

CSE 3035

PRINCIPLES OF CLOUD COMPUTING



Lab Assessment – 4

L15+L16 | SJT501
Dr. Sivaprakash S

FALL SEMESTER 2022-23

by

SHARADINDU ADHIKARI
19BCE2105

Assessment 4 .

Title: Hadoop, MapReduce, Grafana.

Aim:

- To install the Hadoop framework and create an application using the MapReduce programming model.
- To Monitor, visualize and analyse the performance of resource utilization in cloud platforms using the Grafana tool.

Background Theory:

- Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation.
- MapReduce is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster. As the processing component, MapReduce is the heart of Apache Hadoop. The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform.
- Grafana is a multi-platform open-source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.

Requirements for conducting the experiment:

- AWS Licence
- Virtual machine setup in host computer (VMWare, VirtualBox, ParallelsDesktop, etc.)

Procedure & Screenshots:

Create a new Instance and launch it.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (Instances New), Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs New, AMI Catalog), and Elastic Block Store. The main content area shows a table for 'Instances (1/1)'. The table has columns for Instance ID, Name, Instance state, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One row is selected for 'Cloud DA 4' with instance ID i-06c71bfc431d6bbaf, which is currently running. Below the table, there's a detailed view for 'Instance: i-06c71bfc431d6bbaf (Cloud DA 4)' with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, there's an 'Instance summary' section with fields for Instance ID, Public IPv4 address, Private IPv4 addresses, IPv6 address, Instance state, Private IP DNS name, Answer private resource DNS name, Instance type, and Elastic IP addresses.

The screenshot shows the AWS EC2 Connect interface. At the top, there are tabs for "EC2 Instance Connect", "Session Manager", "SSH client", and "EC2 serial console". The "EC2 Instance Connect" tab is selected. Below the tabs, the instance ID is listed as "i-06c71bf431d6bbaf (Cloud DA 4)". The public IP address is "13.127.96.25" and the user name is "ubuntu". A note states: "In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name." At the bottom right are "Cancel" and "Connect" buttons.



The terminal window shows the following output:

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed Oct 5 13:33:29 UTC 2022

System load: 0.3759765625 Processes: 104
Usage of /: 19.6% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.33.47
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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.
```

i-06c71bf431d6bbaf (Cloud DA 4)

PublicIPs: 13.127.96.25 PrivateIPs: 172.31.33.47



PART 1: Hadoop & MapReduce Installation

Steps:

```
~$ sudo apt update
~$ sudo apt install default-jdk
~$ java -version
```

```
~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz
~$ tar -xzvf hadoop-3.3.4.tar.gz
~$ sudo mv hadoop-3.3.4 /usr/local/hadoop
```

```
~$ readlink -f /usr/bin/java | sed "s:bin/java:::"
~$ sudo vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
~$ Copy paste this line: export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
```

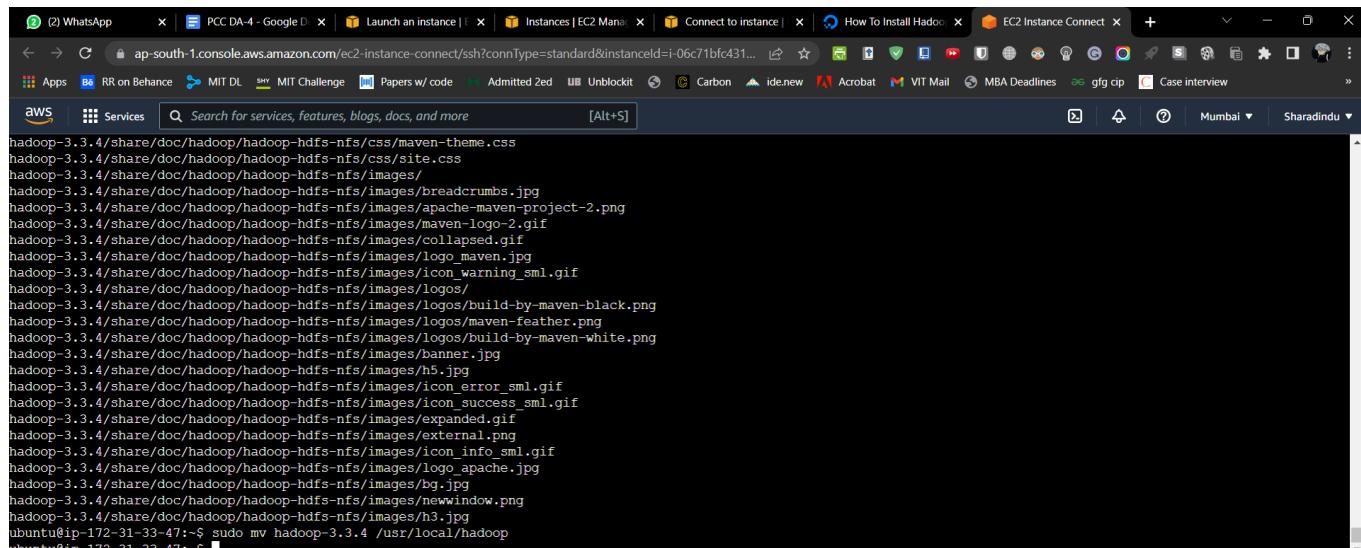
```
:w
:wq!
```

```
~$ /usr/local/hadoop/bin/hadoop
~$ mkdir ~/input
~$ cp /usr/local/hadoop/etc/hadoop/*.xml ~/input
```

```
~$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar grep ~/input ~/grep_example 'allowed[.]*' 
```

```
~$ cat ~/grep_example/*
~$ /usr/local/hadoop/bin/hadoop version
```

Screenshots:



i-06c71bfc431d6bbaf (Cloud DA 4)
PublicIPs: 13.127.96.25 PrivateIPs: 172.31.33.47



