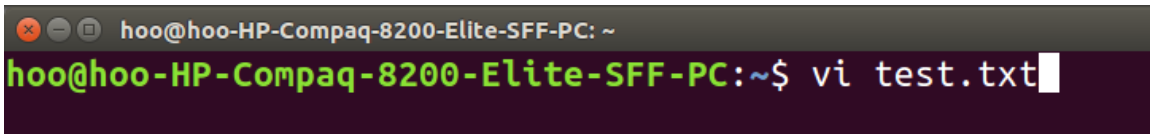
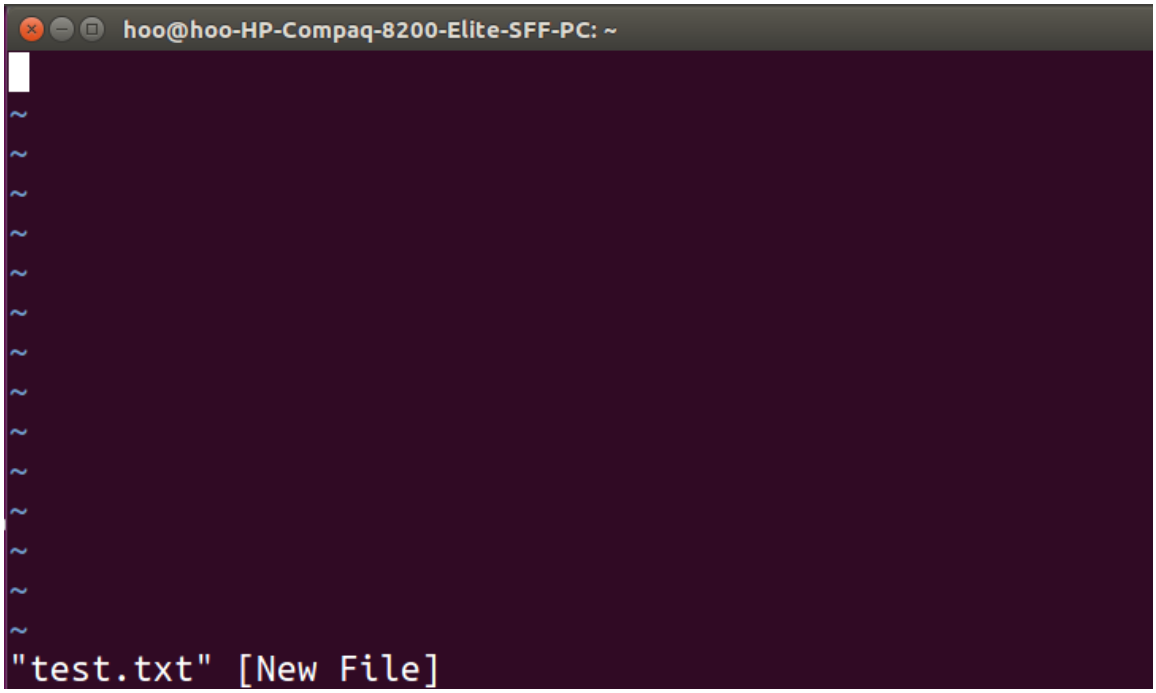
Lab 1 and 2 summary

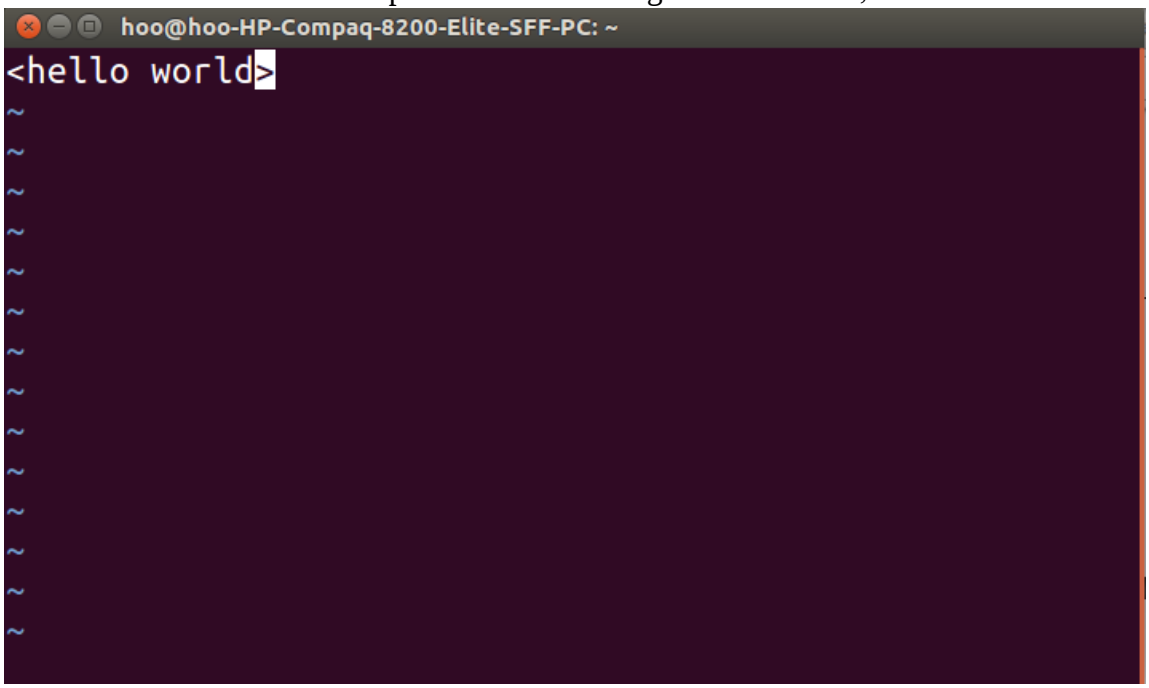
Lab 1:

- How to edit files?
  - ‘vi’

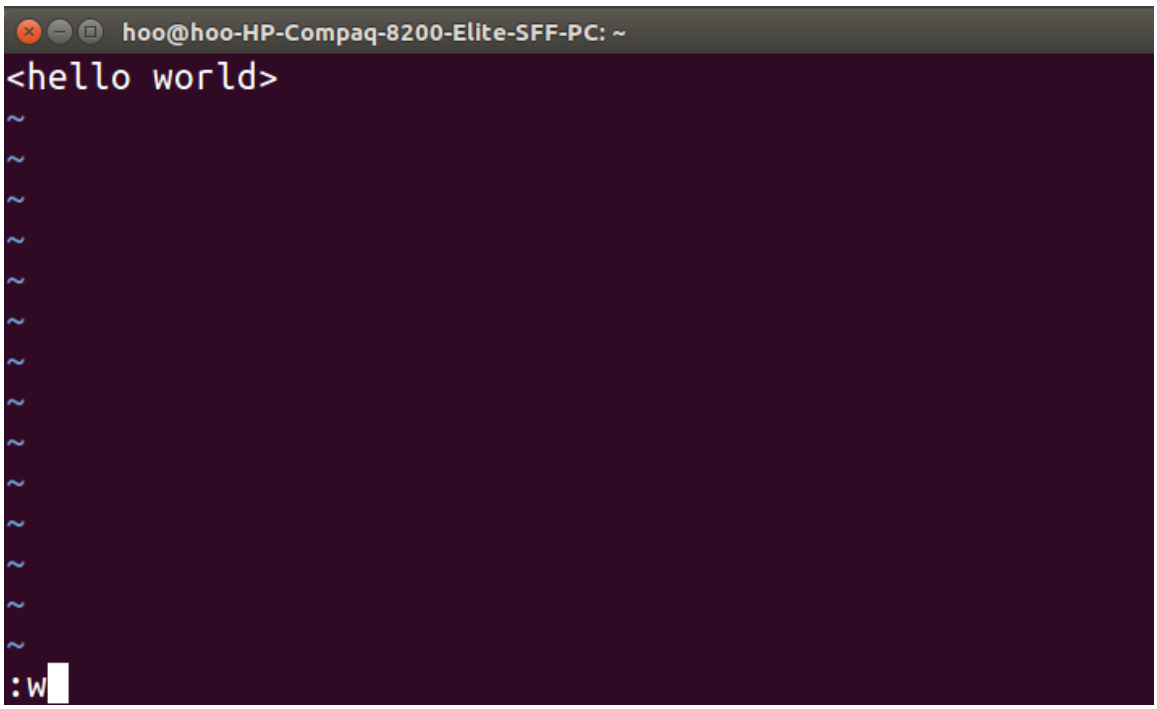




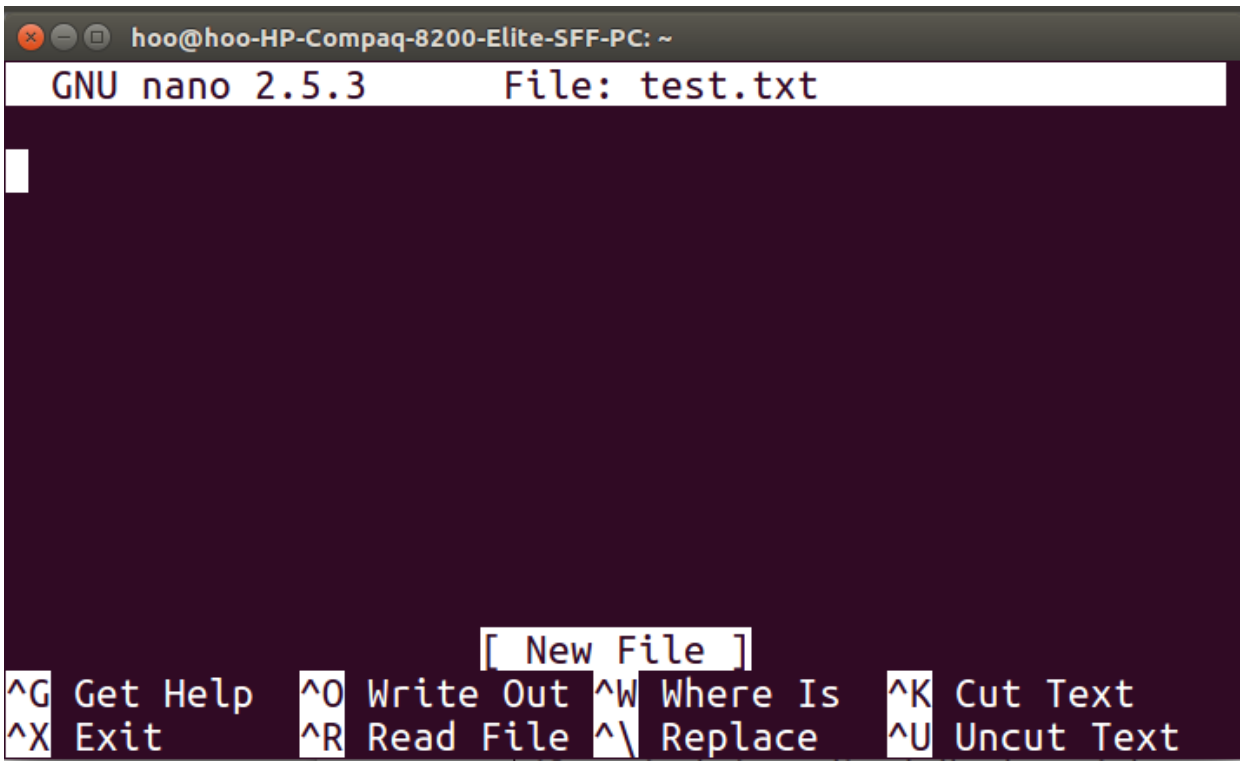
- Type ‘i’ to edit, type “<Hello world>” (sample edit), then ‘esc’ to exit the edit.
  - Don’t use ‘backspace’ to delete wrong words! Instead, use ‘delete’



