
d-site.xml.template /usr/local/hadoop/etc/hadoop/mpred-site.xml
cp: cannot stat '/usr/local/hadoop/etc/hadoop/mpred-site.xml.template': No such
file or directory
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ 
```



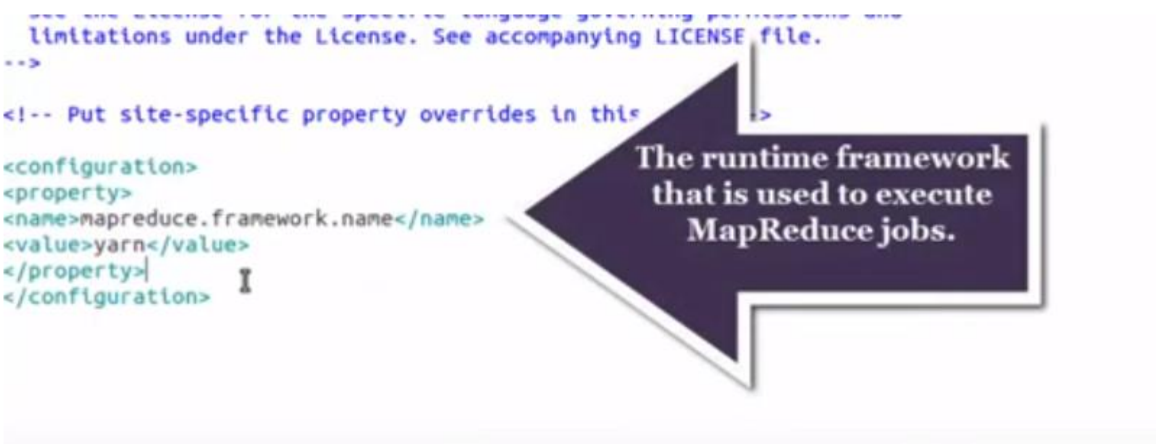
```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cp /usr/local/hadoop/etc/hadoop/mpre
d-site.xml.template /usr/local/hadoop/etc/hadoop/mpred-site.xml
cp: cannot stat '/usr/local/hadoop/etc/hadoop/mpred-site.xml.template': No such
file or directory
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cp /usr/local/hadoop/etc/hadoop/mapr
ed-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```



```
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cp /usr/local/hadoop/etc/hadoop/mpre
d-site.xml.template /usr/local/hadoop/etc/hadoop/mpred-site.xml
cp: cannot stat '/usr/local/hadoop/etc/hadoop/mpred-site.xml.template': No such
file or directory
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cp /usr/local/hadoop/etc/hadoop/mapr
ed-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo gedit mapred-site
```

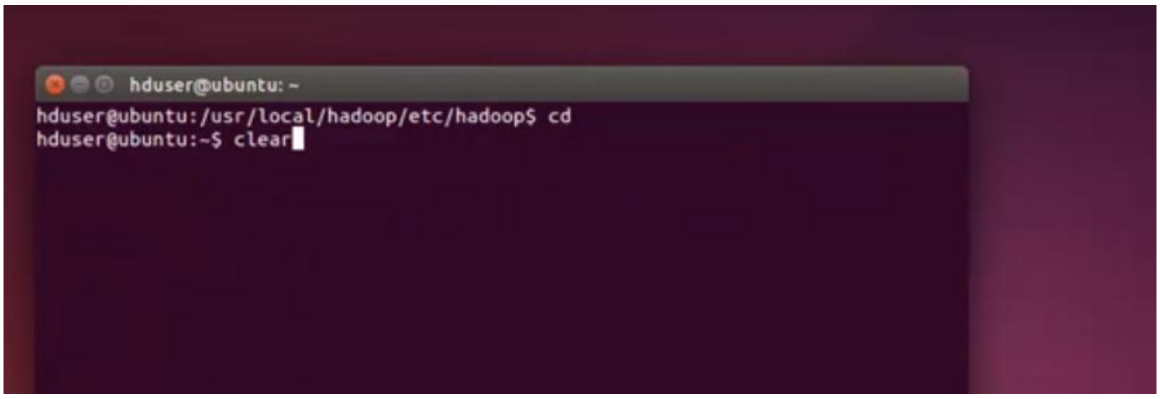
Configuration of mapred-site.xml

mapred-site.xml file enables you to overwrite a number of default values controlling the MapReduce job execution. components.

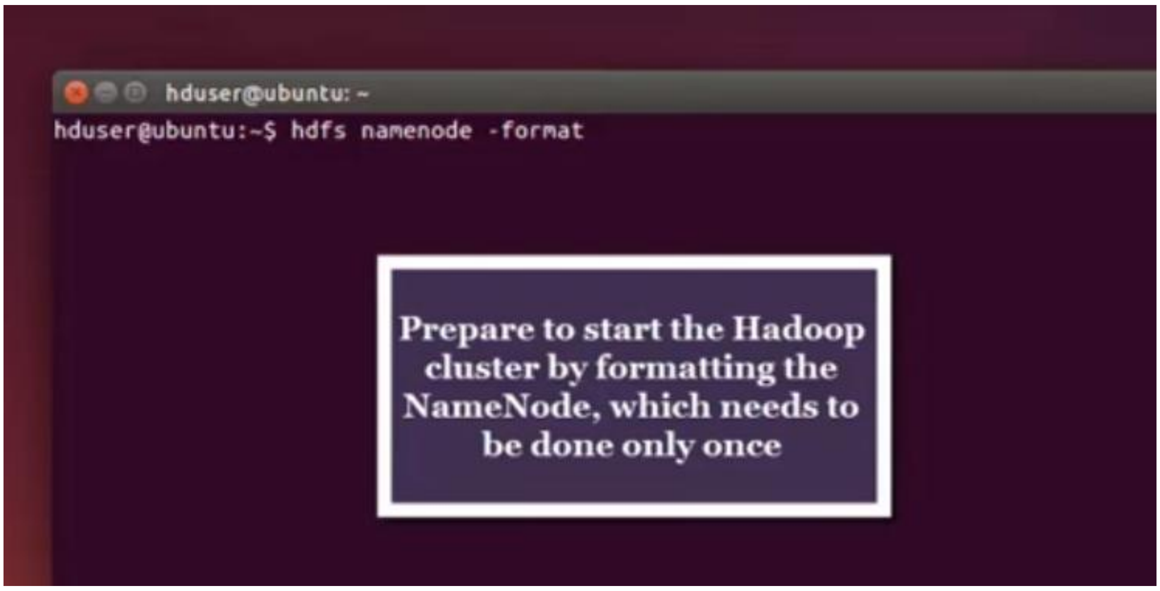


```
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

The runtime framework that is used to execute MapReduce jobs.

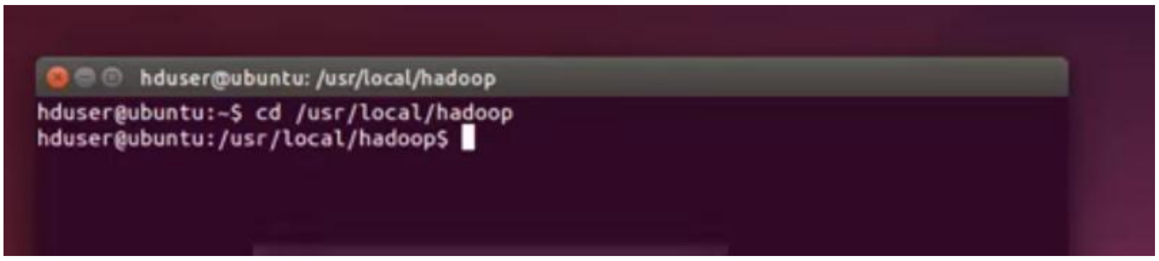


```
hduser@ubuntu: ~
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cd
hduser@ubuntu:~$ clear
```

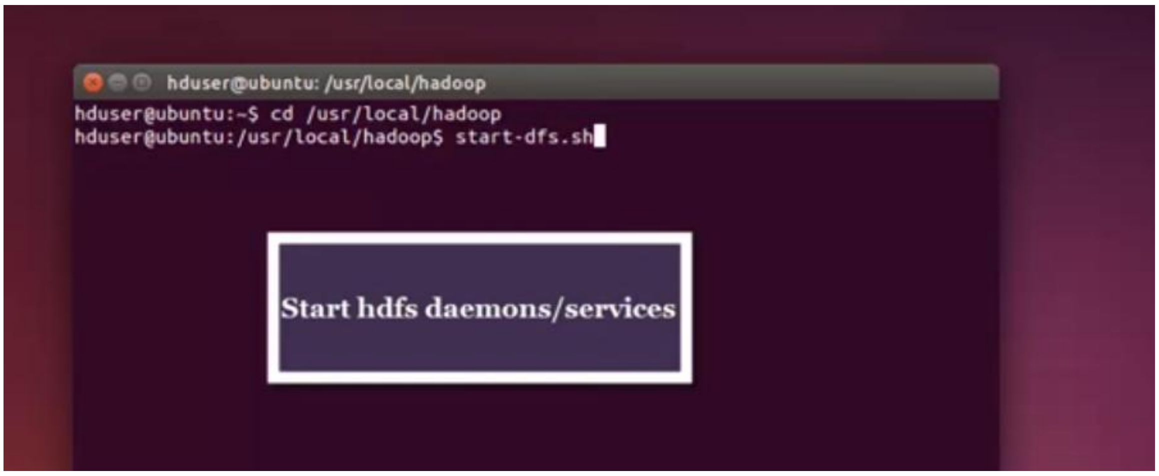


```
hduser@ubuntu: ~
hduser@ubuntu:~$ hdfs namenode -format
```

Prepare to start the Hadoop cluster by formatting the NameNode, which needs to be done only once

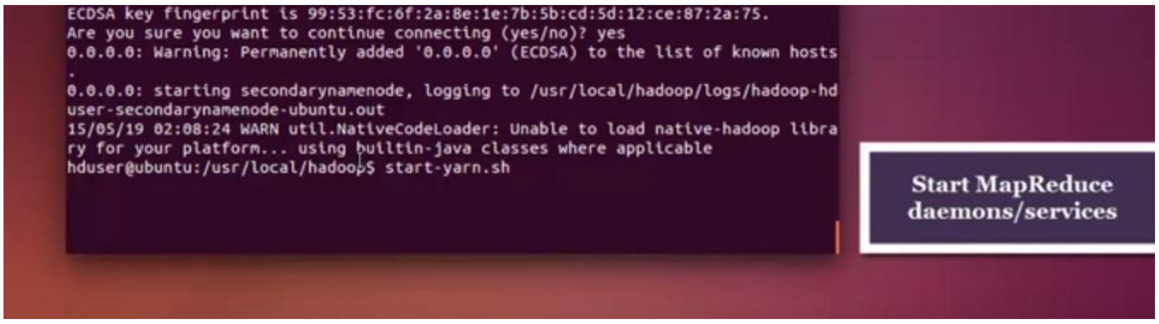


```
hduser@ubuntu: /usr/local/hadoop
hduser@ubuntu:~$ cd /usr/local/hadoop
hduser@ubuntu:/usr/local/hadoop$
```



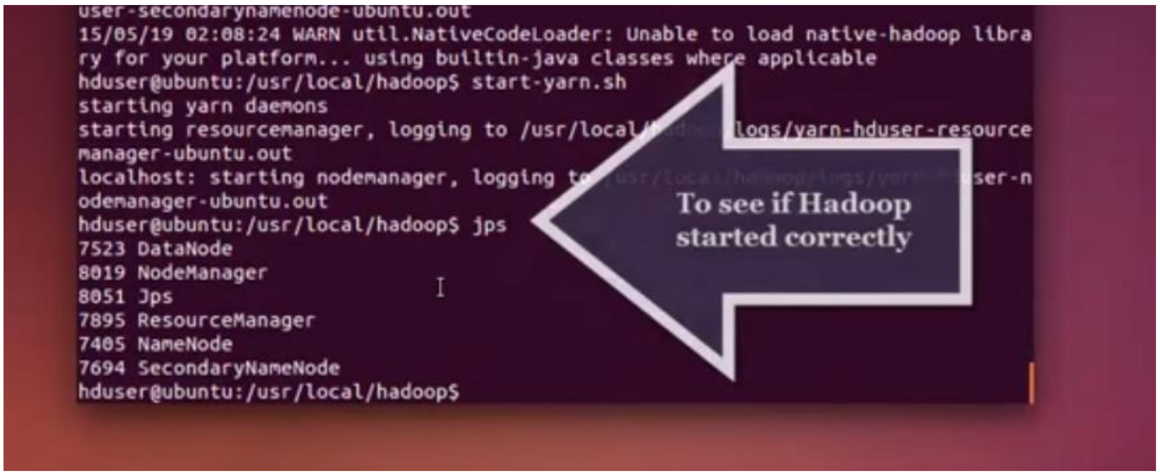
```
hduser@ubuntu: /usr/local/hadoop
hduser@ubuntu:~$ cd /usr/local/hadoop
hduser@ubuntu:/usr/local/hadoop$ start-dfs.sh
```

Start hdfs daemons/services

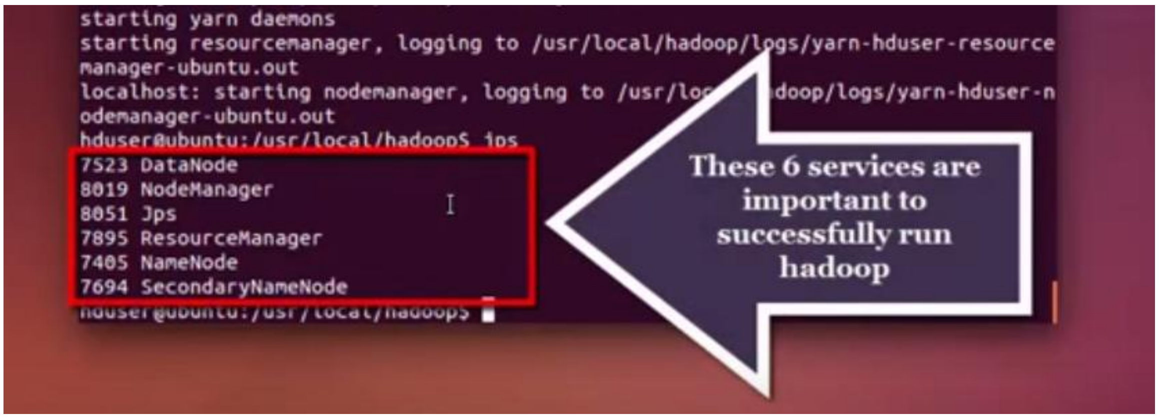


```
ECDSA key fingerprint is 99:53:fc:6f:2a:8e:1e:7b:5b:cd:5d:12:ce:87:2a:75.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts
.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-ubuntu.out
15/05/19 02:08:24 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
hduser@ubuntu:/usr/local/hadoop$ start-yarn.sh
```

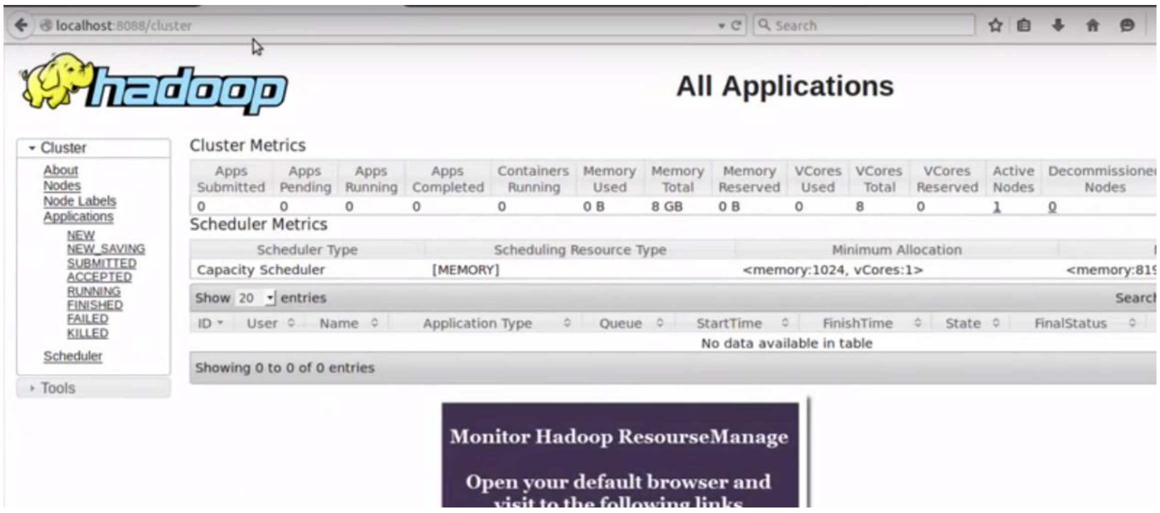
Start MapReduce daemons/services



```
user-secondarynamenode-ubuntu.out
15/05/19 02:08:24 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
hduser@ubuntu:/usr/local/hadoop$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-ubuntu.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-ubuntu.out
hduser@ubuntu:/usr/local/hadoop$ jps
7523 DataNode
8019 NodeManager
8051 Jps
7895 ResourceManager
7405 NameNode
7694 SecondaryNameNode
hduser@ubuntu:/usr/local/hadoop$
```



```
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-ubuntu.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-ubuntu.out
hduser@ubuntu:/usr/local/hadoop$ jps
7523 DataNode
8019 NodeManager
8051 Jps
7895 ResourceManager
7405 NameNode
7694 SecondaryNameNode
hduser@ubuntu:/usr/local/hadoop$
```



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0

Scheduler Metrics

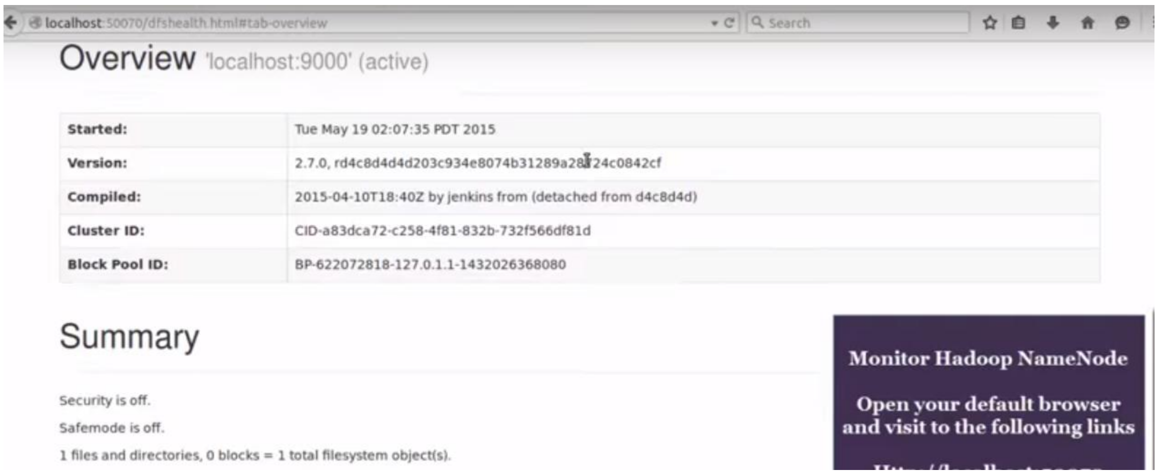
Scheduler Type	Scheduling Resource Type	Minimum Allocation	
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:819

Show 20 entries Search

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
No data available in table								

Showing 0 to 0 of 0 entries

Monitor Hadoop ResourseManage
Open your default browser and visit to the following links



localhost:50070/dfshealth.html#tab-overview

Overview 'localhost:9000' (active)

Started:	Tue May 19 02:07:35 PDT 2015
Version:	2.7.0, rd4c8d4d4d203c934e8074b31289a28f24c0842cf
Compiled:	2015-04-10T18:40Z by jenkins from (detached from d4c8d4d)
Cluster ID:	CID-a83dca72-c258-4f81-832b-732f566df81d
Block Pool ID:	BP-622072818-127.0.1.1-1432026368080

Summary

Security is off.

Safemode is off.

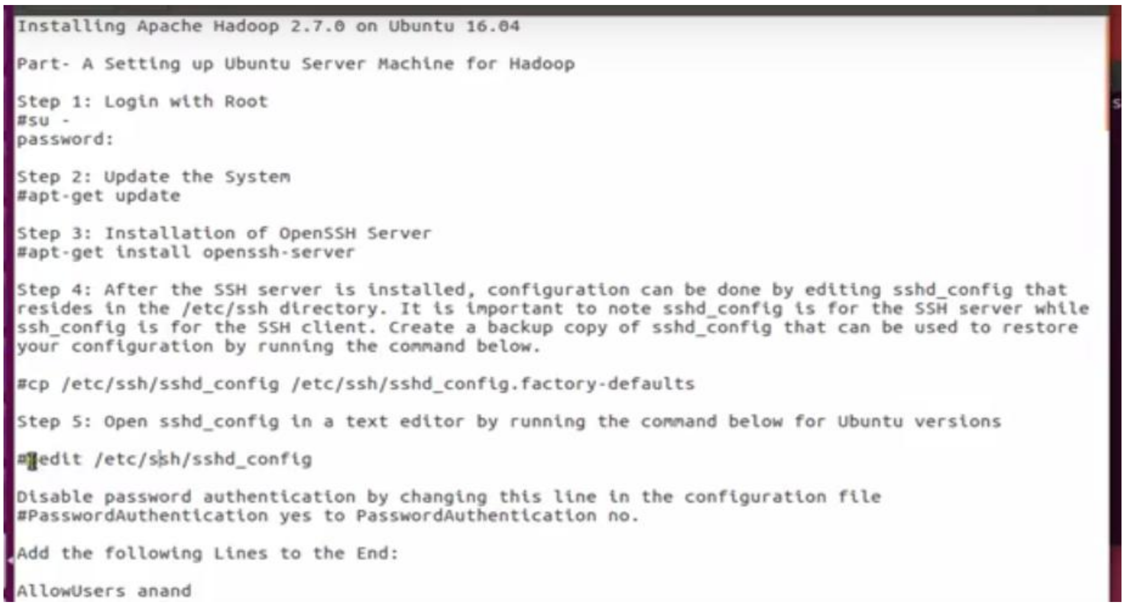
1 files and directories, 0 blocks = 1 total filesystem object(s).

Monitor Hadoop NameNode

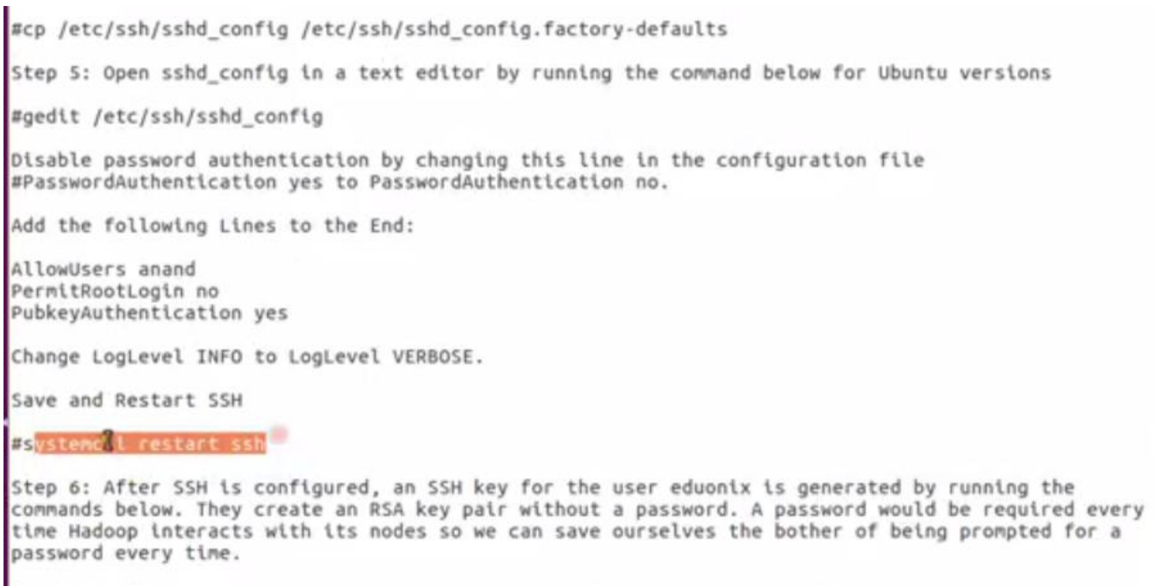
Open your default browser and visit to the following links

- when we format the name node: error: Error: Could not find or load main class ”-Djava.library.path=.usr.local.hadoop.lib” I had installed java-7-opensdk. I dont know hat was the issue, I installed java-8-oracle(which is shown here) and it worked 2. while formating namenode error: hadoop command not found: I modified my Path variable in .bashrc file it was untill hadoop/bin I added one more hadoop, i.e export PATH=$PATH:/usr/local/hadoop/bin/hadoop 3. while starting dfs.sh I faced a problem where is was asking for passphrase for id_rsa, I again generated the ssh key and added it to authorized key as it is shown here.
  change: cat $HOME/.ssh/id_rsa.pub ＞＞ $HOME/.ssh/authorized_keys for

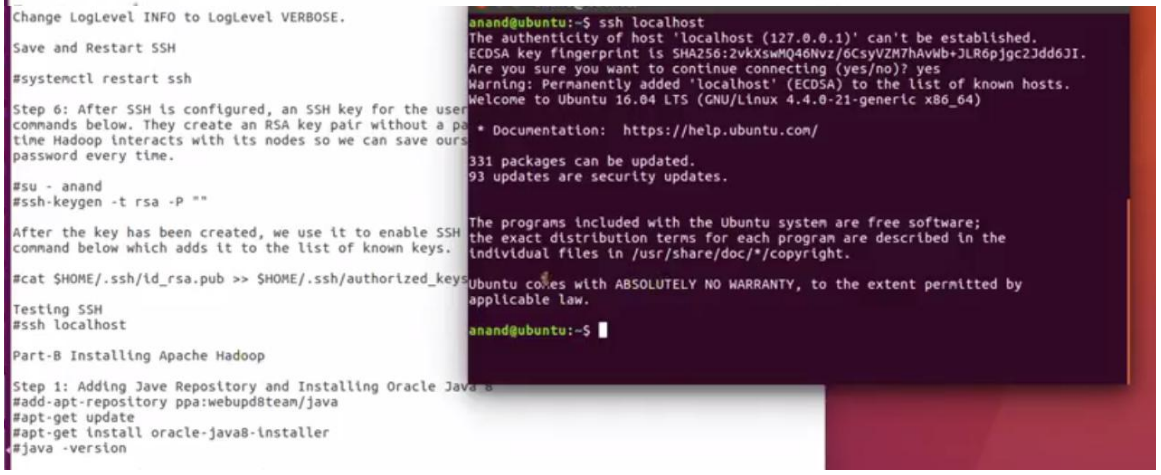
cat
 ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys



```
Installing Apache Hadoop 2.7.0 on Ubuntu 16.04

Part- A Setting up Ubuntu Server Machine for Hadoop

Step 1: Login with Root
#su -
password:

Step 2: Update the System
#apt-get update

Step 3: Installation of OpenSSH Server
#apt-get install openssh-server

Step 4: After the SSH server is installed, configuration can be done by editing sshd_config that
resides in the /etc/ssh directory. It is important to note sshd_config is for the SSH server while
ssh_config is for the SSH client. Create a backup copy of sshd_config that can be used to restore
your configuration by running the command below.

#cp /etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults

Step 5: Open sshd_config in a text editor by running the command below for Ubuntu versions

#gedit /etc/ssh/sshd_config

Disable password authentication by changing this line in the configuration file
#PasswordAuthentication yes to PasswordAuthentication no.

Add the following Lines to the End:

AllowUsers anand
```



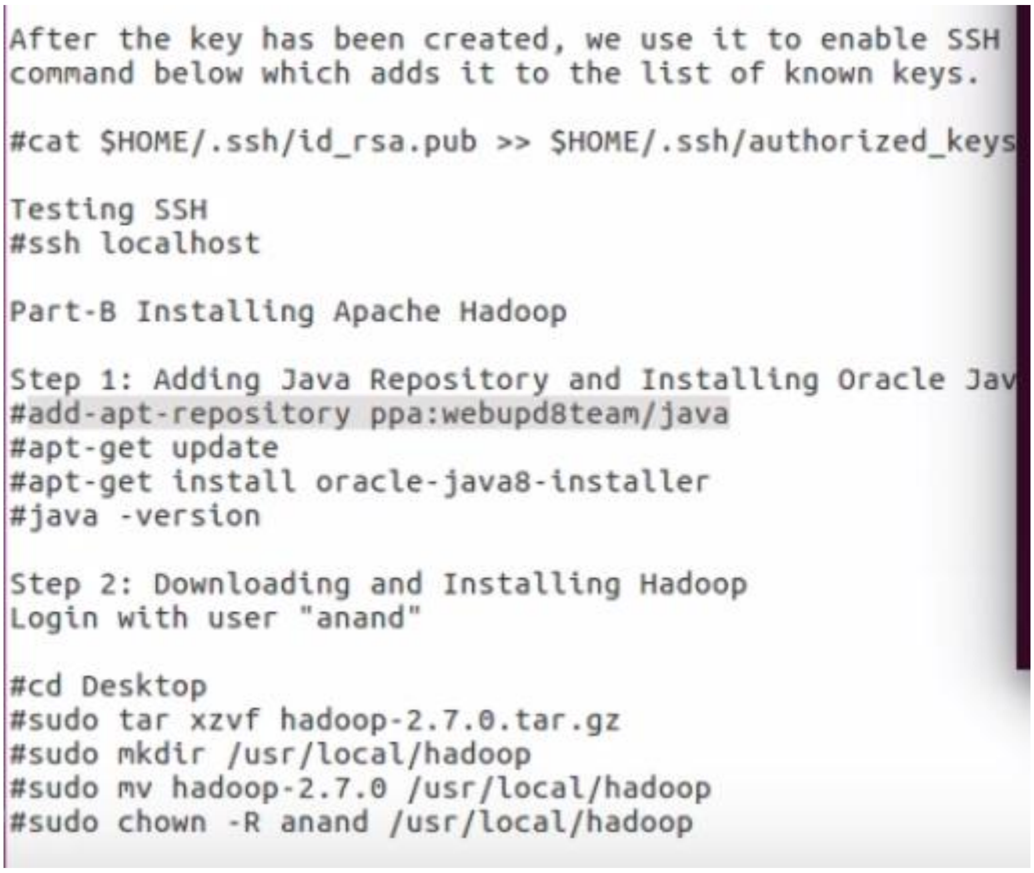
```
#cp /etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults

Step 5: Open sshd_config in a text editor by running the command below for Ubuntu versions

#gedit /etc/ssh/sshd_config

Disable password authentication by changing this line in the configuration file
#PasswordAuthentication yes to PasswordAuthentication no.

