Assignment 2

BIG DATA LAB

Bishwajit Kumar Poddar

Msc MI - 211110

Task performed

1. Installation of Ubuntu
2. Basic commands in Ubuntu
3. Hadoop Installation
4. Word count program
5. Mong Db Installation
6. Basic queries on MongoDB
7. Pig Installation
8. Basic queries on Pig
9. Hbase installation and Basic queries
10. Pyspark installation and queries.

Installation of Ubunutu

To install ubuntu we have to get ubuntu iso file. To download the iso image which is nothing but a disk image kindly refer to the website

```
https://ubuntu.com/
```

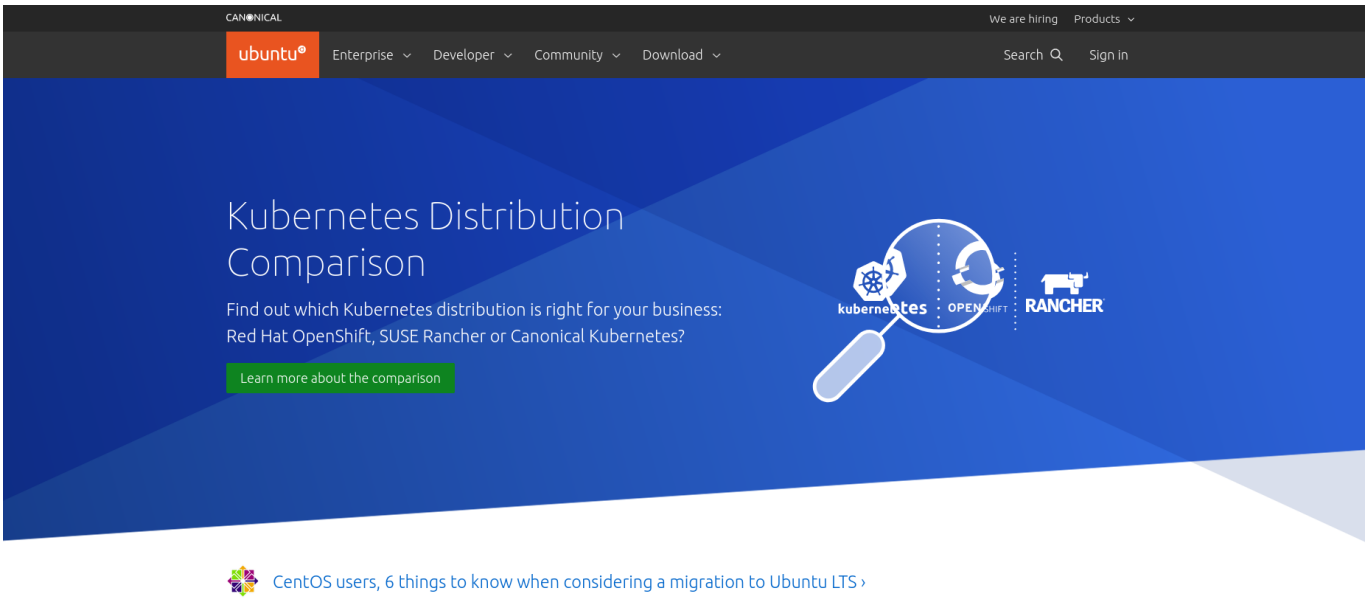

Figure : ubuntu website

1. Download the ISO file
2. Use software like rufus(for windows) / startup disk creator (Linux) `[only needed if installing on current system virtualization doesn't need this step]`
3. For installation in Virtualbox or VmWare use iso file as it is.
4. Mount the iso to the CD drive `for virtualbox` or pendrive `for actual system`
5. Then follow the steps shown bellow

> For this installation guide I have used ubuntu 20.04 Destop version. There might newer version available at the time of this. Also there is one server version which is CLI based system which can also be used for this kind of works.

1. Run the ISO

`might see this kind of screen`

Figure : Booting from the iso

- Wait for the checking system file to complete.
- You can cancel it by using `CTRL + C` if you care confident enough that ISO image you have used is not corrupted any how.

`Then youwill get this boot screen`

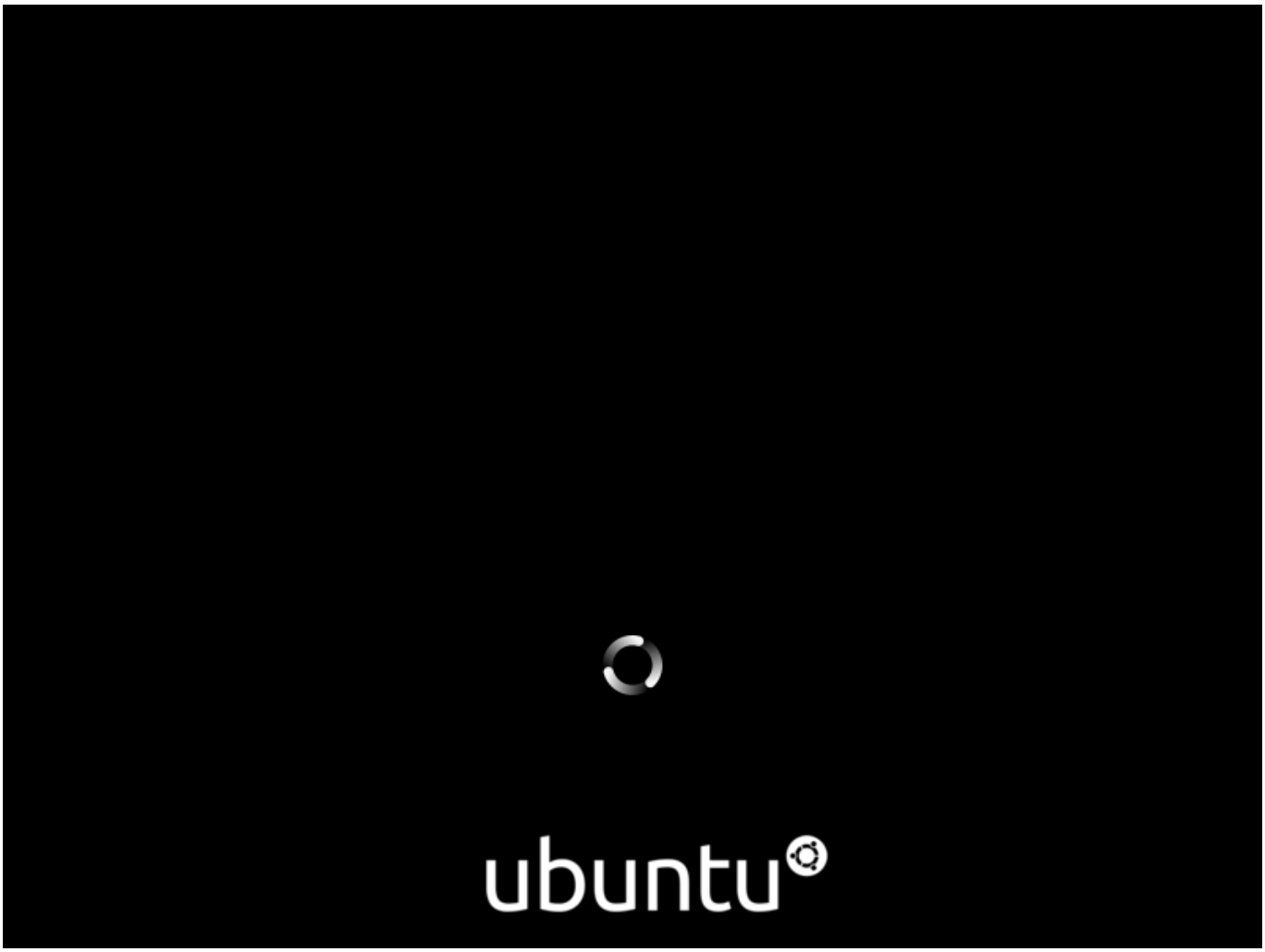

Figure : loading screen ubuntu

2. Installation screen

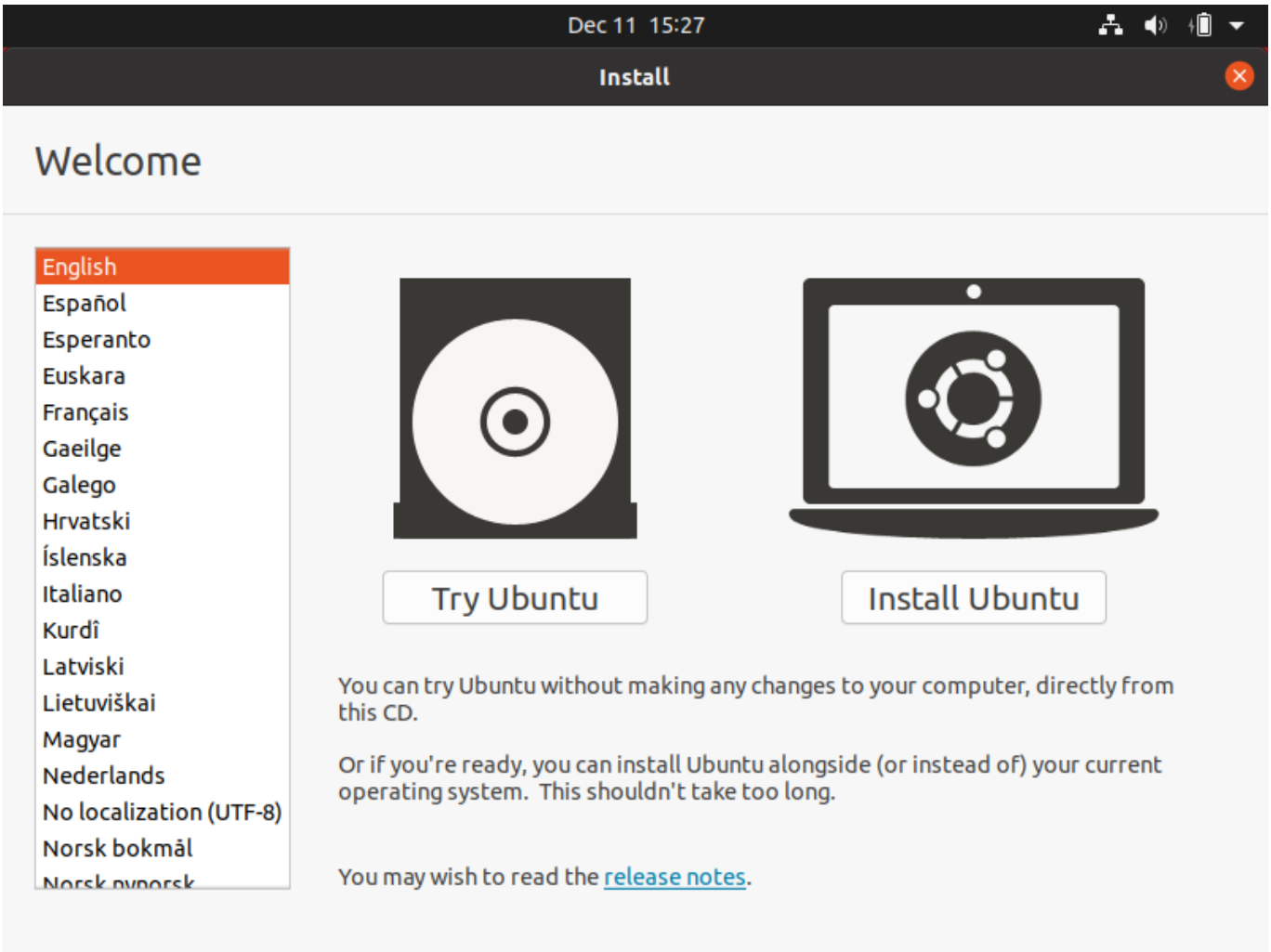

Figure : Installation screen in ubuntu installation

- click on `Install Ubuntu` if you want to install, click on `Try Ubuntu` if you want to try.