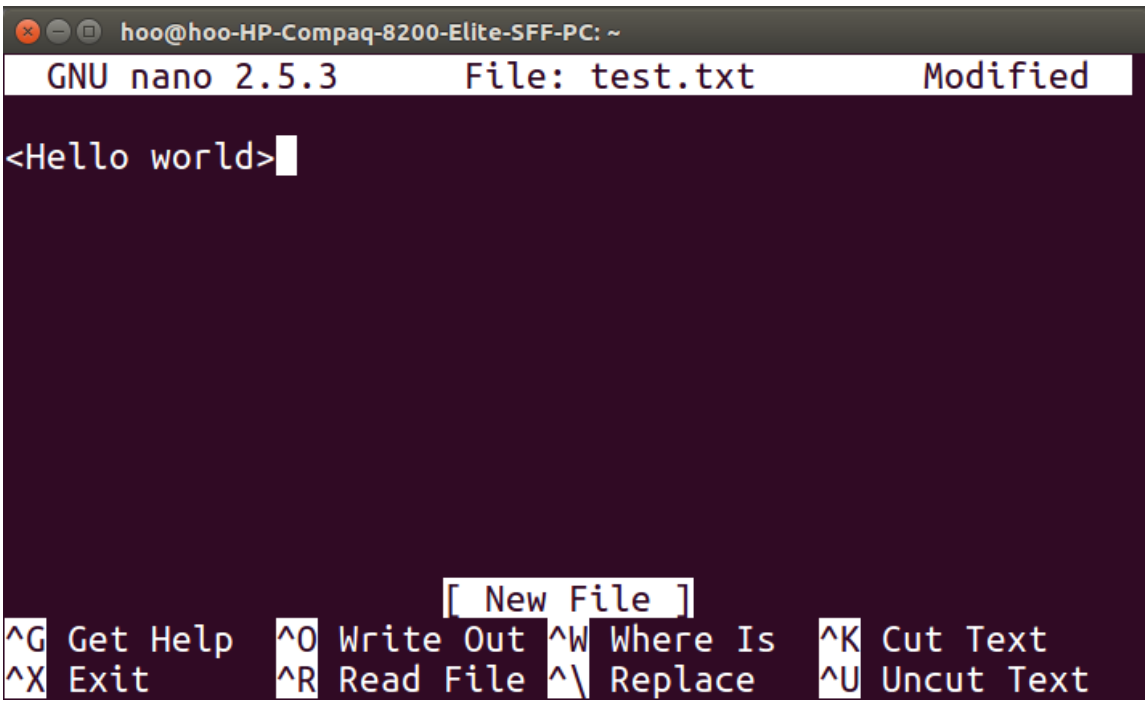
- Type ‘:w’ + ENTER to write, and type ‘:x’ + ENTER to exit the program.
  - The command should show at the bottom left of the screen.



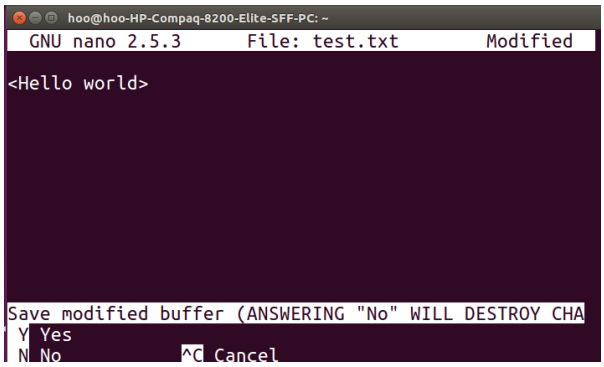
- ‘nano’

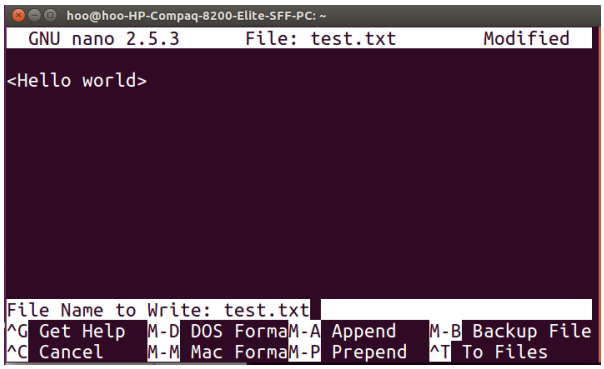


- type ‘<Hello world>’ (sample edit)