Add the following Lines to the End:

AllowUsers anand
PermitRootLogin no
PubkeyAuthentication yes

Change LogLevel INFO to LogLevel VERBOSE.

Save and Restart SSH

#systemctl restart ssh

Step 6: After SSH is configured, an SSH key for the user eduonix is generated by running the
commands below. They create an RSA key pair without a password. A password would be required every
time Hadoop interacts with its nodes so we can save ourselves the bother of being prompted for a
password every time.
```



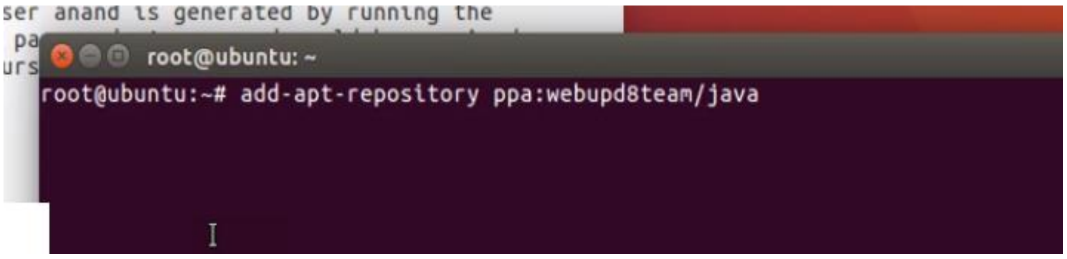
```
Change LogLevel INFO to LogLevel VERBOSE.

Save and Restart SSH

#systemctl restart ssh

Step 6: After SSH is configured, an SSH key for the user
commands below. They create an RSA key pair without a pa
time Hadoop interacts with its nodes so we can save ours
password every time.
```

```
#su - anand
#ssh-keygen -t rsa -P ""

After the key has been created, we use it to enable SSH
command below which adds it to the list of known keys.

#cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

Testing SSH
#ssh localhost

Part-B Installing Apache Hadoop

Step 1: Adding Jave Repository and Installing Oracle Java 8
#add-apt-repository ppa:webupd8team/java
#apt-get update
#apt-get install oracle-java8-installer
#java -version
```

```
anand@ubuntu:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:2vkXswMQ46Nvz/6CsyVZM7hAvWb+JLR6pjgc2Jdd6JI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

331 packages can be updated.
93 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

anand@ubuntu:~$
```

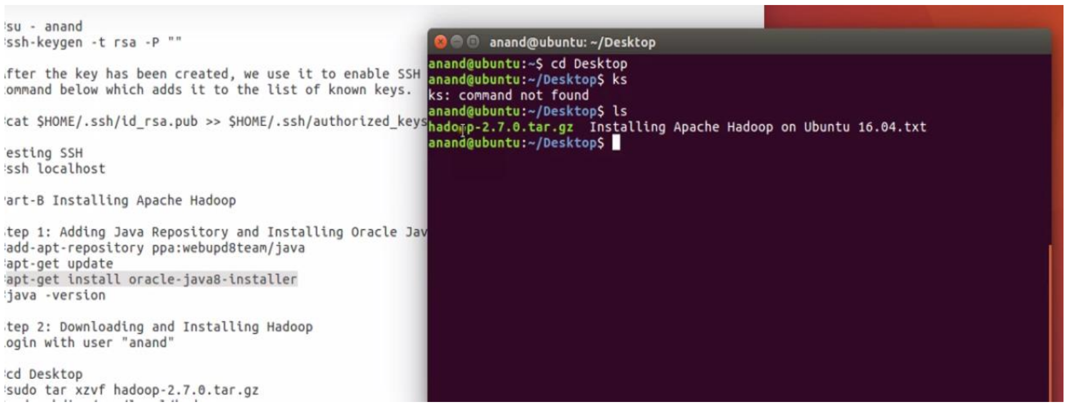


```
After the key has been created, we use it to enable SSH
command below which adds it to the list of known keys.

#cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

Testing SSH
#ssh localhost

Part-B Installing Apache Hadoop

Step 1: Adding Java Repository and Installing Oracle Jav
#add-apt-repository ppa:webupd8team/java
#apt-get update
#apt-get install oracle-java8-installer
#java -version

Step 2: Downloading and Installing Hadoop
Login with user "anand"

#cd Desktop
#sudo tar xzvf hadoop-2.7.0.tar.gz
#sudo mkdir /usr/local/hadoop
#sudo mv hadoop-2.7.0 /usr/local/hadoop
#sudo chown -R anand /usr/local/hadoop
```



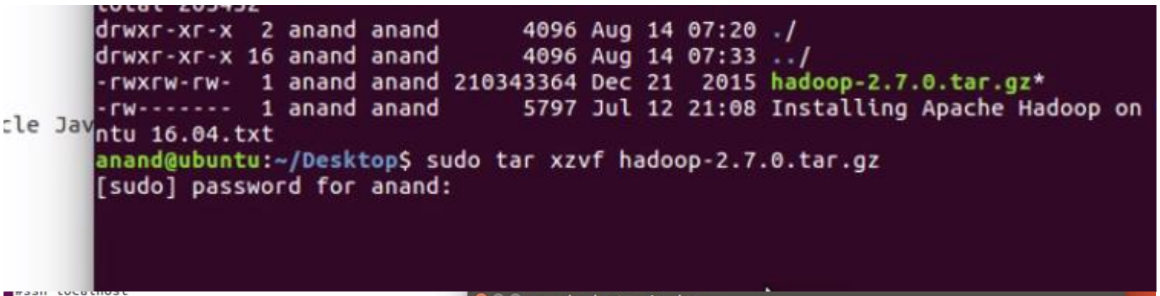
```
ser anand is generated by running the
```

```
root@ubuntu:~# add-apt-repository ppa:webupd8team/java
```

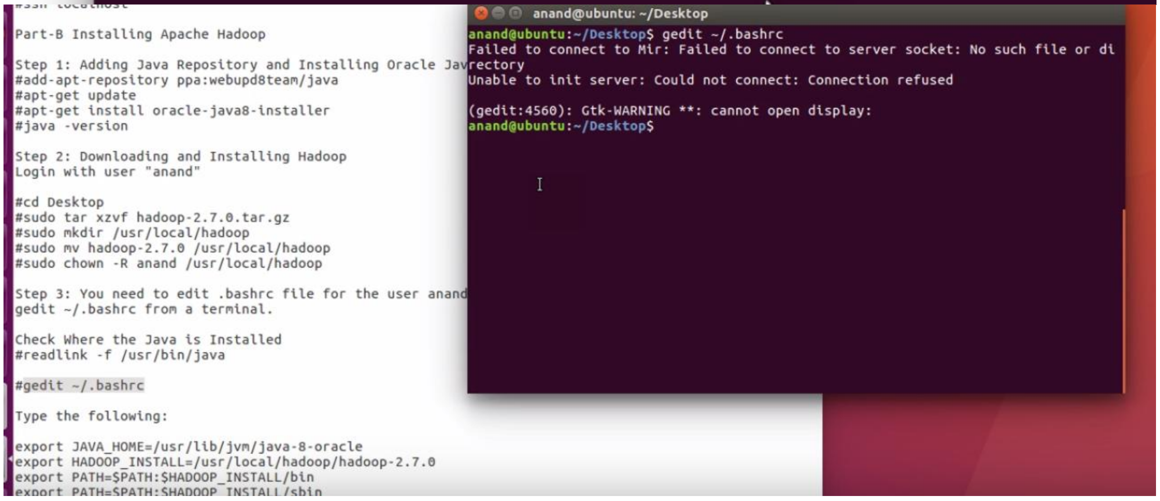


```
su - anand
ssh-keygen -t rsa -P ""

fter the key has been created, we use it to enable SSH
ommand below which adds it to the list of known keys.

cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

esting SSH
ssh localhost

art-B Installing Apache Hadoop

tep 1: Adding Java Repository and Installing Oracle Jav
add-apt-repository ppa:webupd8team/java
apt-get update
apt-get install oracle-java8-installer
java -version

tep 2: Downloading and Installing Hadoop
ogin with user "anand"

cd Desktop
sudo tar xzvf hadoop-2.7.0.tar.gz
```

```
anand@ubuntu:~$ cd Desktop
anand@ubuntu:~/Desktop$ ks
ks: command not found
anand@ubuntu:~/Desktop$ ls
hadoop-2.7.0.tar.gz  Installing Apache Hadoop on Ubuntu 16.04.txt
anand@ubuntu:~/Desktop$
```



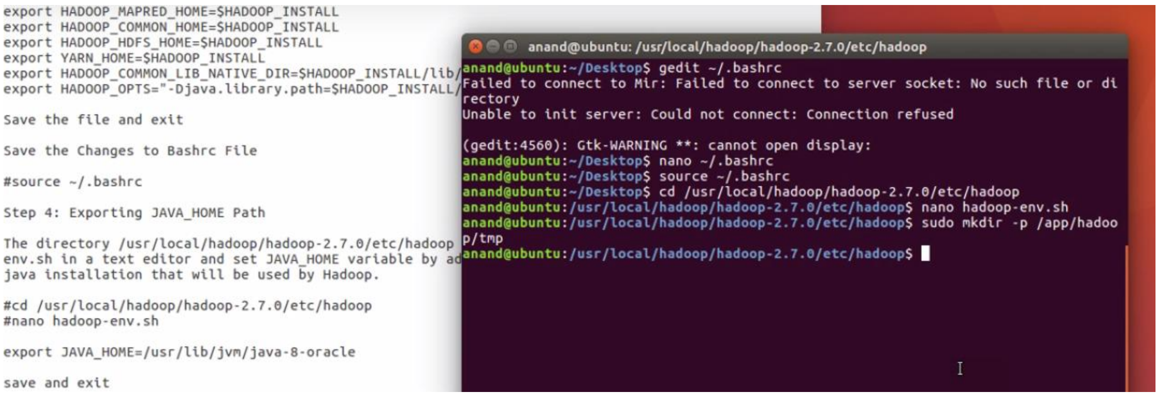
```
drwxr-xr-x  2 anand anand      4096 Aug 14 07:20 ./
drwxr-xr-x 16 anand anand      4096 Aug 14 07:33 ../
-rwxrw-rw-  1 anand anand 210343364 Dec 21  2015 hadoop-2.7.0.tar.gz*
-rw-------  1 anand anand      5797 Jul 12 21:08 Installing Apache Hadoop on
ntu 16.04.txt
anand@ubuntu:~/Desktop$ sudo tar xzvf hadoop-2.7.0.tar.gz
[sudo] password for anand:
```



```
Part-B Installing Apache Hadoop

Step 1: Adding Java Repository and Installing Oracle Jav
#add-apt-repository ppa:webupd8team/java
#apt-get update
#apt-get install oracle-java8-installer
#java -version

Step 2: Downloading and Installing Hadoop
Login with user "anand"

#cd Desktop
#sudo tar xzvf hadoop-2.7.0.tar.gz
#sudo mkdir /usr/local/hadoop
#sudo mv hadoop-2.7.0 /usr/local/hadoop
#sudo chown -R anand /usr/local/hadoop

Step 3: You need to edit .bashrc file for the user anand
gedit ~/.bashrc from a terminal.

Check Where the Java is Installed
#readlink -f /usr/bin/java

#gedit ~/.bashrc

Type the following:

export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_INSTALL=/usr/local/hadoop/hadoop-2.7.0
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
```

anand@ubuntu: ~/Desktop

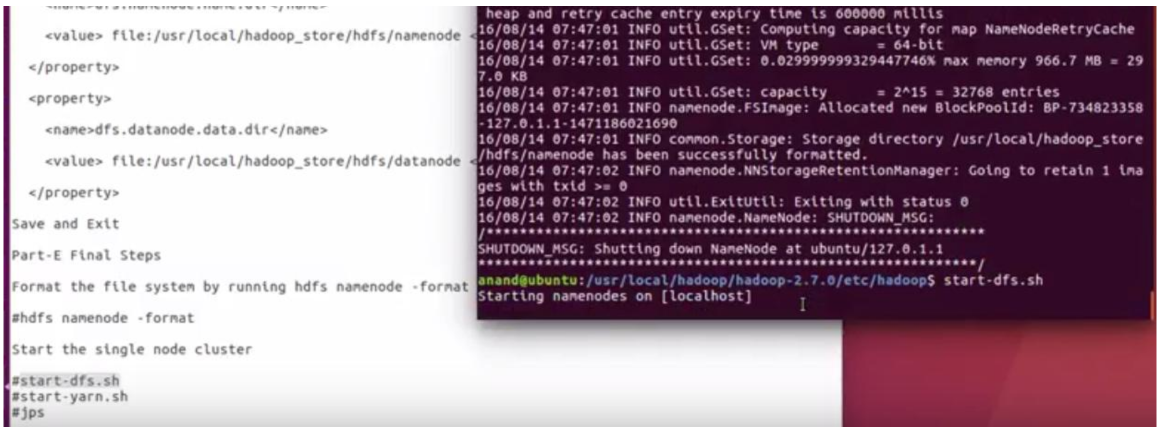
```
anand@ubuntu:~/Desktop$ gedit ~/.bashrc
Failed to connect to Mir: Failed to connect to server socket: No such file or di
rectory
Unable to init server: Could not connect: Connection refused

(gedit:4560): Gtk-WARNING **: cannot open display:
anand@ubuntu:~/Desktop$
```



```
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/

Save the file and exit

Save the Changes to Bashrc File

#source ~/.bashrc

Step 4: Exporting JAVA_HOME Path

The directory /usr/local/hadoop/hadoop-2.7.0/etc/hadoop
env.sh in a text editor and set JAVA_HOME variable by ad
java installation that will be used by Hadoop.

#cd /usr/local/hadoop/hadoop-2.7.0/etc/hadoop
#nano hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-oracle

save and exit
```

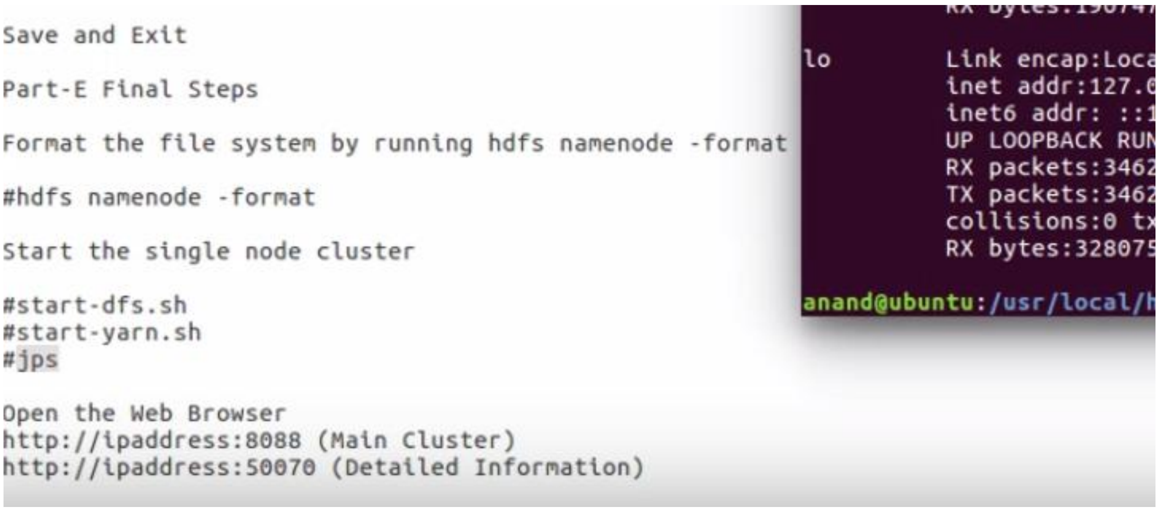
anand@ubuntu: /usr/local/hadoop/hadoop-2.7.0/etc/hadoop

```
anand@ubuntu:~/Desktop$ gedit ~/.bashrc
Failed to connect to Mir: Failed to connect to server socket: No such file or di
rectory
Unable to init server: Could not connect: Connection refused

(gedit:4560): Gtk-WARNING **: cannot open display:
anand@ubuntu:~/Desktop$ nano ~/.bashrc
anand@ubuntu:~/Desktop$ source ~/.bashrc
anand@ubuntu:~/Desktop$ cd /usr/local/hadoop/hadoop-2.7.0/etc/hadoop
anand@ubuntu:/usr/local/hadoop/hadoop-2.7.0/etc/hadoop$ nano hadoop-env.sh
anand@ubuntu:/usr/local/hadoop/hadoop-2.7.0/etc/hadoop$ sudo mkdir -p /app/hadoo
p/tmp
anand@ubuntu:/usr/local/hadoop/hadoop-2.7.0/etc/hadoop$
```



```
    <value> file:/usr/local/hadoop_store/hdfs/namenode <

  </property>

  <property>

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

    <value> file:/usr/local/hadoop_store/hdfs/datanode <

  </property>
Save and Exit

Part-E Final Steps

Format the file system by running hdfs namenode -format

#hdfs namenode -format

Start the single node cluster

#start-dfs.sh
#start-yarn.sh
#jps
```

```
 heap and retry cache entry expiry time is 600000 millis
16/08/14 07:47:01 INFO util.GSet: Computing capacity for map NameNodeRetryCache
16/08/14 07:47:01 INFO util.GSet: VM type       = 64-bit
16/08/14 07:47:01 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 29
7.0 KB
16/08/14 07:47:01 INFO util.GSet: capacity      = 2^15 = 32768 entries
16/08/14 07:47:01 INFO namenode.FSImage: Allocated new BlockPoolId: BP-734823358
-127.0.1.1-1471186021690
16/08/14 07:47:01 INFO common.Storage: Storage directory /usr/local/hadoop_store
/hdfs/namenode has been successfully formatted.
16/08/14 07:47:02 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
16/08/14 07:47:02 INFO util.ExitUtil: Exiting with status 0
16/08/14 07:47:02 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
************************************************************/
anand@ubuntu:/usr/local/hadoop/hadoop-2.7.0/etc/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
```



```
Save and Exit

Part-E Final Steps

Format the file system by running hdfs namenode -format

#hdfs namenode -format

Start the single node cluster

#start-dfs.sh
#start-yarn.sh
#jps

Open the Web Browser
http://ipaddress:8088 (Main Cluster)
http://ipaddress:50070 (Detailed Information)
```

```
lo        Link encap:Loca
          inet addr:127.0
          inet6 addr: ::1
          UP LOOPBACK RUN
          RX packets:3462
          TX packets:3462
          collisions:0 tx
          RX bytes:328075

anand@ubuntu:/usr/local/h
```

PRE LAB VIVA QUESTIONS?

1. What is meant by Hadoop?
2. What is meant by host OS?
3. Why the RAM size to be allocated to your guest OS?
4. Why to provide connectivity between guest OS and host OS?

POST LAB VIVA QUESTIONS?

1. If guest OS takes the 50% of your physical machine's RAM size what happens?
2. Why to create a Virtual Hard disk for your Virtual Machine?
3. What does .vdi extension file contains inside the virtual machine?
4. Define the " dynamically expanding disk"

EXPERIMENT – 5

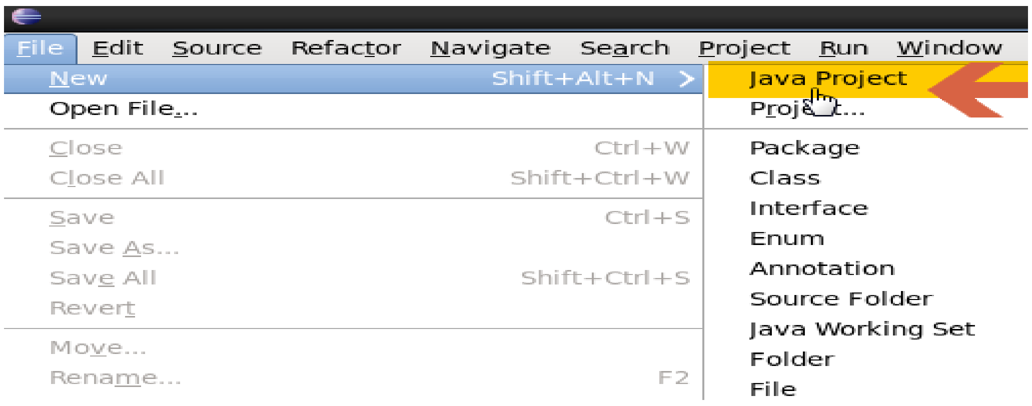
Develop hadoop application to count no of characters, no of words and each character frequency.

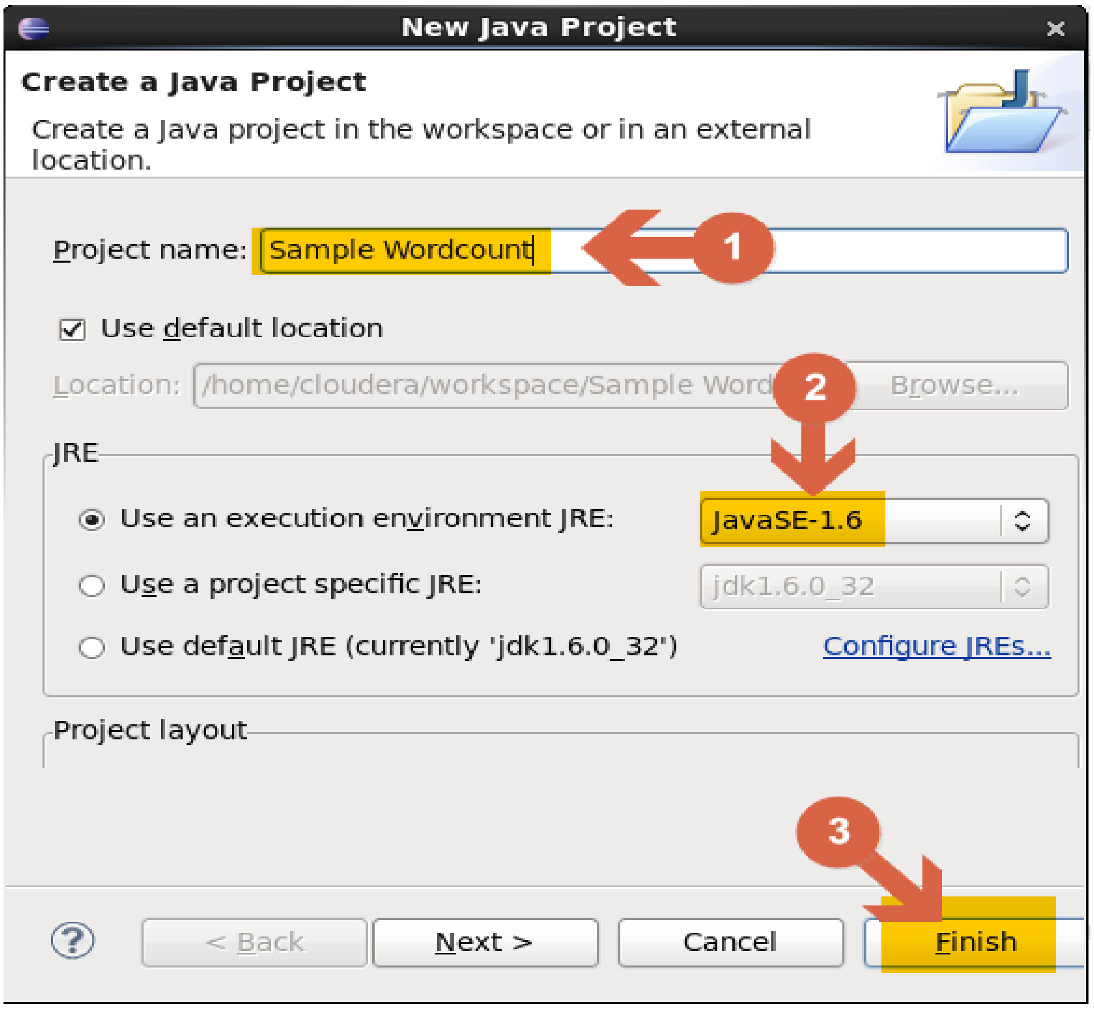
Hadoop WordCount operation occurs in 3 stages –
Mapper Phase
Shuffle Phase
Reducer Phase
Hadoop WordCount Example- Mapper Phase Execution
The text from the input text file is tokenized into words to form a key value pair with all the words present in the input text file. The key is the word from the input file and value is ‘1’.
For instance if you consider the sentence “An elephant is an animal”. The mapper phase in the WordCount example will split the string into individual tokens i.e. words. In this case, the entire sentence will be split into 5 tokens (one for each word) with a value 1 as shown below –
Key-Value pairs from Hadoop Map Phase Execution-
(an,1)
(elephant,1)
(is,1)
(an,1)
(animal,1)
If you would like more information about Big Data and Hadoop Certification, please click the orange "Request Info" button on top of this page.
Hadoop WordCount Example- Shuffle Phase Execution
After the map phase execution is completed successfully, shuffle phase is executed automatically wherein the key-value pairs generated in the map phase are taken as input and then sorted in alphabetical order. After the shuffle phase is executed from the WordCount example code, the output will look like this -
(an,1)
(an,1)
(animal,1)
(elephant,1)
(is,1)
Hadoop WordCount Example- Reducer Phase Execution
In the reduce phase, all the keys are grouped together and the values for similar keys are added up to find the occurrences for a particular word. It is like an aggregation phase for the keys generated by the map phase. The reducer phase takes the output of shuffle phase as input and then reduces the key-value pairs to unique keys with values added up. In our example “An elephant is an animal.” is the only word that appears twice in the sentence. After the execution of the reduce phase of MapReduce

WordCount example program, appears as a key only once but with a count of 2 as shown below
(an,2)
(animal,1)
(elephant,1)
(is,1)

- This is how the MapReduce word count program executes and outputs the number of occurrences of a word in any given input file. An important point to note during the execution of the WordCount example is that the mapper class in the WordCount program will execute completely on the entire input file and not just a single sentence. Suppose if the input file has 15 lines then the mapper class will split the words of all the 15 lines and form initial key value pairs for the entire dataset. The reducer execution will begin only after the mapper phase is executed successfully.
- Learn Hadoop by working on interesting Big Data and Hadoop Projects for just $9.
- Running the WordCount Example in Hadoop MapReduce using Java Project with Eclipse. Now, let's create the WordCount java project with eclipse IDE for Hadoop. Even if you are working on Cloudera VM, creating the Java project can be applied to any environment.

Step 1:
Let's create the java project with the name "Sample WordCount" as shown below -
File > New > Project > Java Project > Next.
"Sample WordCount" as our project name and click "Finish":





New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: Sample Wordcount

☑ Use default location

Location: /home/cloudera/workspace/Sample Word

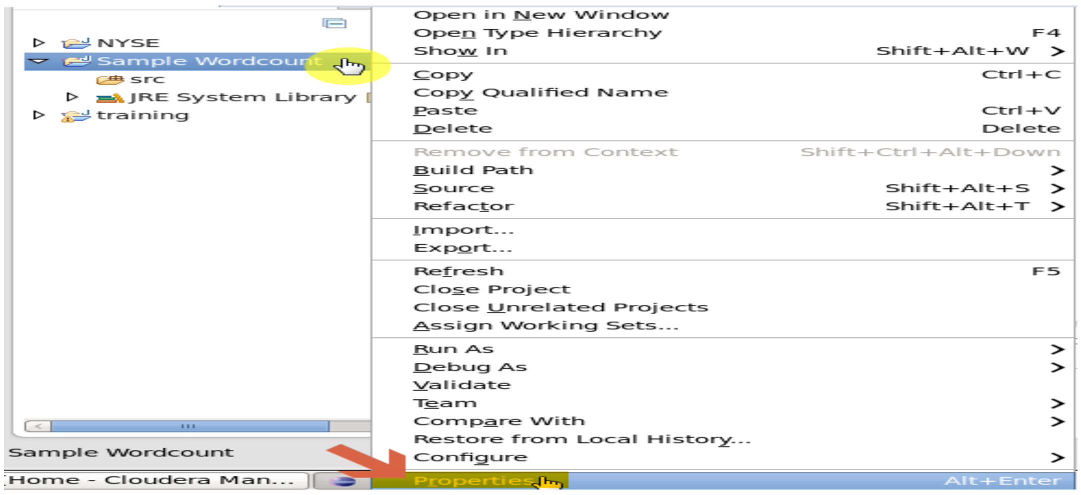
JRE

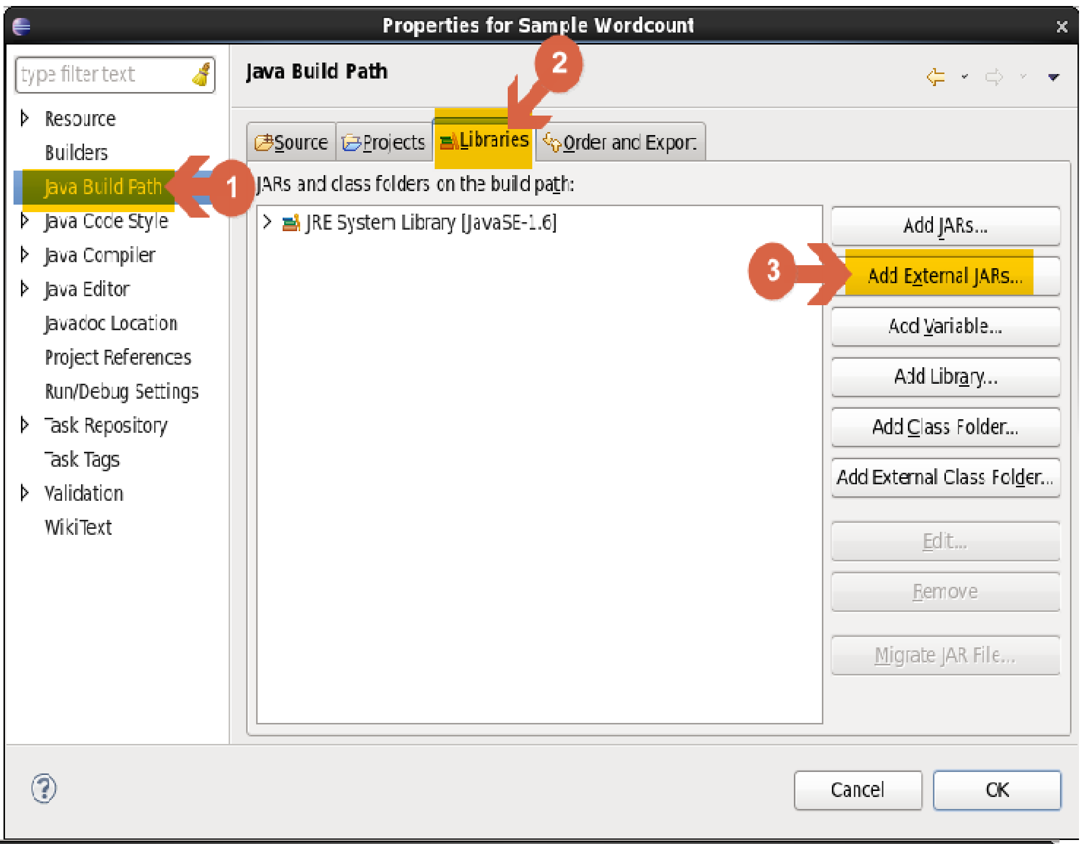
(●) Use an execution environment JRE: JavaSE-1.6

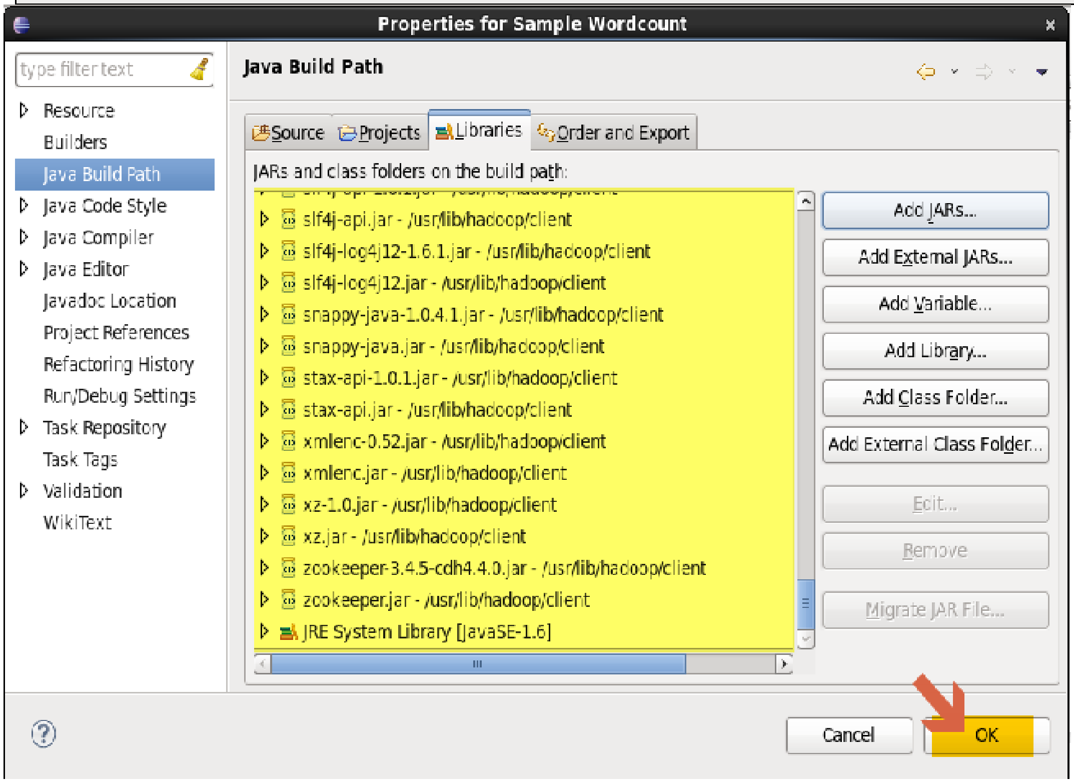
( ) Use a project specific JRE: jdk1.6.0_32

( ) Use default JRE (currently 'jdk1.6.0_32')

Step 2 :
The next step is to get references to hadoop libraries by clicking on Add JARS as follows

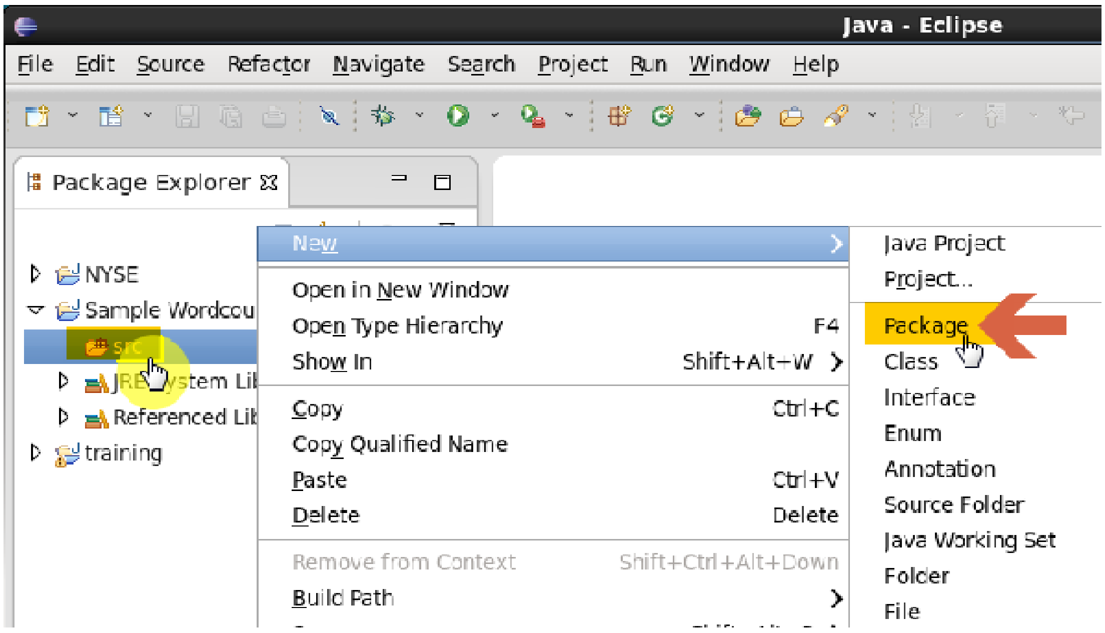


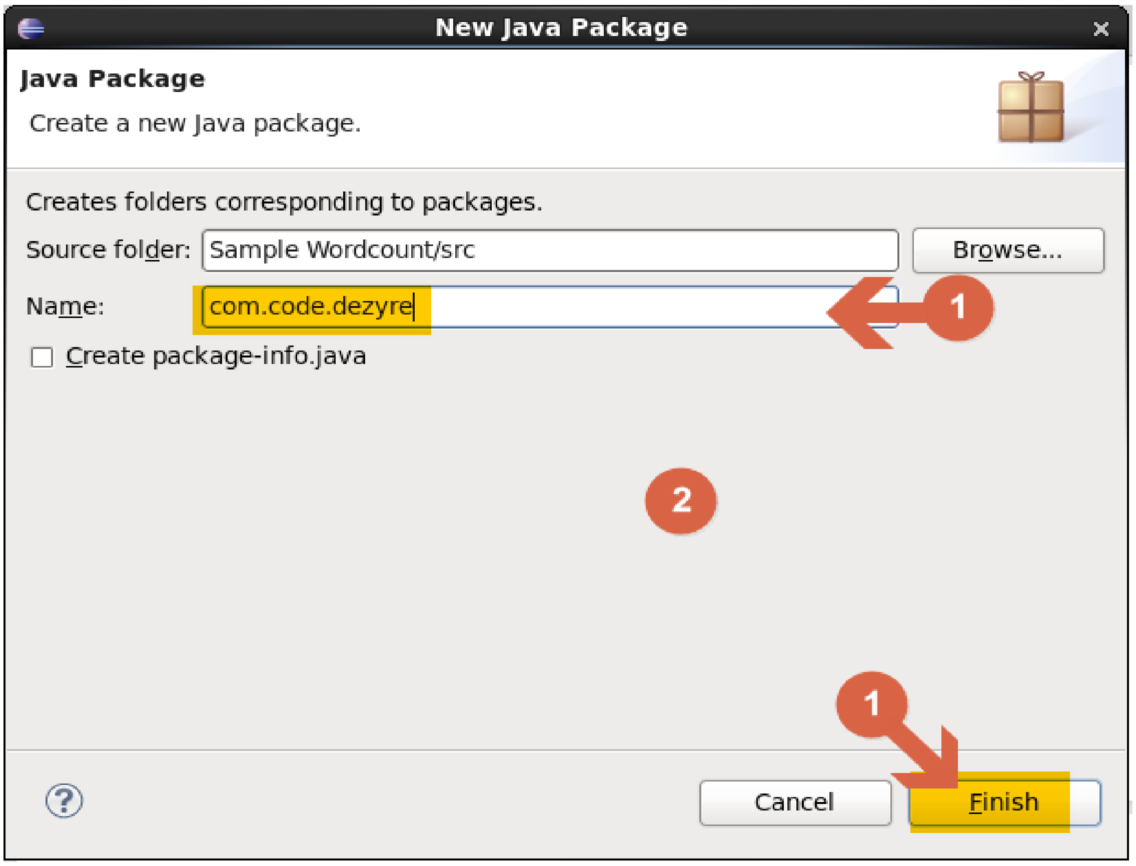