> in this document I'll continue with Install ubuntu

3. Select the keyboard

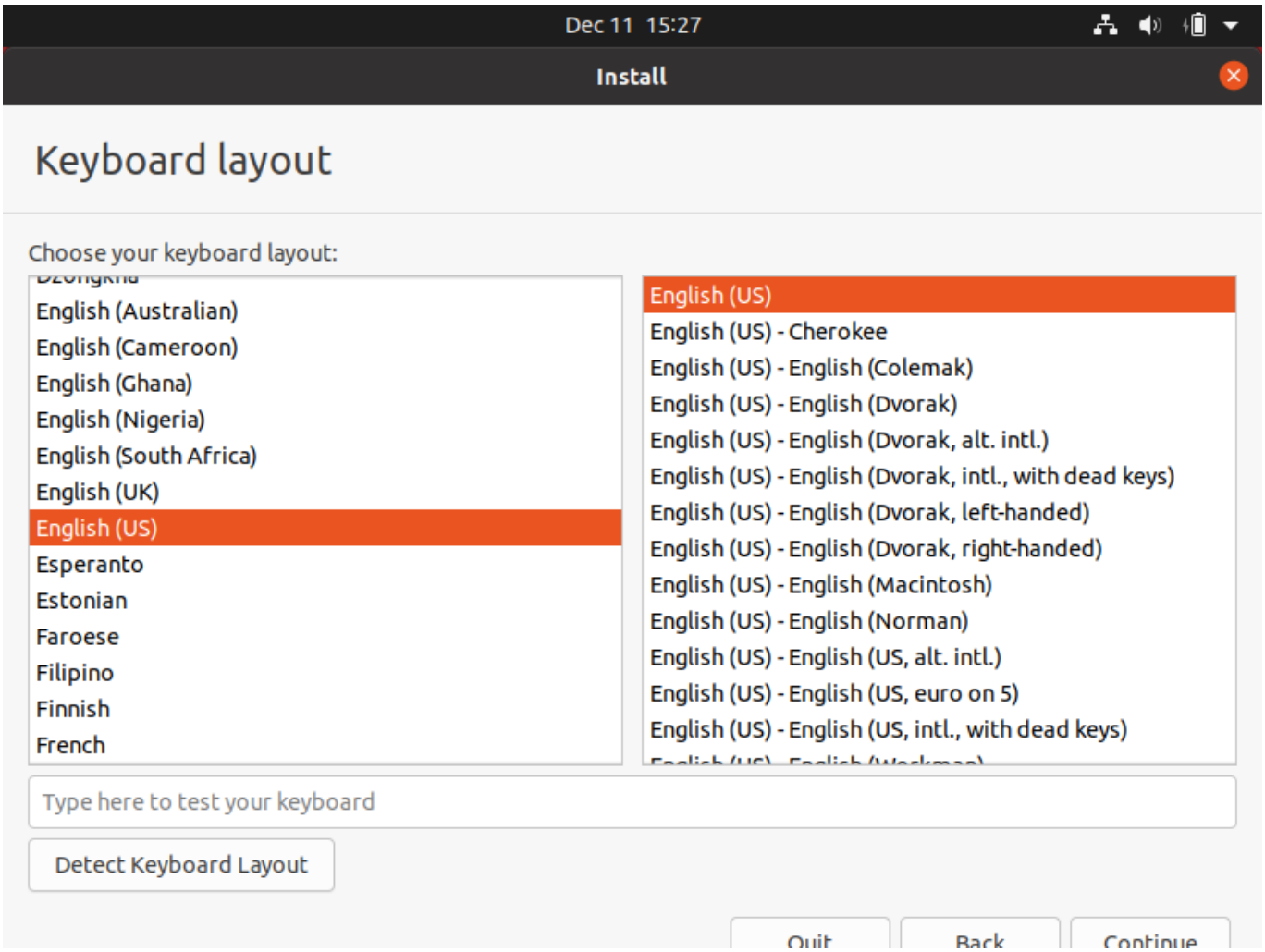

Figure : Keyboard selection in ubuntu

- for my system its US keyboard, check your keyboard layout and confirm this here.

> To understand more about keyboard layout click here

4. Select insllation type

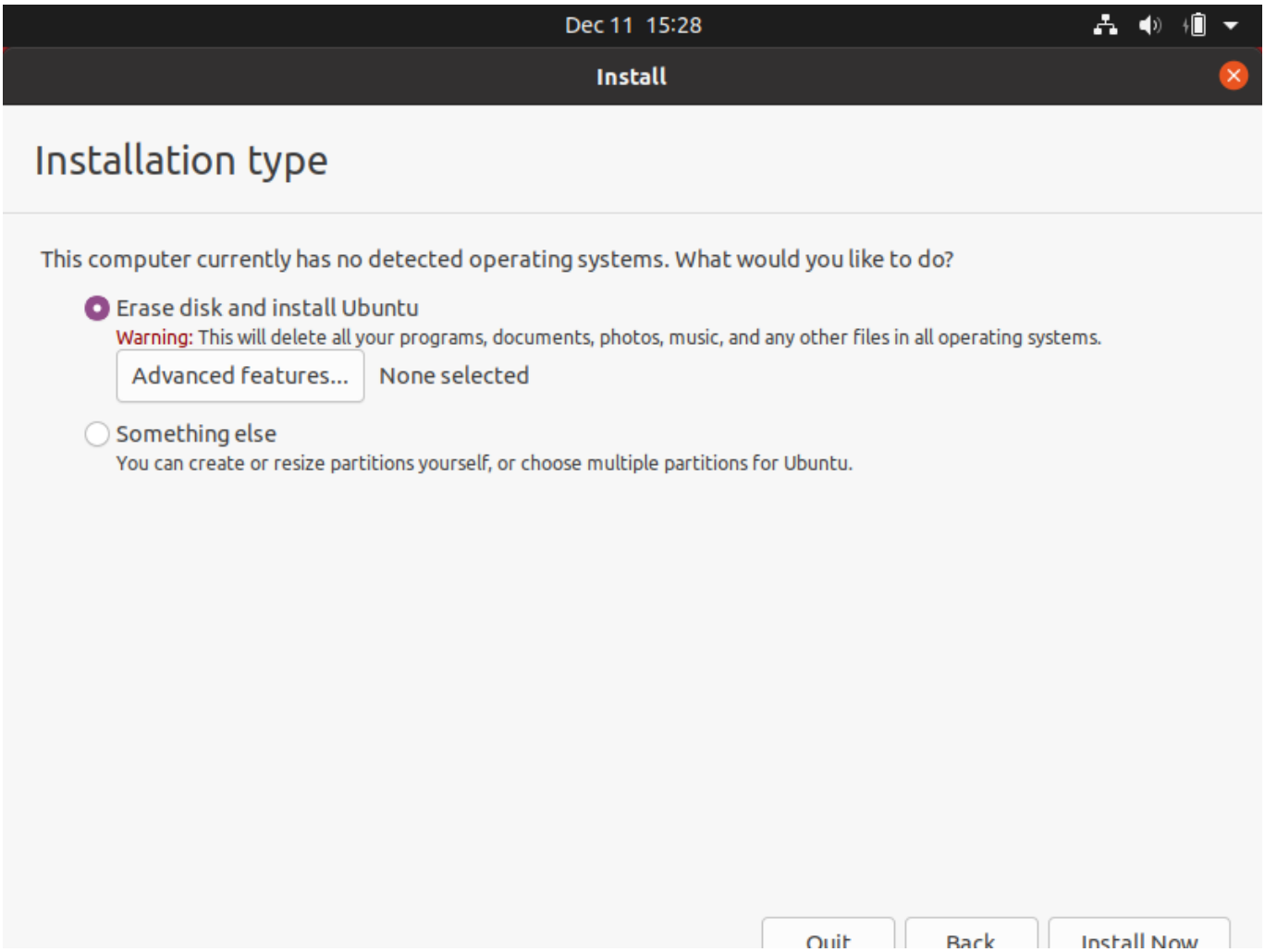

Figure : installation type selection

- select erase disk and install but if you want to install in dual boot or there is some disk configuration you needed then select something else

5. Select region for time and other service sync

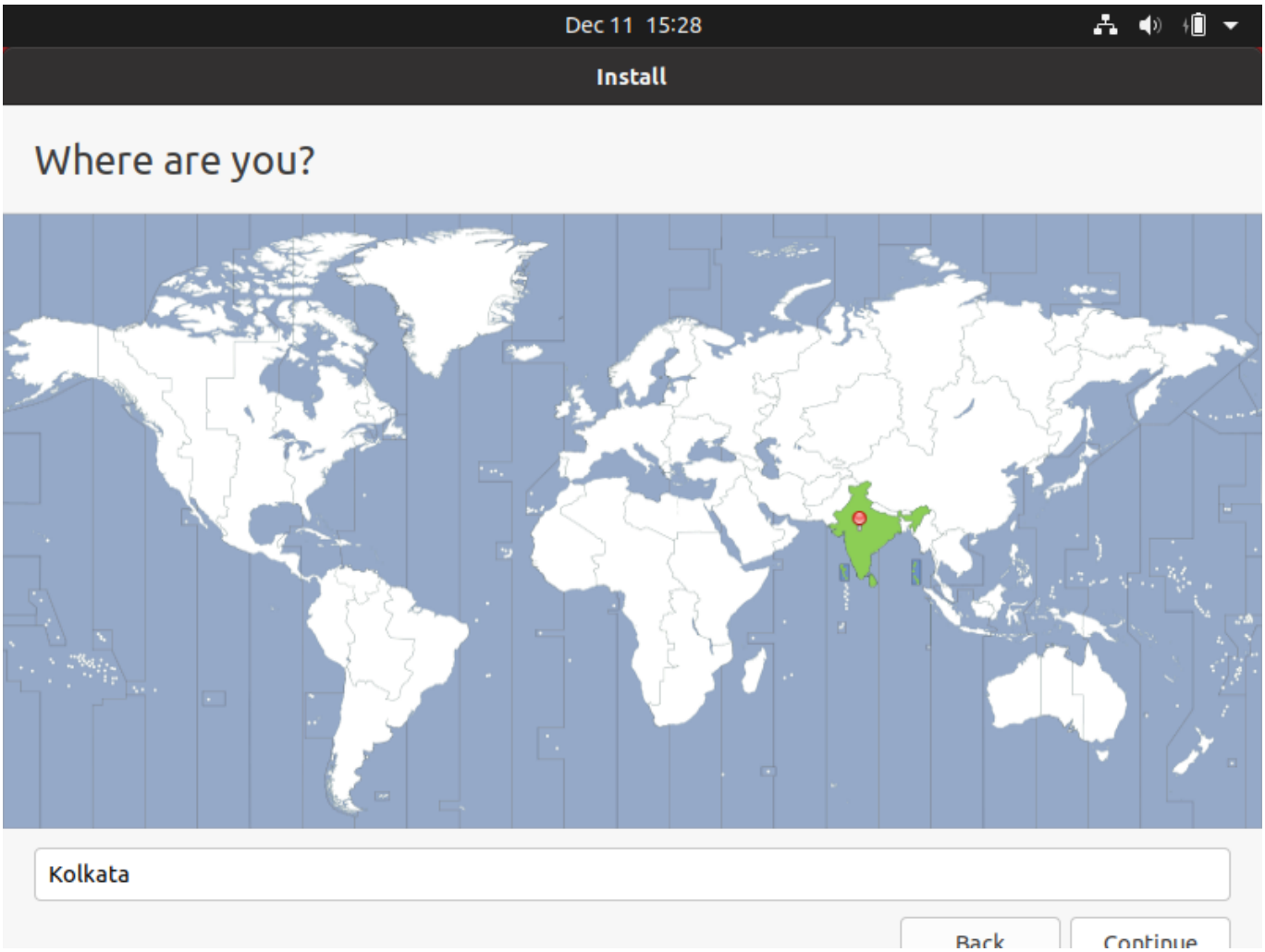

Figure : selection of region

- For my use I'm using Kolkata / IST `(India standard Time)`.