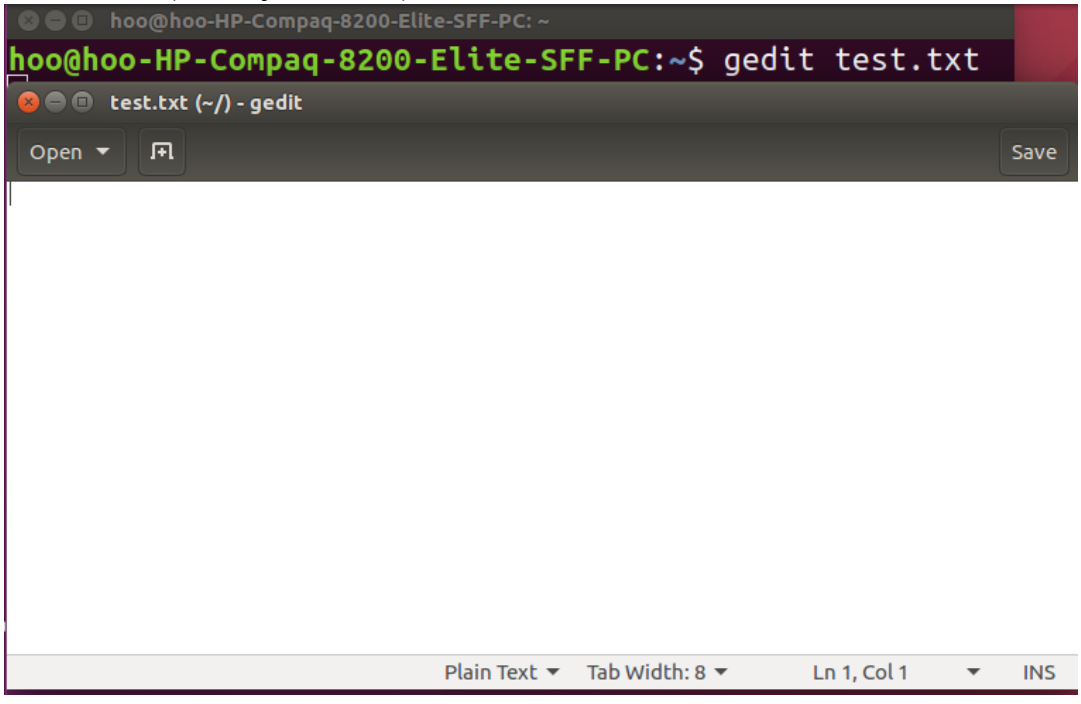
- type CTRL + ‘x’, then ‘y’ to save.





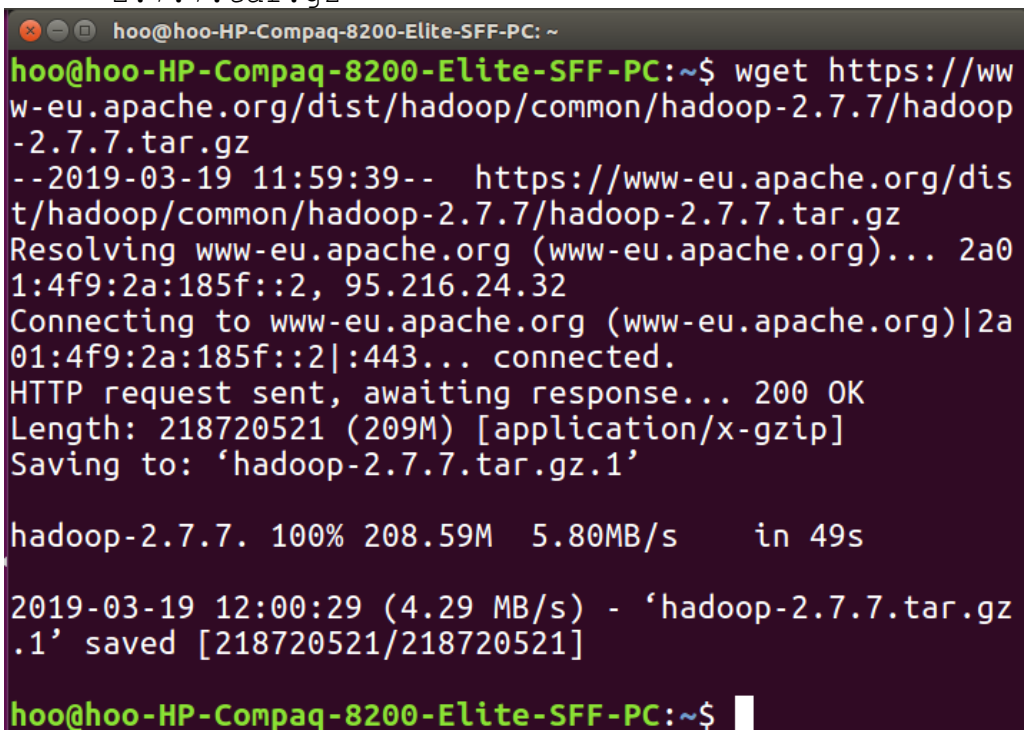
- Then press ENTER to exit.

- ‘`gedit`’ (friendly UI editor)



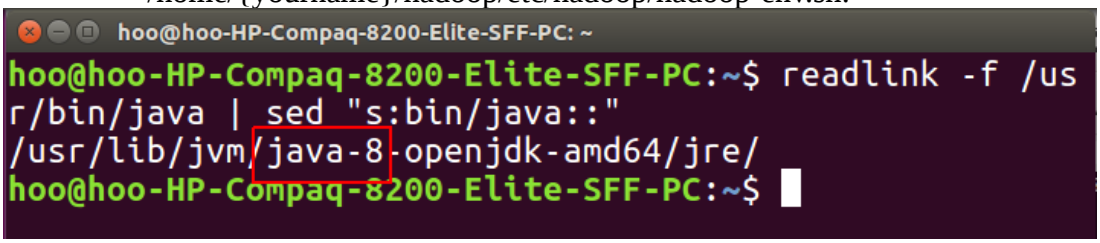
- Installing hadoop in Ubuntu
  - Pre-requisites:
    - `sudo apt-get update`
      - to update latest distributions of software
    - `sudo apt-get install openssh-server`
    - `sudo apt-get install openjdk-8-jdk`
      - hadoop requires java 8 to run. If you have java 11 installed, please install another copy of Java 8. if you run 'java -version', you should see similar output as below. It is OK it the version number is not 191.