```
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/breadcrumbs.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/apache-maven-project-2.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/maven-logo-2.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/collapsed.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logo_maven.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/icon_warning_sml.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logos/
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logos/build-by-maven-black.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logos/maven-feather.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logos/build-by-maven-white.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/banner.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/h3.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/icon_error_sml.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/icon_success_sml.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/expanded.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/external.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/icon_info_sml.gif
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/logo_apache.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/bg.jpg
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/newwindow.png
hadoop-3.3.4/share/doc/hadoop/hdfs-nfs/images/h3.jpg
ubuntu@ip-172-31-33-47:~$ sudo mv hadoop-3.3.4 /usr/local/hadoop
ubuntu@ip-172-31-33-47:~$ readlink -f /usr/bin/java | sed "s:bin/java:::"
/usr/lib/jvm/java-11-openjdk-amd64/
ubuntu@ip-172-31-33-47:~$
```

i-06c71bfc431d6bbaf (Cloud DA 4)

Public IPs: 13.127.96.25 Private IPs: 172.31.33.47



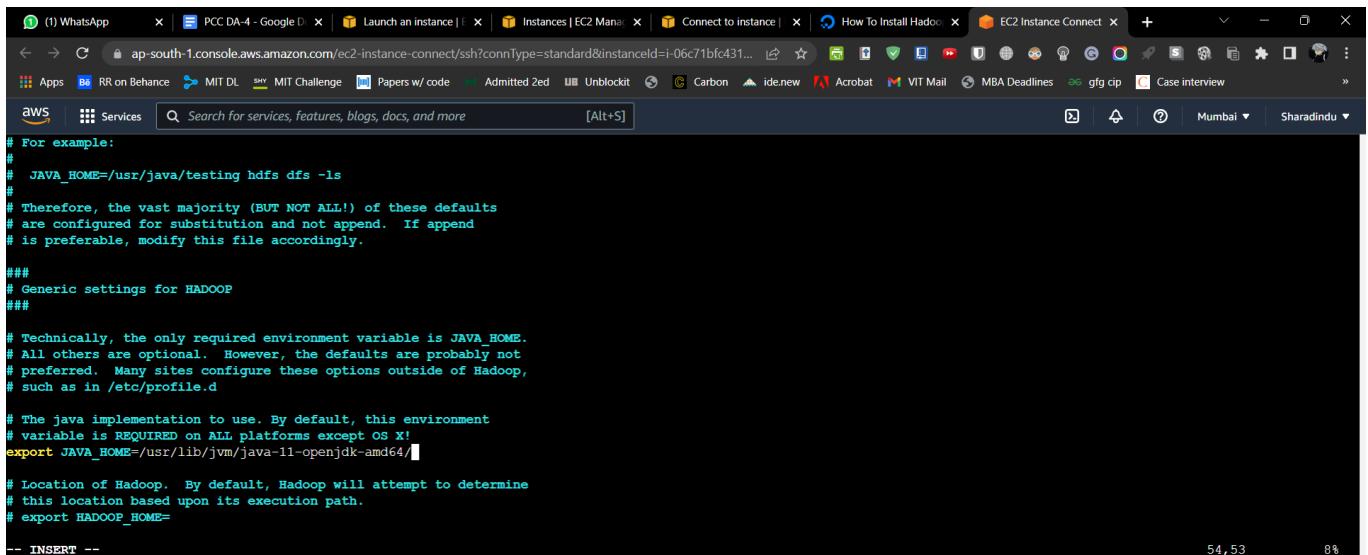
```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); 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.

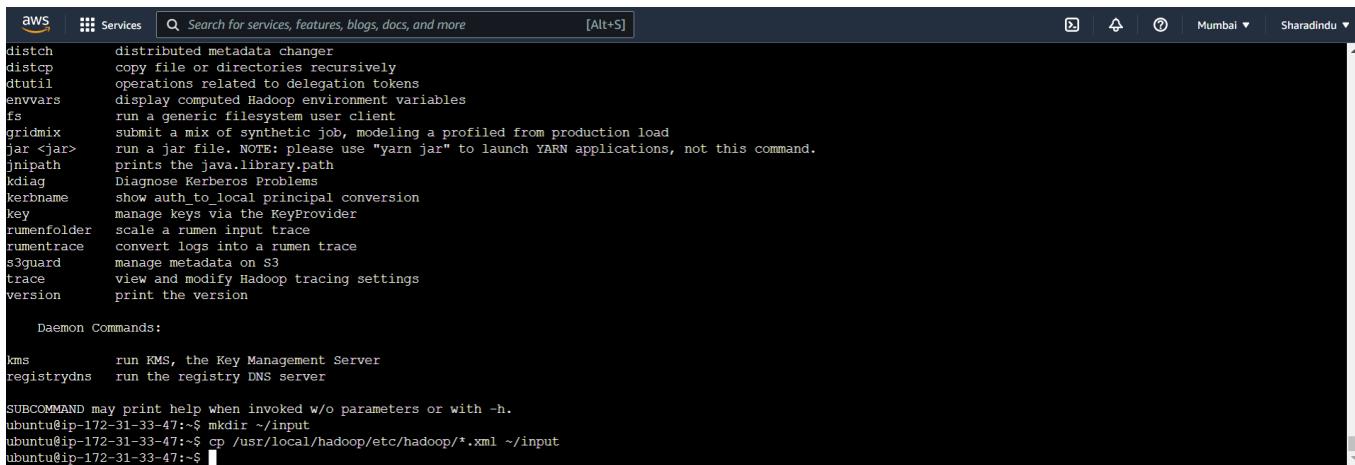
# Set Hadoop-specific environment variables here.

##
## THIS FILE ACTS AS THE MASTER FILE FOR ALL HADOOP PROJECTS.
## SETTINGS HERE WILL BE READ BY ALL HADOOP COMMANDS. THEREFORE,
## ONE CAN USE THIS FILE TO SET YARN, HDFS, AND MAPREDUCE
## CONFIGURATION OPTIONS INSTEAD OF xxx-env.sh.
##
"/usr/local/hadoop/etc/hadoop/hadoop-env.sh" 430L, 16654B
```

10-1

Top





```

aws Services Search for services, features, blogs, docs, and more [Alt+S]
[Alt+S] Mumbai Sharadindu