```
slf4j-api.jar - /usr/lib/hadoop/client
slf4j-log4j12-1.6.1.jar - /usr/lib/hadoop/client
slf4j-log4j12.jar - /usr/lib/hadoop/client
snappy-java-1.0.4.1.jar - /usr/lib/hadoop/client
snappy-java.jar - /usr/lib/hadoop/client
stax-api-1.0.1.jar - /usr/lib/hadoop/client
stax-api.jar - /usr/lib/hadoop/client
xmlenc-0.52.jar - /usr/lib/hadoop/client
xmlenc.jar - /usr/lib/hadoop/client
xz-1.0.jar - /usr/lib/hadoop/client
xz.jar - /usr/lib/hadoop/client
zookeeper-3.4.5-cdh4.4.0.jar - /usr/lib/hadoop/client
zookeeper.jar - /usr/lib/hadoop/client
JRE System Library [JavaSE-1.6]
```

Step 3:
Create a new package within the project with the name com.code.dezyre





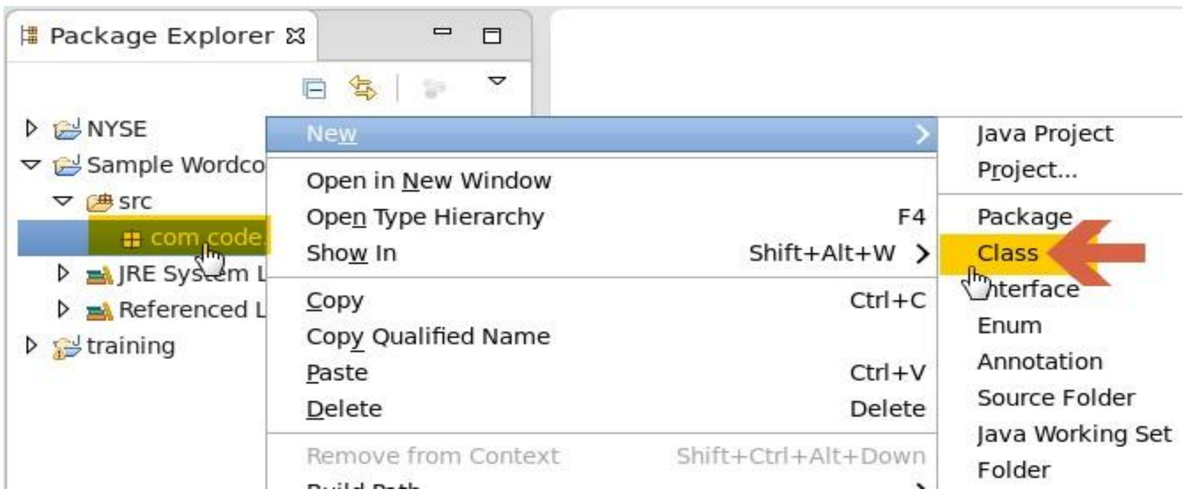
Creates folders corresponding to packages.

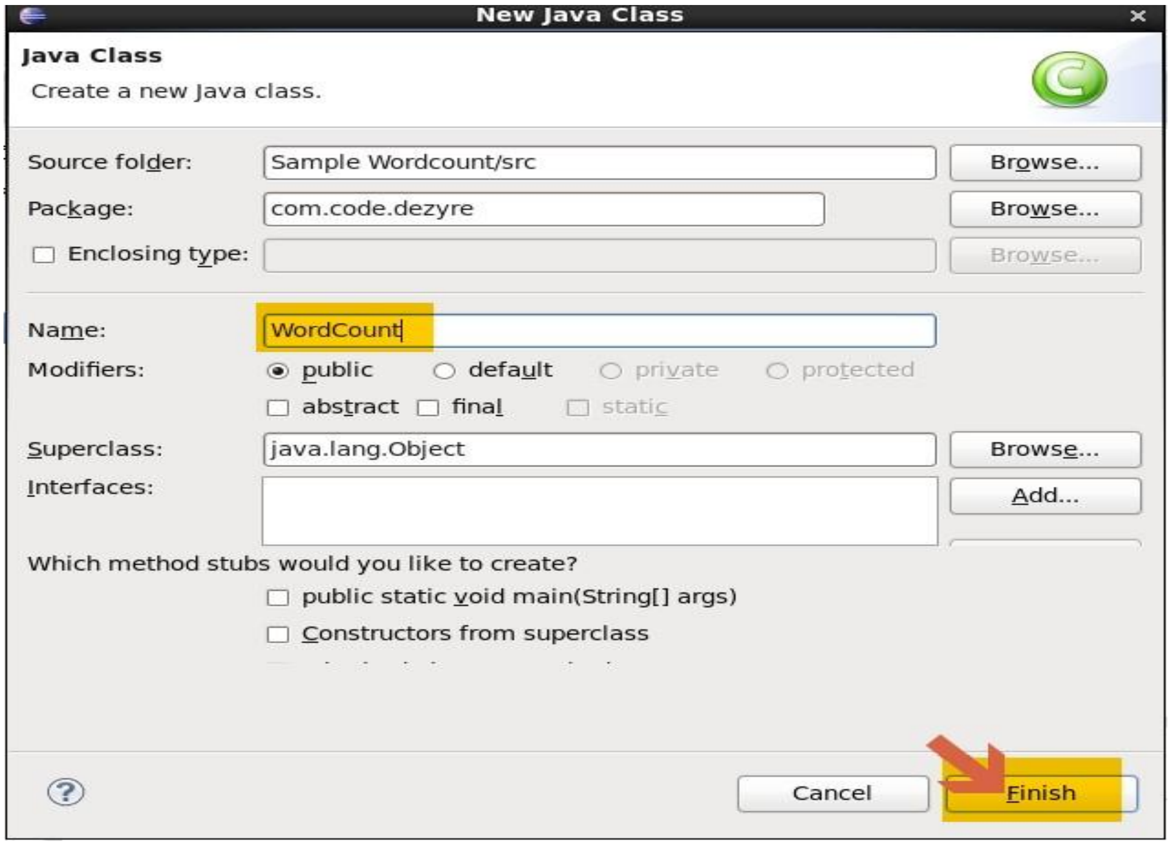
Source folder: Sample Wordcount/src

Name: com.code.dezyre

☐ Create package-info.java

Step 4:
Now let’s implement the WordCount example program by creating a WordCount class under the project com.code.dezyre.





Source folder: Sample Wordcount/src
Package: com.code.dezyre
☐ Enclosing type:
Name: WordCount
Modifiers: [✓ public] default private protected
☐ abstract ☐ final ☐ static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass

Step 5

Create a Mapper class within the WordCount class which extends MapReduceBase Class to implement mapper interface. The mapper class will contain -

1. Code to implement "map" method.
2. Code for implementing the mapper-stage business logic should be written within this method.

Mapper Class Code for WordCount Example in Hadoop MapReduce

```
public static class Map extends MapReduceBase implements Mapper
{
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(LongWritable key, Text value, OutputCollector
        output, Reporter reporter)
        throws IOException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
                }
        }
}
```

In the mapper class code, we have used the String Tokenizer class which takes the entire line and breaks into small tokens (string/word).

Step 6 :

Create a Reducer class within the WordCount class extending MapReduceBase Class to implement reducer interface. The reducer class for the wordcount example in hadoop will contain the -

1. Code to implement "reduce" method
2. Code for implementing the reducer-stage business logic should be written within this method

Reducer Class Code for WordCount Example in Hadoop MapReduce

```
public static class Reduce extends MapReduceBase implements Reducer {
        public void reduce(Text key, Iterator values, OutputCollector
output,
                Reporter reporter) throws IOException {
                int sum = 0;
                while (values.hasNext()) {
                        sum += values.next().get();
                }
                output.collect(key, new IntWritable(sum));
        }
}
```

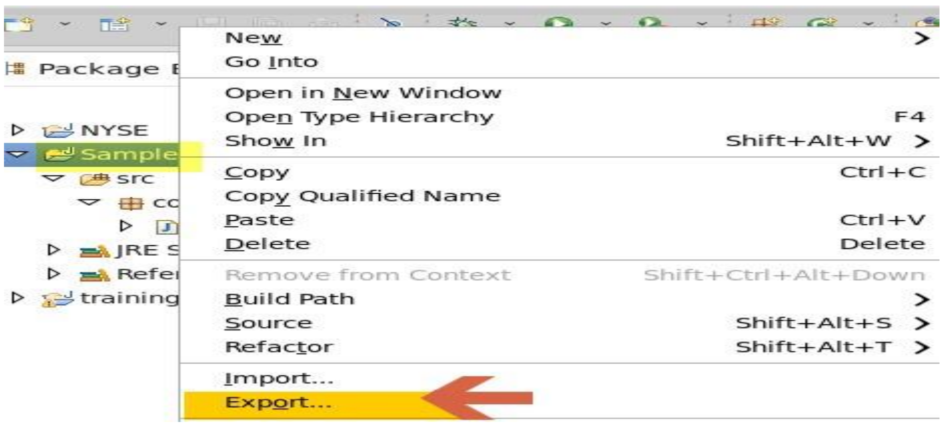
Step 7 :

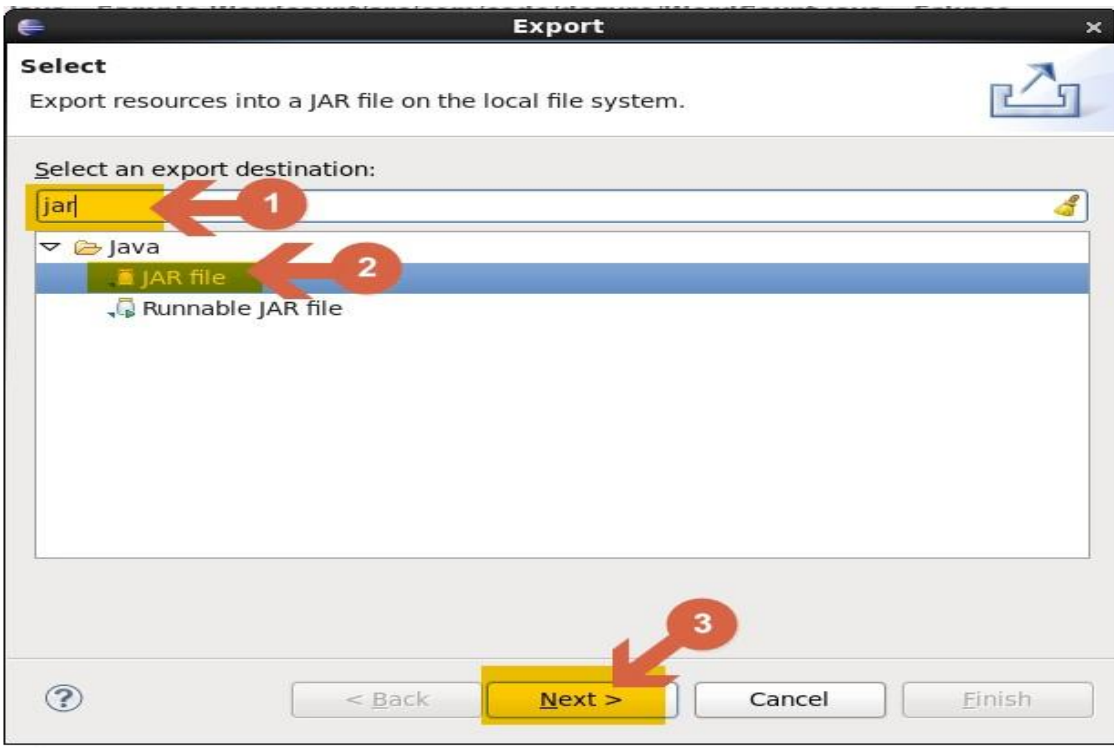
Create main() method within the WordCount class and set the following properties using the JobConf class -

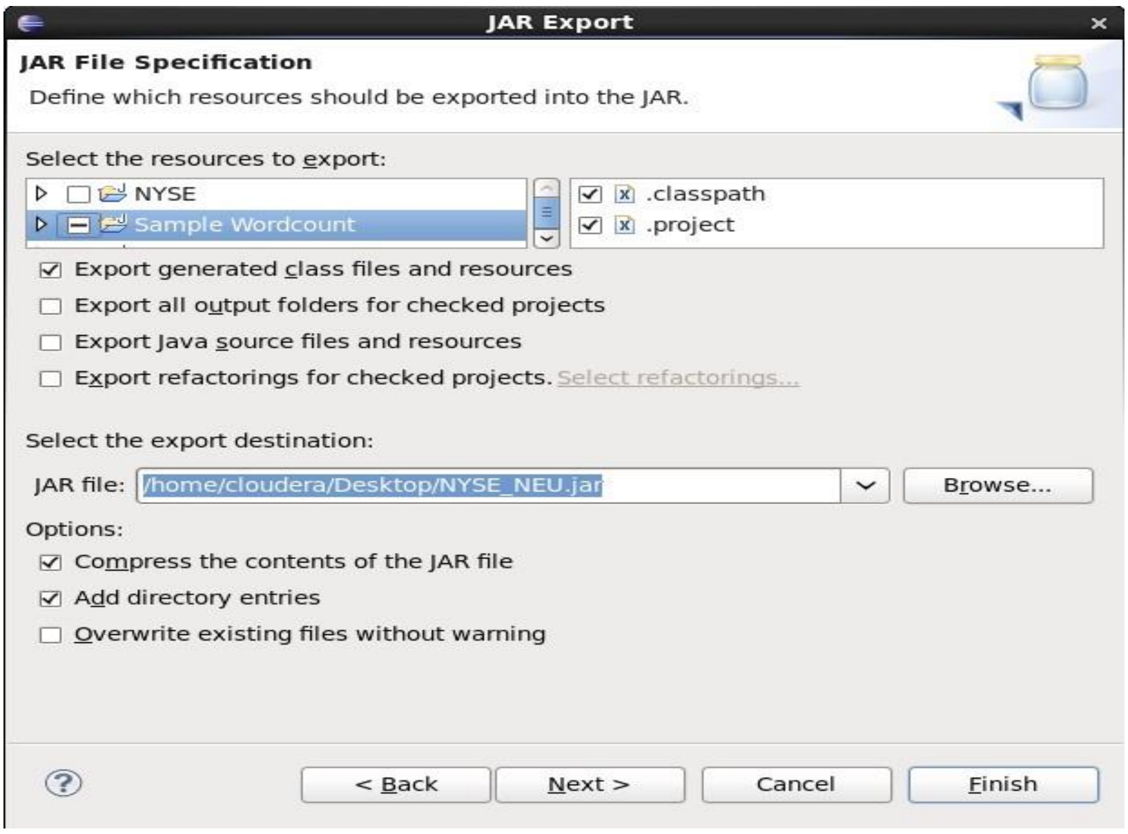
OutputKeyClass
OutputValueClass
Mapper Class
Reducer Class
InputFormat
OutputFormat
InputFilePath
OutputFolderPath

```
  public static void main(String[] args) throws Exception {
                        JobConf conf = new JobConf(WordCount.class);
                        conf.setJobName("WordCount");
                        conf.setOutputKeyClass(Text.class);
                        conf.setOutputValueClass(IntWritable.class);
                        conf.setMapperClass(Map.class);
                        //conf.setCombinerClass(Reduce.class);
                        conf.setReducerClass(Reduce.class);
                        conf.setInputFormat(TextInputFormat.class);
                        conf.setOutputFormat(TextOutputFormat.class);
                        FileInputFormat.setInputPaths(conf, new Path(args[0]));
                        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
                        JobClient.runJob(conf);
             }
}
```

Would you like to work on hands-on Hadoop Projects -CLICK HERE.
Step 8 :
Create the JAR file for the wordcount class –




Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
jar
1
Java
JAR file
2
Runnable JAR file
3
< Back
Next >
Cancel
Finish



Select the resources to export:

- ☐ NYSE
- ☐ Sample Wordcount
- ✓ .classpath
- ✓ .project

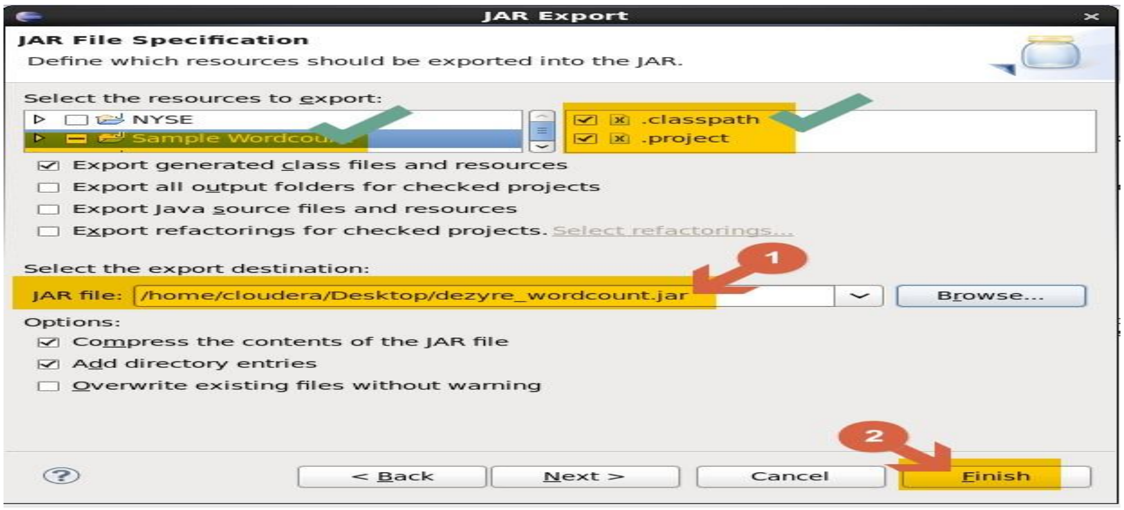
✓ Export generated class files and resources
☐ Export all output folders for checked projects
☐ Export Java source files and resources
☐ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: /home/cloudera/Desktop/NYSE_NEU.jar Browse...

Options:

✓ Compress the contents of the JAR file
✓ Add directory entries
☐ Overwrite existing files without warning


JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- ▷ ☐ NYSE
- ▷ ▬ Sample Wordcou
- ☑ .classpath
- ☑ .project

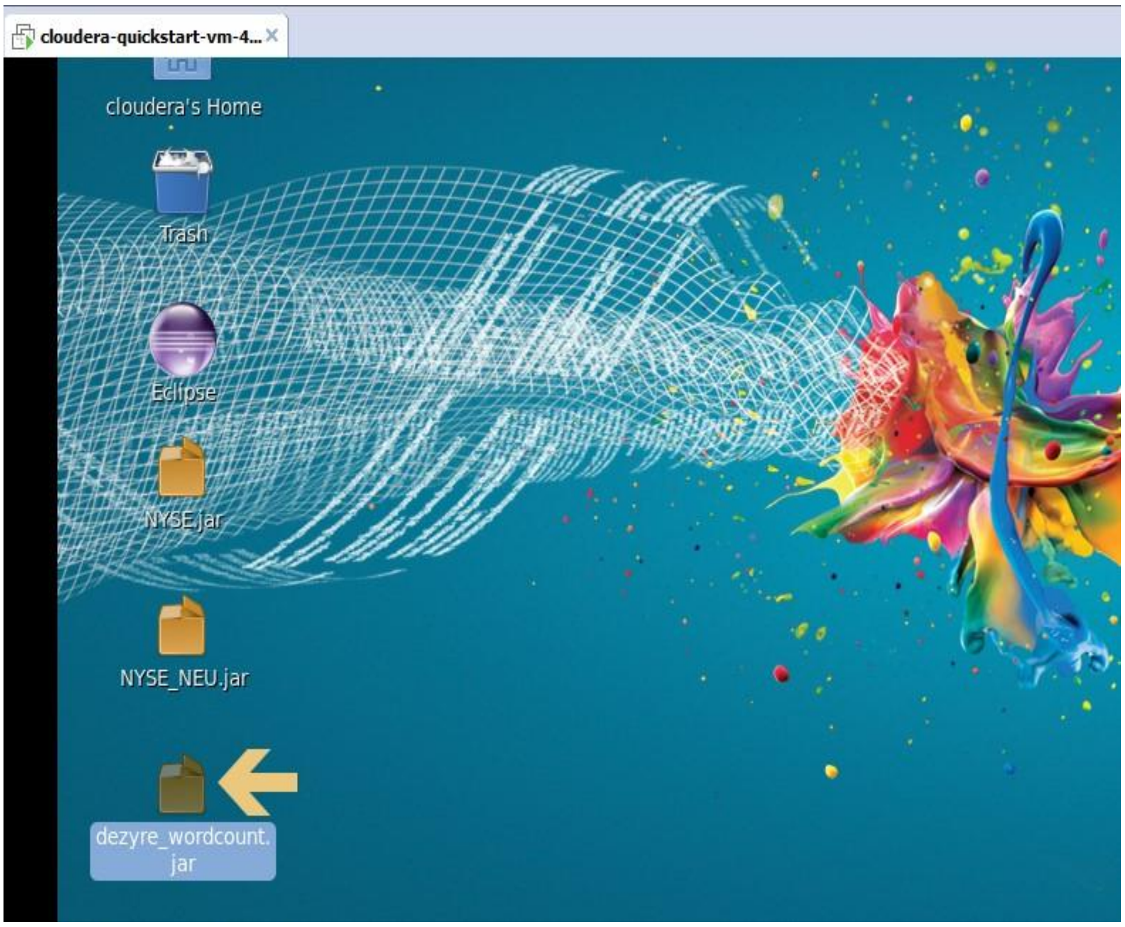
- ☑ Export generated class files and resources
- ☐ Export all output folders for checked projects
- ☐ Export Java source files and resources
- ☐ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: /home/cloudera/Desktop/dezyre_wordcount.jar

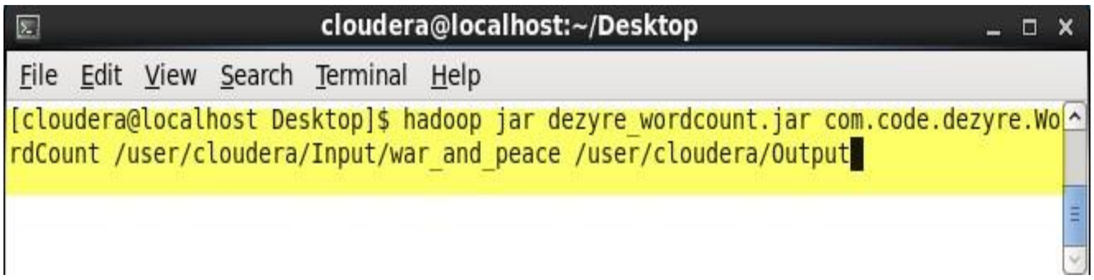
Options:

- ☑ Compress the contents of the JAR file
- ☑ Add directory entries
- ☐ Overwrite existing files without warning



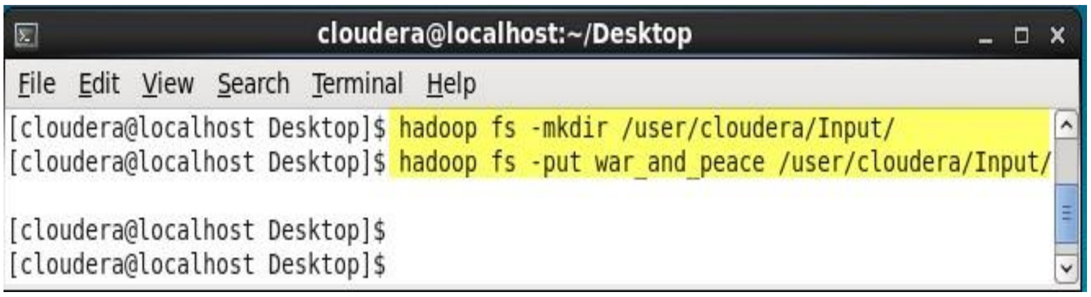
How to execute the Hadoop MapReduce WordCount program ?