      - If you have java version show as '10.0.2', please install Java 8 as mentioned above.
    - Download and extract hadoop
      - `wget https://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz`

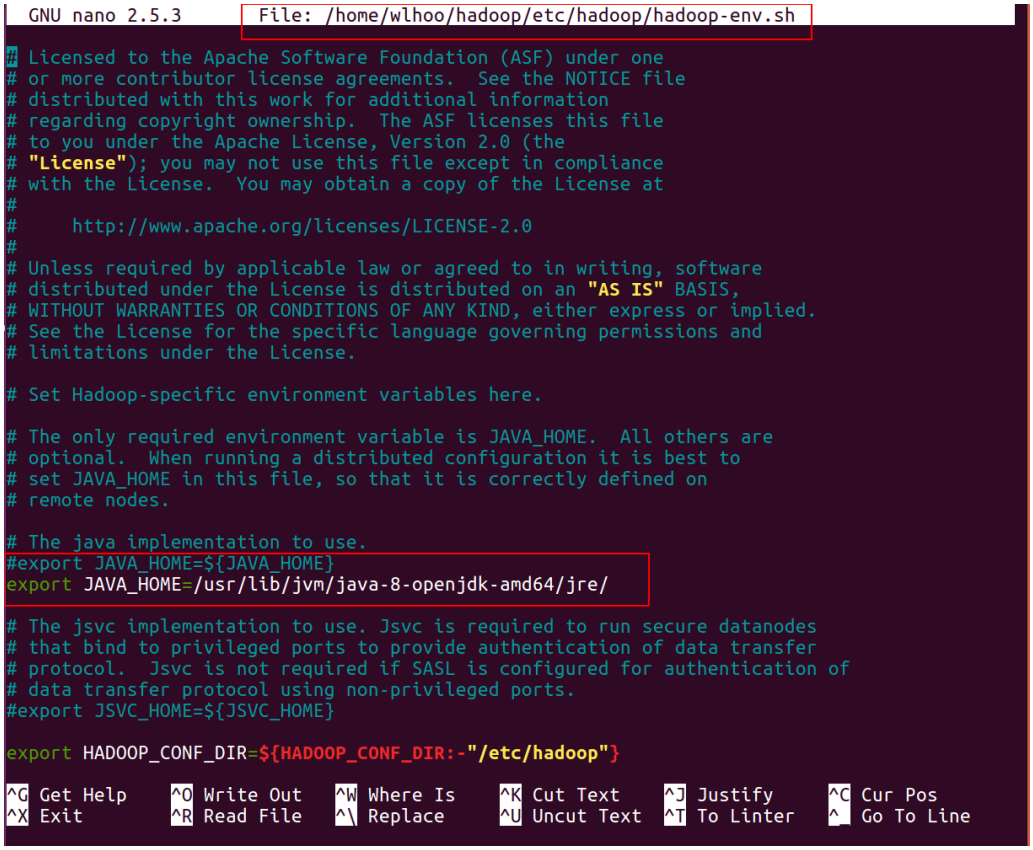


- `tar -xzf hadoop-2.7.7.tar.gz`
  - unzip
- move to selected directory
  - in here, it is recommended to move to your own directory for easy code trace. Please move to '/home/{yourname}/hadoop':
    - `sudo mkdir /home/{yourname}`
    - `sudo mv hadoop-2.7.7 /home/{yourname}/hadoop/`
      - please make sure you type the last slashes. Else, the system will move 'hadoop-2.7.7' under a folder named 'hadoop', rather than rename 'hadoop-2.7.7' to 'hadoop'.

- Set JAVA_HOME for hadoop
  - in order to avoid clashes with other program, it is not recommended to set $JAVA_HOME in Ubuntu variables. Instead, please run '`readlink -f /usr/bin/java | sed "s:bin/java::"`' and set JAVA_HOME in /home/{yourname}/hadoop/etc/hadoop/hadoop-env.sh:



  - if your java version show java-11, please make sure you have installed openjdk8, and manually change from java-11 to java-8. You can also go to the /usr/lib/jvm folder to double check java-8-openjdk-amd64 to make sure it exist.



- Run hadoop
  - Export path for hadoop to allow easy hadoop function access
    - `export PATH=$PATH:/home/{yourname}/hadoop/bin`
      - for hadoop and hdfs functions
    - `export PATH=$PATH:/home/{yourname}/hadoop/sbin`
      - for hadoop scripts execution e.g. start-dfs.sh

```
 GNU nano 2.5.3              File: /home/hoo/.bashrc

alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$$

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# 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

# for hadoop configuration
export PATH=$PATH:/home/wlhoo/hadoop/bin
export PATH=$PATH:/home/wlhoo/hadoop/sbin

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line
```

      - You can copy all these three lines to the end of the ~/.bashrc file to prevent executing these lines again after your system reboot.
  - Type '`hadoop`' in the terminal:

```
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$ hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
  CLASSNAME            run the class named CLASSNAME
 or
  where COMMAND is one of:
  fs                   run a generic filesystem user client
  version              print the version
  jar <jar>            run a jar file
                       note: please use "yarn jar" to launch
                             YARN applications, not this command.
  checknative [-a|-h]  check native hadoop and compression libraries availability
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath            prints the class path needed to get the
  credential           interact with credential providers
                       Hadoop jar and the required libraries
  daemonlog            get/set the log level for each daemon
  trace                view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$
```

- Test installation with sample mapreduce code:
  - `mkdir ~/input`
  - `cp /home/{yourname}/hadoop/etc/hadoop/*.xml ~/input`
  - `hadoop jar /home/{yourname}/hadoop/share/hadoop/mapreduce/hadoop-`

`mapreduce-examples-2.7.7.jar` to see the functions you can use in this compiled java program.

```
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$ hadoop jar /home/wlhoo/hadoop/share/hadoop/mapreduce/
hadoop-mapreduce-examples-2.7.7.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the inp
ut files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of th
e words in the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input fi
les.
  wordmedian: A map/reduce program that counts the median length of the words in the input f
iles.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the leng
th of the words in the input files.
```