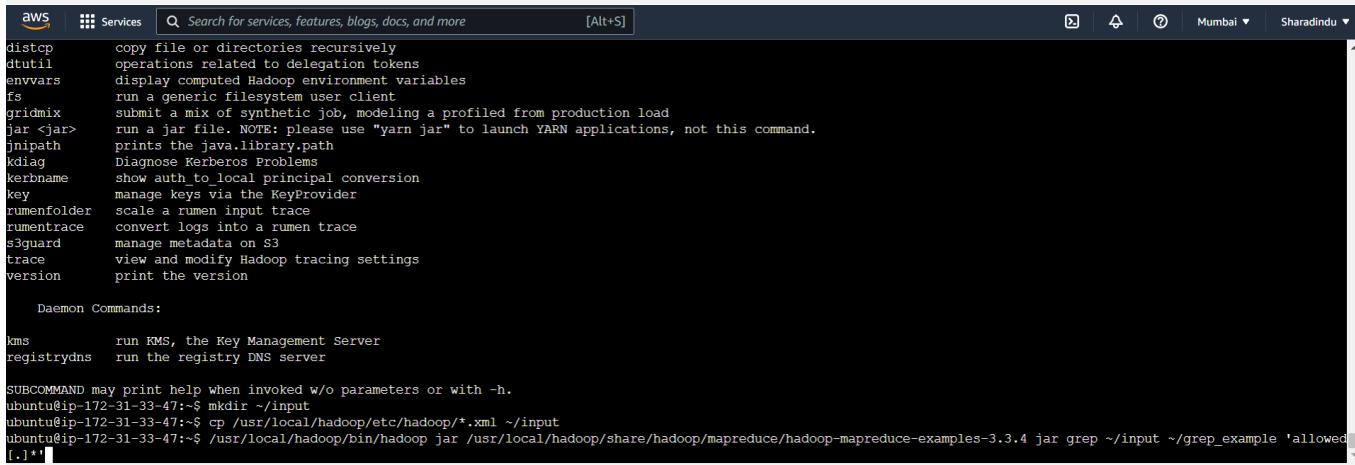
distch      distributed metadata changer
distcp      copy file or directories recursively
dtutil      operations related to delegation tokens
envvars     display computed Hadoop environment variables
fs          run a generic filesystem user client
gridmix    submit a mix of synthetic job, modeling a profiled from production load
jar <jar>   run a jar file. NOTE: please use "yarn jar" to launch YARN applications, not this command.
jnipath     prints the java.library.path
kdiag       Diagnose Kerberos Problems
kerbname    show auth_to_local principal conversion
key         manage keys via the KeyProvider
rumenfolder scale a rumen input trace
rumentrace  convert logs into a rumen trace
s3guard     manage metadata on S3
trace       view and modify Hadoop tracing settings
version     print the version

Daemon Commands:

kms        run KMS, the Key Management Server
registrydns run the registry DNS server

SUBCOMMAND may print help when invoked w/o parameters or with -h.
ubuntu@ip-172-31-33-47:~$ mkdir ~/input
ubuntu@ip-172-31-33-47:~$ cp /usr/local/hadoop/etc/hadoop/*.xml ~/input
ubuntu@ip-172-31-33-47:~$ 

```



```

aws Services Search for services, features, blogs, docs, and more [Alt+S]
[Alt+S] Mumbai Sharadindu

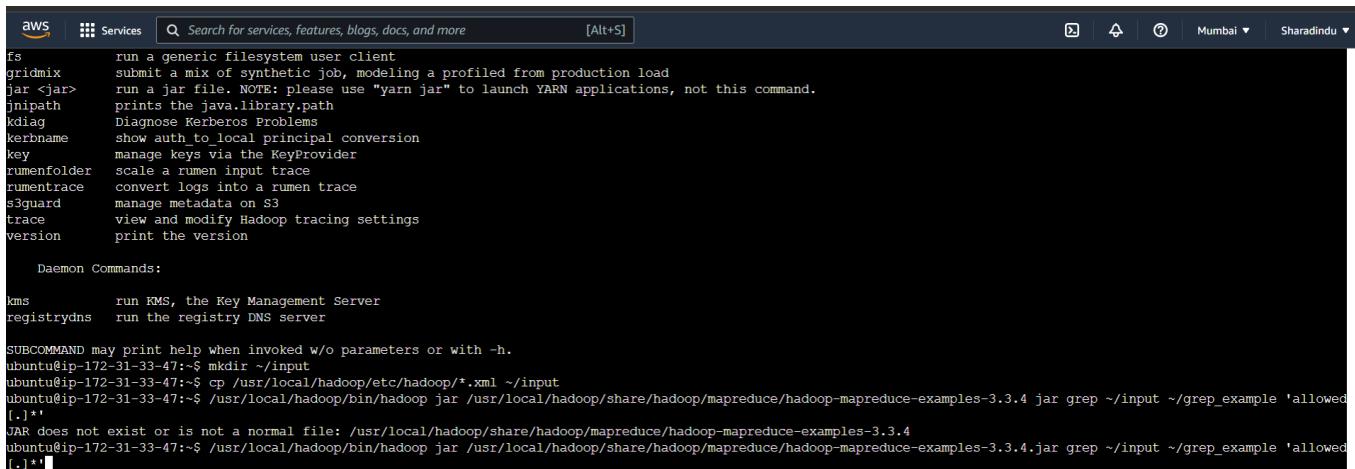
distcp      copy file or directories recursively
dtutil      operations related to delegation tokens
envvars     display computed Hadoop environment variables
fs          run a generic filesystem user client
gridmix    submit a mix of synthetic job, modeling a profiled from production load
jar <jar>   run a jar file. NOTE: please use "yarn jar" to launch YARN applications, not this command.
jnipath     prints the java.library.path
kdiag       Diagnose Kerberos Problems
kerbname    show auth_to_local principal conversion
key         manage keys via the KeyProvider
rumenfolder scale a rumen input trace
rumentrace  convert logs into a rumen trace
s3guard     manage metadata on S3
trace       view and modify Hadoop tracing settings
version     print the version

Daemon Commands:

kms        run KMS, the Key Management Server
registrydns run the registry DNS server

SUBCOMMAND may print help when invoked w/o parameters or with -h.
ubuntu@ip-172-31-33-47:~$ mkdir ~/input
ubuntu@ip-172-31-33-47:~$ cp /usr/local/hadoop/etc/hadoop/*.xml ~/input
ubuntu@ip-172-31-33-47:~$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce-examples-3.3.4 jar grep ~/input ~/grep_example 'allowed
[.]*' 

```



```

aws Services Search for services, features, blogs, docs, and more [Alt+S]
[Alt+S] Mumbai Sharadindu