6. Give username, password & system hostname

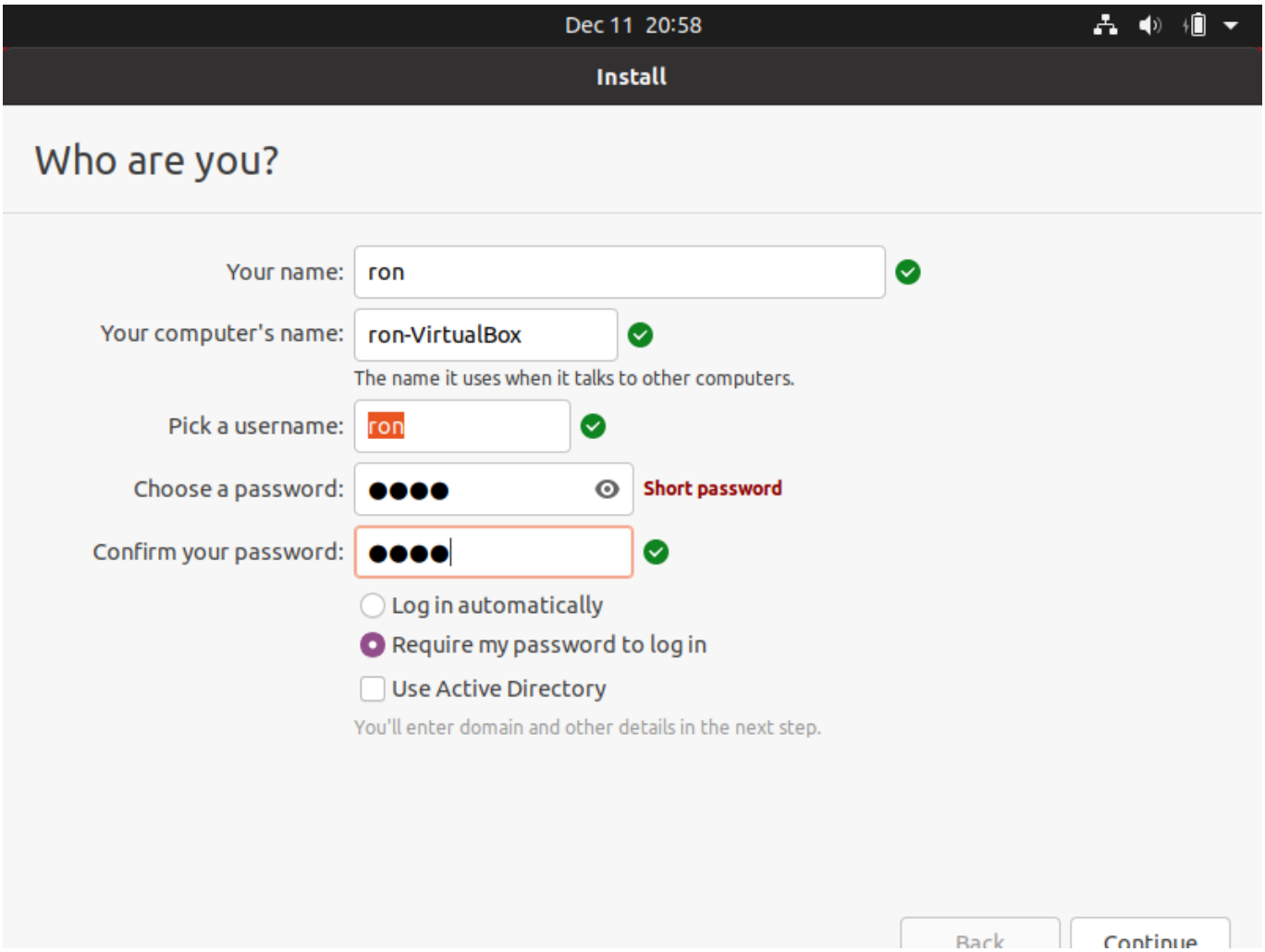

Figure: Password and hostname

- Change this according to your system

7. Installation Completion

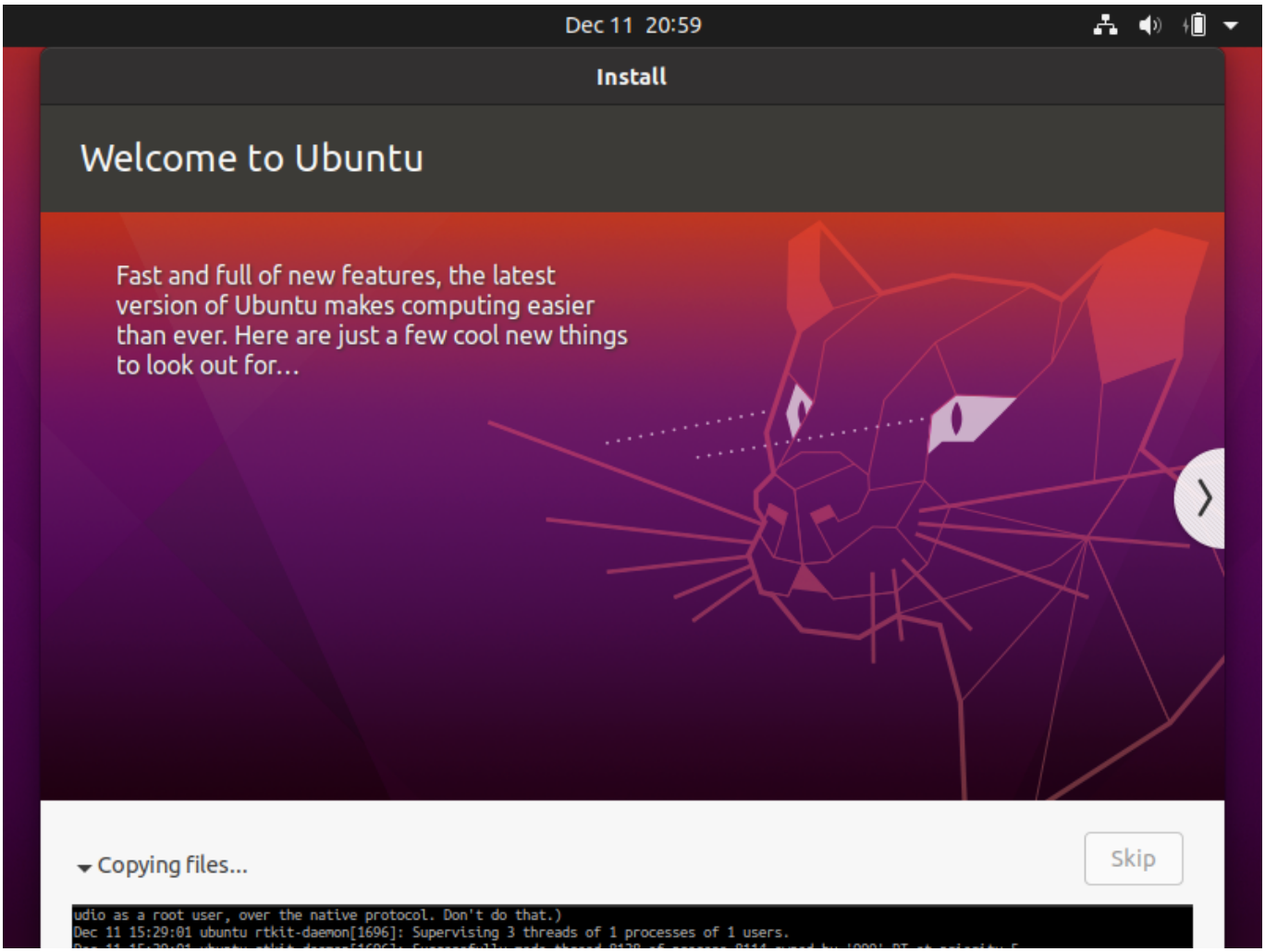

Figure: Started installation screen

> **WAIT FOR THE TOTAL PROCESS TO COMPLETE BY ITSELF**

8. Reboot and remove drive

Figure: after installation screen

- if this screen comes then your installation is complete and you can remove the drive and press enter to reboot to the system

> Congratulation your ubuntu is installated

Basic Ubuntu Command

1. ls

```
$ ls
```

this list the directory.