```
>> hadoop jar  (jar file name) (className_along_with_packageName) (input file)
     (output folderpath)
     hadoop jar dezyre_wordcount.jar com.code.dezyre.WordCount
      /user/cloudera/Input/war_and_peace /user/cloudera/Output
```



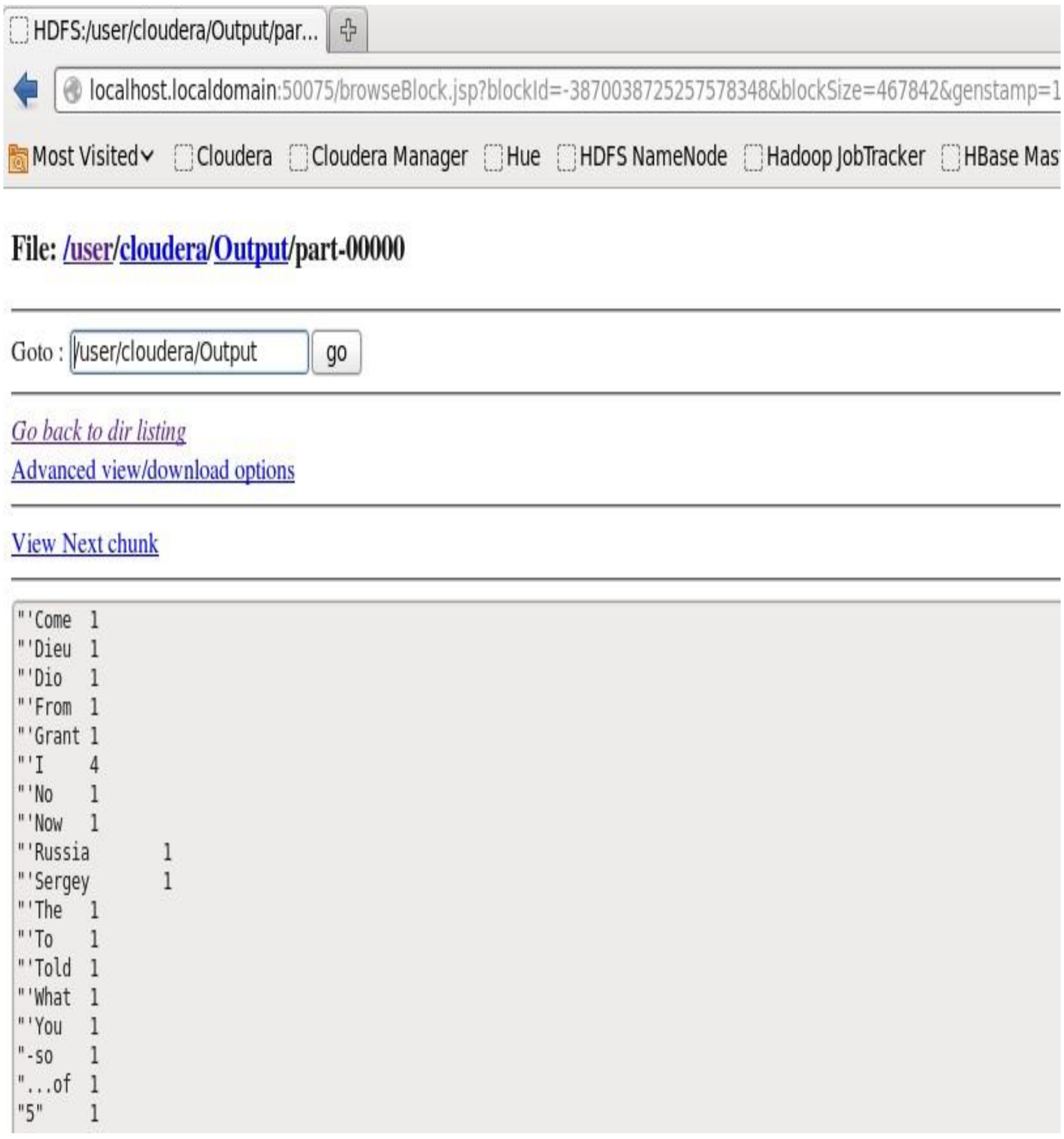
```
[cloudera@localhost Desktop]$ hadoop jar dezyre_wordcount.jar com.code.dezyre.Wo
rdCount /user/cloudera/Input/war_and_peace /user/cloudera/Output
```

Important Note: war_and_peace(Download link) must be available in HDFS at /user/cloudera/Input/war_and_peace.
If not, upload the file on HDFS using the following commands -
hadoop fs –mkdir /user/cloudera/Input
hadoop fs –put war_and_peace /user/cloudera/Input/war_and_peace



```
[cloudera@localhost Desktop]$ hadoop fs -mkdir /user/cloudera/Input/
[cloudera@localhost Desktop]$ hadoop fs -put war_and_peace /user/cloudera/Input/

[cloudera@localhost Desktop]$
[cloudera@localhost Desktop]$
```

Output of Executing Hadoop WordCount Example –


HDFS:/user/cloudera/Output/par...
localhost.localdomain:50075/browseBlock.jsp?blockId=-3870038725257578348&blockSize=467842&genstamp=1
Most Visited
Cloudera
Cloudera Manager
Hue
HDFS NameNode
Hadoop JobTracker
HBase Mas

File: /user/cloudera/Output/part-00000

Goto : /user/cloudera/Output go

Go back to dir listing
Advanced view/download options

View Next chunk

```
"'Come  1
"'Dieu  1
"'Dio   1
"'From  1
"'Grant 1
"'I     4
"'No    1
"'Now   1
"'Russia        1
"'Sergey        1
"'The   1
"'To    1
"'Told  1
"'What  1
"'You   1
"-so    1
"...of  1
"5"     1
```

The program is run with the war and peace input file. To get the war and peace dataset along with the Hadoop example code for the Word count program delivered to your inbox

EXPERIMENT-6

Develop Hadoop application to count no of characters, no of words and each Character Frequency.

Example – (Map function in Word Count) Input Set of data
Bus, Car, bus, car, train, car, bus, car, train, bus,
TRAIN,BUS, buS, caR, CAR, car, BUS, TRAIN
Output
Convert into another set of data
(Key,Value) (Bus,1), (Car,1), (bus,1), (car,1), (train,1),
(car,1), (bus,1), (car,1), (train,1), (bus,1),
(TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1),
(car,1), (BUS,1), (TRAIN,1)

Reduce Function – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.
Example – (Reduce function in Word Count)
Input
(output of Map function) Set of Tuples
(Bus,1), (Car,1), (bus,1), (car,1), (train,1),
(car,1), (bus,1), (car,1), (train,1), (bus,1),
(TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1),
(car,1), (BUS,1), (TRAIN,1)
Output
Converts into smaller set of tuple
(BUS,7), (CAR,7),(TRAIN,4)
Work Flow of the Program

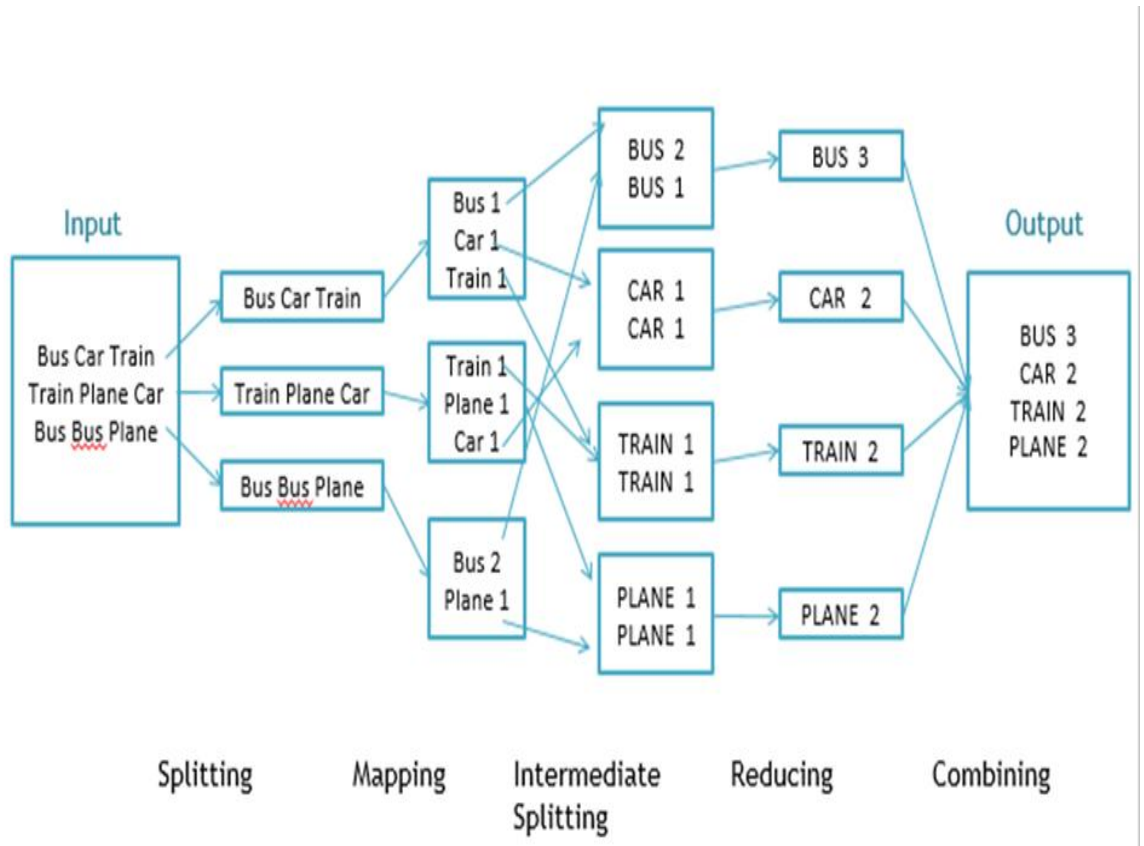


Fig. WorkFlow of MapReducing

Workflow of MapReduce consists of 5 steps:

- Splitting – The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line ('\n').
- Mapping – as explained above.
- Intermediate splitting – the entire process in parallel on different clusters. In order to group them in "Reduce Phase" the similar KEY data should be on the same cluster.
- Reduce – it is nothing but mostly group by phase.
- Combining – The last phase where all the data (individual result set from each cluster) is combined together to form a result.

  Now Let's See the Word Count Program in Java

  Fortunately, we don't have to write all of the above steps, we only need to write the splitting parameter, Map function logic, and Reduce function logic. The rest of the remaining steps will execute automatically.

  Make sure that Hadoop is installed on your system with the Java SDK.

Steps

Open Eclipse> File > New > Java Project >(Name it – MRProgramsDemo) > Finish.
Right Click > New > Package (Name it - PackageDemo) > Finish.
Right Click on Package > New > Class (Name it - WordCount).
Add Following Reference Libraries:
Right Click on Project > Build Path> Add External
/usr/lib/hadoop-0.20/hadoop-core.jar

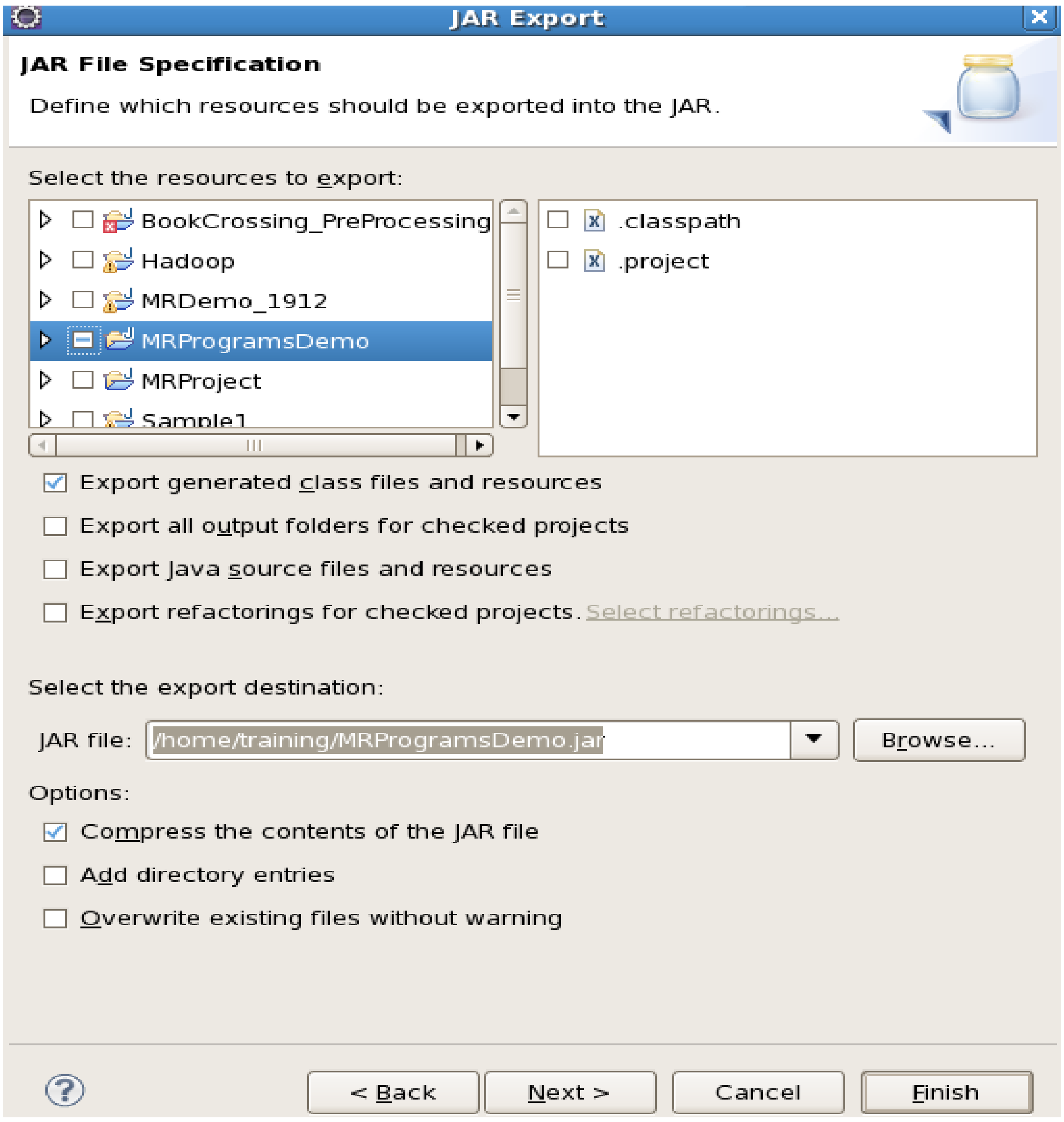
Usr/lib/hadoop-0.20/lib/Commons-cli-1.2.jar

5. Type the following code:

```
package PackageDemo;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration();
String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
Path input=new Path(files[0]);
Path output=new Path(files[1]);
Job j=new Job(c,"wordcount");
j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends Mapper<LongWritable, Text, Text,
IntWritable>{
public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException
{
String line = value.toString();
String[] words=line.split(",");
for(String word: words )
{
  Text outputKey = new Text(word.toUpperCase().trim());
  IntWritable outputValue = new IntWritable(1);
  con.write(outputKey, outputValue);
}
}
}
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text,
IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws
IOException, InterruptedException
{
int sum = 0;
   for(IntWritable value : values)
   {
   sum += value.get();
   }
   con.write(word, new IntWritable(sum));
}
}
}
```

The above program consists of three classes:
Driver class (Public, void, static, or main; this is the entry point).
The Map class which extends the public class
Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT> and implements
the Map function.
The Reduce class which extends the public class
Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT> and implements
the Reduce function.

6. Make a jar file
Right Click on Project> Export> Select export destination as Jar File > next> Finish.


JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- ☐ BookCrossing_PreProcessing
- ☐ Hadoop
- ☐ MRDemo_1912
- ▣ MRProgramsDemo
- ☐ MRProject
- ☐ Sample1

- ☐ .classpath
- ☐ .project

- ☑ Export generated class files and resources
- ☐ Export all output folders for checked projects
- ☐ Export Java source files and resources
- ☐ Export refactorings for checked projects. Select refactorings...

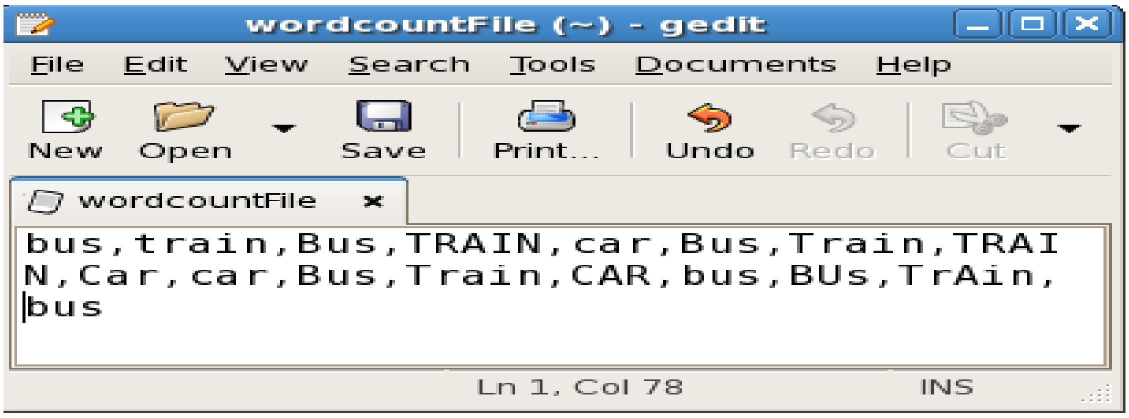
Select the export destination:

JAR file: /home/training/MRProgramsDemo.jar

Options:

- ☑ Compress the contents of the JAR file
- ☐ Add directory entries
- ☐ Overwrite existing files without warning

7. Take a text file and move it into HDFS format:


wordcountFile (~) - gedit

```
bus,train,Bus,TRAIN,car,Bus,Train,TRAI
N,Car,car,Bus,Train,CAR,bus,BUs,TrAin,
bus
```

To move this into Hadoop directly, open the terminal and enter the following commands:

```
[training@localhost ~]$ hadoop fs -put wordcountFile wordCountFile
```

8. Run the jar file:

(Hadoop jar jarfilename.jar packageName.ClassName PathToInputTextFile PathToOutputDirectry)

```
[training@localhost ~]$ hadoop jar MRProgramsDemo.jar PackageDemo.WordCount
wordCountFile MRDir1
```

9. Open the result:

```
[training@localhost ~]$ hadoop fs -ls MRDir1
Found 3 items
-rw-r--r--   1 training supergroup          0 2016-02-23 03:36
/user/training/MRDir1/_SUCCESS
drwxr-xr-x   - training supergroup          0 2016-02-23 03:36
/user/training/MRDir1/_logs
-rw-r--r--   1 training supergroup           20 2016-02-23 03:36
/user/training/MRDir1/part-r-00000
[training@localhost ~]$ hadoop fs -cat MRDir1/part-r-00000
BUS     7
CAR     4
TRAIN   6
```

PRE LAB VIVA QUESTIONS ?

1. Explain About MapReduce?
2. What does meant by splitting in MapReduce?
3. What does it mean by intermediate splitting?
4. Why Reduce Phase is required in MapReduce?

POST LAB VIVA QUESTIONS:

1. Define which resources should be exported to the JAR file ?
2. How does Hadoop works with Java?
3. Define the Map function logic in Hadoop?
4. Define the Reduce function logic in Hadoop?

EXPERIMENT-7

a) Develop Hadoop application to process given data and produce results such as finding the year of maximum usage, year of minimum usage.

Step 1: Following is the application that counts number of lines in a file.

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class CharacterCount {
 public static class CharacterCountMapper extends
 Mapper<Object, Text, Text, IntWritable> {
  private Text word = new Text("Total Characters in file are ");
  public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {
   context.write(word, new IntWritable(value.getLength()));
  }
 }
 public static class IntSumReducer extends
   Reducer<Text, IntWritable, Text, IntWritable> {
  private IntWritable result = new IntWritable();
  public void reduce(Text key, Iterable<IntWritable> values,
  Context context) throws IOException, InterruptedException {
   int sum = 0;
   for (IntWritable val : values) {
    sum += val.get();
   }
   result.set(sum);
   context.write(key, result);
  }
 }

 public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "Character count");
  job.setJarByClass(CharacterCount.class);
  job.setMapperClass(CharacterCountMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
 }
}
```

Step2: Compile above java file.

```
$ hadoop com.sun.tools.javac.Main CharacterCount.java
```

Step 3: Create jar file

```
$ jar cf charcount.jar CharacterCount*class
```

Step 4: Run jar file.

```
hadoop jar charcount.jar CharacterCount /user/harikrishna_gurram/input.txt
/user/harikrishna_gurram/results1
```

Open “/user/harikrishna_gurram/results1” directory, you can see two files.

```
$ hadoop fs -ls /user/harikrishna_gurram/results1
Found 2 items
-rw-r--r--   3 harikrishna_gurram supergroup          0 2015-06-23 09:40
/user/harikrishna_gurram/results1/_SUCCESS
-rw-r--r--   3 harikrishna_gurram supergroup         34 2015-06-23 09:40
/user/harikrishna_gurram/results1/part-r-00000
```

Open “part-r-00000” file; you can see the number of characters of given input file.

```
$ hadoop fs -cat /user/harikrishna_gurram/results1/part-r-00000
```

b) Total Characters in file are 23999453

MapReduce Char Count Example

In MapReduce char count example, we find out the frequency of each character. Here, the role of Mapper is to map the keys to the existing values and the role of Reducer is to aggregate the keys of common values. So, everything is represented in the form of Key-value pair.

Pre-requisite

Java Installation - Check whether the Java is installed or not using the following command.

```
java -version
```

Hadoop Installation - Check whether the Hadoop is installed or not using the following command.

```
hadoop version
```

If any of them is not installed in your system, follow the below link to install it.

www.javatpoint.com/hadoop-installation

Steps to execute MapReduce char count example

Create a text file in your local machine and write some text into it.

```
$ nano info.txt
```

Check the text written in the info.txt file.

```
$ cat info.txt
```

In this example, we find out the frequency of each char value exists in this text file.

Create a directory in HDFS, where to kept text file.

```
$ hdfs dfs -mkdir /count
```

Upload the info.txt file on HDFS in the specific directory.

```
$ hdfs dfs -put /home/codegyani/info.txt /count
```

Write the MapReduce program using eclipse.

File: WC_Mapper.java

```
package com.javatpoint;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WC_Mapper extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable>{
 public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable>
        output,
        Reporter reporter) throws IOException{
        String line = value.toString();
        String  tokenizer[] = line.split("");
        for(String SingleChar : tokenizer)
        {
            Text charKey = new Text(SingleChar);
            IntWritable One = new IntWritable(1);
            output.collect(charKey, One);
        }
    }

}
```

File: WC_Reducer.java

```
 package com.javatpoint;
    import java.io.IOException;
    import java.util.Iterator;
    import org.apache.hadoop.io.IntWritable;
    import org.apache.hadoop.io.Text;
    import org.apache.hadoop.mapred.MapReduceBase;
    import org.apache.hadoop.mapred.OutputCollector;
    import org.apache.hadoop.mapred.Reducer;
    import org.apache.hadoop.mapred.Reporter;
    public class WC_Reducer  extends MapReduceBase implements Reducer
    <Text,IntWritable,Text,IntWritable> {
    public void reduce(Text key, Iterator<IntWritable> values,OutputCollector
    <Text,IntWritable> output,
     Reporter reporter) throws IOException {
    int sum=0;
    while (values.hasNext()) {
    sum+=values.next().get();
    }
```

```
output.collect(key,new IntWritable(sum));
}
}
File: WC_Runner.java
package com.javatpoint;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;
public class WC_Runner {
public static void main(String[] args) throws IOException{
        JobConf conf = new JobConf(WC_Runner.class);
        conf.setJobName("CharCount");
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        conf.setMapperClass(WC_Mapper.class);
        conf.setCombinerClass(WC_Reducer.class);
        conf.setReducerClass(WC_Reducer.class);
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);
        FileInputFormat.setInputPaths(conf,new Path(args[0]));
        FileOutputFormat.setOutputPath(conf,new Path(args[1]));
        JobClient.runJob(conf);
    }
}
```

Download the source code.
Create the jar file of this program and name it charcountdemo.jar.
Run the jar file

```
hadoop jar /home/codegyani/charcountdemo.jar com.javatpoint.WC_Runner
/count/info.txt /char_output
```

The output is stored in /char_output/part-00000
Now execute the command to see the output.

```
hdfs dfs -cat /r_output/part-00000
```

PRE LAB VIVA QUESTIONS?

1. Hadoop has how many layers?
2. What are the different operation modes of Hadoop?
3. What are the platforms can be supported by Hadoop?
4. What are the different operations can be performed by SSH?

POST LAB VIVA QUESTIONS:

1. Define master node?
2. Define data node?
3. What is name note?
4. Define slave node?

EXPERIMENT-9

Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.

Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.

Tasks

To complete this tutorial, perform the following tasks:

1. Launch an Instance
2. Connect to Your Instance
3. Clean Up Your Instance

Related Tutorials

- ➢ If you'd prefer to launch a Windows instance, see this tutorial in the Amazon EC2
User Guide for Windows Instances: Getting Started with Amazon EC2 Windows
Instances.
- ➢ If you'd prefer to use the command line, see this tutorial in the AWS Command
Line Interface User Guide: Using Amazon EC2 through the AWS CLI.

Prerequisites

Before you begin, be sure that you've completed the steps in Setting Up with Amazon EC2.