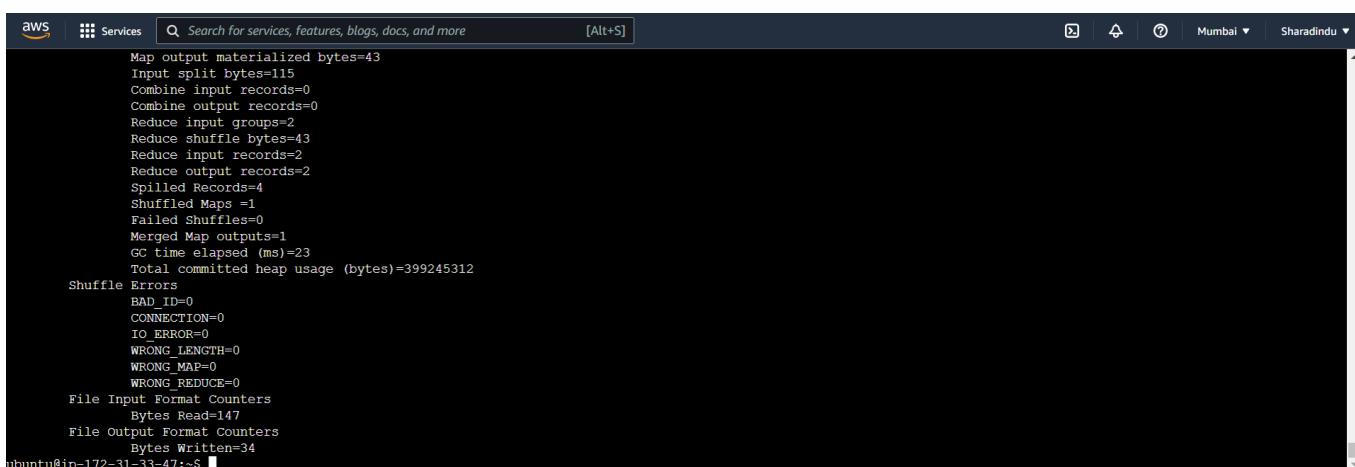
fs          run a generic filesystem user client
gridmix    submit a mix of synthetic job, modeling a profiled from production load
jar <jar>   run a jar file. NOTE: please use "yarn jar" to launch YARN applications, not this command.
jnipath     prints the java.library.path
kdiag       Diagnose Kerberos Problems
kerbname    show auth_to_local principal conversion
key         manage keys via the KeyProvider
rumenfolder scale a rumen input trace
rumentrace  convert logs into a rumen trace
s3guard     manage metadata on S3
trace       view and modify Hadoop tracing settings
version     print the version

Daemon Commands:

kms        run KMS, the Key Management Server
registrydns run the registry DNS server

SUBCOMMAND may print help when invoked w/o parameters or with -h.
ubuntu@ip-172-31-33-47:~$ mkdir ~/input
ubuntu@ip-172-31-33-47:~$ cp /usr/local/hadoop/etc/hadoop/*.xml ~/input
ubuntu@ip-172-31-33-47:~$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce-examples-3.3.4 jar grep ~/input ~/grep_example 'allowed
[.]*' 
JAR does not exist or is not a normal file: /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4
ubuntu@ip-172-31-33-47:~$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce-examples-3.3.4.jar grep ~/input ~/grep_example 'allowed
[.]*' 

```



```

aws Services Search for services, features, blogs, docs, and more [Alt+S]
[Alt+S] Mumbai Sharadindu

Map output materialized bytes=43
Input split bytes=115
Combine input records=0
Combine output records=0
Reduce input groups=2
Reduce shuffle bytes=43
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)=23
Total committed heap usage (bytes)=399245312
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=147
File Output Format Counters
  Bytes Written=34
ubuntu@ip-172-31-33-47:~$ 

```

```

Map output materialized bytes=43
Input split bytes=115
Combine input records=0
Combine output records=0
Reduce input groups=2
Reduce shuffle bytes=43
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)=23
Total committed heap usage (bytes)=399245312
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=147
File Output Format counters
Bytes Written=34
ubuntu@ip-172-31-33-47:~$ cat ~/grep_example/*

```

```

IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=147
File Output Format Counters
Bytes Written=34
ubuntu@ip-172-31-33-47:~$ cat ~/grep_example/*
22 allowed.
1 allowed
ubuntu@ip-172-31-33-47:~$ hdfs version
Command 'hdfs' not found, did you mean:
  command 'hdfls' from deb hdfl4-tools (4.2.15-4)
  command 'hfs' from deb hfstools-tcltk (3.2.6-15build2)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-33-47:~$ hadoop version
hadoop: command not found
ubuntu@ip-172-31-33-47:~$ /usr/local/hadoop/bin/hadoop version
Hadoop 3.3.4
Source code repository https://github.com/apache/hadoop.git -r a585a73c3e02ac62350c136643a5e7f6095a3dbb
Compiled by stevel on 2022-07-29T12:32Z
Compiled with protoc 3.7.1
From source with checksum fb9dd8918a7b8a5b430d61af858f6ec
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.4.jar
ubuntu@ip-172-31-33-47:~$ 

```

```

ubuntu@ip-172-31-33-47:~$ /usr/local/hadoop/bin/hadoop version
Hadoop 3.3.4
Source code repository https://github.com/apache/hadoop.git -r a585a73c3e02ac62350c136643a5e7f6095a3dbb
Compiled by stevel on 2022-07-29T12:32Z
Compiled with protoc 3.7.1
From source with checksum fb9dd8918a7b8a5b430d61af858f6ec
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.4.jar
ubuntu@ip-172-31-33-47:~$ 

```

```

Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-33-47:~$ sudo apt-get install -y software-properties-common wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.2-2ubuntul).
wget set to manually installed.
software-properties-common is already the newest version (0.99.22.3).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
ubuntu@ip-172-31-33-47:~$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-33-47:~$ echo "deb https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/enterprise/deb stable main
ubuntu@ip-172-31-33-47:~$ 

```

```

(1) WhatsApp x | Microsoft Word x | PCC DA-4 - G x | Launch an inst x | Instances | EC x | Connect to inst x | How To Install x | EC2 Instance x | Not getting T x | + 
[Alt+S] 
aws Services Search for services, features, blogs, docs, and more 
Selecting previously unselected package grafana-enterprise. 
(Reading database ... 66096 files and directories currently installed.) 
Preparing to unpack .../grafana-enterprise_9.1.7_amd64.deb ... 
Unpacking grafana-enterprise (9.1.7) ... 
Setting up grafana-enterprise (9.1.7) ... 
Adding system user 'grafana' (UID 114) ... 
Adding new user 'grafana' (UID 114) with group `grafana` ... 
Not creating home directory '/usr/share/grafana'. 
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd 
sudo /bin/systemctl daemon-reload 
sudo /bin/systemctl enable grafana-server 
### You can start grafana-server by executing 
sudo /bin/systemctl start grafana-server 
Scanning processes... 
Scanning linux images... 