`expected output`

```
ron@ron-linux:~/SECONDARY_SSD/IOT LAB$ ls
aws-raspberrypi   flask_app   group6@10.10.14.86   index.html
testing.ipynb  'ultrasonic sensor.py'
```

> But this command doesn't show you hidden files and folder. For getting those you can use

```
$ ll
```

expected output

```
ron@ron-linux:~/SECONDARY_SSD/IOT LAB$ ll
total 33
drwxrwxrwx 1 root root 4096 Dec 11 21:34  ./
drwxrwxrwx 1 root root 4096 Dec 11 20:53  ../
drwxrwxrwx 1 root root 4096 Nov 15 16:16  aws-raspberrypi/
drwxrwxrwx 1 root root 4096 Nov  9 19:18  flask_app/
drwxrwxrwx 1 root root 4096 Nov 15 16:20 'group6@10.10.14.86'/
-rwxrwxrwx 1 root root 2860 Nov  9 19:06  index.html*
-rwxrwxrwx 1 root root  498 Dec 11 21:34  .something.txt*
-rwxrwxrwx 1 root root 1457 Nov 10 10:35  testing.ipynb*
-rwxrwxrwx 1 root root 1036 Nov 16 10:12 'ultrasonic sensor.py'*
```

> here you can see the .something.txt file which was not visible there

- understand more about this bellow

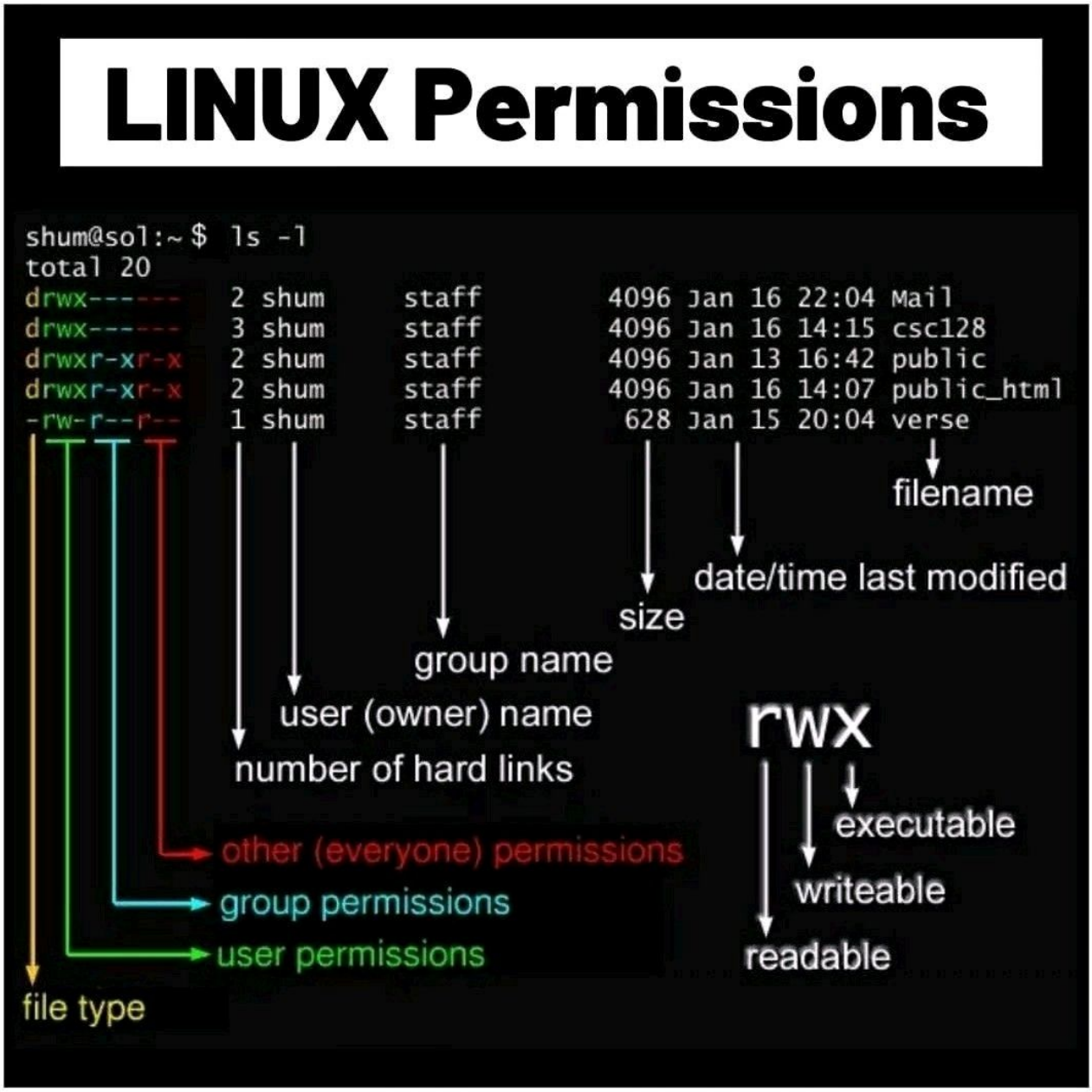

Figure: Understand the linux permission

2. pwd

```
$ pwd
```

- Print working directory command in Linux `expected output`

```
ron@ron-linux:~/SECONDARY_SSD/IOT LAB$ pwd
/home/ron/SECONDARY_SSD/IOT LAB
```

3. cd

```
$ cd
```

- Linux command to navigate through directories

expected output

```
ron@ron-linux:~$ pwd
/home/ron
ron@ron-linux:~$ cd Documents/
ron@ron-linux:~/Documents$ pwd
/home/ron/Documents
ron@ron-linux:~/Documents$
```

- current on /home/ron directory
- from there change directory to /home/ron/Documents

4. ## mkdir

```
$ mkdir
```

- Command used to create directories in Linux (basically creates a folder)

5. ## mv

```
$ mv
```

- Move or rename files in Linux

6. ## cp

```
$ cp
```

- Similar usage as mv but for copying files in Linux

7. ## rm

- Delete files or directories

8. ## touch

- Create blank/empty files

9. ln

```
$ ln
```

- Create symbolic links (shortcuts) to other files

10. cat

```
$ cat
```

- Display file contents on the terminal

11. clear

```
$ clear
```

- Clear the terminal display

12. echo

```
$ echo
```

- Print any text that follows the command

13. less

```
$ less
```

- Linux command to display paged outputs in the terminal

14. man

```
$ man
```

- Access manual pages for all Linux commands

15. uname

```
$ uname
```

- Linux command to get basic information about the OS

16. ## whoami

```
$ whoami
```

- Get the active username

17. ## tar

```
$ tar
```

- Command to extract and compress files in Linux

18. ## grep

```
$ grep
```

- Search for a string within an output

19. ## head

```
$ head
```

- Return the specified number of lines from the top

20. ## tail

```
$ tail
```

- Return the specified number of lines from the bottom

21. ## diff

```
$ diff
```

- Find the difference between two files

22. ## cmp

```
$ cmp
```

- Allows you to check if two files are identical

23. ## comm

```
$ comm
```

- Combines the functionality of diff and cmp

24. ## sort

```
$ sort
```

- Linux command to sort the content of a file while outputting

25. ## export

```
$ export
```

- Export environment variables in Linux

26. ## zip

```
$ zip
```

- Zip files in Linux

27. ## unzip

```
$ unzip
```

- Unzip files in Linux

28. ## ssh

```
$ ssh
```

- Secure Shell command in Linux

29. ## service

```
$ service
```

- Linux command to start and stop services

30. ## ps

```
$ ps
```

- Display active processes

31. ## kill and killall

```
$ kill and killall
```

- Kill active processes by process ID or name

32. ## df

```
$ df
```

- Display disk filesystem information

expected output

```
ron@ron-linux:~$ df
Filesystem     1K-blocks      Used Available Use% Mounted on
udev             3696096         0   3696096   0% /dev
tmpfs             746920      2436    744484   1% /run
/dev/nvme0n1p5 250870640 170290420  67763836  72% /
tmpfs            3734584     20880   3713704   1% /dev/shm
tmpfs               5120         4      5116   1% /run/lock
tmpfs            3734584         0   3734584   0% /sys/fs/cgroup
/dev/loop0           128       128         0 100% /snap/bare/5
/dev/loop1          2048      2048         0 100% /snap/btop/578
/dev/loop2         74624     74624         0 100% /snap/core22/310
/dev/loop3         83456     83456         0 100% /snap/discord/145
/dev/loop4         74624     74624         0 100% /snap/core22/444
/dev/loop5        424320    424320         0 100% /snap/gnome-42-2204/29
/dev/loop6         47104     47104         0 100% /snap/snap-store/599
/dev/loop7         47104     47104         0 100% /snap/snap-store/638
/dev/loop8        119552    119552         0 100% /snap/core/14399
/dev/loop15       119296    119296         0 100% /snap/jdownloader2/17
/dev/loop16        93952     93952         0 100% /snap/gtk-common-themes/1535
/dev/loop12        56448     56448         0 100% /snap/cups/836
/dev/loop10       457088    457088         0 100% /snap/gnome-42-2204/44
/dev/loop9          1408      1408         0 100% /snap/nvtop/66
/dev/loop14       573696    573696         0 100% /snap/pycharm-community/307
/dev/loop13       354688    354688         0 100% /snap/gnome-3-38-2004/115
/dev/loop11       354688    354688         0 100% /snap/gnome-3-38-2004/119
/dev/loop22        50816     50816         0 100% /snap/snapd/17883
/dev/loop19         2048      2048         0 100% /snap/btop/583
/dev/loop20       148864    148864         0 100% /snap/chromium/2193
/dev/loop18        64768     64768         0 100% /snap/core20/1695
/dev/loop17        83200     83200         0 100% /snap/discord/143
/dev/loop23        64768     64768         0 100% /snap/core20/1738
/dev/loop21        56960     56960         0 100% /snap/core18/2620
/dev/loop26        56960     56960         0 100% /snap/core18/2632
/dev/loop25          640       640         0 100% /snap/quadrapassel/481
/dev/loop24       604416    604416         0 100% /snap/pycharm-community/310
/dev/nvme0n1p1    262144     63300    198844  25% /boot/efi
/dev/sda2      234413052 144785180  89627872  62% /home/ron/SECONDARY_SSD
tmpfs             746916        16    746900   1% /run/user/125
tmpfs             746916        60    746856   1% /run/user/1000
/dev/nvme0n1p3 242775036  60054348 182720688  25% /media/ron/Acer
```

Figure: expected output of `df`

- for more human readable view use `-h` flag

```
$ df -h
```

`expected output`

```
ron@ron-linux:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.6G     0  3.6G   0% /dev
tmpfs           730M  2.4M  728M   1% /run
/dev/nvme0n1p5  240G  163G   65G  72% /
tmpfs           3.6G   21M  3.6G   1% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.6G     0  3.6G   0% /sys/fs/cgroup
/dev/loop0      128K  128K     0 100% /snap/bare/5
/dev/loop1      2.0M  2.0M     0 100% /snap/btop/578
/dev/loop2       73M   73M     0 100% /snap/core22/310
/dev/loop3       82M   82M     0 100% /snap/discord/145
/dev/loop4       73M   73M     0 100% /snap/core22/444
/dev/loop5      415M  415M     0 100% /snap/gnome-42-2204/29
/dev/loop6       46M   46M     0 100% /snap/snap-store/599
/dev/loop7       46M   46M     0 100% /snap/snap-store/638
/dev/loop8      117M  117M     0 100% /snap/core/14399
/dev/loop15     117M  117M     0 100% /snap/jdownloader2/17
/dev/loop16      92M   92M     0 100% /snap/gtk-common-themes/1535
/dev/loop12      56M   56M     0 100% /snap/cups/836
/dev/loop10     447M  447M     0 100% /snap/gnome-42-2204/44
/dev/loop9      1.4M  1.4M     0 100% /snap/nvtop/66
/dev/loop14     561M  561M     0 100% /snap/pycharm-community/307
/dev/loop13     347M  347M     0 100% /snap/gnome-3-38-2004/115
/dev/loop11     347M  347M     0 100% /snap/gnome-3-38-2004/119
/dev/loop22      50M   50M     0 100% /snap/snapd/17883
/dev/loop19     2.0M  2.0M     0 100% /snap/btop/583
/dev/loop20     146M  146M     0 100% /snap/chromium/2193
/dev/loop18      64M   64M     0 100% /snap/core20/1695
/dev/loop17      82M   82M     0 100% /snap/discord/143
/dev/loop23      64M   64M     0 100% /snap/core20/1738
/dev/loop21      56M   56M     0 100% /snap/core18/2620
/dev/loop26      56M   56M     0 100% /snap/core18/2632
/dev/loop25     640K  640K     0 100% /snap/quadrapassel/481
/dev/loop24     591M  591M     0 100% /snap/pycharm-community/310
/dev/nvme0n1p1  256M   62M  195M  25% /boot/efi
/dev/sda2       224G  139G   86G  62% /home/ron/SECONDARY_SSD
tmpfs           730M   16K  730M   1% /run/user/125
tmpfs           730M   60K  730M   1% /run/user/1000
/dev/nvme0n1p3  232G   58G  175G  25% /media/ron/Acer
```