Step 1: Launch an Instance

You can launch a Linux instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see Launching an Instance.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the console dashboard, choose Launch Instance.
3. The Choose an Amazon Machine Image (AMI) page displays a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. Select an HVM version of Amazon Linux 2.Notice that these AMIs are marked "Free tier eligible."
5. On the Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default. Notice that this instance type is eligible for the free tier.
6. Choose Review and Launch to let the wizard complete the other configuration settings for you.
7. On the Review Instance Launch page, under Security Groups, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:

a. Choose Edit security groups.

b. On the Configure Security Group page, ensure that Select an existing security group is selected.
c. Select your security group from the list of existing security groups, and then choose Review and Launch.

7. On the Review Instance Launch page, choose Launch.
8. When prompted for a key pair, select Choose an existing key pair, then select the key pair that you created when getting set up.

   Alternatively, you can create a new key pair. Select Create a new key pair, enter a name for the key pair, and then choose Download Key Pair. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

   Warning

   Don't select the Proceed without a key pair option. If you launch your instance without a key pair, then you can't connect to it.

   When you are ready, select the acknowledgement check box, and then choose Launch Instances.

8. A confirmation page lets you know that your instance is launching. Choose View Instances to close the confirmation page and return to the console.
9. On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to runningand it receives a public DNS name. (If the Public DNS (IPv4) column is hidden, choose Show/Hide Columns (the gear-shaped icon) in the top right corner of the page and then select Public DNS (IPv4).)
10. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the Status Checks column.

Step 2: Connect to Your Instance

There are several ways to connect to your Linux instance. For more information,see Connect to Your Linux Instance.

Important

You can't connect to your instance unless you launched it with a key pair for which you have the .pem file and you launched it with a security group that allows SSH access from your computer. If you can't connect to your instance, see Troubleshooting Connecting to Your Instance for assistance.

Step 3: Clean Up Your Instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see Next Steps.

Important

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the AWS Free Tier, you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see Stopping Instances.

To terminate your instance

1. In the navigation pane, choose Instances. In the list of instances, select the instance.
2. Choose Actions, Instance State, Terminate.
3. Choose Yes, Terminate when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.
Next Steps.

After you start your instance, you might want to try some of the following exercises:

- Learn how to remotely manage your EC2 instance using Run Command. For more information, see AWS Systems Manager Run Command in the AWS Systems Manager User Guide.
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see Create a Billing Alarm in the AWS Billing and Cost Management User Guide.
- Add an EBS volume. For more information, see Creating an Amazon EBS Volume and Attaching an Amazon EBS Volume to an Instance.
- Install the LAMP stack. For more information, see Tutorial: Install a LAMP Web Server on Amazon Linux 2.

PRE LAB VIVA QUESTIONS

1. What is AWS?

2. What are the key components of AWS?

3. The fundamental elements of AWS ?

4. What is the importance of buffer in Amazon Web Services?

POST LAB VIVA QUESTIONS

1. What is lambda@edge in AWS??

2. Distinguish between scalability and flexibility?

3. What are the different types of events triggered by Amazon Cloud Front?

4. Which automation gears can help with spinup services?

EXPERIMENT- 11

Before installing ZooKeeper, make sure your system is running on any of the following operating systems

- Any of Linux OS − Supports development and deployment. It is preferred for demo applications.
- Windows OS − Supports only development.
- Mac OS − Supports only development.

ZooKeeper server is created in Java and it runs on JVM. You need to use JDK 6 or greater.

Now, follow the steps given below to install ZooKeeper framework on your machine.

Step 1: Verifying Java Installation

We believe you already have a Java environment installed on your system. Just verify it using the following command.

```
$ java -version
```

If you have Java installed on your machine, then you could see the version of installed Java. Otherwise, follow the simple steps given below to install the latest version of Java.

Step 1.1: Download JDK

Download the latest version of JDK by visiting the following link and download the latest version. Java

The latest version (while writing this tutorial) is JDK 8u 60 and the file is “jdk-8u60-linuxx64.tar.gz”. Please download the file on your machine.

```
$ mkdir /opt/jdk
$ mv jdk-1.8.0_60 /opt/jdk/
```

Step 1.4: Set path

To set path and JAVA_HOME variables, add the following commands to ~/.bashrc file.

```
export JAVA_HOME = /usr/jdk/jdk-1.8.0_60
export PATH=$PATH:$JAVA_HOME/bin
```

Now, apply all the changes into the current running system.

```
$ source ~/.bashrc
```

Step 1.5: Java alternatives

Use the following command to change Java alternatives.

```
update-alternatives --install /usr/bin/java java /opt/jdk/jdk1.8.0_60/bin/java 100
```

Step 1.6

Verify the Java installation using the verification command (java -version)explained in Step 1.Step 2: ZooKeeper Framework Installation

Step 2.1: Download ZooKeeper

To install ZooKeeper framework on your machine, visit the following link and download the latest version of ZooKeeper. http://zookeeper.apache.org/releases.html

As of now, the latest version of ZooKeeper is 3.4.6 (ZooKeeper-3.4.6.tar.gz).

Step 2.2: Extract the tar file

Extract the tar file using the following commands −

```
$ cd opt/
$ tar -zxf zookeeper-3.4.6.tar.gz
$ cd zookeeper-3.4.6
$ mkdir data
```

Step 2.3: Create configuration file
Open the configuration file named conf/zoo.cfg using the command vi conf/zoo.cfg and all the following parameters to set as starting point.

```
$ vi conf/zoo.cfg
tickTime = 2000
dataDir = /path/to/zookeeper/data
clientPort = 2181
initLimit = 5
syncLimit = 2
```

Once the configuration file has been saved successfully, return to the terminal again. You can now start the zookeeper server.
Step 2.4: Start ZooKeeper server
Execute the following command −

```
$ bin/zkServer.sh start
```

After executing this command, you will get a response as follows −

```
$ JMX enabled by default
$ Using config: /Users/../zookeeper-3.4.6/bin/../conf/zoo.cfg
$ Starting zookeeper ... STARTED
```

Step 2.5: Start CLI
Type the following command

```
$ bin/zkCli.sh
```

After typing the above command, you will be connected to the ZooKeeper server and you should get the following response.

```
Connecting to localhost:2181
................
................
................
Welcome to ZooKeeper!
................
................
WATCHER::
WatchedEvent state:SyncConnected type: None path:null
[zk: localhost:2181(CONNECTED) 0]
```

Stop ZooKeeper Server
After connecting the server and performing all the operations, you can stop the zookeeper server by using the following command.
EXAMPLE: ZooKeeper has an official API binding for Java and C. The ZooKeeper community provides unofficial API for most of the languages (.NET, python, etc.). Using ZooKeeper API, an application can connect, interact, manipulate data, coordinate, and finally disconnect from a ZooKeeper ensemble.
ZooKeeper API has a rich set of features to get all the functionality of the ZooKeeper ensemble in a simple and safe manner. ZooKeeper API provides both synchronous and asynchronous methods.
ZooKeeper ensemble and ZooKeeper API completely complement each other in every aspect and it benefits the developers in a great way. Let us discuss Java binding in this chapter.

Basics of ZooKeeper API

Application interacting with ZooKeeper ensemble is referred as ZooKeeper Client or simply Client.

Znode is the core component of ZooKeeper ensemble and ZooKeeper API provides a small set of methods to manipulate all the details of znode with ZooKeeper ensemble.

A client should follow the steps given below to have a clear and clean interaction with **ZooKeeper ensemble.**

- Connect to the ZooKeeper ensemble. ZooKeeper ensemble assign a Session ID for
  the client.
- Send heartbeats to the server periodically. Otherwise, the ZooKeeper ensemble expires the Session ID and the client needs to reconnect.
- Get / Set the znodes as long as a session ID is active.
- Disconnect from the ZooKeeper ensemble, once all the tasks are completed. If the

client is inactive for a prolonged time, then the ZooKeeper ensemble will automatically disconnect the client.

Java Binding

Let us understand the most important set of ZooKeeper API in this chapter. The central part of the ZooKeeper API is ZooKeeper class. It provides options to connect the ZooKeeper ensemble in its constructor and has the following methods

- connect − connect to the ZooKeeper ensemble
- create − create a znode
- exists − check whether a znode exists and its information
- getData − get data from a particular znode
- setData − set data in a particular znode
- getChildren − get all sub-nodes available in a particular znode
- delete − get a particular znode and all its children
- close − close a connection

Connect to the ZooKeeper Ensemble

The ZooKeeper class provides connection functionality through its constructor. The signature of the constructor is as follows −

ZooKeeper(String connectionString, int sessionTimeout, Watcher watcher)

Where,

- connectionString − ZooKeeper ensemble host.
- sessionTimeout − session timeout in milliseconds.
- watcher − an object implementing “Watcher” interface. The ZooKeeper ensemble returns the connection status through the watcher object.

Let us create a new helper class ZooKeeperConnection and add a method connect. The connect method creates a ZooKeeper object, connects to the ZooKeeper ensemble, and then returns the object.

Here CountDownLatch is used to stop (wait) the main process until the client connects with the ZooKeeper ensemble.

The ZooKeeper ensemble replies the connection status through the Watcher callback. The Watcher callback will be called once the client connects with the ZooKeeper ensemble and the Watcher callback calls the countDown method of the CountDownLatch to release the lock, await in the main process.

Here is the complete code to connect with a ZooKeeper ensemble.
Coding: ZooKeeperConnection.java

```
// import java classes
import java.io.IOException;
import java.util.concurrent.CountDownLatch;
// import zookeeper classes
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.AsyncCallback.StatCallback;
import org.apache.zookeeper.KeeperException.Code;
import org.apache.zookeeper.data.Stat;
public class ZooKeeperConnection
 {
   // declare zookeeper instance to access ZooKeeper ensemble
   private ZooKeeper zoo;
   final CountDownLatch connectedSignal = new CountDownLatch(1);
   // Method to connect zookeeper ensemble.
   public ZooKeeper connect(String host) throws IOException,InterruptedException
   {
       zoo = new ZooKeeper(host,5000,new Watcher()
       {
        public void process(WatchedEvent we)
       {
           if (we.getState() == KeeperState.SyncConnected)
            {
              connectedSignal.countDown();
            }
         }
      });
     connectedSignal.await();
      return zoo;
   }
   // Method to disconnect from zookeeper server
   public void close() throws InterruptedException
   {
      zoo.close();
   }
}
```

Save the above code and it will be used in the next section for connecting the ZooKeeper ensemble.

Create a Znode

The ZooKeeper class provides create method to create a new znode in the ZooKeeper ensemble. The signature of the create method is as follows −

create(String path, byte[] data, List<ACL> acl, CreateMode createMode)

Where,

- path − Znode path. For example, /myapp1, /myapp2, /myapp1/mydata1,
- myapp2/mydata1/myanothersubdata
- data − data to store in a specified znode path
- acl − access control list of the node to be created. ZooKeeper API provides a
- static interface ZooDefs.Ids to get some of basic acl list. For example, ZooDefs.Ids.OPEN_ACL_UNSAFE returns a list of acl for open znodes.
- createMode − the type of node, either ephemeral, sequential, or both. This is an enum.

Let us create a new Java application to check the create functionality of the ZooKeeper API. Create a file ZKCreate.java. In the main method, create an object of type ZooKeeperConnection and call the connect method to connect to the ZooKeeper ensemble.
The connect method will return the ZooKeeper object zk. Now, call the createmethod of zk object with custom path and data.
The complete program code to create a znode is as follows −
Coding: ZKCreate.java

```
import java.io.IOException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.CreateMode;
import org.apache.zookeeper.ZooDefs;
public class ZKCreate {
  // create static instance for zookeeper class.
  private static ZooKeeper zk;
  // create static instance for ZooKeeperConnection class.
  private static ZooKeeperConnection conn;
  // Method to create znode in zookeeper ensemble
   public static void create(String path, byte[] data) throws
    KeeperException,InterruptedException {
    zk.create(path, data, ZooDefs.Ids.OPEN_ACL_UNSAFE,
    CreateMode.PERSISTENT);
  }
  public static void main(String[] args) {

    // znode path
    String path = "/MyFirstZnode"; // Assign path to znode
    // data in byte array
    byte[] data = "My first zookeeper app".getBytes(); // Declare data
    try {
      conn = new ZooKeeperConnection();
      zk = conn.connect("localhost");
      create(path, data); // Create the data to the specified path
      conn.close();
    }
    catch (Exception e)
     {
      System.out.println(e.getMessage()); //Catch error message
    }
  }
}
```

Once the application is compiled and executed, a znode with the specified data will be created in the ZooKeeper ensemble. You can check it using the ZooKeeper CLI zkCli.sh.

```
cd /path/to/zookeeper
bin/zkCli.sh
>>> get /MyFirstZnode
```

Exists – Check the Existence of a Znode

The ZooKeeper class provides the exists method to check the existence of a znode. It returns the metadata of a znode, if the specified znode exists. The signature of the exists method is as follows −

```
exists(String path, boolean watcher)
```

Where,

- path − Znode path
- watcher − boolean value to specify whether to watch a specified znode or not

Let us create a new Java application to check the "exists" functionality of the ZooKeeper API. Create a file "ZKExists.java". In the main method, create ZooKeeper object, "zk" using "ZooKeeperConnection" object. Then, call "exists" method of "zk" object with custom "path". The complete listing is as follow −

Coding: ZKExists.java

```
import java.io.IOException;
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import org.apache.zookeeper.data.Stat;
public class ZKExists {
  private static ZooKeeper zk;
  private static ZooKeeperConnection conn;
  // Method to check existence of znode and its status, if znode is available.
  public static Stat znode_exists(String path) throws
    KeeperException,InterruptedException {
    return zk.exists(path, true);
  }
  public static void main(String[] args) throws InterruptedException,KeeperException
{
     String path = "/MyFirstZnode"; // Assign znode to the specified path
    try
      {
       conn = new ZooKeeperConnection();
       zk = conn.connect("localhost");
       Stat stat = znode_exists(path); // Stat checks the path of the znode
         if(stat != null)
          {
          System.out.println("Node exists and the node version is " +
          stat.getVersion());
          }
          else
          {
          System.out.println("Node does not exists");
       }
  }
 catch(Exception e)
 {
       System.out.println(e.getMessage()); // Catches error messages
   }
  }
}
```

Once the application is compiled and executed, you will get the below output.

```
Node exists and the node version is 1.
```

getData Method

The ZooKeeper class provides getData method to get the data attached in a specified znode and its status. The signature of the getData method is as follows −

```
getData(String path, Watcher watcher, Stat stat)
```

Where,

- path − Znode path.
- watcher − Callback function of type Watcher. The ZooKeeper ensemble will notify through the Watcher callback when the data of the specified znode changes.
  This is one time notification.
- stat − Returns the metadata of a znode.

Let us create a new Java application to understand the getData functionality of the ZooKeeper API. Create a file ZKGetData.java. In the main method, create a ZooKeeper object zk using he ZooKeeperConnection object. Then, call the getData method of zk object with custom path.

Here is the complete program code to get the data from a specified node −

Coding: ZKGetData.java

```
import java.io.IOException;
import java.util.concurrent.CountDownLatch;
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import org.apache.zookeeper.data.Stat;
public class ZKGetData {
private static ZooKeeper zk;
private static ZooKeeperConnection conn;
public static Stat znode_exists(String path) throws
    KeeperException,InterruptedException {
    return zk.exists(path,true);
  }
  public static void main(String[] args) throws InterruptedException,
KeeperException {
    String path = "/MyFirstZnode";
    final CountDownLatch connectedSignal = new CountDownLatch(1);
   try {
      conn = new ZooKeeperConnection();
      zk = conn.connect("localhost");
      Stat stat = znode_exists(path);
       if(stat != null) {
        byte[] b = zk.getData(path, new Watcher() {
        public void process(WatchedEvent we) {
        if (we.getType() == Event.EventType.None) {
          switch(we.getState()) {
          case Expired:
          connectedSignal.countDown();
               break;
              }
          } else
           {
              String path = "/MyFirstZnode";
            try {
                byte[] bn = zk.getData(path,
                false, null);
                String data = new String(bn,
                "UTF-8");
                System.out.println(data);
                connectedSignal.countDown();
             }
            catch(Exception ex) {
                System.out.println(ex.getMessage());
              }
            }
          }
        }, null);
        String data = new String(b, "UTF-8");
        System.out.println(data);
        connectedSignal.await();
        } else {
        System.out.println("Node does not exists");
      }
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }
  }
}
```

Once the application is compiled and executed, you will get the following output

```
My first zookeeper app
```

And the application will wait for further notification from the ZooKeeper ensemble.
Change the data of the specified znode using ZooKeeper CLI zkCli.sh.

```
cd /path/to/zookeeper
bin/zkCli.sh
>>> set /MyFirstZnode Hello
```

Now, the application will print the following output and exit.

```
Hello
```

setData Method

The ZooKeeper class provides setData method to modify the data attached in a specified znode. The signature of the setData method is as follows −

```
setData(String path, byte[] data, int version)
```

Where,

- path − Znode path
- data − data to store in a specified znode path.
- version − Current version of the znode. ZooKeeper updates the version number of the znode whenever the data gets changed.

Let us now create a new Java application to understand the setDatafunctionality of the ZooKeeper API. Create a file ZKSetData.java. In the main method, create a ZooKeeper object zk using the ZooKeeperConnectionobject. Then, call the setData method of zk object with the specified path, new data, and version of the node.
Here is the complete program code to modify the data attached in a specified znode.
Code: ZKSetData.java

```
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import java.io.IOException;
public class ZKSetData {
  private static ZooKeeper zk;
  private static ZooKeeperConnection conn;
  // Method to update the data in a znode. Similar to getData but without watcher.
  public static void update(String path, byte[] data) throws
    KeeperException,InterruptedException {
    zk.setData(path, data, zk.exists(path,true).getVersion());
  }
 public static void main(String[] args) throws   InterruptedException,KeeperException
   {
    String path= "/MyFirstZnode";
    byte[] data = "Success".getBytes(); //Assign data which is to be updated.

    try {
      conn = new ZooKeeperConnection();
      zk = conn.connect("localhost");
      update(path, data); // Update znode data to the specified path
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }
  }
}
```

Once the application is compiled and executed, the data of the specified znode will be changed and it can be checked using the ZooKeeper CLI, zkCli.sh.

```
cd /path/to/zookeeper
bin/zkCli.sh
>>> get /MyFirstZnode
```

getChildren Method

The ZooKeeper class provides getChildren method to get all the sub-node of a particular znode. The signature of the getChildren method is as follows −

getChildren(String path, Watcher watcher)

Where,

- path − Znode path.
- watcher − Callback function of type “Watcher”. The ZooKeeper ensemble will notify when the specified znode gets deleted or a child under the znode gets created / deleted. This is a one-time notification.

Coding: ZKGetChildren.java

```
import java.io.IOException;
import java.util.*;
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
import org.apache.zookeeper.WatchedEvent;
import org.apache.zookeeper.Watcher;
import org.apache.zookeeper.Watcher.Event.KeeperState;
import org.apache.zookeeper.data.Stat;
public class ZKGetChildren {
  private static ZooKeeper zk;
  private static ZooKeeperConnection conn;
  // Method to check existence of znode and its status, if znode is available.
  public static Stat znode_exists(String path) throws
    KeeperException,InterruptedException {
    return zk.exists(path,true);
  }
  public static void main(String[] args) throws InterruptedException,KeeperException
{
  String path = "/MyFirstZnode"; // Assign path to the znode
   try {
      conn = new ZooKeeperConnection();
      zk = conn.connect("localhost");
      Stat stat = znode_exists(path); // Stat checks the path
      if(stat!= null) {
        //"getChildren" method- get all the children of znode.It has two
        args, path and watch
        List <String> children = zk.getChildren(path, false);
        for(int i = 0; i < children.size(); i++)
        System.out.println(children.get(i)); //Print children's
      } else {
        System.out.println("Node does not exists");
      }
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }
  }
}
```

Before running the program, let us create two sub-nodes for /MyFirstZnodeusing the ZooKeeper CLI, zkCli.sh.

```
cd /path/to/zookeeper
bin/zkCli.sh
>>> create /MyFirstZnode/myfirstsubnode Hi
>>> create /MyFirstZnode/mysecondsubmode Hi
```

Now, compiling and running the program will output the above created znodes.

```
myfirstsubnode
mysecondsubnode
```

Delete a Znode

The ZooKeeper class provides delete method to delete a specified znode. The signature of the delete method is as follows −

```
delete(String path, int version)
```

Where,

- path − Znode path.
- version − Current version of the znode.

Let us create a new Java application to understand the delete functionality of the ZooKeeper API. Create a file ZKDelete.java. In the main method, create a ZooKeeper object zk using ZooKeeperConnection object. Then, call the delete method of zk object with the specified path and version of the node.

The complete program code to delete a znode is as follows −

Coding: ZKDelete.java

```
import org.apache.zookeeper.ZooKeeper;
import org.apache.zookeeper.KeeperException;
public class ZKDelete {
   private static ZooKeeper zk;
   private static ZooKeeperConnection conn;
   // Method to check existence of znode and its status, if znode is available.
   public static void delete(String path) throws KeeperException,InterruptedException
{
      zk.delete(path,zk.exists(path,true).getVersion());
   }
   public static void main(String[] args) throws InterruptedException,KeeperException
{
   String path = "/MyFirstZnode"; //Assign path to the znode
   try {
         conn = new ZooKeeperConnection();
         zk = conn.connect("localhost");
         delete(path); //delete the node with the specified path
      } catch(Exception e) {
         System.out.println(e.getMessage()); // catches error messages
      }
   }
}
```

PRE LAB VIVA QUESTIONS?

1. What exactly is ZooKeeper?
2. What is ZooKeeper server and client?
3. How does ZooKeeper work?
4. What is ZooKeeper used for?

POST LAB VIVA QUESTIONS?

5. What type of connection is made with ZooKeeper by client?
6. Is ZooKeeper mandatory for Kafka?
7. What does Apache ZooKeeper mean?
8. What is Apache ZooKeeper?

EXPERIMENT-12

Create a Java Hello World Example with Google App Engine in Eclipse

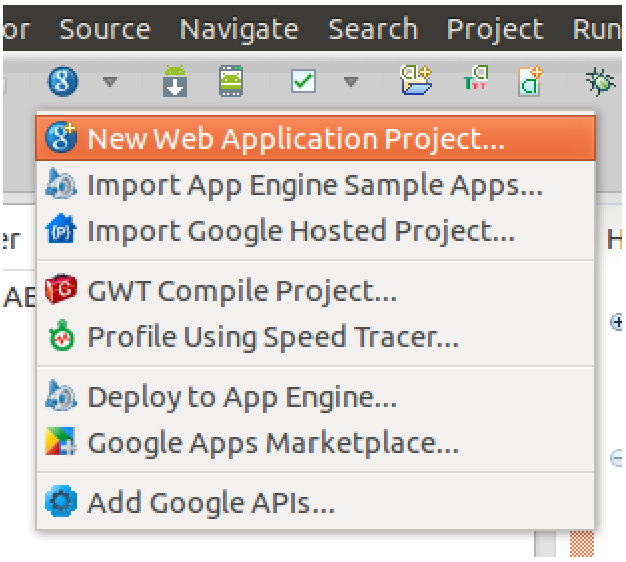
In the following tutorial we will create a Java Google App Engine example. The application will display the typical “Hello World” greeting when it is invoked. We’ll develop all the source code in Eclipse with the help of the GAE plugin for Eclipse, test the application through Google App Engine Runtime (included within the plug in) and finally we will deploy it to the GAE account.

1. Prerequisites

1. Eclipse IDE.
2. GAE Eclipse Plugin for Eclipse.
3. A Google App Engine account (to test the app in GAE)

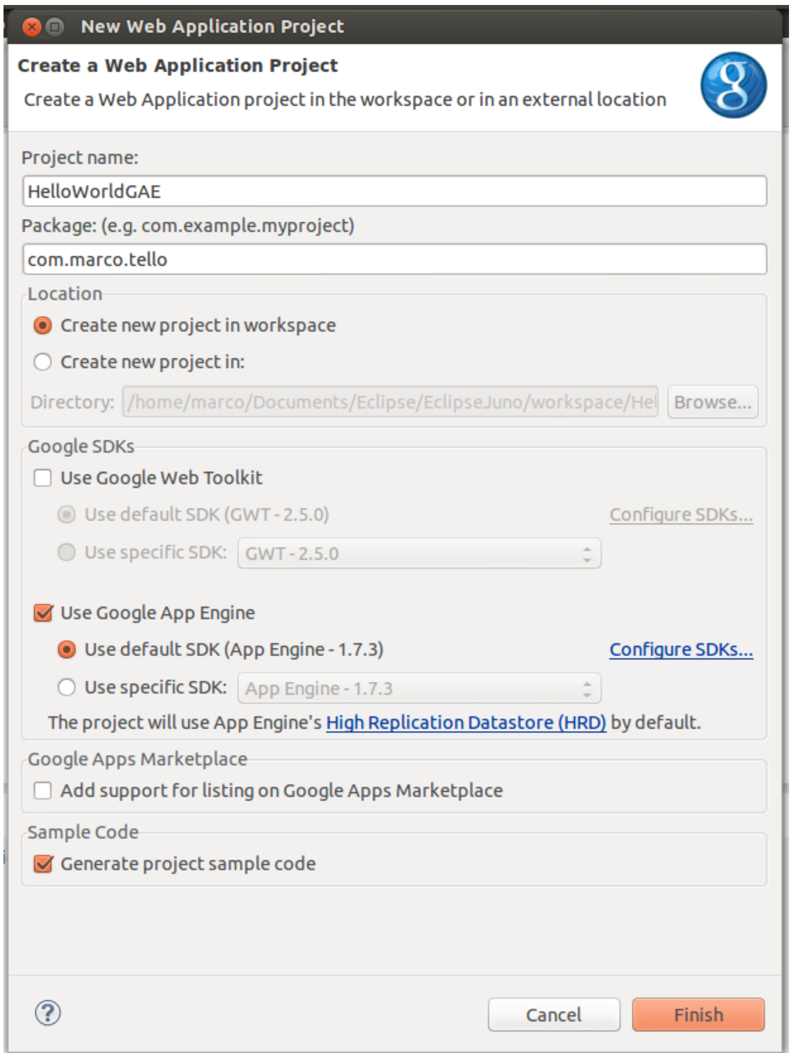
2. Create the GAE Project

Once you have installed the GAE Plugin for Eclipse, a new icon (a blue “g”) will be shown in the tool bar. Click it and a menu will be displayed.


or Source Navigate Search Project Run
New Web Application Project...
Import App Engine Sample Apps...
Import Google Hosted Project...
GWT Compile Project...
Profile Using Speed Tracer...
Deploy to App Engine...
Google Apps Marketplace...
Add Google APIs...

Select New Web Application Project.

A new wizard appears and you have to put the information about your project.


New Web Application Project

Create a Web Application Project

Create a Web Application project in the workspace or in an external location

Project name: HelloWorldGAE

Package: (e.g. com.example.myproject) com.marco.tello

Location

[✓ Create new project in workspace]

Create new project in:

Directory: /home/marco/Documents/Eclipse/EclipseJuno/workspace/Hel

Google SDKs

Use Google Web Toolkit

Use default SDK (GWT - 2.5.0)

Use specific SDK: GWT - 2.5.0

[✓ Use Google App Engine]

[✓ Use default SDK (App Engine - 1.7.3)]

Use specific SDK: App Engine - 1.7.3

The project will use App Engine's High Replication Datastore (HRD) by default.

Google Apps Marketplace

Add support for listing on Google Apps Marketplace

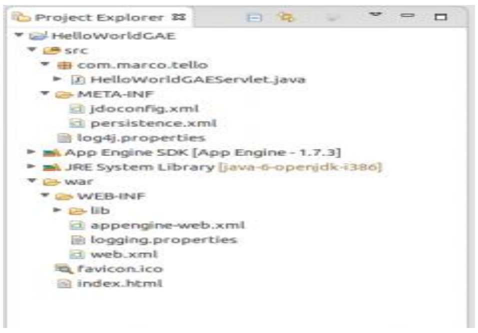
Sample Code

[✓ Generate project sample code]

Make sure that the **Use Google Web Toolkit** is unchecked, then click Finish.

3.-Code your Project

The created project will have this structure:


```
Project Explorer
▾ HelloWorldGAE
  ▾ src
    ▾ com.marco.tello
      ▸ HelloWorldGAEServlet.java
    ▾ META-INF
        jdoconfig.xml
        persistence.xml
      log4j.properties
  ▸ App Engine SDK [App Engine - 1.7.3]
  ▸ JRE System Library [java-6-openjdk-i386]
  ▾ war
    ▾ WEB-INF
      ▸ lib
        appengine-web.xml
        logging.properties
        web.xml
      favicon.ico
      index.html
```

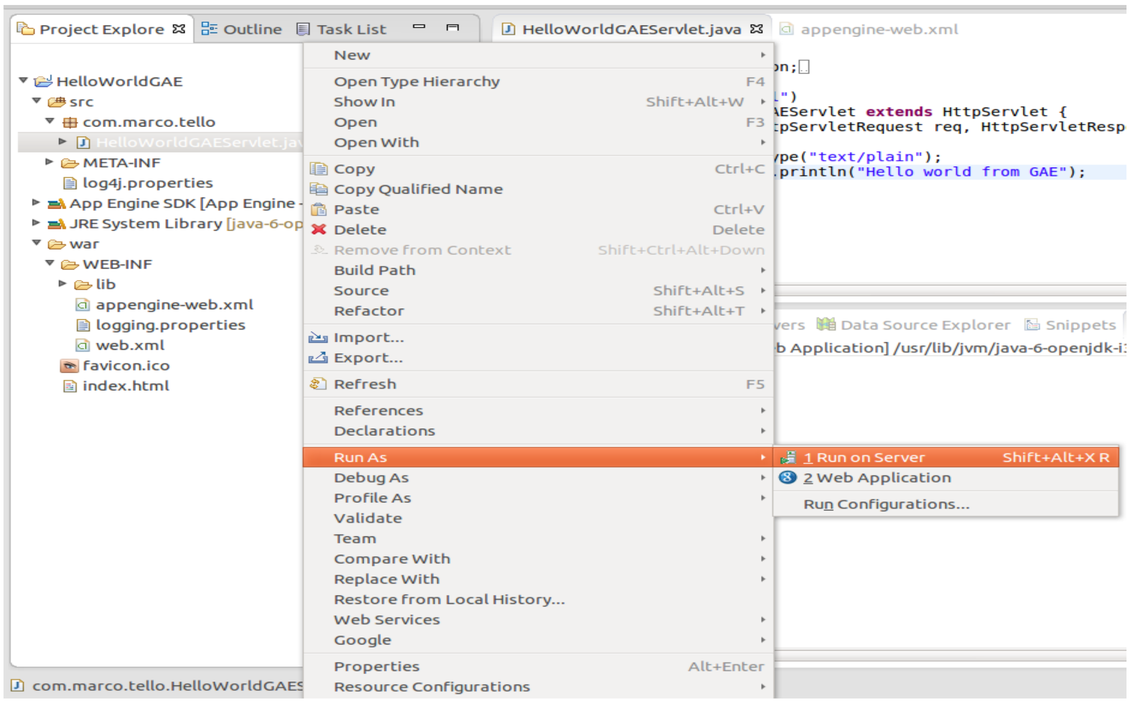
The structure of the project is like a typical Web project with some extra libraries and a **appengine-web.xml** , which we will look into later.