- Test ‘grep’ function:
  - `hadoop jar /home/{yourname}/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar grep ~/input ~/grep_example ‘principal[.]’`

```
19/03/19 12:52:22 INFO mapreduce.Job:  map 100% reduce 100%
19/03/19 12:52:22 INFO mapreduce.Job: Job job_local1203238571_0002 completed successfully
19/03/19 12:52:22 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=1255192
                FILE: Number of bytes written=2355260
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=2
                Map output records=2
                Map output bytes=37
                Map output materialized bytes=47
                Input split bytes=112
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=47
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1054867456
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=151
        File Output Format Counters
                Bytes Written=37
```

- If you tried to copy the and paste the code to the console, please make sure you re-type open and close quote ‘’ IN the terminal. Else, the terminal won’t recognize these quotes, and return null result.

```
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$ cat ~/grep_example/*
6       principal
1       principal.
```

    - `cat ~/grep_example/*`
- Test ‘wordcount’ function:
    - `hadoop jar /home/{yourname}/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar wordcount ~/input ~/grep_example2`
    - `cat ~/grep_example2/*`

```
to      53
traffic.        1
type    1
type,   1
type="text/xsl" 4
u:%user:%user   1
under   28
updating        1
use     10
used    22
used.   1
user    40
user.   2
user?   1
users   21
users,wheel".   18
uses    2
using   3
value   19
values  1
version 1
version="1.0"   5
version="1.0"?> 3
via     1
when    4
where   1
which   5
while   1
who     2
will    7
window  1
window, 1
with    28
within  1
without 1
work    1
writing,        9
you     10
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$
```

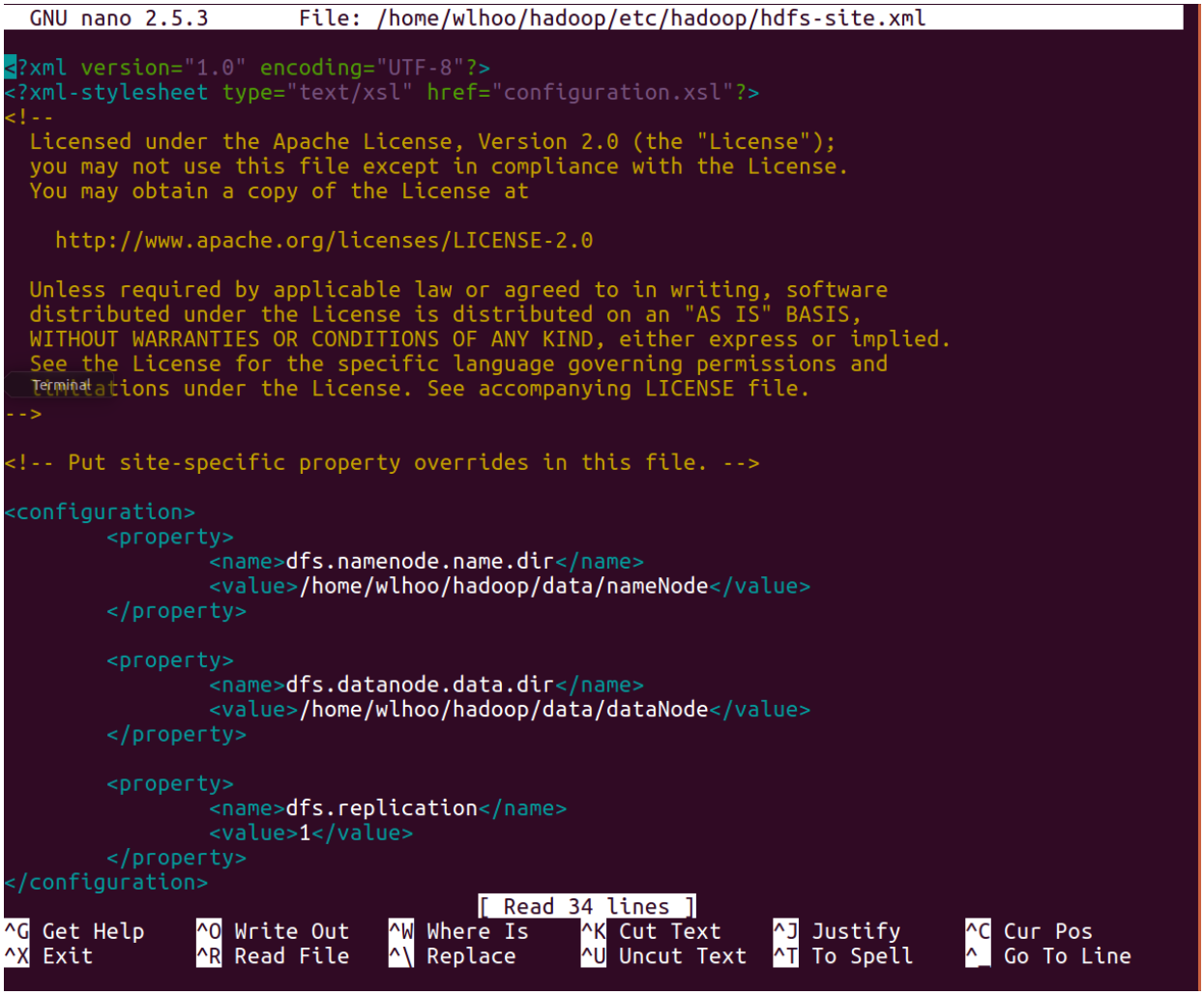
Lab 2:

- Update hdfs-site.xml in /home/{yourname}/hadoop/etc/hadoop folder:

```
<configuration>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/home/wlhoo/hadoop/data/nameNode</value>
    </property>

    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/home/wlhoo/hadoop/data/dataNode</value>
    </property>

    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```



- Update core-site.xml in /home/{yourname}/hadoop/etc/hadoop:

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

    - You can change ‘localhost’ to your PC’s IP address.
    - Alternatively, you can put ‘file:///’ instead to indicate local file system.
- change the name of mapred-site.xml.template in /home/{yourname}/hadoop/etc/hadoop to mapred-site.xml. Then, update mapred-site.xml:

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

    <property>
        <name>yarn.app.mapreduce.am.resource.mb</name>
        <value>512</value>
    </property>

    <property>
        <name>mapreduce.map.memory.mb</name>
        <value>256</value>
    </property>

    <property>
        <name>mapreduce.reduce.memory.mb</name>
        <value>256</value>
    </property>
</configuration>
```

- Update yarn-site.xml in /home/{yourname}/hadoop/etc/hadoop:

```
<configuration>
    <property>
        <name>yarn.acl.enable</name>
        <value>0</value>
    </property>

    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>localhost</value>
    </property>

    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>

    <property>
        <name>yarn.nodemanager.resources.memory-mb</name>
        <value>1536</value>
    </property>

    <property>
        <name>yarn.scheduler.maximum -allocation-mb</name>
        <value>1536</value>
    </property>

    <property>
        <name>yarn-scheduler.minimum-allocation-mb</name>
        <value>128</value>
    </property>

    <property>
        <name>yarn.nodemanager.vmem-check-enabled</name>
        <value>false</value>
    </property>
</configuration>
```

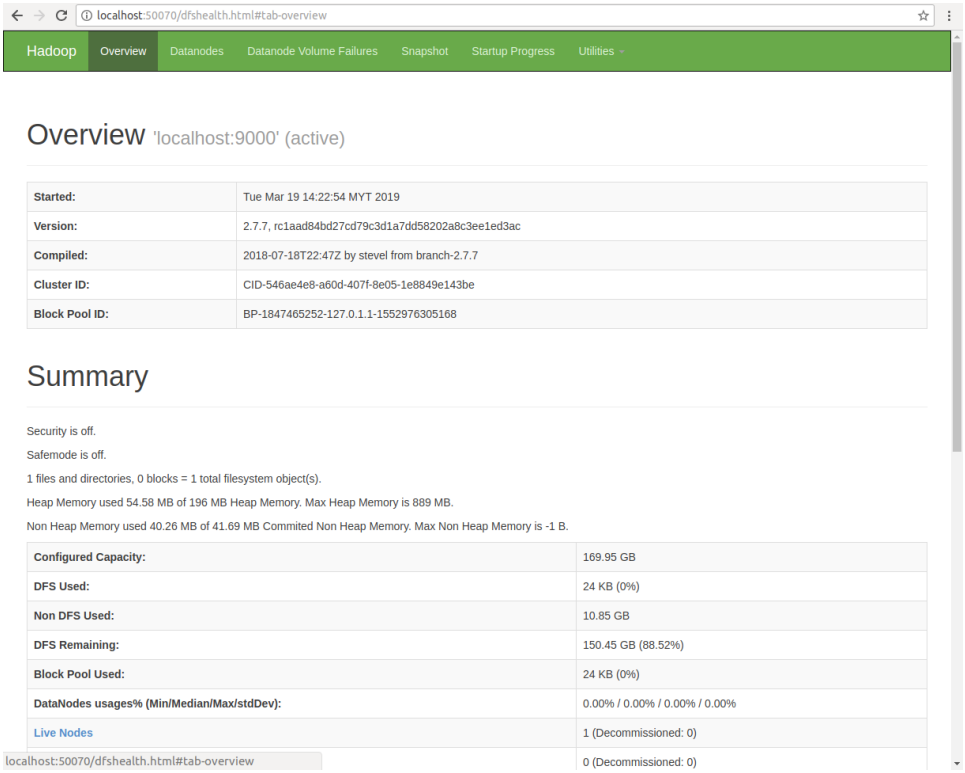
- run `hdfs namenode -format`

```
19/03/19 14:18:25 INFO namenode.FSDirectory: XAttrs enabled? true
19/03/19 14:18:25 INFO namenode.FSDirectory: Maximum size of an xattr: 16384
19/03/19 14:18:25 INFO namenode.NameNode: Caching file names occuring more than 10 times
19/03/19 14:18:25 INFO util.GSet: Computing capacity for map cachedBlocks
19/03/19 14:18:25 INFO util.GSet: VM type       = 64-bit
19/03/19 14:18:25 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
19/03/19 14:18:25 INFO util.GSet: capacity      = 2^18 = 262144 entries
19/03/19 14:18:25 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.999000
0128746033
19/03/19 14:18:25 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
19/03/19 14:18:25 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
19/03/19 14:18:25 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets =
 10
19/03/19 14:18:25 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
19/03/19 14:18:25 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,
5,25
19/03/19 14:18:25 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
19/03/19 14:18:25 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and re
try cache entry expiry time is 600000 millis
19/03/19 14:18:25 INFO util.GSet: Computing capacity for map NameNodeRetryCache
19/03/19 14:18:25 INFO util.GSet: VM type       = 64-bit
19/03/19 14:18:25 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
19/03/19 14:18:25 INFO util.GSet: capacity      = 2^15 = 32768 entries
19/03/19 14:18:25 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1847465252-127.0.1.1-
1552976305168
19/03/19 14:18:25 INFO common.Storage: Storage directory /home/wlhoo/hadoop/data/nameNode ha
s been successfully formatted.
19/03/19 14:18:25 INFO namenode.FSImageFormatProtobuf: Saving image file /home/wlhoo/hadoop/
data/nameNode/current/fsimage.ckpt_0000000000000000000 using no compression
19/03/19 14:18:25 INFO namenode.FSImageFormatProtobuf: Image file /home/wlhoo/hadoop/data/na
meNode/current/fsimage.ckpt_0000000000000000000 of size 320 bytes saved in 0 seconds.
19/03/19 14:18:25 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txi
d >= 0
19/03/19 14:18:25 INFO util.ExitUtil: Exiting with status 0
19/03/19 14:18:25 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at hoo-HP-Compaq-8200-Elite-SFF-PC/127.0.1.1
************************************************************/
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$
```