Figure: expected output of `df -h`

33. ## mount

```
$ mount
```

- Mount file systems in Linux

34. ## chmod

```
$ chmod
```

- Command to change file permissions

35. ## chown

```
$ chown
```

- Command for granting ownership of files or folders

36. ## ifconfig

```
$ ifconfig
```

- Display network interfaces and IP addresses

37. ## traceroute

```
$ traceroute
```

- Trace all the network hops to reach the destination

38. ## wget

```
$ wget
```

- Direct download files from the internet

39. ## ufw

```
$ ufw
```

- Firewall command

40. ## iptables

```
$ iptables
```

- Base firewall for all other firewall utilities to interface with

41. ## apt, pacman, yum, rpm

```
$ apt
$ pacman
$ yum
$ rpm
```

- Package managers depending on the distro

42. ## sudo

```
$ sudo
```

- Command to escalate privileges in Linux

43. ## cal

```
$ cal
```

- View a command-line calendar

44. ## alias

```
$ alias
```

- Create custom shortcuts for your regularly used commands

45. ## dd

```
$ dd
```

- Majorly used for creating bootable USB sticks

46. ## whereis

```
$ whereis
```

- Locate the binary, source, and manual pages for a command

47. whatis

```
$ whatis
```

- Find what a command is used for

48. top

```
$ top
```

- View active processes live with their system usage
- there is also better alternative with more information like htop, btop
- to install htop run this command

```
$ sudo apt install htop
OR
$ sudo apt install btop
```

expected view

```
top - 23:18:24 up  3:43,  1 user,  load average: 0.43, 0.72, 0.70
Tasks: 411 total,   1 running, 410 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.1 us,  0.4 sy,  0.0 ni, 98.4 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :   7294.1 total,   1100.3 free,   4816.7 used,   1377.1 buff/cache
MiB Swap:   2048.0 total,    371.9 free,   1676.1 used.   2104.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2663 ron       20   0 6795624 161548  29504 S   9.9   2.2  10:45.12 gnome-shell
   2130 ron       20   0   25.5g  73212  31732 S   2.3   1.0   6:15.02 Xorg
    270 root     -51   0       0      0      0 S   0.7   0.0   0:14.35 irq/43-SYNA7DB5
   3290 ron       20   0 5163364 415624 151848 S   0.7   5.6  35:38.95 firefox
  19637 ron       20   0   83268   1556   1432 S   0.7   0.0   0:03.41 VBoxXPCOMIPCD
  24712 root      20   0       0      0      0 D   0.7   0.0   0:20.22 kworker/u32:3+events_unbou
  45413 ron       20   0   20884   4148   3076 R   0.7   0.1   0:00.10 top
     14 root      20   0       0      0      0 I   0.3   0.0   0:15.66 rcu_sched
     75 root      rt   0       0      0      0 S   0.3   0.0   0:00.28 migration/10
   1129 root      20   0 1740956  19988   6648 S   0.3   0.3  14:00.58 warp-svc
   3905 ron       20   0 2476388  52484  33672 S   0.3   0.7   0:38.70 Isolated Servic
   4409 ron       20   0 3224936 355576  72244 S   0.3   4.8   1:47.11 Isolated Web Co
   4412 ron       20   0 2998792 298836  49644 S   0.3   4.0   1:17.28 Isolated Web Co
   8642 ron       20   0  822476  20512  13660 S   0.3   0.3   0:05.85 gnome-terminal-
```

49. useradd and usermod

```
$ useradd
$ usermod
```

- Add new user or change existing users data

50. passwd

```
$ passwd
```

- Create or update passwords for existing users

Hadoop installation

Prerequisite Test

```
sudo apt update
sudo apt install openjdk-8-jdk -y

java -version; javac -version
sudo apt install openssh-server openssh-client -y
sudo adduser hdoop
su - hdoop
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
ssh localhost
```

Downloading Hadoop (Please note link is updated to new version of hadoop here on 6th May 2022)

```
$ wget https://downloads.apache.org/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz
$ tar xzf hadoop-3.2.3.tar.gz
```

Editng 6 important files

1st file

```
$ sudo nano .bashrc
```

- here you might face issue saying hdoop is not sudo user if this issue comes then

```
$ su - ron
$ sudo adduser hdoop sudo

$ sudo nano .bashrc
```

#Add below lines in this file

```
#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/hadoop-3.2.3
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$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 PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS"-Djava.library.path=$HADOOP_HOME/lib/nativ"
```

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

2nd File

```
$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- Add below line in this file in the end

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

3rd File

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

- Add below lines in this file(between "" and "<"/configuration>")

```
   <property>
        <name>hadoop.tmp.dir</name>
        <value>/home/hdoop/tmpdata</value>
        <description>A base for other temporary directories.</description>
    </property>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
        <description>The name of the default file system></description>
    </property>
```

4th File

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- Add below lines in this file(between "" and "<"/configuration>")

```
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
```

5th File

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Add below lines in this file(between "" and "<"/configuration>")

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

6th File

```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

- Add below lines in this file(between "" and "<"/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>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>127.0.0.1</value>
  </property>
  <property>
    <name>yarn.acl.enable</name>
    <value>0</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>

  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSP
  ATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
  </property>
```

Launching Hadoop

```
$ hdfs namenode -format

$ ./start-dfs.sh
```

or

```
$ bash start-all.sh
```

- After launching if you run `$ jps` command then you will get this output

```
hadoop@ron-VirtualBox:~$ jps
3858 SecondaryNameNode
4563 Jps
4052 ResourceManager
3641 DataNode
3453 NameNode
4287 NodeManager
```

Word count program

- To use the word count program in the hadoop we need to upload a file in the `HADOOP DFS`.
- TO upload so use command

```
$ hdfs dfs -put /directory/to/file /directory/target/folder
```

- if there is no directory in `DFS` then create using the command create a folder

```
$ hdfs dfs -mkdir /<Folder Name>
```

- using `ls` command we can see the folders and files

```
$ hdfs dfs -ls /
```

1. ## check and create folders present in the `DFS`

   - use `ls` command to check folders

`expected output`

```
hadoop@ron-VirtualBox:~$ hdfs dfs -ls  /
hadoop@ron-VirtualBox:~$
```

   - as there is no folder is not showing any. So, lets create directory using `mkdir`

```
hadoop@ron-VirtualBox:~$ hdfs dfs -mkdir /test
hadoop@ron-VirtualBox:~$ hdfs dfs -ls  /
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2022-12-12 10:57 /test
hadoop@ron-VirtualBox:~$
```

   - now in the output you can see `drwxr-xr-x - hadoop supergroup 0 2022-12-12 10:57 /test` which indicates a folder name test is there

2. ## Put files inside folder

   - to put the file inside Hadoop dfs we need to use `put` command as discussed above

   - but first create a test file using `nano` after than use `cat` to read the file

expected output

```
hadoop@ron-VirtualBox:~$ nano something.txt
hadoop@ron-VirtualBox:~$ cat something.txt
a quick brown fox jumps over the lazy dog.
The most lazy people decline things based on their interest.
But hard working people accepts things based on their need.

hadoop@ron-VirtualBox:~$
```

- now copy / put the file inside the `/test` folder of dfs

expected output

```
hadoop@ron-VirtualBox:~$ hdfs dfs -put something.txt /test/
hadoop@ron-VirtualBox:~$ hdfs dfs -ls /test/
Found 1 items
-rw-r--r--   1 hadoop supergroup        165 2022-12-12 11:11
/test/something.txt
hadoop@ron-VirtualBox:~$
```

3. Run hadoop jar with the folder

- to run jar in hadoop then use this command

```
$ hadoop jar /location/to/jar/file / <OPERATION NAME>
/directory/to/file /directory/to/output/folder
```

expected output

```
hadoop@ron-VirtualBox:~$ hadoop jar hadoop-
3.2.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar
wordcount /test/something.txt /output/
2022-12-12 11:18:36,095 WARN util.NativeCodeLoader: Unable to load
native-hadoop library for your platform... using builtin-java classes
where applicable
2022-12-12 11:18:36,506 INFO client.RMProxy: Connecting to
ResourceManager at /127.0.0.1:8032
2022-12-12 11:18:36,833 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path: /tmp/hadoop-
yarn/staging/hadoop/.staging/job_1670822333931_0001
2022-12-12 11:18:36,998 INFO input.FileInputFormat: Total input files
to process : 1
2022-12-12 11:18:37,068 INFO mapreduce.JobSubmitter: number of
splits:1
2022-12-12 11:18:37,202 INFO mapreduce.JobSubmitter: Submitting tokens
```

```
for job: job_1670822333931_0001
2022-12-12 11:18:37,203 INFO mapreduce.JobSubmitter: Executing with
tokens: []
2022-12-12 11:18:37,331 INFO conf.Configuration: resource-types.xml
not found
2022-12-12 11:18:37,331 INFO resource.ResourceUtils: Unable to find
'resource-types.xml'.
2022-12-12 11:18:37,696 INFO impl.YarnClientImpl: Submitted
application application_1670822333931_0001
2022-12-12 11:18:37,723 INFO mapreduce.Job: The url to track the job:
http://ron-VirtualBox:8088/proxy/application_1670822333931_0001/
2022-12-12 11:18:37,724 INFO mapreduce.Job: Running job:
job_1670822333931_0001
2022-12-12 11:18:43,797 INFO mapreduce.Job: Job job_1670822333931_0001
running in uber mode : false
2022-12-12 11:18:43,798 INFO mapreduce.Job:  map 0% reduce 0%
2022-12-12 11:18:46,842 INFO mapreduce.Job:  map 100% reduce 0%
2022-12-12 11:18:50,865 INFO mapreduce.Job:  map 100% reduce 100%
2022-12-12 11:18:51,882 INFO mapreduce.Job: Job job_1670822333931_0001
completed successfully
2022-12-12 11:18:51,941 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=274
        FILE: Number of bytes written=473053
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=270
        HDFS: Number of bytes written=176
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=1499
        Total time spent by all reduces in occupied slots (ms)=1544
        Total time spent by all map tasks (ms)=1499
        Total time spent by all reduce tasks (ms)=1544
        Total vcore-milliseconds taken by all map tasks=1499
        Total vcore-milliseconds taken by all reduce tasks=1544
        Total megabyte-milliseconds taken by all map tasks=1534976
        Total megabyte-milliseconds taken by all reduce tasks=1581056
    Map-Reduce Framework
        Map input records=4
        Map output records=29
        Map output bytes=280
        Map output materialized bytes=274
        Input split bytes=105
        Combine input records=29
        Combine output records=23
        Reduce input groups=23
```