As you can see, a **Servlet** is created. In this Servlet you will put all the logic for the incoming requests that your application will have.
In this example we will return **Hello world from GAE** , so let's take a look into the code.

```
package com.marco.tello;
import java.io.IOException;
import javax.servlet.http.*;
 @SuppressWarnings("serial")
public class HelloWorldGAEServlet extends HttpServlet {
public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws IOException {
                resp.setContentType("text/plain");
                resp.getWriter().println("Hello world from GAE");
        }
}
```

4. Test your Application Local

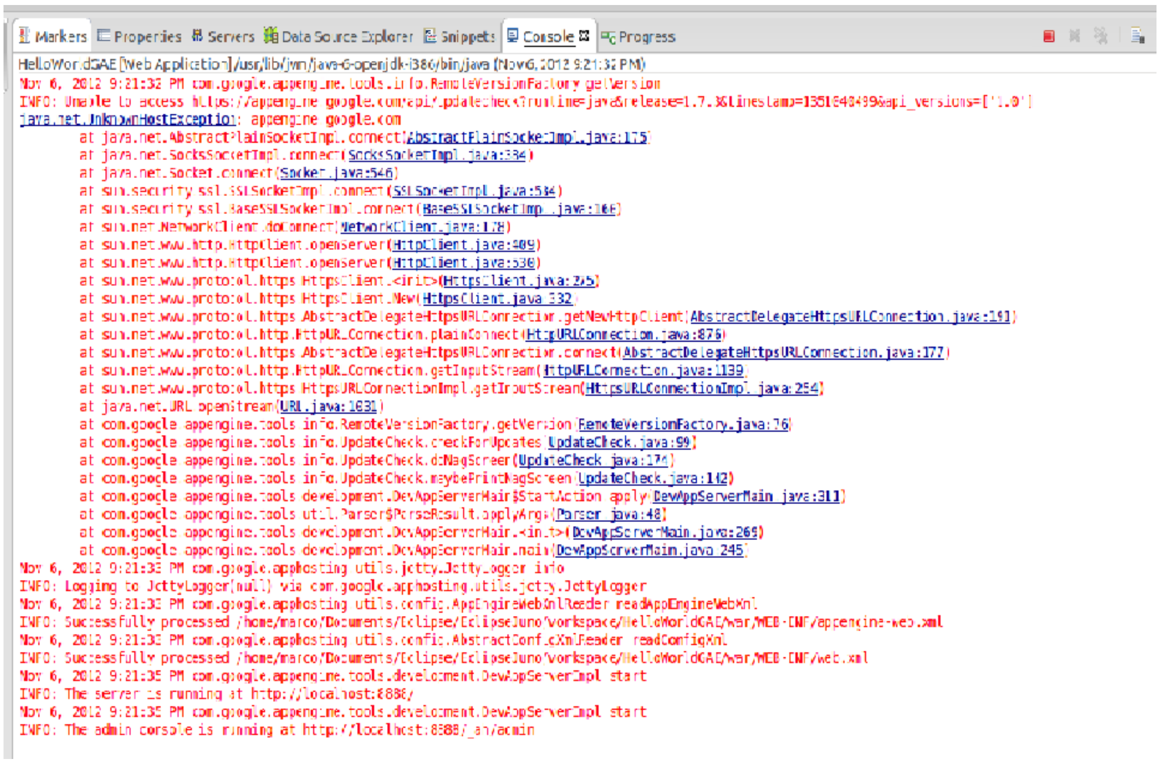
At this point you can test your application in your local environment. Right click on **HelloWorldGAEServlet.java->Run As->Web Application.**



```
on;
")
EServlet extends HttpServlet {
pServletRequest req, HttpServletResp
pe("text/plain");
.println("Hello world from GAE");
```

```
b Application] /usr/lib/jvm/java-6-openjdk-i
```

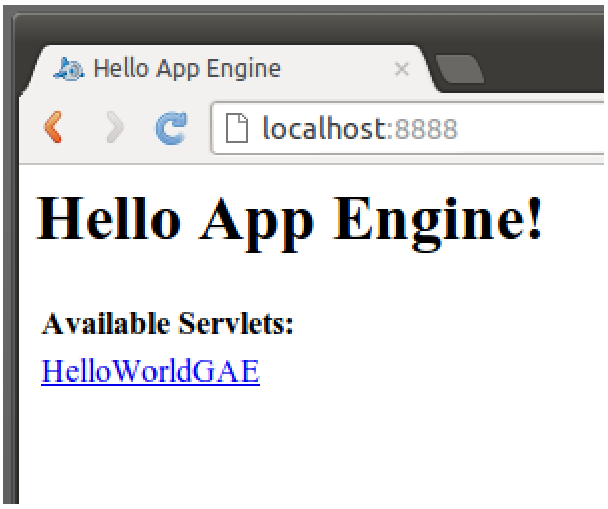
You will see the embedded server starting and when the deploy is ready, something like



```
HelloWorldGAE [Web Application] /usr/lib/jvm/java-6-openjdk-i386/bin/java (Nov 6, 2012 9:21:32 PM)
Nov 6, 2012 9:21:32 PM com.google.appengine.tools.info.RemoteVersionFactory getVersion
INFO: Unable to access https://appengine.google.com/api/updatecheck?runtime=java&release=1.7.3&timestamp=1351648499&api_versions=['1.0']
java.net.UnknownHostException: appengine.google.com
	at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:175)
	at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:384)
	at java.net.Socket.connect(Socket.java:546)
	at sun.security.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:584)
	at sun.security.ssl.BaseSSLSocketImpl.connect(BaseSSLSocketImpl.java:166)
	at sun.net.NetworkClient.doConnect(NetworkClient.java:178)
	at sun.net.www.http.HttpClient.openServer(HttpClient.java:409)
	at sun.net.www.http.HttpClient.openServer(HttpClient.java:530)
	at sun.net.www.protocol.https.HttpsClient.<init>(HttpsClient.java:275)
	at sun.net.www.protocol.https.HttpsClient.New(HttpsClient.java:332)
	at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.getNewHttpClient(AbstractDelegateHttpsURLConnection.java:191)
	at sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:876)
	at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDelegateHttpsURLConnection.java:177)
	at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1139)
	at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(HttpsURLConnectionImpl.java:254)
	at java.net.URL.openStream(URL.java:1031)
	at com.google.appengine.tools.info.RemoteVersionFactory.getVersion(RemoteVersionFactory.java:76)
	at com.google.appengine.tools.info.UpdateCheck.checkForUpdates(UpdateCheck.java:99)
	at com.google.appengine.tools.info.UpdateCheck.doNagScreen(UpdateCheck.java:174)
	at com.google.appengine.tools.info.UpdateCheck.maybePrintNagScreen(UpdateCheck.java:142)
	at com.google.appengine.tools.development.DevAppServerMain$StartAction.apply(DevAppServerMain.java:311)
	at com.google.appengine.tools.util.Parser$ParseResult.applyArgs(Parser.java:48)
	at com.google.appengine.tools.development.DevAppServerMain.<init>(DevAppServerMain.java:265)
	at com.google.appengine.tools.development.DevAppServerMain.main(DevAppServerMain.java:245)
Nov 6, 2012 9:21:33 PM com.google.apphosting.utils.jetty.JettyLogger info
INFO: Logging to JettyLogger(null) via com.google.apphosting.utils.jetty.JettyLogger
Nov 6, 2012 9:21:33 PM com.google.apphosting.utils.config.AppEngineWebXmlReader readAppEngineWebXml
INFO: Successfully processed /home/marco/Documents/Eclipse/EclipseJuno/workspace/HelloWorldGAE/war/WEB-INF/appengine-web.xml
Nov 6, 2012 9:21:33 PM com.google.apphosting.utils.config.AbstractConfigXmlReader readConfigXml
INFO: Successfully processed /home/marco/Documents/Eclipse/EclipseJuno/workspace/HelloWorldGAE/war/WEB-INF/web.xml
Nov 6, 2012 9:21:35 PM com.google.appengine.tools.development.DevAppServerImpl start
INFO: The server is running at http://localhost:8888/
Nov 6, 2012 9:21:35 PM com.google.appengine.tools.development.DevAppServerImpl start
INFO: The admin console is running at http://localhost:8888/_ah/admin
```

is will show in the Console tab.

Open a browser and go to http://localhost:8888/. The main Google App Engine screen comes up.

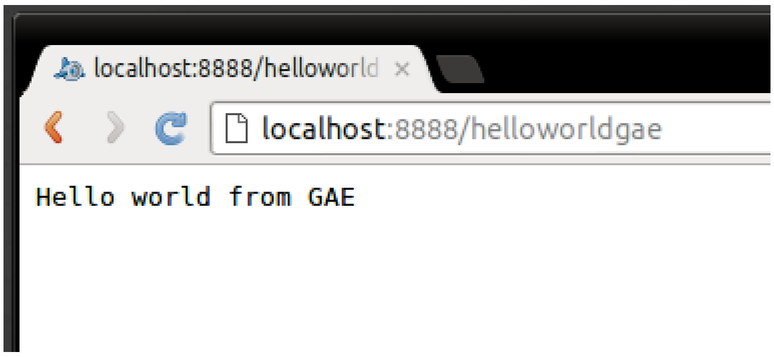


Hello App Engine!

Available Servlets:

HelloWorldGAE

If you click on HelloWorldGAE you will see the greeting from the servlet you just modified.



```
Hello world from GAE
```

That’s it!

5. Deploy to Google App Engine

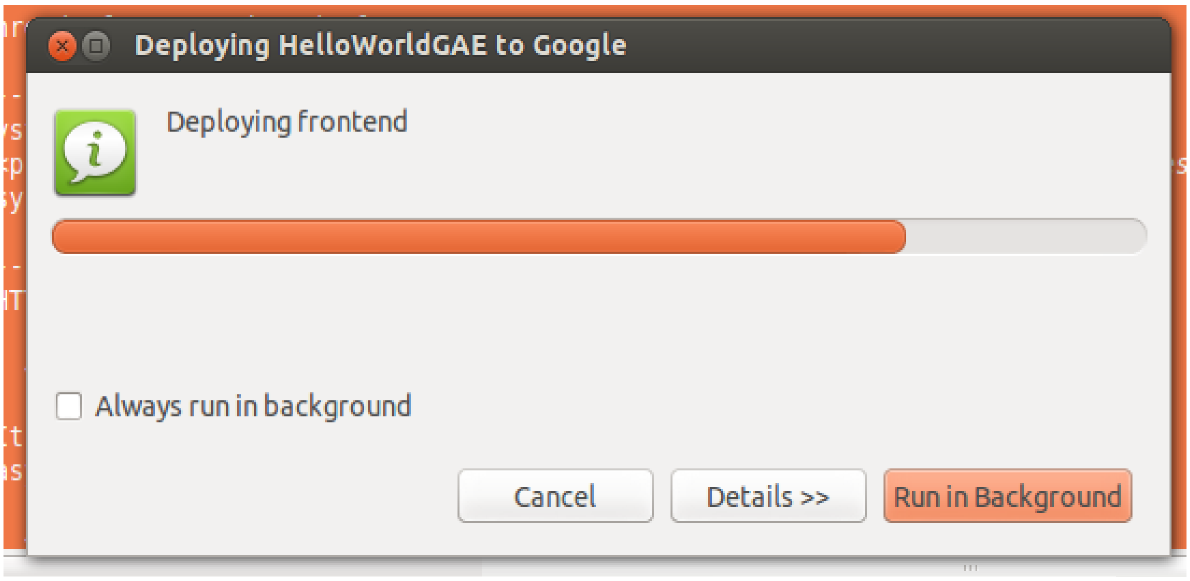
For this step you have to create an account in https://appengine.google.com/ and register you application. Google will only let you register 10 application, so be careful.
Modify the appengine-web.xml with the name of your registered application. In my case it is marcotello-test.

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>marcotello-test</application>
  <version>1</version>
   <!--
    Allows App Engine to send multiple requests to one instance in parallel:
  -->
  <threadsafe>true</threadsafe>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB
INF/logging.properties"/>
  </system-properties>
   <!--
    HTTP Sessions are disabled by default. To enable HTTP sessions specify:
     <sessions-enabled>true</sessions-enabled>
     It's possible to reduce request latency by configuring your application to
    asynchronously write HTTP session data to the datastore:
    <async-session-persistence enabled="true" />
     With this feature enabled, there is a very small chance your app will see
    stale session data. For details, see
    http://code.google.com/appengine/docs/java/config/appconfig.html#Enabling_Sessi
ons
  -->
 </appengine-web-app>
```

Once you save the changes, click on the **g** icon in the tool bar and select **Deploy to**

**App
Engine…**

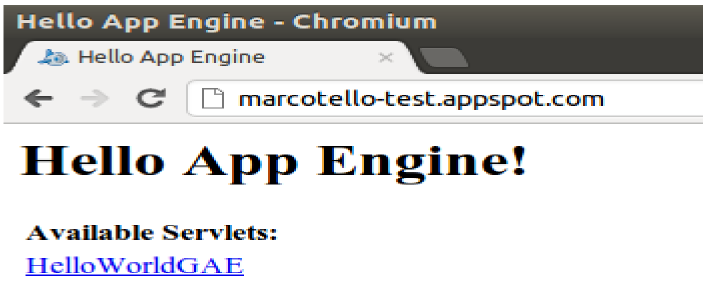
At this point you will have to log in with your Google account and the deployment will
start.



Deploying HelloWorldGAE to Google

Deploying frontend

When the deployment is finished, the following message will show up in the Eclipse console:
Test the application in the browser with the name of your registered application followed by **appspot.com**. In my case it's http://marcotello-test.appspot.com/



Hello App Engine!

Available Servlets:
HelloWorldGAE

PRE LAB VIVA QUESTIONS

1. What is meant by google App Engine?
2. How does google App Engine work ?
3. What are different versions of google App Engine?
4. What is Eclipse IDE?

POST LAB VIVA QUESTIONS

1. What does Eclipse IDE work ?
2. What is the Eclipse Plugin?
3. What is the use of Eclipse Plugin ?
4. Define the java persistence API ?

WEEK 13

Creating a Guestbook Application

This tutorial shows you how to build and run a sample Python application for App Engine and provides a code walkthrough of the sample code. The sample is a simple guestbook that lets users post messages to a public message board.

Objectives

- Build and test an App Engine app using Python.
- Integrate your application with Google Accounts for user authentication.
- Use the webapp2 framework.
- Use Jinja2 templates.
- Store data in Cloud Datastore.
- Deploy your app to App Engine.
- Costs

App Engine has generous free quotas that will cover your testing this tutorial in a live production environment.

Before you begin

1. Create a new GCP Console project or retrieve the project ID of an existing project from the Google Cloud Platform Console:
   GO TO THE PROJECTS PAGE

   Tip: Retrieve a list of your existing project IDs with gcloud.
2. Install the Google Cloud SDK and then initialize the gcloud tool:
   DOWNLOAD THE SDK

Cloning the project from GitHub

1. Clone the Guestbook application repository to your local machine:

```
git clone https://github.com/GoogleCloudPlatform/appengine-guestbook-python.git
```

Go to the directory that contains the sample code:

```
cd appengine-guestbook-python
```

Building and running locally

To build and run the sample locally:

1. Start the local development web server by running the following command from the appengine-guestbook-python directory: dev_appserver.py ./
   The development web server runs and listens for requests on port 8080.
   Note: The dev_appserver.py tool is installed globally with the App Engine SDK whether you installed with the Cloud SDK or the standalone SDK.
2. Visit http://localhost:8080/ in your web browser to view the app.

   Click **Login**, then sign in with any email address. The development server accepts any email you supply, valid or not. This same code requires a valid Google Account and email when deployed to production.
3. Stop the development server by pressing Control+C.

Authenticating Users

This part of the Python Guestbook code walkthrough shows how to authenticate users and display a customized greeting for the signed-in user.

This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

Signing in users

The `MainPage` class defines a handler for HTTP `GET` requests to the root path `'/'`. The handler checks to see whether a user is signed in:

guestbook.py

```
class MainPage(webapp2.RequestHandler):
    def get(self):
    guestbook_name = self.request.get('guestbook_name',
                                      DEFAULT_GUESTBOOK_NAME)
    greetings_query = Greeting.query(
    ancestor=guestbook_key(guestbook_name)).order(-Greeting.date)
    greetings = greetings_query.fetch(10)
    user = users.get_current_user()
    if user:
       url = users.create_logout_url(self.request.uri)
       url_linktext = 'Logout'
    else:
       url = users.create_login_url(self.request.uri)
       url_linktext = 'Login'
       template_values = {
       'user': user,
       'greetings': greetings,
       'guestbook_name': urllib.quote_plus(guestbook_name),
       'url': url,
       'url_linktext': url_linktext,
    }
    template = JINJA_ENVIRONMENT.get_template('index.html')
    self.response.write(template.render(template_values))
```

If the user is already signed in to your application, the get_current_user() method returns a Userobject, and the app displays the users's nickname. If the user has not signed in, the code redirects the user's browser to the Google account sign-in screen. The Google account sign-in mechanism sends the user back to the app after the user has signed in.

For more information, see the Users API.

Handling User Input in a Form

This part of the Python Guestbook code walkthrough shows how to handle user input.

This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

Configuring the app to use webapp2

The Guestbook sample uses the webapp2 framework, which is included in the App Engine environment and the App Engine Python SDK. You don't need to bundle webapp2 with your application code to use it.

The app.yaml file specifies that the app uses the webapp2 framework:

app.yaml

```
libraries:
name: webapp2
version: latest
name: jinja2
version: latest
```

A webapp2 application has two parts:

- One or more RequestHandler classes that process requests and build responses.
- A WSGIApplication instance that routes incoming requests to handlers based on the URL.

The app.yaml file specifies the app object in guestbook.py as the handler for all URLs:
app.yaml

```
handlers:
  url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico
  url: /bootstrap
  static_dir: bootstrap
  url: /.*
  script: guestbook.app
```

Defining a handler for form submission

The app object in guestbook.py is a WSGIApplication that defines which scripts handle requests for given URLs.
guestbook.py

```
app = webapp2.WSGIApplication([
    ('/', MainPage),
    ('/sign', Guestbook),
], debug=True)
```

The debug=True parameter tells webapp2 to print stack traces to the browser output if a handler encounters an error or raises an uncaught exception. This option should be removed before deploying the final version of your application, otherwise you will inadvertently expose the internals of your application.

The Guestbook handler has a post() method instead of a get() method. This is because the form displayed by Main Page uses the HTTP POST method to submit the form data.
guestbook.py

```
class Guestbook(webapp2.RequestHandler):
    def post(self):
        # We set the same parent key on the 'Greeting' to ensure each
        # Greeting is in the same entity group. Queries across the
        # single entity group will be consistent. However, the write
        # rate to a single entity group should be limited to
        # ~1/second.
        guestbook_name = self.request.get('guestbook_name',
                                          DEFAULT_GUESTBOOK_NAME)
        greeting = Greeting(parent=guestbook_key(guestbook_name))
        if users.get_current_user():
            greeting.author = Author(
                    identity=users.get_current_user().user_id(),
                    email=users.get_current_user().email())
        greeting.content = self.request.get('content')
        greeting.put()
        query_params = {'guestbook_name': guestbook_name}
        self.redirect('/?' + urllib.urlencode(query_params))
```

The post() method gets the form data from self.request.

Generating Dynamic Content from Templates

This part of the Python Guestbook code walkthrough shows how to use Jinja templates to generate dynamic web content.

This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

HTML embedded in code is messy and difficult to maintain. It's better to use a templating system, where the HTML is kept in a separate file with special syntax to indicate where the data from the application appears. There are many templating systems for Python: EZT, Cheetah, ClearSilver,Quixote, Django, and Jinja2 are just a few. You can use your template engine of choice by bundling it with your application code.

For your convenience, App Engine includes the Django and Jinja2 templating engines.

Using Jinja2 Templates

The app.yaml file lists the latest version of jinja2 as a required library. Production applications should use an actual version number rather than version: latest.

app.yaml

```
libraries:
 name: webapp2
 version: latest
name: jinja2
version: latest
```

The app imports jinja2 and creates a jinja2.Environment object.

guestbook.py

```
import os
import urllib
from google.appengine.api import users
from google.appengine.ext import ndb
import jinja2
import webapp2
JINJA_ENVIRONMENT = jinja2.Environment(
    loader=jinja2.FileSystemLoader(os.path.dirname(__file__)),
    extensions=['jinja2.ext.autoescape'],
    autoescape=True)
```

The get method for the MainPage request handler forms a dictionary of key/value pairs and passes it to template.render.
guestbook.py

```
class MainPage(webapp2.RequestHandler):
    def get(self):
        guestbook_name = self.request.get('guestbook_name',
                                          DEFAULT_GUESTBOOK_NAME)
        greetings_query = Greeting.query(
            ancestor=guestbook_key(guestbook_name)).order(-Greeting.date)
        greetings = greetings_query.fetch(10)
        user = users.get_current_user()
        if user:
            url = users.create_logout_url(self.request.uri)
            url_linktext = 'Logout'
        else:
            url = users.create_login_url(self.request.uri)
            url_linktext = 'Login'
            template_values = {
            'user': user,
            'greetings': greetings,
            'guestbook_name': urllib.quote_plus(guestbook_name),
            'url': url,
            'url_linktext': url_linktext,
        }
        template = JINJA_ENVIRONMENT.get_template('index.html')
        self.response.write(template.render(template_values))
```

The page is rendered according to the index.html template, which receives the dictionary as input.
index.html

```
{% for greeting in greetings %}
<div class="row">
  {% if greeting.author %}
    <b>{{ greeting.author.email }}
      {% if user and user.user_id() == greeting.author.identity %}
        (You)
      {% endif %}
    </b> wrote:
  {% else %}
    An anonymous person wrote:
  {% endif %}
  <blockquote>{{ greeting.content }}</blockquote>
</div>
{% endfor %}
```

The JINJA_ENVIRONMENT.get_template(name) method takes the name of a template file and returns a template object. The template.render(template_values) call

takes a dictionary of values, and returns the rendered text. The template uses Jinja2 templating syntax to access and iterate over the values, and can refer to properties of those values.
< PREVNEXT >

Storing Data in Cloud Datastore
This part of the Python Guestbook code walkthrough shows how to store structured data in Cloud Datastore. With App Engine and Cloud Datastore, you don't have to worry about distribution, replication, and load balancing of data. That is done for you behind a simple API—and you get a powerful query engine and transactions as well.
Note: Cloud Datastore is only one option you have for storing application data. If you prefer to use relational data and SQL instead of data structures, try walking through the instructions for using Google Cloud SQL, which also use the Guestbook example application.
This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

Storing the submitted greetings
Data is written to Cloud Datastore in objects known as entities. Each entity has a key that uniquely identifies it. An entity can optionally designate another entity as its parent; the first entity is a child of the parent entity. The entities in the data store thus form a hierarchically-structured space similar to the directory structure of a file system. For detailed information, see Structuring Data for Strong Consistency.
App Engine includes a data modeling API for Python. To use the data modeling API, the sample app imports the google.appengine.ext.ndb module. Each greeting includes the author's name, the message content, and the date and time the message was posted. The app displays messages in chronological order. The following code defines the data model:

```
guestbook.py
class Author(ndb.Model):
"""Sub model for representing an author."""
identity = ndb.StringProperty(indexed=False)
email = ndb.StringProperty(indexed=False)
class Greeting(ndb.Model):
"""A main model for representing an individual Guestbook entry."""
author = ndb.StructuredProperty(Author)
content = ndb.StringProperty(indexed=False)
date = ndb.DateTimeProperty(auto_now_add=True)
```

- The code defines a Greeting model with three properties: author whose value is an Authorobject with the email address and the author's identity, content whose value is a string, and datewhose value is a datetime.datetime.
- Some property constructors take parameters to further configure their behavior. Passing the ndb.StringProperty constructor the indexed=False parameter says

that values for this property will not be indexed. This saves writes which aren't needed because the app never uses that property in a query.

- Passing the ndb.DateTimeProperty constructor
  an auto_now_add=True parameter configures the model to automatically give new objects a datetime stamp of the time the object is created, if the application doesn't otherwise provide a value. For a complete list of property types and their options, see NDB Properties.

The application uses the data model to create new Greeting objects and put them into Cloud Datastore. The Guestbook handler creates new greetings and saves them to the data store:

guestbook.py

```
class Guestbook(webapp2.RequestHandler):
 def post(self):
        # We set the same parent key on the 'Greeting' to ensure each
        # Greeting is in the same entity group. Queries across the
        # single entity group will be consistent. However, the write
        # rate to a single entity group should be limited to
        # ~1/second.
        guestbook_name = self.request.get('guestbook_name',
        greeting = Greeting(parent=guestbook_key(guestbook_name))
        if users.get_current_user():
         greeting.author = Author(
         identity=users.get_current_user().user_id(),
                    email=users.get_current_user().email())
                   greeting.content = self.request.get('content')
        greeting.put()
        query_params = {'guestbook_name': guestbook_name}
        self.redirect('/?' + urllib.urlencode(query_params))
```

This Guestbook handler creates a new Greeting object, then sets its author and contentproperties with the data posted by the user. The parent of Greeting is a Guestbook entity. There's no need to create the Guestbook entity before setting it to be the parent of another entity. In this example, the parent is used as a placeholder for transaction and consistency purposes. See theTransactions page for more information. Objects that share a common ancestor belong to the same entity group. The code does not set the date property, so date is automatically set to the present, using auto_now_add=True.

Finally, greeting.put() saves the new object to the data store. If we had acquired this object from a query, put() would have updated the existing object. Because we created this object with the model constructor, put() adds the new object to the data store.

Because querying in Cloud Datastore is strongly consistent only within entity groups, the code assigns all of one book's greetings to the same entity group by setting the same parent for each greeting. This means the user always sees a greeting immediately after it is written. However, the rate at which you can write to the same entity group is limited to one write to the entity group per second. When you design a real application you'll need to keep this fact in mind. Note that by using services such as Memcache, you can lower the chance that a user sees stale results when querying across entity groups after a write.

Retrieving submitted greetings

Cloud Datastore has a sophisticated query engine for data models. Because Cloud Datastore is not a traditional relational database, queries are not specified using SQL. Instead, data is queried one of two ways: either by using Datastore queries, or by using an SQL-like query language called GQL. To access the full range of Cloud Datastore's query capabilities, we recommend using queries over GQL.

The MainPage handler retrieves and displays previously submitted greetings.

The greetings_query.fetch(10) call performs the query.

More about Cloud Datastore indexes

Every query in Cloud Datastore is computed from one or more indexes—tables that map ordered property values to entity keys. This is how App Engine is able to serve results quickly regardless of the size of your application's data store. Many queries can be computed from the built-in indexes, but for queries that are more complex, Cloud Datastore requires a custom index. Without a custom index, Cloud Datastore can't execute these queries efficiently.For example, the Guestbook application filters by guestbook, and orders by date, using an ancestor query and a sort order. This requires a custom index to be specified in the application's index.yamlfile. You can edit this file manually, or you can take care of it automatically by running the queries in the application locally. After the index is defined in index.yaml, deploying the application will also deploy the custom index information.

The definition for the query in index.yaml looks like this:

index.yaml

```
  indexes:
  kind: Greeting
  ancestor: yes
  properties:
  name: date
  direction: desc
```

You can read all about Cloud Datastore indexes in the Datastore Indexes page. You can read about the proper specification for index.yaml files in Python Datastore Index Configuration.

Serving Static Files

This part of the Python Guestbook code walkthrough shows how to serve static files. App Engine does not serve files directly out of your application's source directory unless configured to do so. But there are many cases where you want to serve static files directly to the web browser. Images, CSS stylesheets, JavaScript code, movies, and Flash animations are all typically stored with a web application and served directly to the browser.

This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

Configuring the app to use static files

The CSS files for the Guestbook app are in the bootstrap/css directory. The template for the app's web page, index.html, instructs the browser to load bootstrap.css and bootstrap-responsive.css, which are static files:

index.html

```
<link type="text/css" rel="stylesheet" href="/bootstrap/css/bootstrap.css">
<link type="text/css" rel="stylesheet" href="/bootstrap/css/bootstrap-responsive.css">
```

The app.yaml file specifies the bootstrap directory as the location for static files:

app.yaml

```
handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico
- url: /bootstrap
  static_dir: bootstrap
- url: /.*
  script: guestbook.app
```

The handlers section defines two handlers for URLs. When App Engine receives a request for a URL beginning with /bootstrap, it maps the remainder of the path to files in the bootstrap directory, and if an appropriate file is found, the contents of the file are returned to the client. All other URLs match the /.* pattern, and are handled by the app object in the guestbook module.

URL path patterns are tested in the order they appear in app.yaml. In this case, the /bootstrappattern matches before the /.* pattern for the appropriate paths. For more information on URL mapping and other options you can specify in app.yaml, see the app.yaml reference.

Serving Static Files

This part of the Python Guestbook code walkthrough shows how to serve static files. App Engine does not serve files directly out of your application's source directory unless configured to do so. But there are many cases where you want to serve static files directly to the web browser. Images, CSS stylesheets, JavaScript code, movies, and Flash animations are all typically stored with a web application and served directly to the browser.

This page is part of a multi-page tutorial. To start from the beginning and see instructions for setting up, go to Creating a Guestbook.

Configuring the app to use static files

The CSS files for the Guestbook app are in the bootstrap/css directory. The template for the app's web page, index.html, instructs the browser to load bootstrap.css and bootstrap-responsive.css, which are static files:

index.html