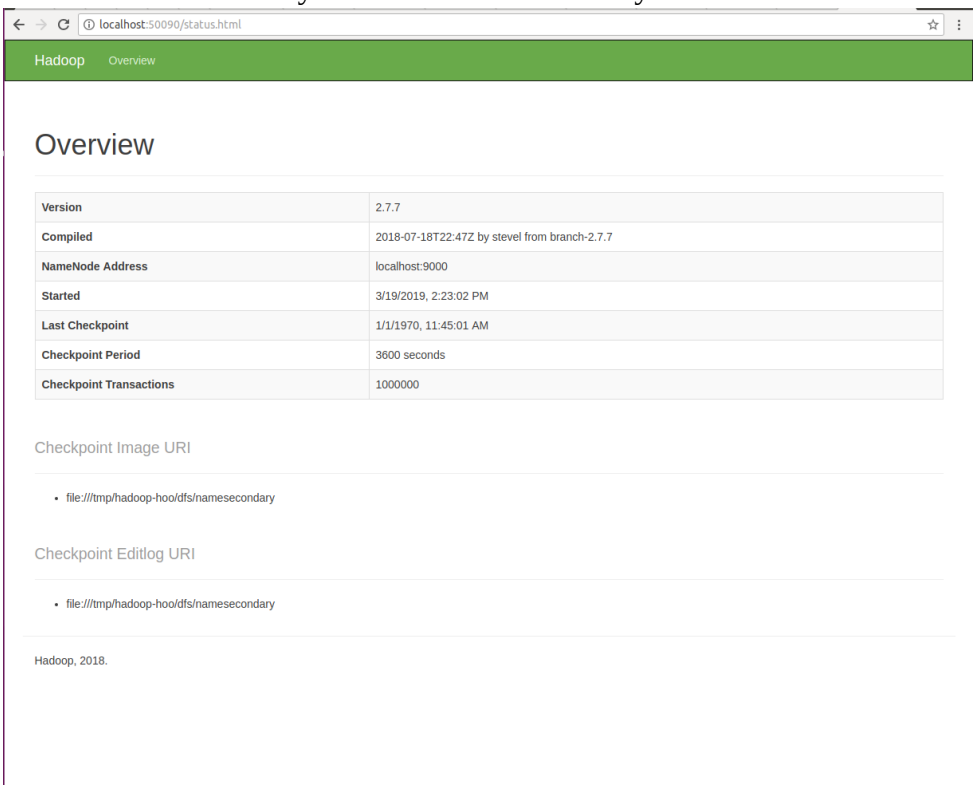
- run `start-all.sh` (or run start-dfs.sh and start-yarn.sh separately)

```
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/wlhoo/hadoop/logs/hadoop-hoo-namenode-hoo-HP-
Compaq-8200-Elite-SFF-PC.out
localhost: starting datanode, logging to /home/wlhoo/hadoop/logs/hadoop-hoo-datanode-hoo-HP-
Compaq-8200-Elite-SFF-PC.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/wlhoo/hadoop/logs/hadoop-hoo-secondary
namenode-hoo-HP-Compaq-8200-Elite-SFF-PC.out
starting yarn daemons
starting resourcemanager, logging to /home/wlhoo/hadoop/logs/yarn-hoo-resourcemanager-hoo-HP
-Compaq-8200-Elite-SFF-PC.out
localhost: starting nodemanager, logging to /home/wlhoo/hadoop/logs/yarn-hoo-nodemanager-hoo
-HP-Compaq-8200-Elite-SFF-PC.out
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$
```

- browse localhost:50070 in your browser for namenode:



- browser localhost:50090 in your browser for secondary namenode:



- After these settings, the same mapreduce code (`hadoop jar /home/{yourname}/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar grep ~/input ~/grep_example 'principal[.]'`) will not work anymore, because the default path for hadoop is now point to hdfs file system:

```
org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input path does not exist: hdfs
://localhost:9000/home/hoo/input
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(Fi
leInputFormat.java:323)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.
java:265)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getSplits(FileInputFormat.j
ava:387)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:301)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:318)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:196)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1290)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1287)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:17
62)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1287)
        at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1308)
        at org.apache.hadoop.examples.Grep.run(Grep.java:78)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
        at org.apache.hadoop.examples.Grep.main(Grep.java:103)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java
:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(ProgramDriver.java
:71)
        at org.apache.hadoop.util.ProgramDriver.run(ProgramDriver.java:144)
        at org.apache.hadoop.examples.ExampleDriver.main(ExampleDriver.java:74)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java
:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:226)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:141)
hoo@hoo-HP-Compaq-8200-Elite-SFF-PC:~$
```

  - to solve this, whenever local filesystem is pointed (except using -put and -get), you will need to add 'file:///' as prefix and cannot use abbreviation of ~ anymore, for example: '~/input' need to change to 'file:///home/hoo/input'. For output, you will need to indicate full path e.g. '/user/hdfs/grep-example':
    - `hadoop jar /home/{yourname}/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar grep file:///home/hoo/input /user/hdfs/grep-example 'principal[.]'`
      - you will need to make sure '/user/hdfs' path is exist as well