```
        Reduce shuffle bytes=274
        Reduce input records=23
        Reduce output records=23
        Spilled Records=46
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=60
        CPU time spent (ms)=750
        Physical memory (bytes) snapshot=508264448
        Virtual memory (bytes) snapshot=5101694976
        Total committed heap usage (bytes)=354942976
        Peak Map Physical memory (bytes)=316489728
        Peak Map Virtual memory (bytes)=2547867648
        Peak Reduce Physical memory (bytes)=191774720
        Peak Reduce Virtual memory (bytes)=2553827328
    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=165
    File Output Format Counters
        Bytes Written=176
```

- check folder now

```
hadoop@ron-VirtualBox:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2022-12-12 11:18 /output
drwxr-xr-x   - hadoop supergroup          0 2022-12-12 11:11 /test
drwx------   - hadoop supergroup          0 2022-12-12 11:18 /tmp
```

- check the output folder

```
hadoop@ron-VirtualBox:~$ hdfs dfs -ls /output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2022-12-12 11:18
/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup        176 2022-12-12 11:18
/output/part-r-00000
```

- read the file `part-r-00000

```
hadoop@ron-VirtualBox:~$ hdfs dfs -cat /output/part-r-00000
But 1
The 1
a   1
accepts 1
based   2
brown   1
decline 1
dog.    1
fox 1
hard    1
interest.   1
jumps   1
lazy    2
most    1
need.   1
on  2
over    1
people  2
quick   1
the 1
their   2
things  2
working 1
hadoop@ron-VirtualBox:~$
```

> As here you can see that output tells about the text and the number of occurance

Mongodb installation

To install mongodb we will use docker container system.

- we will install mongodb + mongo express
- mongo gives the direct access to express form express you get a webUI for the mongo

> Follow the steps

1. ## install docker

```
$ sudo apt install docker.io
```

2. ## portainer

   - its webUI for maintaining docker

```
$ sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --
restart=always -v /var/run/docker.sock:/var/run/docker.sock -v
portainer_data:/data portainer/portainer-ce:latest
```

- go to

```
https://localhost:9443
```

- you will get a webUI asking to set username password / set accordingly
- after that login into gui and you will see this screen

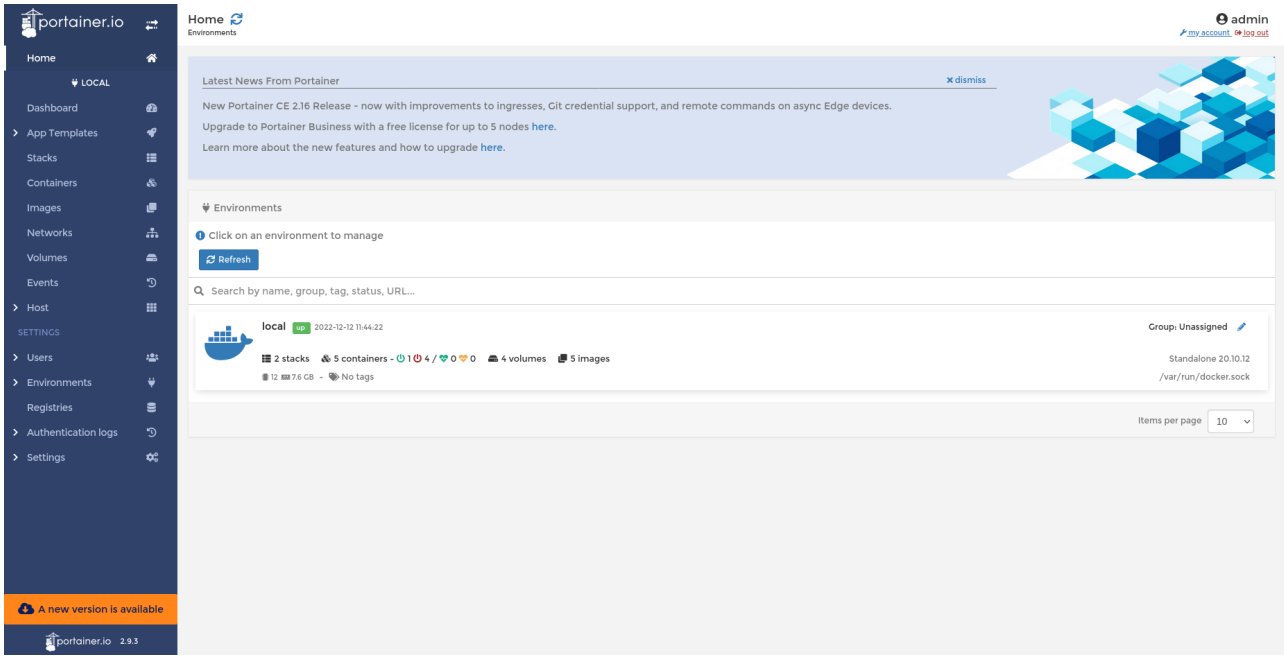

Figure : Portainer after login screen

- select the local and you will see this UI

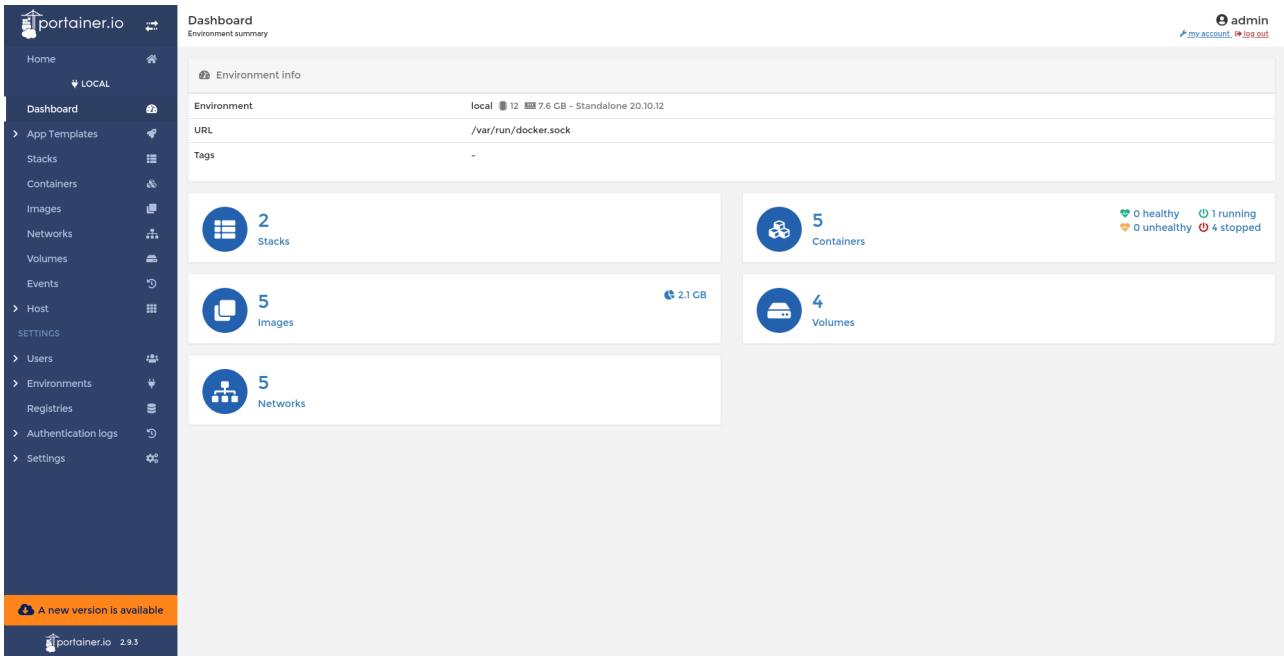

Figure: Local container in WEB UI

3. install mongodb

- select stacks
- select `+ Add stack`

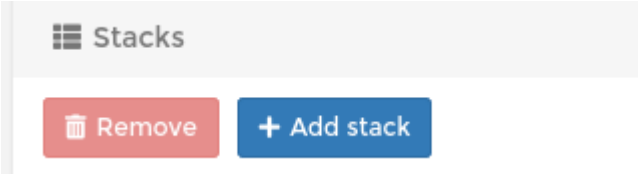

Figure: Stack addition

- copy and paste this yml text

```
# Use root/example as user/password credentials
version: '3.1'

services:

mongo:
    image: mongo
    #restart: None
    ports:
    - 27017:27017
    environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: root

mongo-express:
    image: mongo-express
    #restart: None
    ports:
```

```
      - 8081:8081
      environment:
      ME_CONFIG_MONGODB_ADMINUSERNAME: root
      ME_CONFIG_MONGODB_ADMINPASSWORD: root
      ME_CONFIG_MONGODB_URL: mongodb://root:root@mongo:27017/
```

- set a name

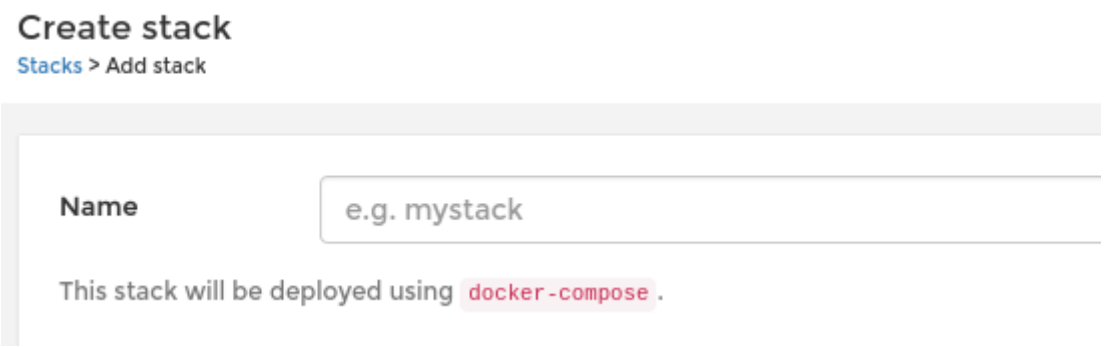

Figure: stack name

- deploy the stack

- after that this kind of screen will be seen

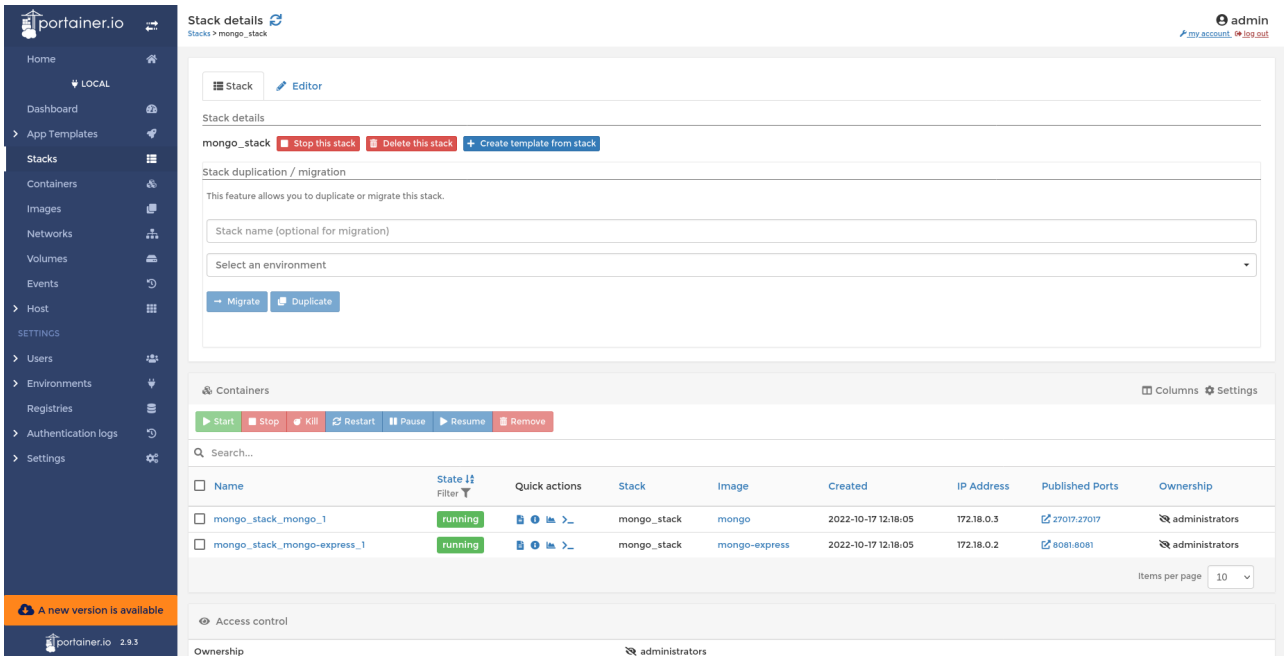

Figure: Deployed Mongo

- got to url