Running kernel seems to be up-to-date. 
No services need to be restarted. 
No containers need to be restarted. 
No user sessions are running outdated binaries. 
No VM guests are running outdated hypervisor (qemu) binaries on this host. 
ubuntu@ip-172-31-33-47:~$ 

```

```

(1) WhatsApp x | Microsoft Word x | PCC DA-4 - G x | Launch an inst x | Instances | EC x | Connect to inst x | How To Install x | EC2 Instance x | Not getting T x | + 
[Alt+S] 
aws Services Search for services, features, blogs, docs, and more 
No VM guests are running outdated hypervisor (qemu) binaries on this host. 
ubuntu@ip-172-31-33-47:~$ sudo systemctl daemon-reload 
ubuntu@ip-172-31-33-47:~$ sudo systemctl start grafana-server 
ubuntu@ip-172-31-33-47:~$ sudo systemctl status grafana-server 
● grafana-server.service - Grafana instance 
   Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled) 
     Active: active (running) since Wed 2022-10-05 14:03:49 UTC; 11s ago 
       Docs: http://docs.grafana.org 
   Main PID: 5824 (grafana-server) 
     Tasks: 7 (limit: 1143) 
    Memory: 87.4M 
      CPU: 991ms 
     CGroup: /system.slice/grafana-server.service 
             └─ 5824 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:default.paths.logs=/var/log/grafana-server.log grafana-server[5824]: logger=ticker t=2022-10-05T14:03:52.911264634Z level=info msg=starting first tick=2022-10-05T14:04:00Z 
Oct 05 14:03:52 ip-172-31-33-47 grafana-server[5824]: logger=infra.usagestats.collector t=2022-10-05T14:03:52.980738947Z level=info msg="registering usage stat providers" 
Oct 05 14:03:52 ip-172-31-33-47 grafana-server[5824]: logger=server t=2022-10-05T14:03:52.98116787Z level=info msg="Writing PID file" path=/run/grafana/grafana-server.pid 