```
<link type="text/css" rel="stylesheet" href="/bootstrap/css/bootstrap.css">
<link type="text/css" rel="stylesheet" href="/bootstrap/css/bootstrap-responsive.css">
```

The app.yaml file specifies the bootstrap directory as the location for static files:

app.yaml

```
handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico
- url: /bootstrap
  static_dir: bootstrap
- url: /.*
  script: guestbook.app
```

The handlers section defines two handlers for URLs. When App Engine receives a request for a URL beginning with /bootstrap, it maps the remainder of the path to files in the bootstrap directory, and if an appropriate file is found, the contents of the file are returned to the client. All other URLs match the /.* pattern, and are handled by the app object in the guestbook module.

URL path patterns are tested in the order they appear in app.yaml. In this case, the /bootstrap pattern matches before the /.* pattern for the appropriate paths. For more information on URL mapping and other options you can specify in app.yaml, see the app.yaml reference.

PRE LAB VIVA QUESTIONS?

1. What is meant by Google App Engine?
2. How does Google App Engine work ?
3. What are different versions of google App Engine?
4. What is Eclipse IDE?

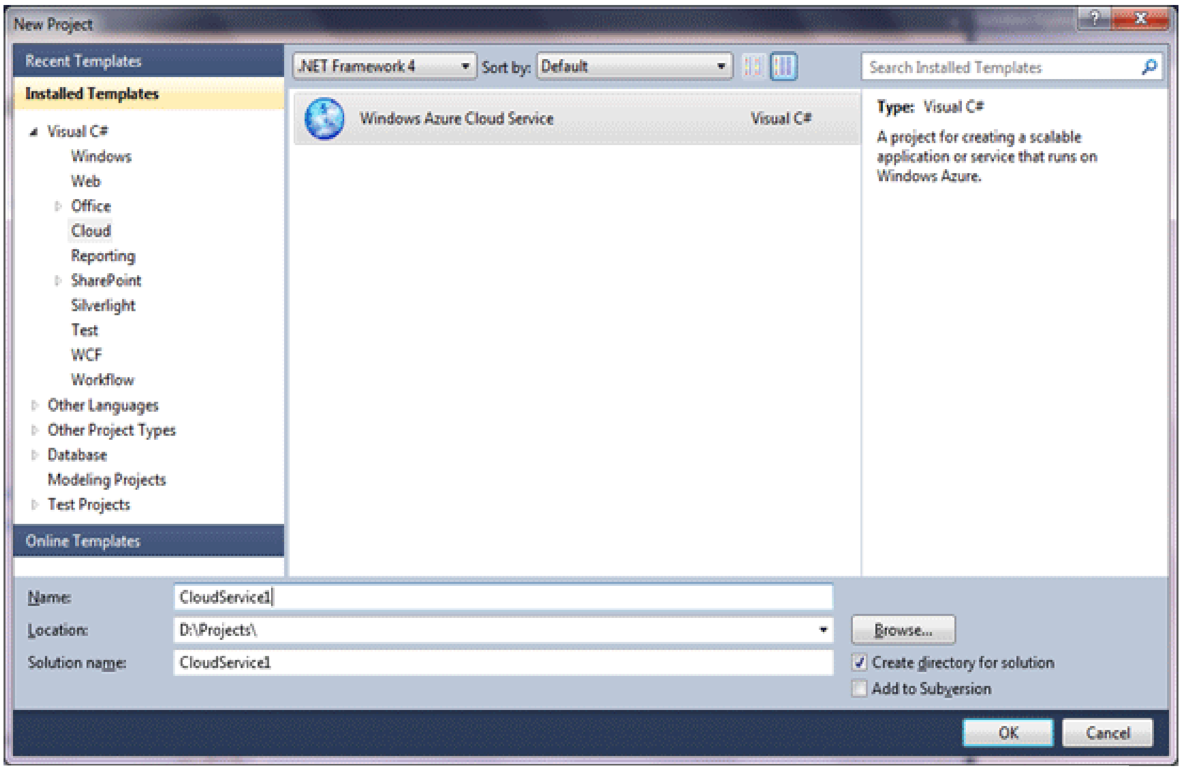
POST LAB VIVA QUESTIONS?

1. What does Eclipse IDE work ?
2. What is the Eclipse Plugin?
3. What is the use of Eclipse Plugin ?
4. Define the java persistence API ?

WEEK-14

Develop a Windows Azure Hello World application

In this experiment, we will see how to create a "Hello World!!!" application in Azure using .Net. I used Visual Studio 2010 along with the Azure SDK 1.3 installed. Start Visual Studio and select a new project. In the cloud template select Windows Azure Cloud Service.



Type: Visual C#

A project for creating a scalable application or service that runs on Windows Azure.

Name: CloudService1

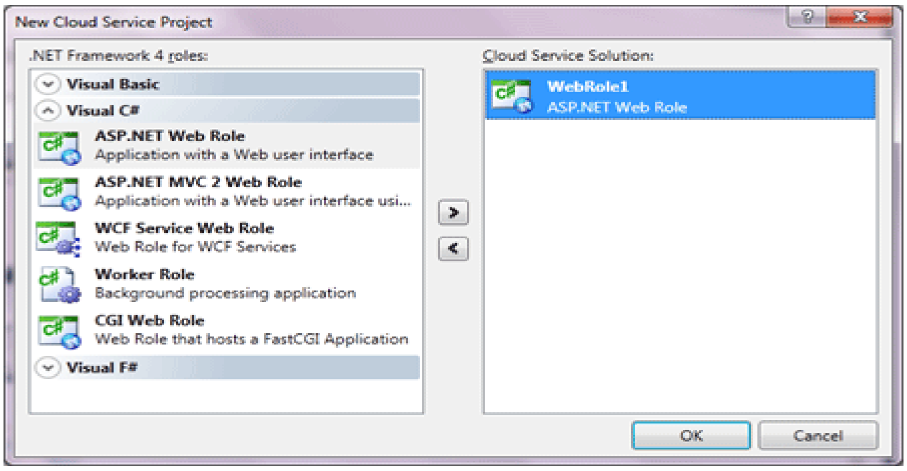
Location: D:\Projects\

Solution name: CloudService1

☑ Create directory for solution

☐ Add to Subversion

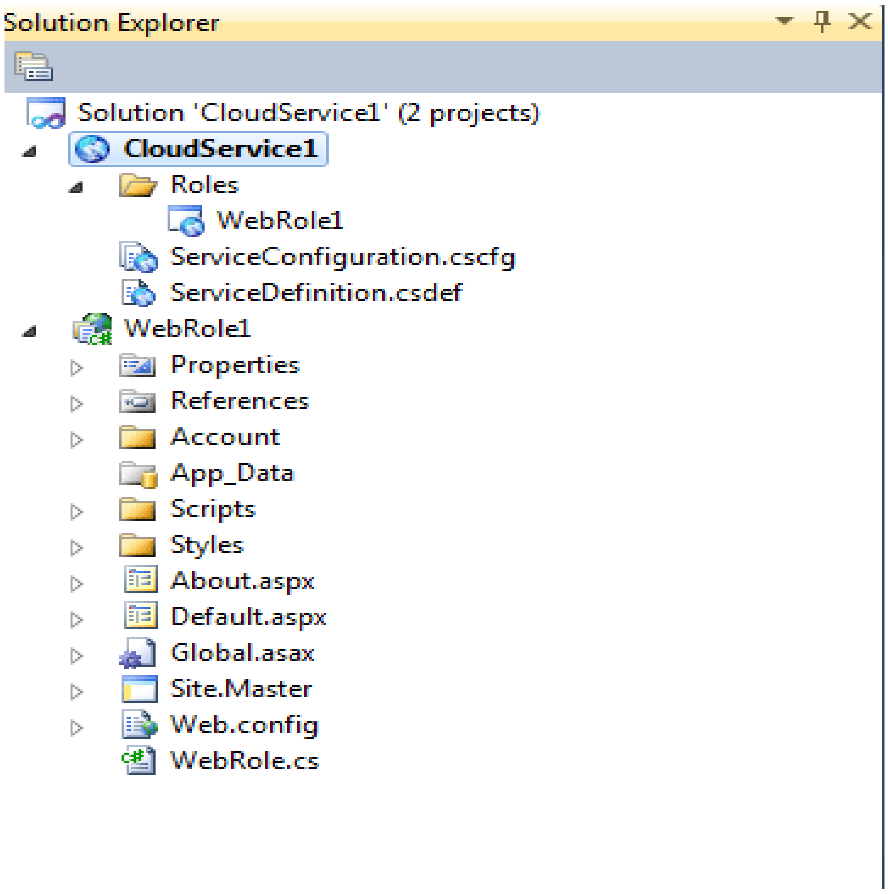
Once you click OK, it asks for selecting a Role. Here we are adding an ASP.Net Web role.


New Cloud Service Project
.NET Framework 4 roles:
Visual Basic
Visual C#
ASP.NET Web Role
Application with a Web user interface
ASP.NET MVC 2 Web Role
Application with a Web user interface usi...
WCF Service Web Role
Web Role for WCF Services
Worker Role
Background processing application
CGI Web Role
Web Role that hosts a FastCGI Application
Visual F#
Cloud Service Solution:
OK
Cancel

WebRole1
ASP.NET Web Role

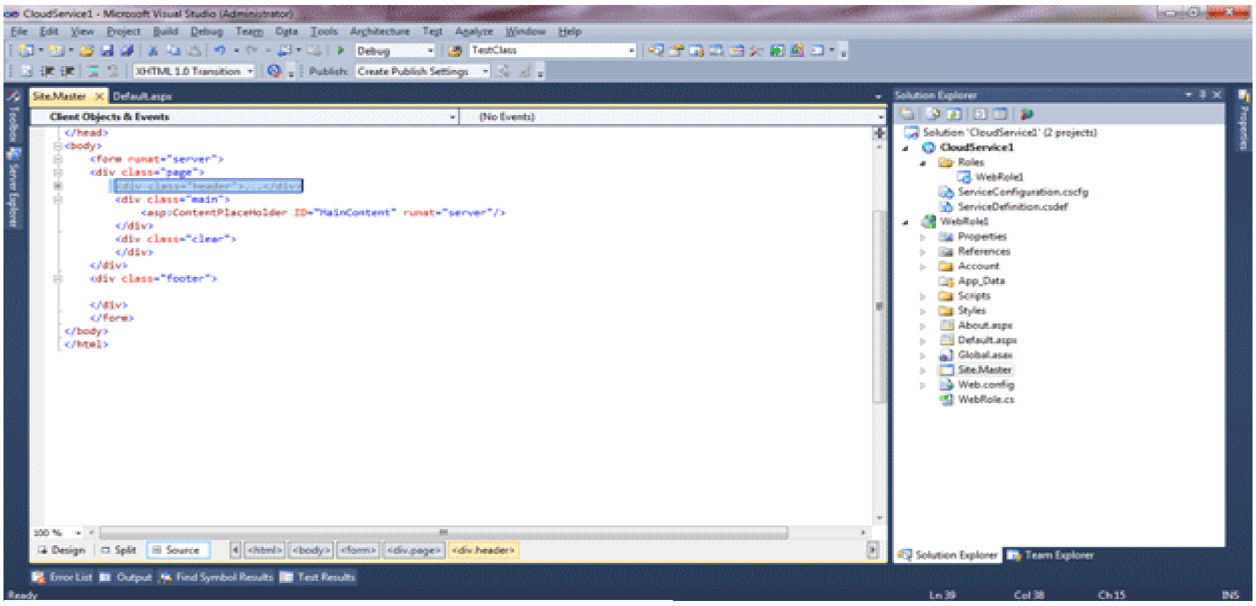
It will create two projects, CloudService1 (Azure Service Project) and WebRole1

(ASP.Net Project). The Azure service project is used to configure the application, and to create a deployment package. The ASP.Net project is like a normal web project.



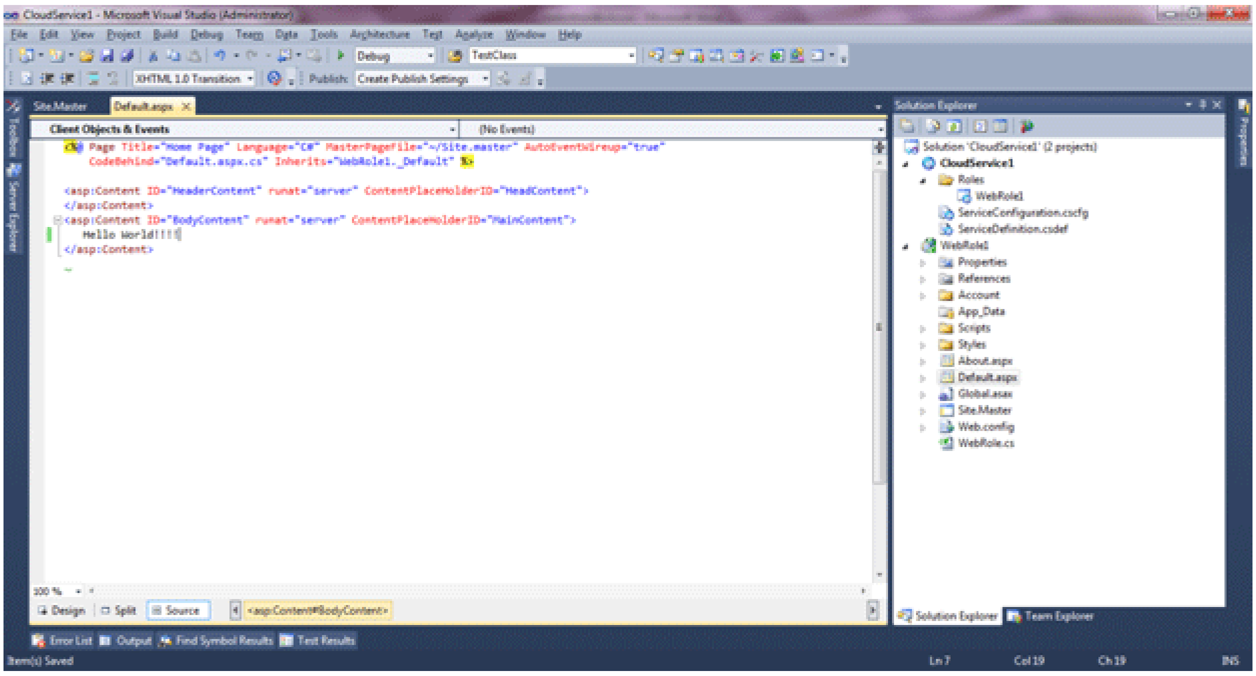
```
Solution Explorer
Solution 'CloudService1' (2 projects)
◢ CloudService1
    ◢ Roles
        WebRole1
      ServiceConfiguration.cscfg
      ServiceDefinition.csdef
◢ WebRole1
    ▷ Properties
    ▷ References
    ▷ Account
      App_Data
    ▷ Scripts
    ▷ Styles
    ▷ About.aspx
    ▷ Default.aspx
    ▷ Global.asax
    ▷ Site.Master
    ▷ Web.config
      WebRole.cs
```

Here i removed header div from the SiteMaster.



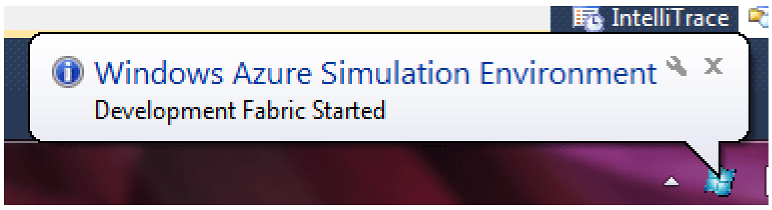
```
</head>
<body>
    <form runat="server">
    <div class="page">
        <div class="header">...</div>
        <div class="main">
            <asp:ContentPlaceHolder ID="MainContent" runat="server"/>
        </div>
        <div class="clear">
        </div>
    </div>
    <div class="footer">

    </div>
    </form>
</body>
</html>
```

And add a text "Hello World!!!!" in the default.aspx



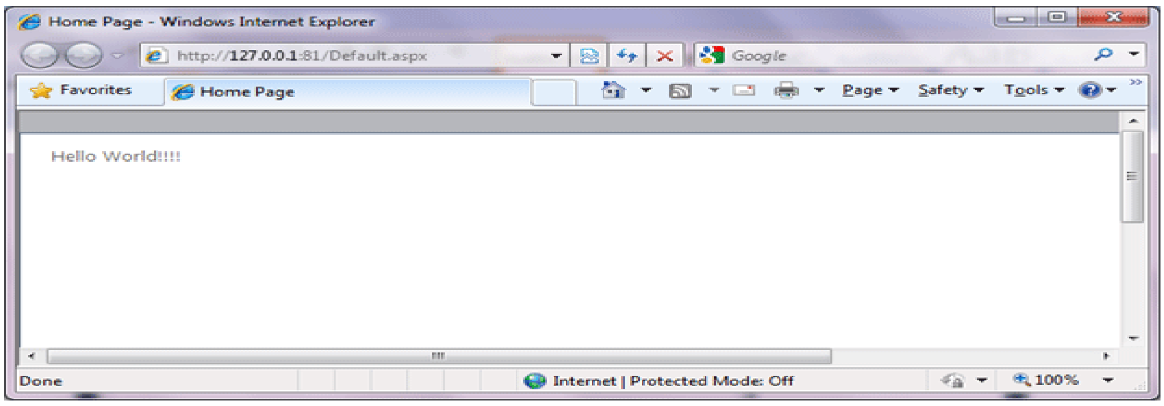
```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="WebRole1._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    Hello World!!!!
</asp:Content>
```

And we are done. Now Select Cloud Service as your startup project and run it. When you
run it, the Azure simulation environment is initialized, and it starts development storage
and development fabric.



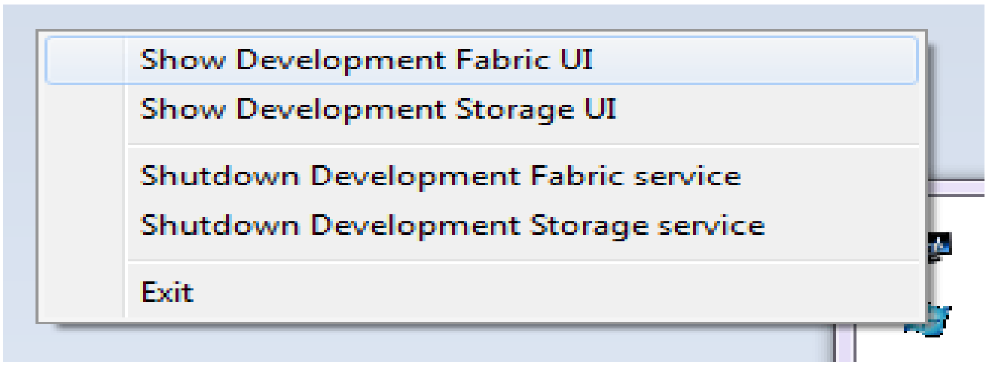
Windows Azure Simulation Environment
Development Fabric Started

And here you can see your Hello World!!!

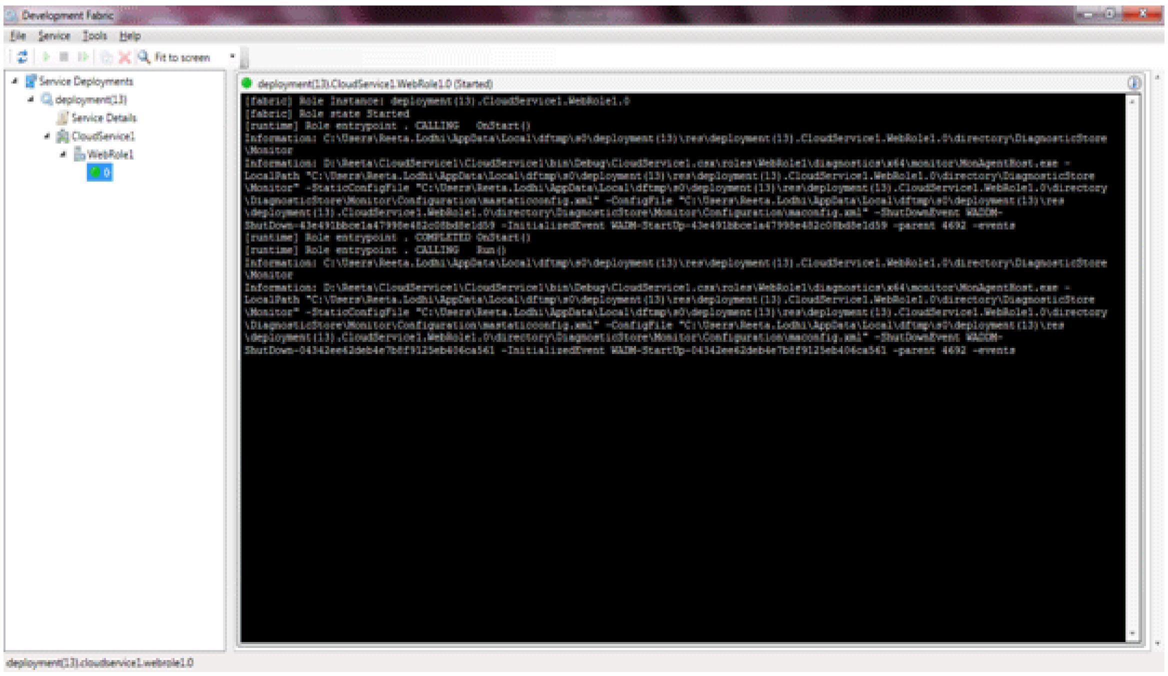


Hello World!!!!

If you select WebRole1 as your startup project and run the solution, it will run as normalASP.Net project and hosted locally on localhost.
If you click on the Azure Simulation and select "Show development Fabric UI" then it will show the Azure services running on your machine.


Show Development Fabric UI
Show Development Storage UI
Shutdown Development Fabric service
Shutdown Development Storage service
Exit

Here My CloudService1 is running having a single instance of WebRole.



```
[fabric] Role Instance: deployment(13).CloudService1.WebRole1.0
[fabric] Role state Started
[runtime] Role entrypoint . CALLING   OnStart()
Information: C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore
\Monitor
Information: D:\Reeta\CloudService1\CloudService1\bin\Debug\CloudService1.csx\roles\WebRole1\diagnostics\x64\monitor\MonAgentHost.exe -
LocalPath "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore
\Monitor" -StaticConfigFile "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory
\DiagnosticStore\Monitor\Configuration\mastaticconfig.xml" -ConfigFile "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res
\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore\Monitor\Configuration\maconfig.xml" -ShutDownEvent WADDM-
ShutDown-43e491bbce1a47998e482c08bd8e1d59 -InitializedEvent WADM-StartUp-43e491bbce1a47998e482c08bd8e1d59 -parent 4692 -events
[runtime] Role entrypoint . COMPLETED OnStart()
[runtime] Role entrypoint . CALLING   Run()
Information: C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore
\Monitor
Information: D:\Reeta\CloudService1\CloudService1\bin\Debug\CloudService1.csx\roles\WebRole1\diagnostics\x64\monitor\MonAgentHost.exe -
LocalPath "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore
\Monitor" -StaticConfigFile "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res\deployment(13).CloudService1.WebRole1.0\directory
\DiagnosticStore\Monitor\Configuration\mastaticconfig.xml" -ConfigFile "C:\Users\Reeta.Lodhi\AppData\Local\dftmp\s0\deployment(13)\res
\deployment(13).CloudService1.WebRole1.0\directory\DiagnosticStore\Monitor\Configuration\maconfig.xml" -ShutDownEvent WADDM-
ShutDown-04342ee62deb4e7b8f9125eb406ca561 -InitializedEvent WADM-StartUp-04342ee62deb4e7b8f9125eb406ca561 -parent 4692 -events
```

PRE LAB VIVA QUESTIONS?

1. What is Microsoft Azure and why it is used?
2. Which service in Azure is used to manage resource in Azure ?
3. What is Availability set?
4. Why is Azure Active Directory used?

POST LAB VIVA QUESTIONS?

1. What is Azure Redis Cache?
2. What is Azure Search?
3. What is the meaning of Application Partitions?
4. What are redis databases?

WEEK 15
Create a Mashup using Yahoo! Pipes.

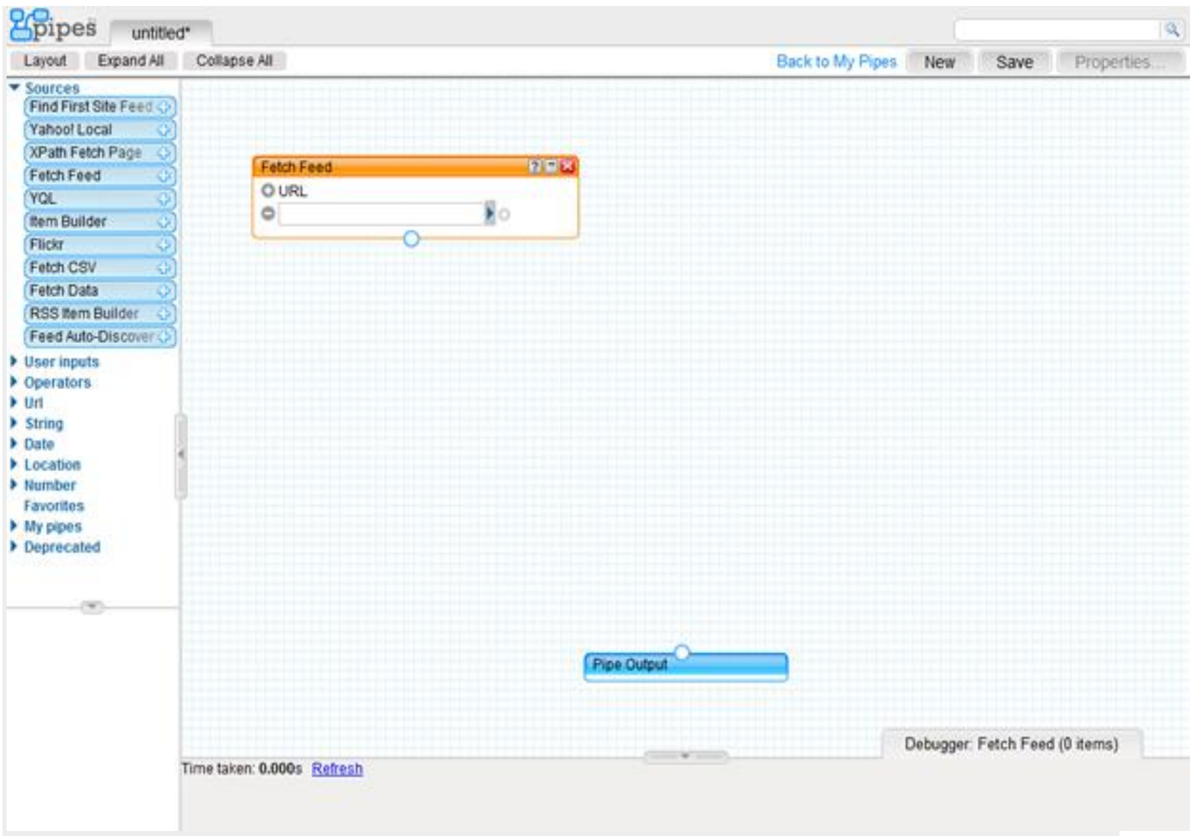
The internet is a great resource for news and updates, and no matter what you're looking to keep track of, you're sure to be able to find countless sites that will be able to keep you up to date with the latest information. To help make it easier to keep track of new developments, you might make use of an RSS feed to save you having to look things up manually. You might already be used to using RSS in apps like Google Reader, but there's so much more you can do with RSS feeds.

Pipes is a tool from Yahoo that enables you to take things a step further so you can, amongst other things, create your own custom RSS feeds that pull in content from a variety of sources and filter it so that you only see the most relevant news stories. It's a venerable web app, starting off life in a rather Google-ish way of being in a lengthy period of beta but then living on for years, long enough that many of us have likely forgotten about it. But it's still a great tool, even in 2012, so let's dig in and see what you can do with it.

Getting Started

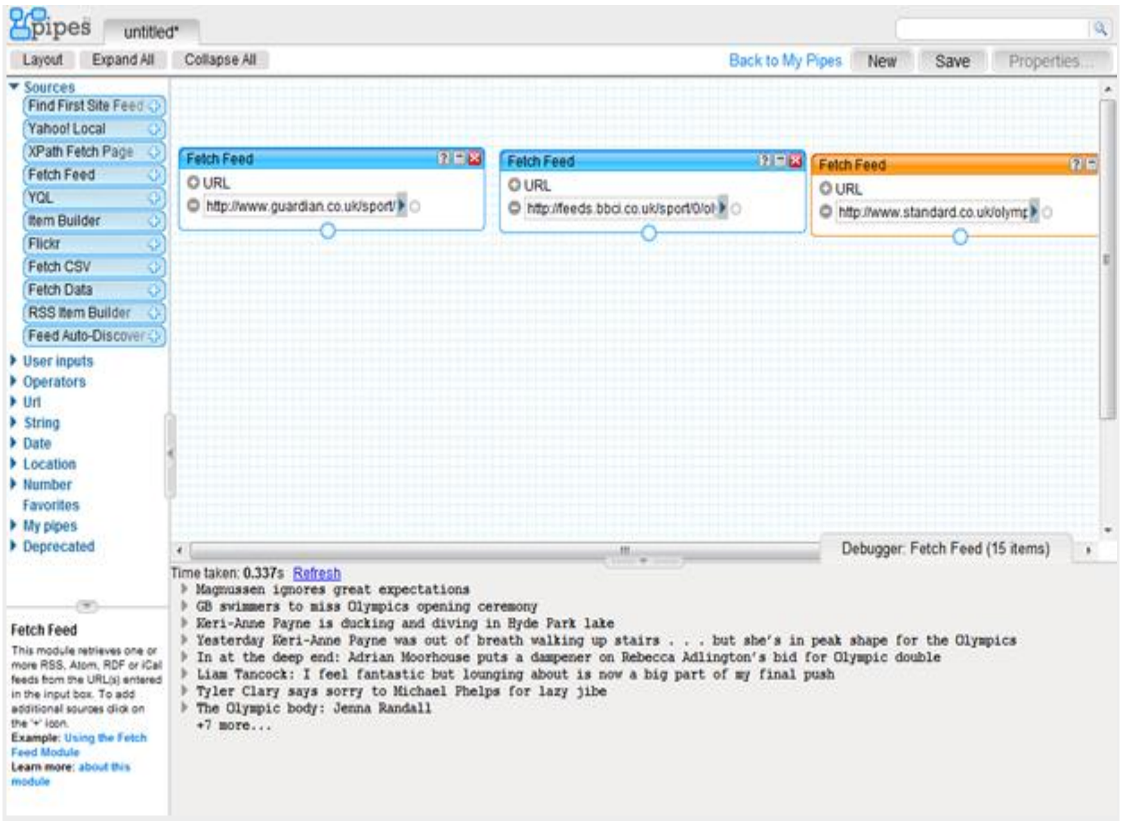
Ready to get started? Make sure you've got a Yahoo! account (something you likely already have if you've ever used Flickr). Then, fire up your preferred web browser and pay a visit to the Yahoo Pipes. Pipes is a tool that lets you take RSS feeds and mix them together, like pipes mixing two fluids together. Just sign in, and you'll be ready to get started.

You'll be presented with a blank workspace and this is where you will be creating your pipes in a visual, drag and drop based environment. To the left you'll see a list of **Sources** and these are what you will use to pull in data from other web sites. The pipe we're going to create is going to filter news from existing RSS feeds, so start by dragging a **Fetch Feed** module from the left on the workspace.



Yahoo Pipes provides you with a blank canvas on which to work on your creation

In the text field, enter the URL of an RSS feed you would like to work with and then repeat the process of adding a **Fetch Feed** module and a feed for as many feeds as you need. You can check that the feeds are working correctly by dragging the debugger pane up from the bottom of the screen; click on of the **Fetch Feed** modules and the output of the feed will be displayed here.



Fetch Feed

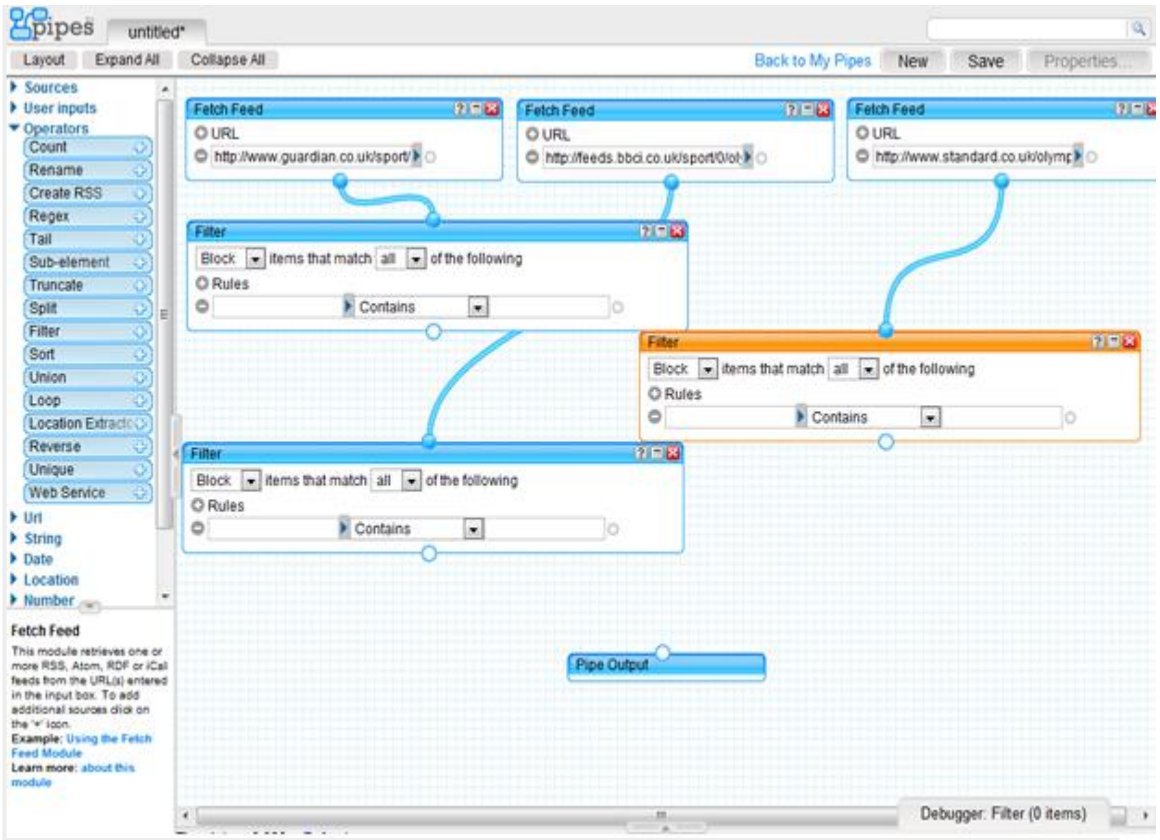
This module retrieves one or more RSS, Atom, RDF or iCal feeds from the URL(s) entered in the input box. To add additional sources click on the '+' icon.
Example: Using the Fetch Feed Module
Learn more: about this module