```
http://localhost:8081
```

- you will get web ui for mongo

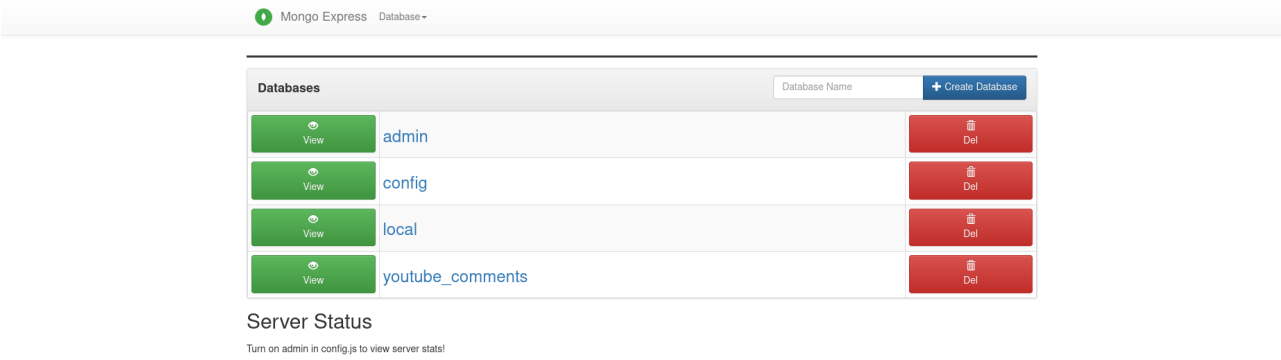

Figure: Mongo express web UI

> Its like this for me, might not be same for you. As I have created a `youtube_comments` manually

Congratulation your mongodb is inatlled and running

Basic queries of MongoDB

MongoDB Cheat Sheet

1. Check `monosh` Version

```
>mongosh --version


MongoDB shell version v5.0.13
```

2. Start the Mongo Shell

```
>mongosh "YOUR_CONNECTION_STRING" --username YOUR_USER_NAME
```

Show Current Database

```
>db


test
```

3. ### Show All Databases

```
>show dbs


admin   0.000GB
blog    0.000GB
config  0.000GB
local   0.000GB
```

4. ### Create Or Switch Database

```
>use Big_Data


switched to db big_data
```

5. ### Drop Database

```
>db.dropDatabase()


{ "ok" : 1 }
```

6. ### Create Collection

```
>db.createCollection('Students')


{ "ok" : 1 }
```

7. ## Show Collections

```
>show collections


Students
```

8. ## Insert Document

```
>db.Students.insertOne({
Name: 'Ardra',
Age: 22,
Course: 'MSc DA',
No: 8,
Interest: ['Reading', 'Music'],
date: Date()
})



        "acknowledged" : true,
        "insertedId" : ObjectId("63638a03c3e198ff6a8392cf")
```

9. ## Insert Multiple Documents

```
>db.Students.insertMany([
{
Name: 'Aleena',
Age: 22,
Course: 'MSc DA',
No: 9,
Interest: ['Reading', 'Writing'],
date: Date()
},
{
Name: 'Stalin',
Age: 22,
Course: 'MSc GA',
No: 4,
Interest: ['Dance', 'Music'],
date: Date()
},
{
Name: 'Navas',
Age: 22,
Course: 'MSc MI',
```

```
No: 16,
Interest: ['Sports'],
date: Date()
},
{ Name: 'Ajmala',
Age: 22,
Course: 'MSc MI',
No: 18,
Interest: ['Reading', 'Music'],
date: Date()
}
])



{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("63638a28c3e198ff6a8392d0"),
                ObjectId("63638a28c3e198ff6a8392d1"),
                ObjectId("63638a28c3e198ff6a8392d2"),
                ObjectId("63638a28c3e198ff6a8392d3")
        ]
}
```

10. ## Find All Documents

```
>db.Students.find()



{ "_id" : ObjectId("63638a03c3e198ff6a8392cf"), "Name" : "Ardra",
"Age" : 22, "Course" : "MSc DA", "No" : 8, "Interest" : [ "Reading",
"Music" ], "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)" }
{ "_id" : ObjectId("63638af8c3e198ff6a8392d4"), "Name" : "Aleena",
"Age" : 22, "Course" : "MSc DA", "No" : 9, "Interest" : [ "Reading",
"Writing" ], "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India
Standard Time)" }
{ "_id" : ObjectId("63638af8c3e198ff6a8392d5"), "Name" : "Stalin",
"Age" : 22, "Course" : "MSc GA", "No" : 4, "Interest" : [ "Dance",
"Music" ], "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)" }
{ "_id" : ObjectId("63638af8c3e198ff6a8392d6"), "Name" : "Navas",
"Age" : 22, "Course" : "MSc MI", "No" : 16, "Interest" : [ "Sports" ],
"date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard Time)" }
{ "_id" : ObjectId("63638af8c3e198ff6a8392d7"), "Name" : "Ajmala",
"Age" : 22, "Course" : "MSc MI", "No" : 18, "Interest" : [ "Reading",
"Music" ], "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
```

```
Time)" }
```

11. Find All Documents with pretty

```
>db.Students.find().pretty()



{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 22,
        "Course" : "MSc GA",
        "No" : 4,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d6"),
        "Name" : "Navas",
```

```
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 16,
        "Interest" : [
                "Sports"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d7"),
        "Name" : "Ajmala",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 18,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

12. Find Documents with Query

```
>db.Students.find({ Name:'Aleena' })




{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

13. Sort Documents

Ascending

```
>db.Students.find().sort({ No: 1 }).pretty()




{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 22,
        "Course" : "MSc GA",
        "No" : 4,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d6"),
        "Name" : "Navas",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 16,
        "Interest" : [
                "Sports"
        ],
```

```
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d7"),
        "Name" : "Ajmala",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 18,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

Descending

```
>db.Students.find().sort({ No: -1 }).pretty()




{
        "_id" : ObjectId("63638af8c3e198ff6a8392d7"),
        "Name" : "Ajmala",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 18,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d6"),
        "Name" : "Navas",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 16,
        "Interest" : [
                "Sports"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
```

```
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 22,
        "Course" : "MSc GA",
        "No" : 4,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

14. Count Documents

```
>db.Students.find().count()


5


>db.Students.find({ Course: 'MSc DA' }).count()
```

```
2
```

15. Limit Documents

```
>db.Students.find().limit(2).pretty()




{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

16. Chaining

```
>db.Students.find().limit(3).sort({Name:1}).pretty()




{
```

```
        "_id" : ObjectId("63638af8c3e198ff6a8392d7"),
        "Name" : "Ajmala",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 18,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "Interest" : [
                "Reading",
                "Writing"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
```

17. Find One Document

```
>db.Students.findOne({ Age: { $gt: 20 } })



{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra",
        "Age" : 22,
        "Course" : "MSc DA",
```

```
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 14:59:39 GMT+0530 (India Standard
Time)"
}
```

18. Update Document

```
>db.Students.updateOne({ Name: 'Navas' },
{
$set: {
    Age: 23
}
})



{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

19. After Updation

```
>db.Students.findOne({ Name:'Navas' })



{
        "_id" : ObjectId("63638af8c3e198ff6a8392d6"),
        "Name" : "Navas",
        "Age" : 23,
        "Course" : "MSc MI",
        "No" : 16,
        "Interest" : [
                "Sports"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

20. Update Document or Insert if not Found

```
>db.Students.updateOne({ Name: 'Ardra' },
{
$set: {
Name: 'Ardra Rajeesh',
Age: 22,
Course: 'MSc DA',
No: 7,
Interest: ['Reading', 'Music','Travel'],
date: Date()
}
},
{
upsert: true
})




{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

21. Increment Field ($inc)

```
>db.Students.updateOne({ Name:'Stalin' },
{
$inc: {
    Age:1
}
})




{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }


>db.Students.findOne({ Name:'Stalin' })




{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 23,
        "Course" : "MSc GA",
        "No" : 4,
        "Interest" : [
                "Dance",
                "Music"
```

```
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

22. Update Multiple Documents

```
>db.Students.updateMany({}, {
$inc: {
    No: 1
}
})



{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 5 }



>db.Students.find().pretty()



{
        "_id" : ObjectId("63638a03c3e198ff6a8392cf"),
        "Name" : "Ardra Rajeesh",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 8,
        "Interest" : [
                "Reading",
                "Music",
                "Travel"
        ],
        "date" : "Thu Nov 03 2022 16:30:11 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d4"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 10,
        "Interest" : [
                "Reading",
                "Writing"
```

```
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 23,
        "Course" : "MSc GA",
        "No" : 5,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d6"),
        "Name" : "Navas",
        "Age" : 23,
        "Course" : "MSc MI",
        "No" : 17,
        "Interest" : [
                "Sports"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d7"),
        "Name" : "Ajmala",
        "Age" : 22,
        "Course" : "MSc MI",
        "No" : 19,
        "Interest" : [
                "Reading",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

23. ## Rename Field

```
>db.Students.updateOne({ Name: 'Aleena' },
{
$rename: {
    Interest: 'Hobby'
}
```

```
})



{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }




>db.Students.find({Name:'Aleena'}).pretty()



{
        "_id" : ObjectId("6363a94f7ce8d232c5d49067"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "date" : "Thu Nov 03 2022 17:13:11 GMT+0530 (India Standard
Time)",
        "Hobby" : [
                "Reading",
                "Writing"
        ]
}
```

24. ## Delete a Document