Oct 05 14:03:52 ip-172-31-33-47 grafana-server[5824]: logger=provisioning.alerting t=2022-10-05T14:03:52.998535003Z level=info msg="starting to provision alerting" 
Oct 05 14:03:52 ip-172-31-33-47 grafana-server[5824]: logger=provisioning.alerting t=2022-10-05T14:03:52.998722952Z level=info msg="finished to provision alerting" 
Oct 05 14:03:53 ip-172-31-33-47 grafana-server[5824]: logger=http.server t=2022-10-05T14:03:53.003066329Z level=info msg="HTTP Server Listen" address=[::]:3000 protocol=http 
Oct 05 14:03:53 ip-172-31-33-47 grafana-server[5824]: logger=ngalert t=2022-10-05T14:03:53.003317478Z level=info msg="warning cache for startup" 
Oct 05 14:03:53 ip-172-31-33-47 grafana-server[5824]: logger=grafanaStorageLogger t=2022-10-05T14:03:53.022263721Z level=info msg="storage starting" 
Oct 05 14:03:53 ip-172-31-33-47 grafana-server[5824]: logger=report t=2022-10-05T14:03:53.022877686Z level=warn msg="Scheduling and sending of reports disabled, SMTP is not configured" 
Oct 05 14:03:53 ip-172-31-33-47 grafana-server[5824]: logger=ngalert.multiorg.alertmanager t=2022-10-05T14:03:53.03304758Z level=info msg="starting MultiOrg Alertmanager" 
ubuntu@ip-172-31-33-47:~$ 

```

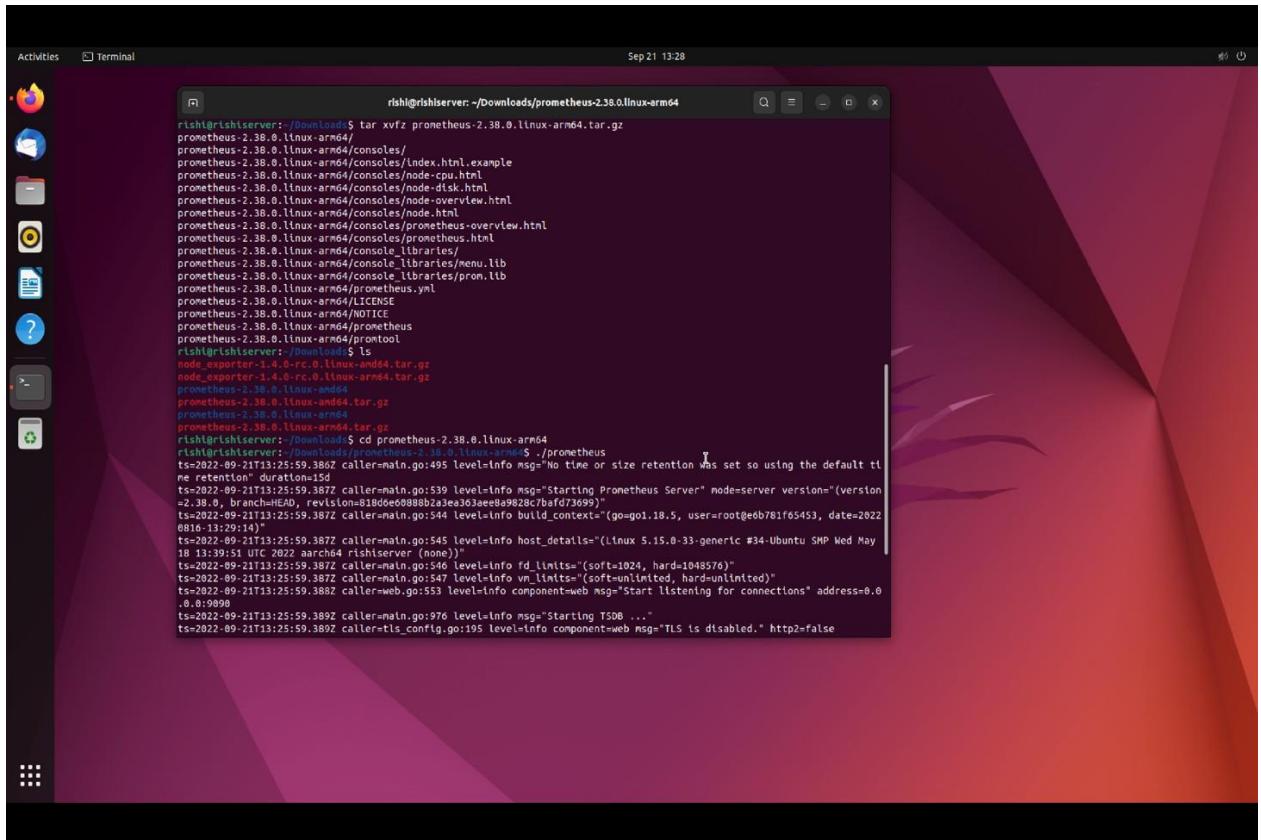


PART 2: Grafana

Steps and Screenshots:

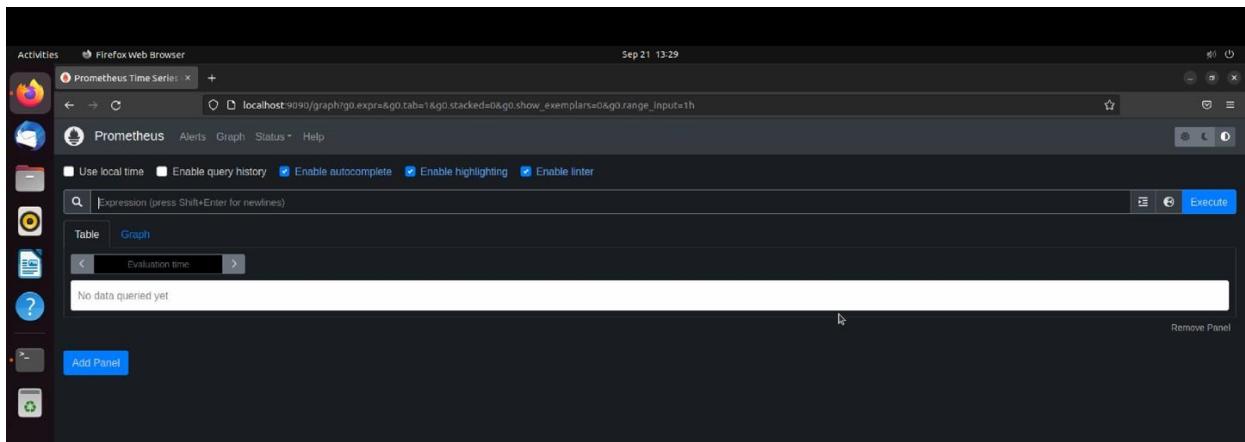
Step 1: Installation of Prometheus

- Goto Download Page of Prometheus and select the prometheus-x.xx.x.linux-amd64.tar.gz file for download.
- Extract the download binary file. (Note- Replace x.xx.x with the downloaded version of Prometheus)
- Go into the extracted directory cd prometheus-x.xx.x.linux-amd64.tar.gz
- Start the prometheus server with the command ./Prometheus

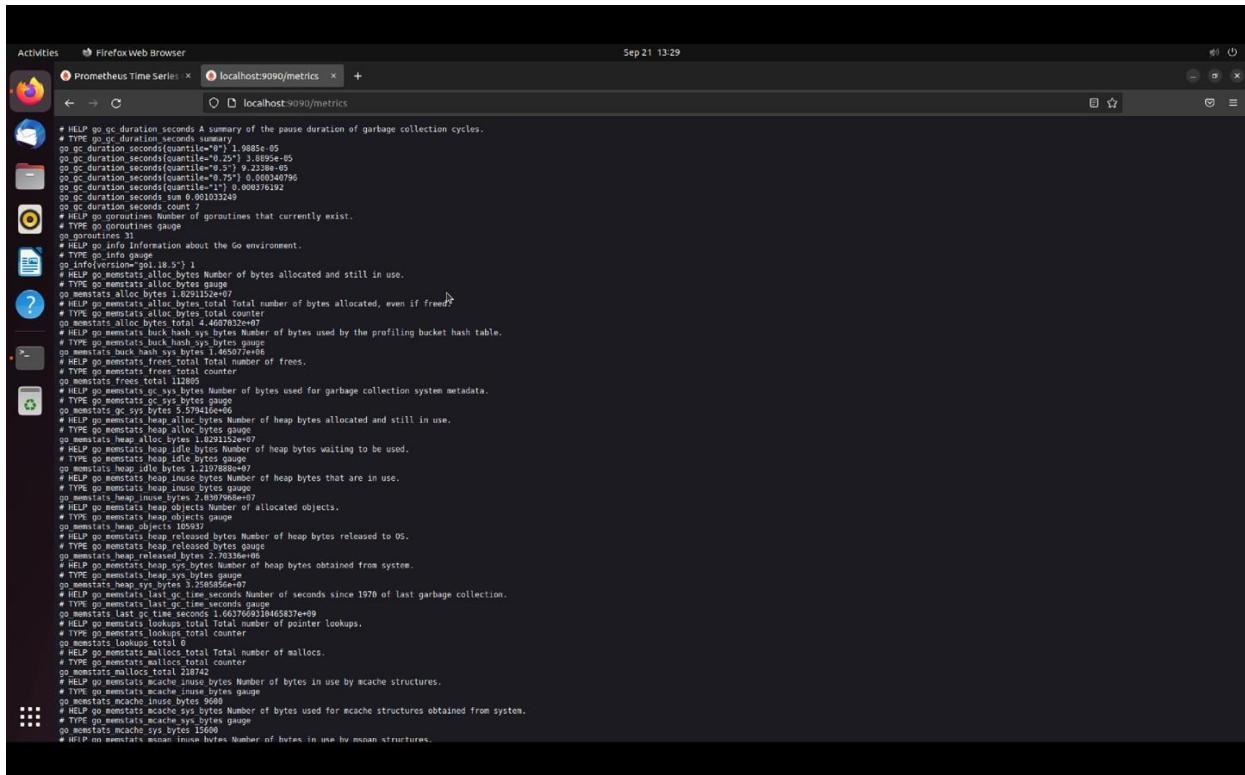