```
Time taken: 0.337s  Refresh
 ▸ Magnussen ignores great expectations
 ▸ GB swimmers to miss Olympics opening ceremony
 ▸ Keri-Anne Payne is ducking and diving in Hyde Park lake
 ▸ Yesterday Keri-Anne Payne was out of breath walking up stairs . . . but she's in peak shape for the Olympics
 ▸ In at the deep end: Adrian Moorhouse puts a dampener on Rebecca Adlington's bid for Olympic double
 ▸ Liam Tancock: I feel fantastic but lounging about is now a big part of my final push
 ▸ Tyler Clary says sorry to Michael Phelps for lazy jibe
 ▸ The Olympic body: Jenna Randall
   +7 more...
```

Use the debugger to check that your feeds are working as expected

Filtering Feeds

Now we're going to add a filter to each feed to control which news stories are displayed. Click the **Operators** link to the left to expand the group and then drag and drop three **Filter** modules to the workspace. You will now need to join each of the **Fetch Feed** modules to a **Filter**operator – just click on the white dot at the bottom of a **Fetch Feed** module and drag to a white dot at the top of a **Filter** box to establish a link.

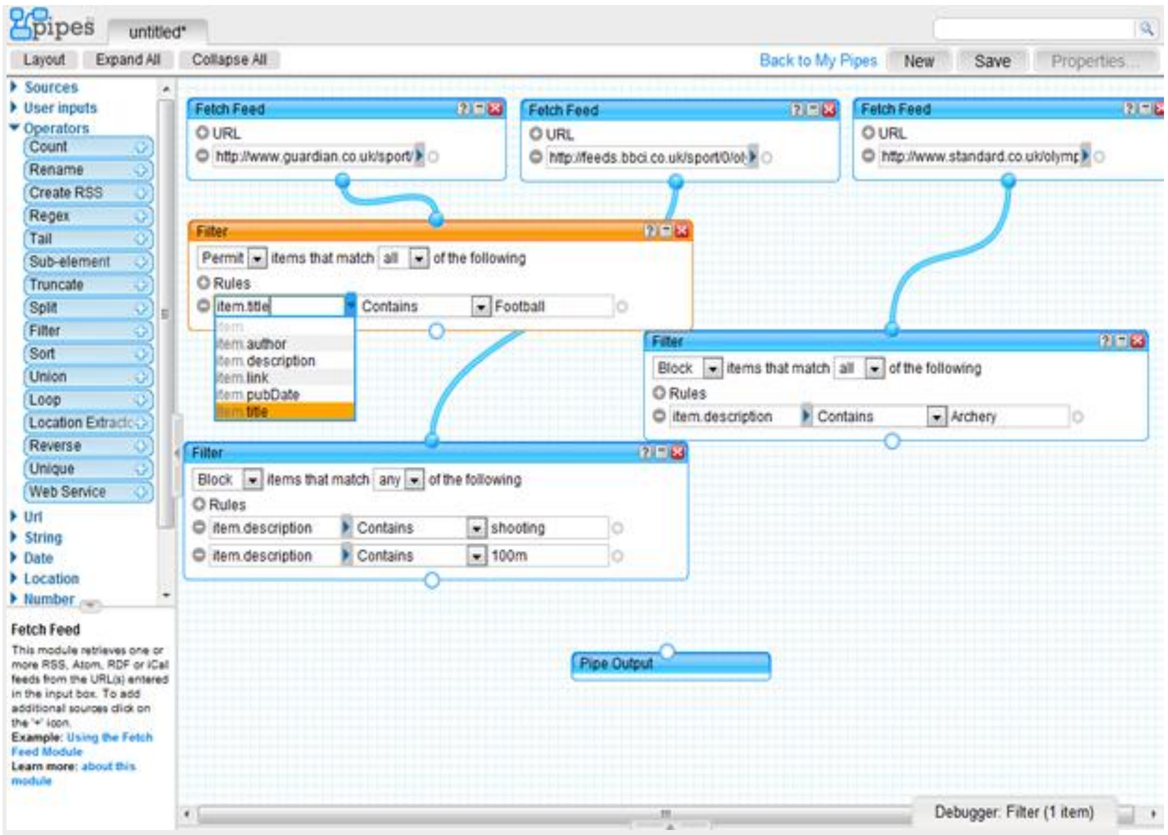


Fetch Feed

This module retrieves one or more RSS, Atom, RDF or iCal feeds from the URL(s) entered in the input box. To add additional sources click on the '+' icon.
Example: Using the Fetch Feed Module
Learn more: about this module

Modules need to be linked together in order for filters and other operators to take effect

Once links have been set up you can use the drop down menu in each of the **Filter** boxes to choose to block or permit different content and you can then specify keywords that will be looked for in titles, authors and other parts of feed items. As well as permitting and block content based on individual keywords, you can also configure multiple rules that must be matched before content of displayed or blocked.


Pipes
untitled*
Layout
Expand All
Collapse All
Back to My Pipes
New
Save
Properties...
Sources
User inputs
Operators
Count
Rename
Create RSS
Regex
Tail
Sub-element
Truncate
Split
Filter
Sort
Union
Loop
Location Extractor
Reverse
Unique
Web Service
Url
String
Date
Location
Number
Fetch Feed
URL
http://www.guardian.co.uk/sport/
http://feeds.bbci.co.uk/sport/0/ol
http://www.standard.co.uk/olymp
Filter
Permit items that match all of the following
Rules
item.title Contains Football
item.author
item.description
item.link
item.pubDate
item.title
Block items that match all of the following
item.description Contains Archery
Block items that match any of the following
item.description Contains shooting
item.description Contains 100m
Pipe Output
Debugger: Filter (1 item)

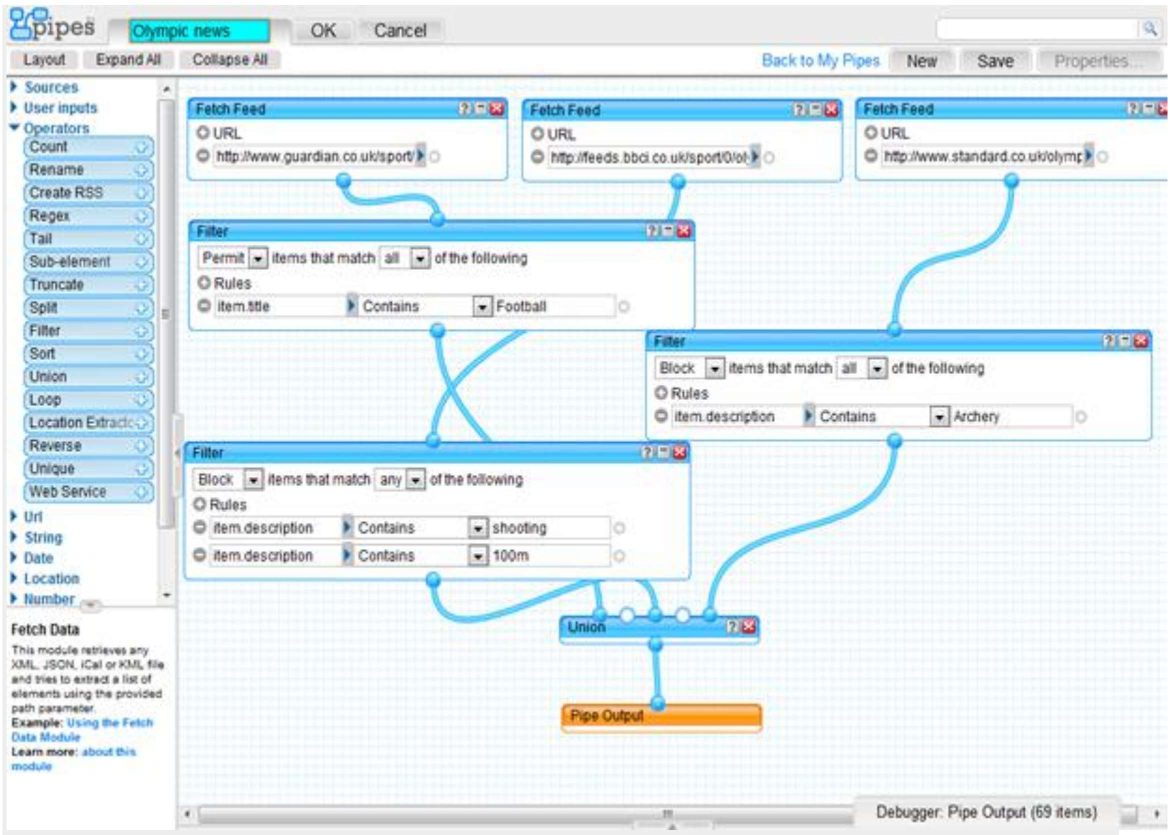
Fetch Feed

This module retrieves one or more RSS, Atom, RDF or iCal feeds from the URL(s) entered in the input box. To add additional sources click on the '+' icon.
Example: Using the Fetch Feed Module
Learn more: about this module

Filter modules can be used to tightly control which news items appear in your pipe feeds

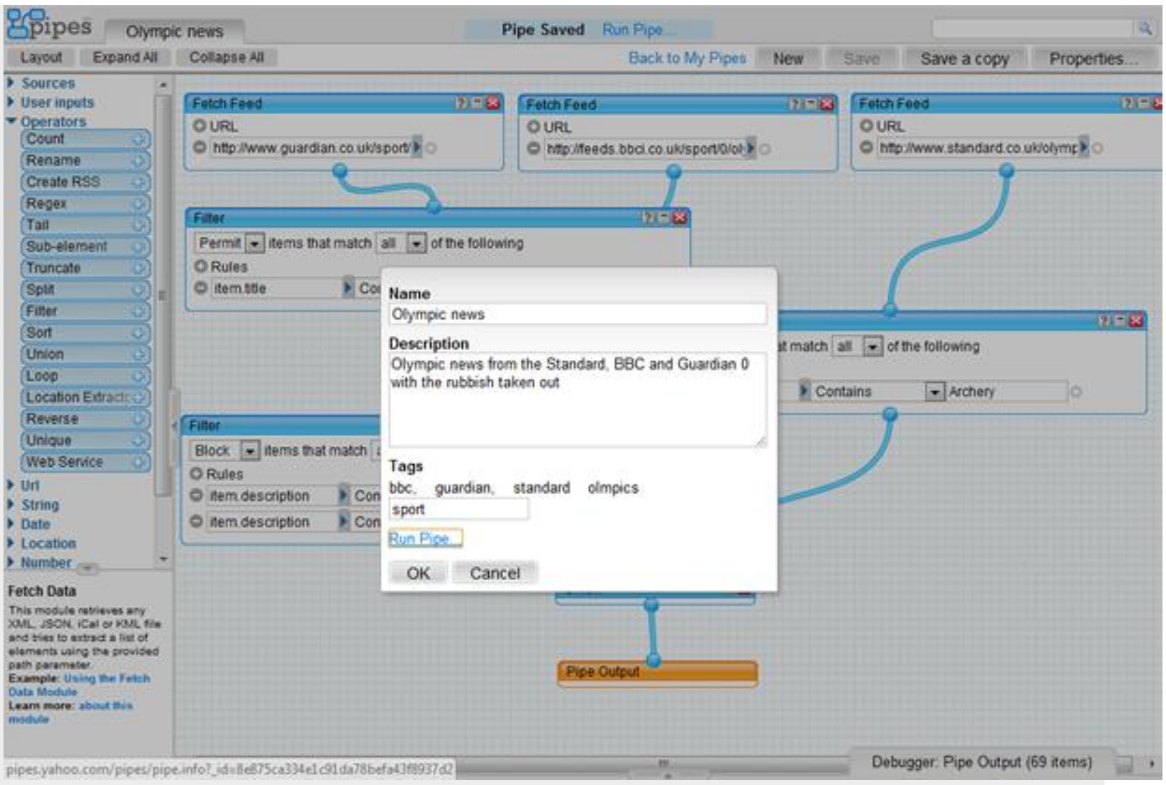
Unifying Filtered Feeds
Additional operations can be applied to feeds, but for the purposes of this guide, we’ll start to tie thing up. Back in the **Operators** section to the left, drag a **Union** module onto the workspace. You can then join each of the **Filter** modules to the nodes on the top of the Union module, before joining this to the **Pipe Output** module at the bottom of the workspace.

You should use the debugger to check that your filters are working correctly and if you’re happy with the output, click the untitled tab at the top of page and enter a suitable new name before clicking **OK**.



A Union module can be used to bring all of your filtered content together

Click the **Save** button to the upper right of the page and then click the Properties button. In the pop up window that appears you can enter a description for your pipe as well as a number of keywords to enable other people to search for and make use of it.



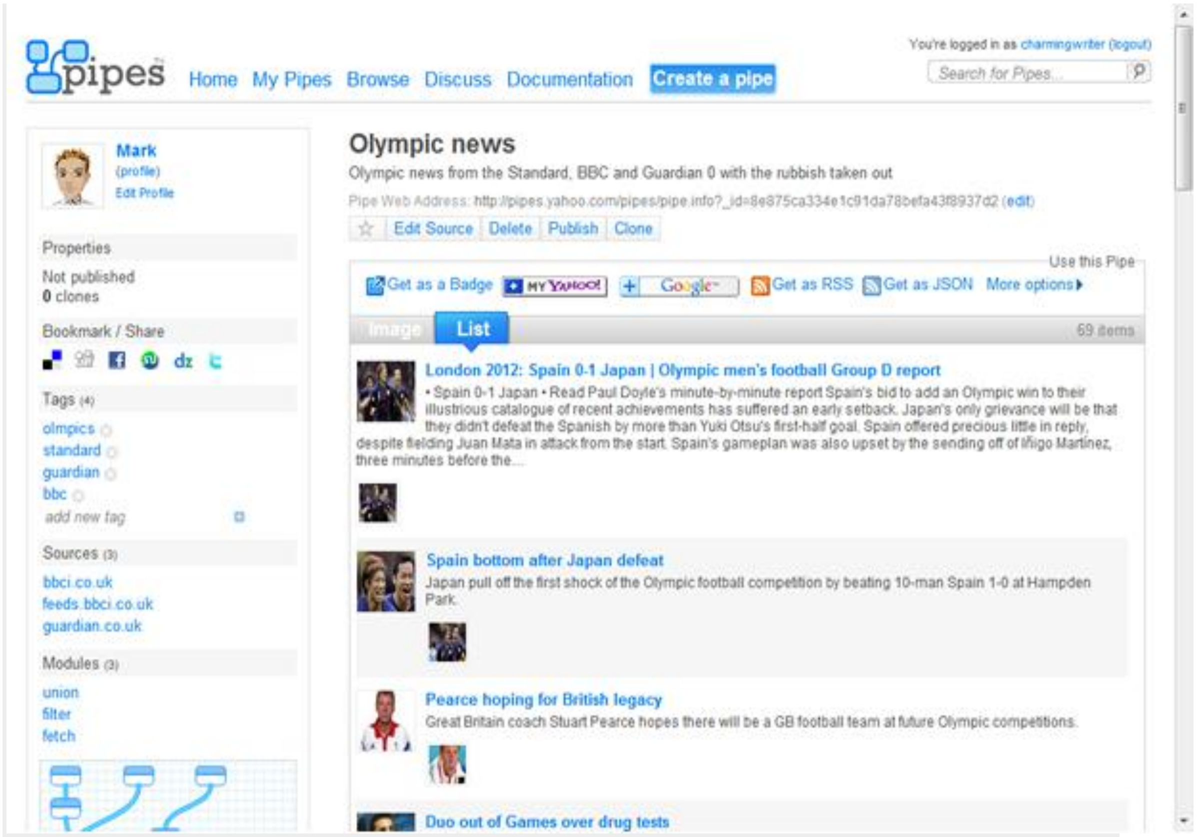
Name
Olympic news

Description
Olympic news from the Standard, BBC and Guardian 0 with the rubbish taken out

Tags
bbc, guardian, standard olmpics
sport

A tags and a description so that other people can track down the pipes you make.

To check how your pipe looks, click the **Run Pipe** link – you can also visit the **My Pipes**section of the web site and then click the pipe you are interested in. If you find that you need to make any changes or additions, just click the **Edit Source** link, but otherwise you can make use of the pipe as a feed by clicking the **Get as RSS** link



Olympic news

Olympic news from the Standard, BBC and Guardian 0 with the rubbish taken out

Pipe Web Address: http://pipes.yahoo.com/pipes/pipe.info?_id=8e875ca334e1c91da78befa43f8937d2 (edit)

London 2012: Spain 0-1 Japan | Olympic men's football Group D report

• Spain 0-1 Japan • Read Paul Doyle's minute-by-minute report Spain's bid to add an Olympic win to their illustrious catalogue of recent achievements has suffered an early setback. Japan's only grievance will be that they didn't defeat the Spanish by more than Yuki Otsu's first-half goal. Spain offered precious little in reply, despite fielding Juan Mata in attack from the start. Spain's gameplan was also upset by the sending off of Iñigo Martínez, three minutes before the...

Spain bottom after Japan defeat

Japan pull off the first shock of the Olympic football competition by beating 10-man Spain 1-0 at Hampden Park.

Pearce hoping for British legacy

Great Britain coach Stuart Pearce hopes there will be a GB football team at future Olympic competitions.

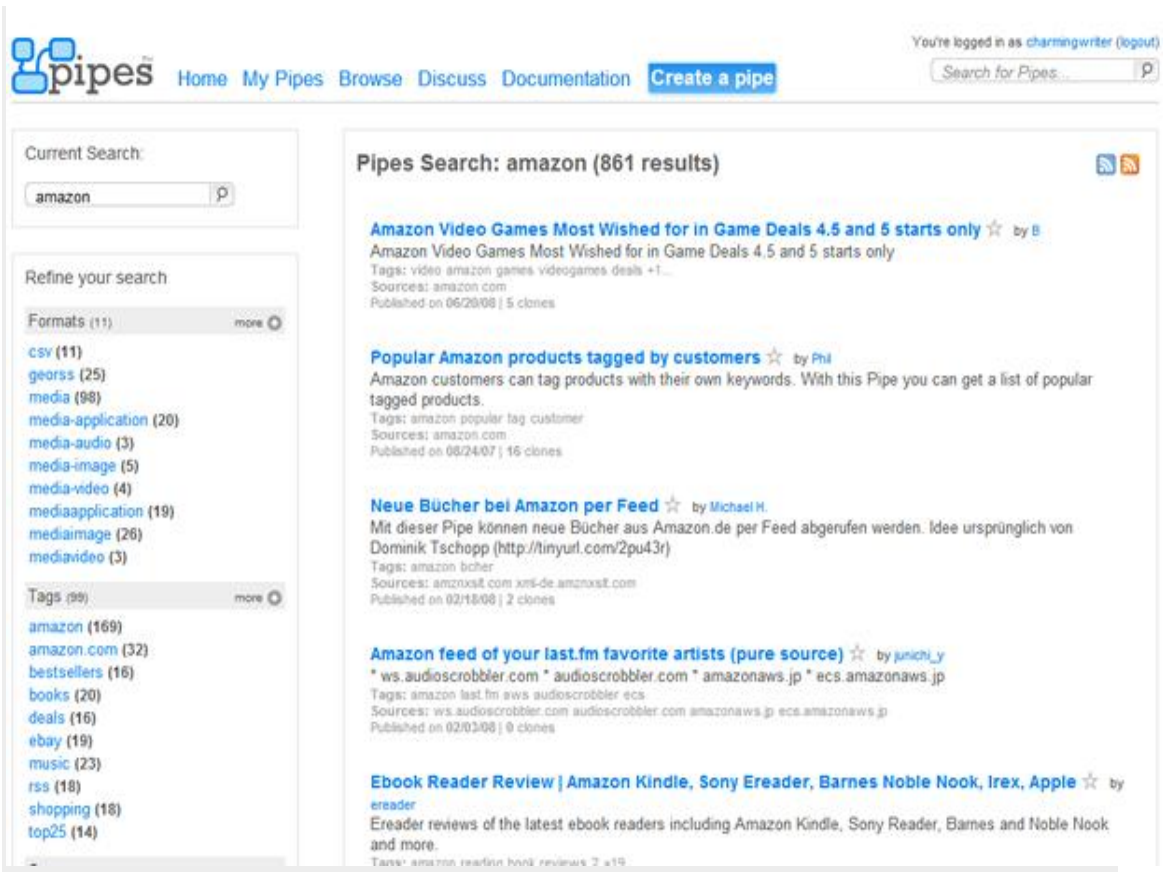
Duo out of Games over drug tests

The My Pipes section of Yahoo Pipes provides access to your creations

Browsing Premade Pipes

Creating your own pipes can be fun and rewarding but, depending on what you are looking for them to do, it can also be a complex and frustrating experience. Before you spend an age creating a complicated pipe, it is worth taking look through what other users have put together – even if you do not find something that precisely meets your needs, you may well stumble across something that could be adapted, or at least gives you an idea of how to achieve what you're looking to create.

Once you're logged into your account you can either use the search box to the upper right of the page to look for something matching certain criteria, or you can click the **Browse** button in the upper navigation bar to take a look through other people's creations.



Current Search:

amazon

Refine your search

Formats (11)	
csv	(11)
georss	(25)
media	(98)
media-application	(20)
media-audio	(3)
media-image	(5)
media-video	(4)
mediaapplication	(19)
mediaimage	(26)
mediavideo	(3)

Tags (99)	
amazon	(169)
amazon.com	(32)
bestsellers	(16)
books	(20)
deals	(16)
ebay	(19)
music	(23)
rss	(18)
shopping	(18)
top25	(14)

Pipes Search: amazon (861 results)

Amazon Video Games Most Wished for in Game Deals 4.5 and 5 starts only ☆ by B
Amazon Video Games Most Wished for in Game Deals 4.5 and 5 starts only
Tags: video amazon games videogames deals +1...
Sources: amazon.com
Published on 06/20/08 | 5 clones

Popular Amazon products tagged by customers ☆ by Phil
Amazon customers can tag products with their own keywords. With this Pipe you can get a list of popular tagged products.
Tags: amazon popular tag customer
Sources: amazon.com
Published on 08/24/07 | 16 clones

Neue Bücher bei Amazon per Feed ☆ by Michael H.
Mit dieser Pipe können neue Bücher aus Amazon.de per Feed abgerufen werden. Idee ursprünglich von Dominik Tschopp (http://tinyurl.com/2pu43r)
Tags: amazon bcher
Sources: amznxslt.com xml-de.amznxslt.com
Published on 02/18/08 | 2 clones

Amazon feed of your last.fm favorite artists (pure source) ☆ by junichi_y
* ws.audioscrobbler.com * audioscrobbler.com * amazonaws.jp * ecs.amazonaws.jp
Tags: amazon last.fm aws audioscrobbler ecs
Sources: ws.audioscrobbler.com audioscrobbler.com amazonaws.jp ecs.amazonaws.jp
Published on 02/03/08 | 0 clones

Ebook Reader Review | Amazon Kindle, Sony Ereader, Barnes Noble Nook, Irex, Apple ☆ by ereader
Ereader reviews of the latest ebook readers including Amazon Kindle, Sony Reader, Barnes and Noble Nook and more.
Tags: amazon reading book reviews 2 +19...

There are a wealth of readymade pipes that you can use as-is or adapt to suit your needs

Yahoo Pipes is an extremely versatile service that can be twisted and tweaked to work in a huge variety of ways. If you are a programmer, there is great scope for getting your hands dirty with interactive pipes that enable you to get more form the web, but even the most simple creations are very useful. But even if you've never coded in your life, Yahoo Pipes makes it easy to get the data you want from the web, mixed up and sorted just the way you want it.

Have you ever used Yahoo Pipes, or is this your first time to try it? It'd be fun to hear some of the ways our readers are using Yahoo Pipes to make feeds that work best for them.

PRE LAB VIVA QUESTIONS?

1. What are good alternatives to yahoo pipe?
2. How to delete RSS feeds from yahoo pipes?
3. What kind of things can you build with yahoo pipes?
4. How to salvage logic and data from yahoo pipes?

POST LAB VIVA QUESTIONS?

1. Which source model in yahoo pipe is most suitable for feeding data to a yahoo filter?
2. How can I setup multiple tabs/RSS Feeds in blogger?
3. Is it possible to create keywords based RSS feeds?
4. How can I get a RSS Feed for a custom page?