```
>db.Students.deleteOne({ Name: 'Ajmala' })



{ "acknowledged" : true, "deletedCount" : 1 }
```

Delete Multiple Documents

```
>db.Students.deleteMany({ Course: 'MSc MI' })
```

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

25. Greater & Less Than

```
>db.Students.find({ Age: { $gt: 20 } }).pretty()




{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 23,
        "Course" : "MSc GA",
        "No" : 5,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("6363a8e45881447be22f55b8"),
        "Name" : "Ardra Rajeesh",
        "Age" : 22,
        "Course" : "MSc DA",
        "Interest" : [
                "Reading",
                "Music",
                "Travel"
        ],
        "No" : 7,
        "date" : "Thu Nov 03 2022 17:11:24 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("6363a94f7ce8d232c5d49067"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "date" : "Thu Nov 03 2022 17:13:11 GMT+0530 (India Standard
Time)",
        "Hobby" : [
                "Reading",
                "Writing"
        ]
}
```

```
>db.Students.find({ No: { $gte: 8 } }).pretty()
```

```
{
        "_id" : ObjectId("6363a94f7ce8d232c5d49067"),
        "Name" : "Aleena",
        "Age" : 22,
        "Course" : "MSc DA",
        "No" : 9,
        "date" : "Thu Nov 03 2022 17:13:11 GMT+0530 (India Standard
Time)",
        "Hobby" : [
                "Reading",
                "Writing"
        ]
}
```

```
>db.Students.find({ No: { $lt: 7 } }).pretty()
```

```
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
        "Age" : 23,
        "Course" : "MSc GA",
        "No" : 5,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
```

```
>db.Students.find({ No: { $lte: 7 } }).pretty()
```

```
{
        "_id" : ObjectId("63638af8c3e198ff6a8392d5"),
        "Name" : "Stalin",
```

```
        "Age" : 23,
        "Course" : "MSc GA",
        "No" : 5,
        "Interest" : [
                "Dance",
                "Music"
        ],
        "date" : "Thu Nov 03 2022 15:03:44 GMT+0530 (India Standard
Time)"
}
{
        "_id" : ObjectId("6363a8e45881447be22f55b8"),
        "Name" : "Ardra Rajeesh",
        "Age" : 22,
        "Course" : "MSc DA",
        "Interest" : [
                "Reading",
                "Music",
                "Travel"
        ],
        "No" : 7,
        "date" : "Thu Nov 03 2022 17:11:24 GMT+0530 (India Standard
Time)"
}
```

Pig Installation

- To install pig in system first download the pig tar

```
https://dlcdn.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz
```

- This URL redirects to the latest one currently available at time of documentation.
- use this command to download it in ubuntu

```
$ wget https://dlcdn.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz
```

- then un-tar the tar file using this command

```
$ tar -xvf pig-0.17.0.tar.gz
```

- add the path for `.bashrc`

```
export PIG_HOME=/home/hadoop/pig
export PATH=$PATH:/home/hadoop/pig/bin
export PIG_CLASSPATH=$HADOOP_HOME/conf
```

> for my use case the pig file is in `home/hadoop`. It might be different.

- run pig using `pig` in terminal

```
$ pig
```

`Expected output`

```
hadoop@ron-VirtualBox:~$ pig
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop-
3.2.3/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hbase-1.4.9/lib/slf4j-
log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2022-12-12 14:45:58,772 INFO pig.ExecTypeProvider: Trying ExecType :
LOCAL
2022-12-12 14:45:58,774 INFO pig.ExecTypeProvider: Trying ExecType :
MAPREDUCE
2022-12-12 14:45:58,774 INFO pig.ExecTypeProvider: Picked MAPREDUCE as
the ExecType
2022-12-12 14:45:58,817 [main] INFO  org.apache.pig.Main - Apache Pig
version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2022-12-12 14:45:58,817 [main] INFO  org.apache.pig.Main - Logging
error messages to: /home/hadoop/pig_1670836558812.log
2022-12-12 14:45:58,833 [main] INFO  org.apache.pig.impl.util.Utils -
Default bootup file /home/hadoop/.pigbootup not found
2022-12-12 14:45:58,995 [main] WARN
org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop
library for your platform... using builtin-java classes where
applicable
2022-12-12 14:45:59,010 [main] INFO
org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker
is deprecated. Instead, use mapreduce.jobtracker.address
2022-12-12 14:45:59,010 [main] INFO
org.apache.pig.backend.hadoop.executionengine.HExecutionEngine -
Connecting to hadoop file system at: hdfs://localhost:9000
2022-12-12 14:45:59,398 [main] INFO  org.apache.pig.PigServer - Pig
Script ID for the session: PIG-default-ebc2de17-41ce-433d-9d3e-
ce5b372f0cc2
2022-12-12 14:45:59,399 [main] WARN  org.apache.pig.PigServer - ATS is
```

```
disabled since yarn.timeline-service.enabled set to false
grunt>
```

- if you get `grunt >` then its working for you
- use `quit` to get out of grunt runtime

Basic queries on Pig

1. ## Fs

   - This will list all the file in the HDFS

   ```
   grunt> fs -ls
   ```

2. ## Clear

   - This will clear the interactive Grunt shell.

   ```
   grunt> clear
   ```

3. ## History

   - This command shows the commands executed so far.

   ```
   grunt> history
   ```

4. ## Reading Data

   - Assuming the data resides in HDFS, and we need to read data to Pig.

   ```
   grunt> college_students = LOAD
   'hdfs://localhost:9000/pig_data/college_data.txt'

   USING PigStorage(',')

   as ( id:int, firstname:chararray, lastname:chararray,
   phone:chararray,

   city:chararray );

   PigStorage() is the function that loads and stores data as
   structured text files.
   ```

5. ## Storing Data

   - Store operator is used to storing the processed/loaded data.

```
grunt> STORE college_students INTO ‘
hdfs://localhost:9000/pig_Output/ ‘ USING PigStorage (‘,’);

Here, “/pig_Output/” is the directory where relation needs to be
stored.
```

6. ## Dump Operator

   - This command is used to display the results on screen. It usually helps in debugging.

```
grunt> Dump college_students;
```

7. ## Describe Operator

   - It helps the programmer to view the schema of the relation.

```
grunt> describe college_students;
```

8. ## Explain

   - This command helps to review the logical, physical and map-reduce execution plans.

```
grunt> explain college_students;
```

9. ## Illustrate operator

   - This gives step-by-step execution of statements in Pig Commands.

```
grunt> illustrate college_students;
```

10. ## Group

    - This command works towards grouping data with the same key.

```
grunt> group_data = GROUP college_students by first name;
```

11. ## COGROUP

-It works similarly to the group operator. The main difference between Group & Cogroup operator is that group operator usually used with one relation, while cogroup is used with more than one relation.

12. ## Join

    - This is used to combine two or more relations.

> Example: In order to perform self-join, let's say relation "customer" is loaded from HDFS tp pig commands in two relations customers1 & customers2.

```
grunt> customers3 = JOIN customers1 BY id, customers2 BY id;
```

Join could be self-join, Inner-join, Outer-join.

13. ## Cross

    - This pig command calculates the cross product of two or more relations.

```
grunt> cross_data = CROSS customers, orders;
```

14. ## Union

    - It merges two relations. The condition for merging is that both the relation's columns and domains must be identical.

```
grunt> student = UNION student1, student2;
```

Hbase installation and Basic queries

- To install Hbase in ubuntu use download hbase from

```
https://archive.apache.org/dist/hbase/1.4.9/hbase-1.4.9-bin.tar.gz
```

- After download use this command to un-tar the file

```
$ tar -xvf hbase-1.4.9-bin.tar.gz
```

- after that add paths to `hbase/conf/hbase-env.sh` file
- run this command

```
$ nano hbase/conf/hbase-env.sh
```

- then add the bellow after this `[# The java implementation to use. Java 1.7+ required. ]` line

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export HADOOP_home=/home/hadoop/hadoop-3.2.3
```

> HADOOP MIGHT BE DIFFERENT FOR YOU KINDLY CHECK

- then change the add the bellow lines in `.bashrc`

```
# Hbase home
export HBASE_HOME=/home/hadoop/hbase-1.4.9
export PATH=$PATH:$HBASE_HOME/bin
```

- change hbase-site.xml which is situated hbase/conf/hbase-site.xml

```
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:9000/hbase</value>
</property>

<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>

<property>
<name>hbase.zookeeper.quorum</name>
<value>localhost</value>
</property>

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

<property>
```

```
<name>hbase.zookeeper.property.clientPort</name>
<value>2181</value>
</property>

<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/home/hduser/hbase/zookeeper</value>
</property>
```

- to start the base start hadoop first

```
$ bash hadoop-3.2.3/sbin/start-all.sh
```

- check using `jps`

```
hadoop@ron-VirtualBox:~$ jps
3653 Jps
3477 NodeManager
2726 DataNode
2538 NameNode
2940 SecondaryNameNode
3134 ResourceManager
```

- then start hbase

```
$ bash hbase-1.4.9/bin/start-hbase.sh
```

- then run `jps`

```
hadoop@ron-VirtualBox:~$ jps
4321 Jps
3477 NodeManager
4054 HMaster
2726 DataNode
2538 NameNode
2940 SecondaryNameNode
4188 HRegionServer
3134 ResourceManager
```

- to access the shell change directory to `hbase1.4.9/bin/`
- then use `hbase shell` to initiate the interactive shell of hbase

Pyspark installation and queries.

- Pyspark is basically python-spark library
- To install it run this command `Need to have python and pip`

```
$ pip install pyspark
```

- start a basic pyspark.sql session using this python code

```
from pyspark.sql import SparkSession
```

- user builder to start running the pyspark in the machine

```
spark = SparkSession.builder.appName('practise').getOrCreate()
```

- now if you run spark in the runtime you will get this output

```
spark
```

OUPUT

```
SparkSession - in-memory

SparkContext

Spark UI

Version
v3.3.1
Master
local[*]
AppName
practise
```

- Load csv using

```
df_spark = spark.read.csv('/home/ron/Downloads/guns - guns.csv')
df_spark =
spark.read.format('csv').option('header','true').load('/home/ron/Downl
oads/guns - guns.csv')
```

- Check basic queries
  - Check shema

    ```
    df_spark.printSchema()
    ```

  - show first 5 rows

    ```
    df_spark.show(5)
    ```

  - show months column in the output

    ```
    df_spark.select('month').show(5)
    ```

  - Filter by street

    ```
    df_spark.filter(df_spark.place == 'Street').show(4)
    ```

  - Filter using regex

    ```
    df_spark.filter(df_spark.race.like('W%')).filter((df_spark.age==6
    0) | (df_spark.age==31)).show()
    ```

  - To display the count of values (To show the total number of events that occurred in each month)

    ```
    df_spark.groupBy('month').count().show()
    ```

  - To display in ascending order (to display the places in ascending order)

    ```
    df_spark.orderBy('month').show(10)
    ```

  - To create a subset of people whose age is between 20 and 50

    ```
    subset = df_spark.filter((df_spark.age > 20 ) & (df_spark.age <
    50))
    subset.show()
    ```