```
rishi@rishiServer:~/Downloads/prometheus-2.38.0.linux-arm64$ tar xvfz prometheus-2.38.0.linux-arm64.tar.gz
prometheus-2.38.0.linux-arm64/
prometheus-2.38.0.linux-arm64/consoles/
prometheus-2.38.0.linux-arm64/consoles/Index.html.example
prometheus-2.38.0.linux-arm64/consoles/node-cpu.html
prometheus-2.38.0.linux-arm64/consoles/node-disk.html
prometheus-2.38.0.linux-arm64/consoles/node-overview.html
prometheus-2.38.0.linux-arm64/consoles/node.html
prometheus-2.38.0.linux-arm64/consoles/prometheus-overview.html
prometheus-2.38.0.linux-arm64/consoles/prometheus.html
prometheus-2.38.0.linux-arm64/console_libraries/
prometheus-2.38.0.linux-arm64/console_libraries/menu.lib
prometheus-2.38.0.linux-arm64/console_libraries/prometheus.yml
prometheus-2.38.0.linux-arm64/LICENSE
prometheus-2.38.0.linux-arm64/NOTICE
prometheus-2.38.0.linux-arm64/prometheus
prometheus-2.38.0.linux-arm64/prometheus-tool
rishi@rishiServer: ~ $ ls
node_exporter-1.4.0-rc.0.linux-amd64.tar.gz
node_exporter-1.4.0-rc.0.linux-arm64.tar.gz
prometheus-2.38.0.linux-amd64.tar.gz
prometheus-2.38.0.linux-arm64.tar.gz
prometheus-2.38.0.linux-arm64.tar.gz
rishi@rishiServer:~/Downloads$ cd prometheus-2.38.0.linux-arm64
rishi@rishiServer:~/Downloads/prometheus-2.38.0.linux-arm64$ ./prometheus
ts=2022-09-21T13:25:59.380Z caller=main.go:495 level=Info msg="No time or size retention was set so using the default time retention duration=15d"
ts=2022-09-21T13:25:59.387Z caller=main.go:539 level=Info msg="Starting Prometheus Server" mode=server version="(version 2.38.0, branch=HEAD, revision=81806e0b9888b2a3ea3633ee8a9828c7bfd73699)"
ts=2022-09-21T13:25:59.387Z caller=main.go:544 level=Info build_context="(go=go1.18.5, user=root@e6b781f65453, date=2022-08-16-13:29:14)"
ts=2022-09-21T13:25:59.387Z caller=main.go:545 level=Info host_details="(Linux 5.15.0-33-generic #34-Ubuntu SMP Wed May 18 13:45:40 UTC 2022 aarch64 rishiServer (none))"
ts=2022-09-21T13:25:59.387Z caller=main.go:546 level=Info fd_limits="(soft=1024, hard=1048576)"
ts=2022-09-21T13:25:59.387Z caller=main.go:547 level=Info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2022-09-21T13:25:59.388Z caller=web.go:553 level=Info component=web msg="Start listening for connections" address=0.0.0.9999
ts=2022-09-21T13:25:59.389Z caller=main.go:976 level=Info msg="Starting TSDB ..."
ts=2022-09-21T13:25:59.389Z caller=tls_config.go:195 level=Info component=web msg="TLS is disabled." http=false
```

- We can access the Prometheus graph UI by visiting <http://localhost:9090/graph>

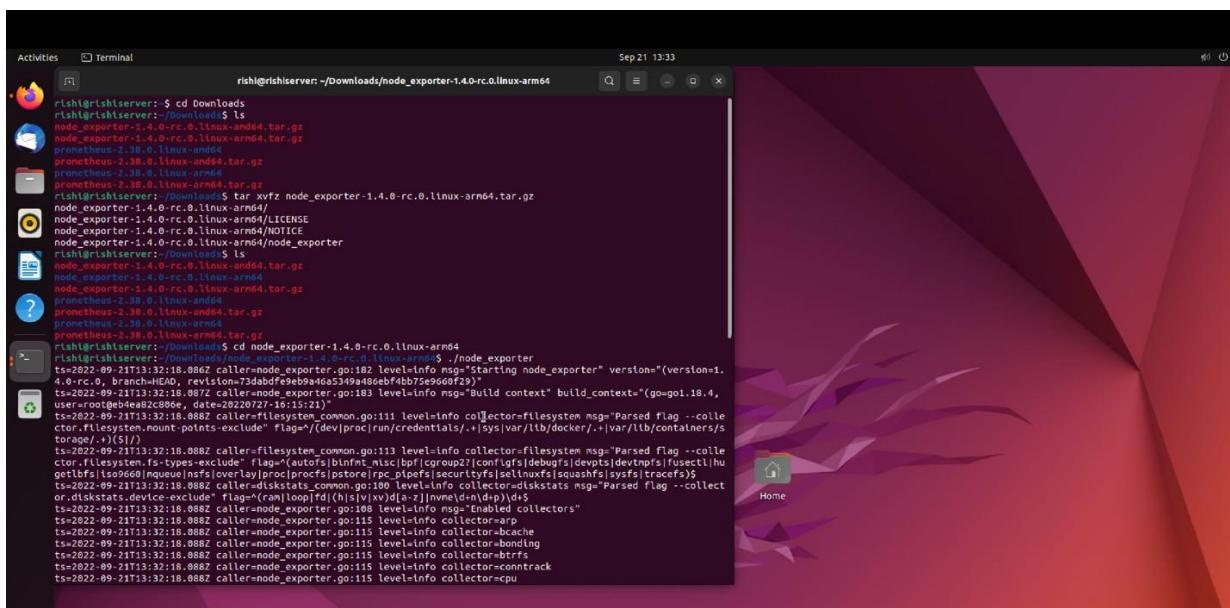


- We can access the Prometheus metrics UI by visiting <http://localhost:9090/metrics>

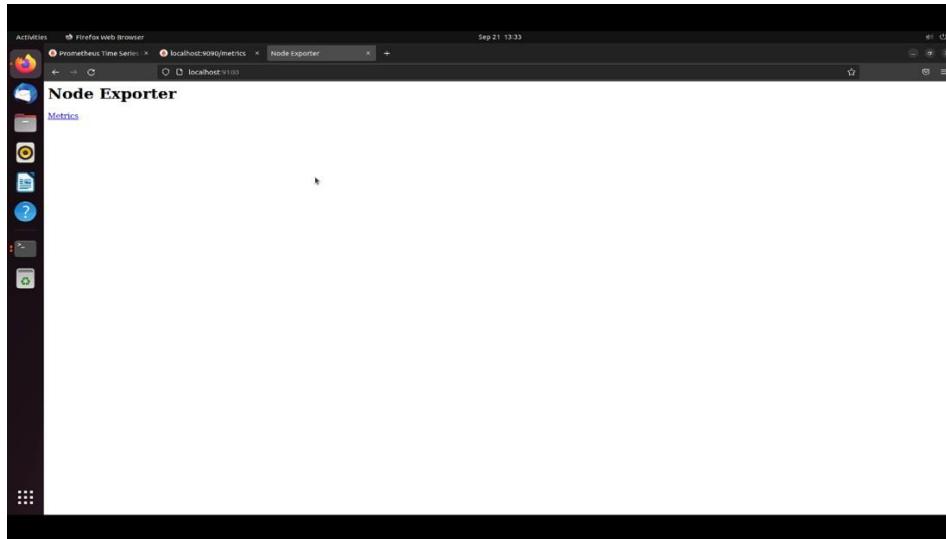


Step 2: Install Node exporter

- Download the binary of Node exporter based on the operating system.(In my case I am using Ubuntu linux machine)
 - Extract the download node exporter binary file. (Note- Replace x.xx.x with the downloaded version of node exporter)
 - Go into the extracted directory cd xvzf node_exporter-*.*-amd64.tar.gz
 - Start the node exported with the command ./node_exporter



- Access the Node exporter metrics on the browser with URL - <http://localhost:9100>



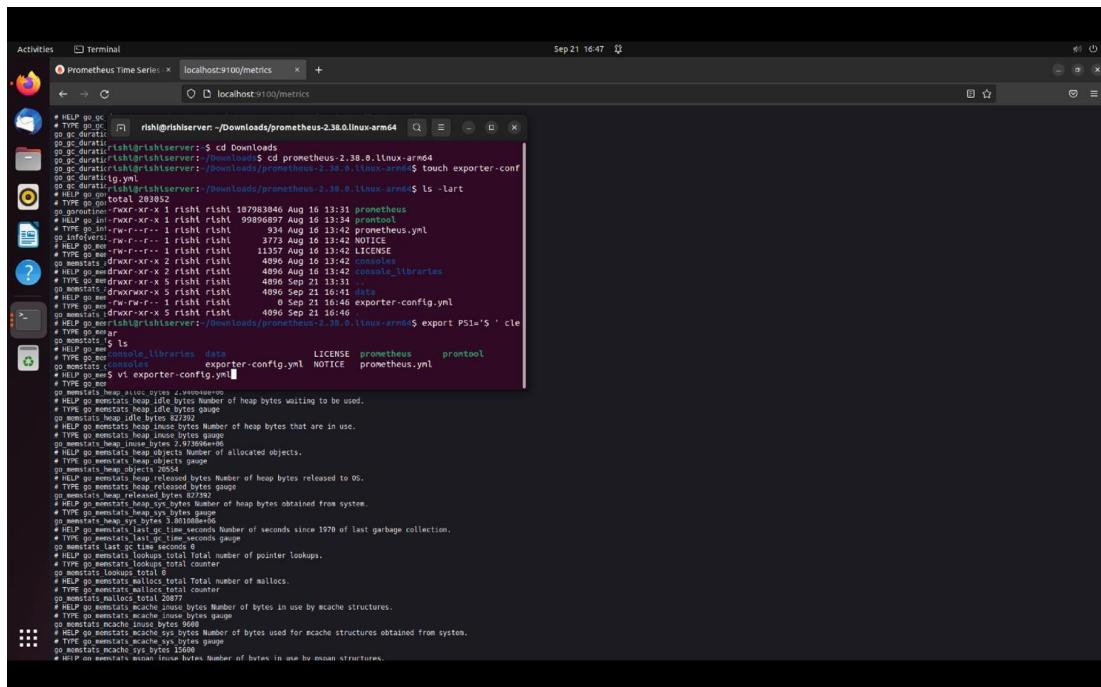
Step 3: Adding node exporter scrap_configs to prometheus as a YAML Configuration

- In the previous node export installation, we had Prometheus and node exporter running on the same server. But if you want to add node exporter for any remote server then you need to define scrap_configs with target hosts inside the YAML configuration.
 - Here is an example of scrap_configs –
- ```

global:
 scrape_interval: 15s

scrape_configs:
 - job_name: node
 static_configs:
 - targets: ['localhost:9100', '192.168.64.2:9100']

```
- Save the above configuration as YAML(ex. exporter.yaml)
  - Start the Prometheus server bypassing scrap\_config with the help of --config.file flag



- After adding the remote host into the node exporter configuration verify it by accessing the prometheus target URL localhost:9090/targets

## Step 5: Installing the Grafana

- In this installation, we are going to install the latest enterprise edition Grafana.
- Update the package info –

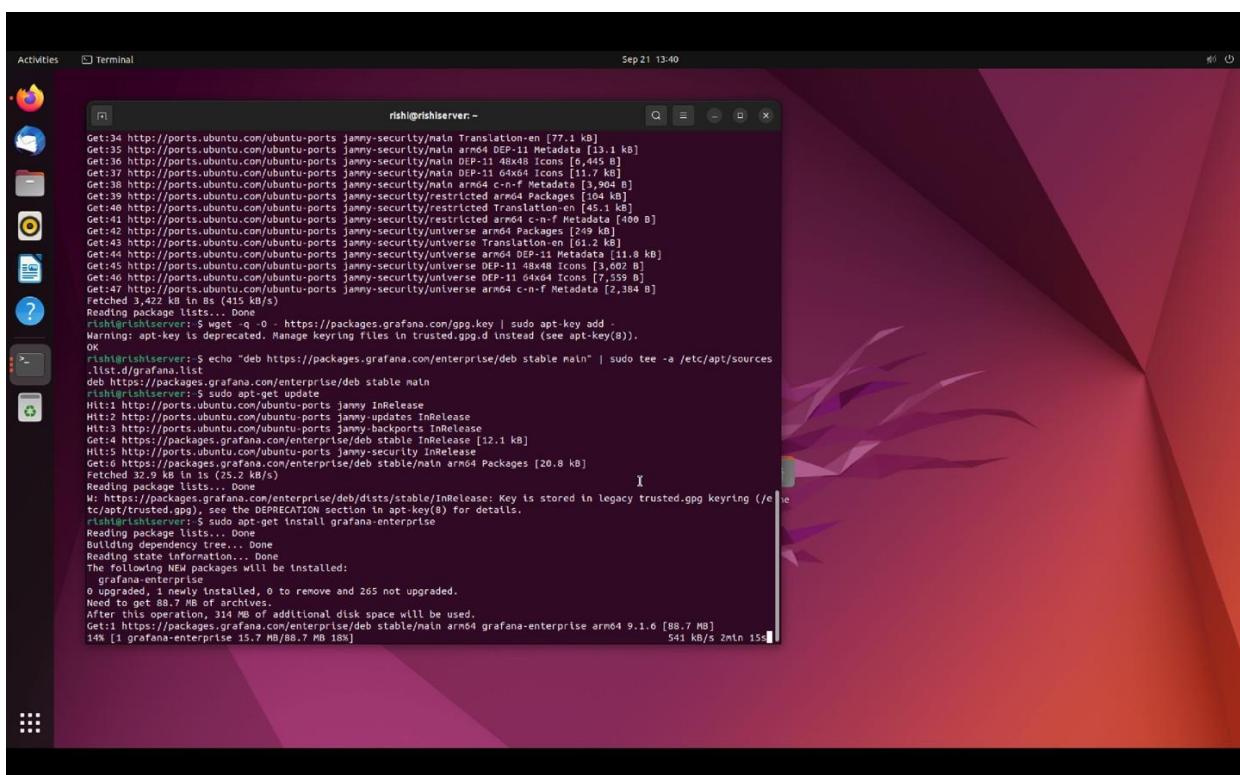
```
sudo apt-get install -y apt-transport-https
sudo apt-get install -y software-properties-common wget
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

- Add stable repository of Grafana –

```
echo "deb https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

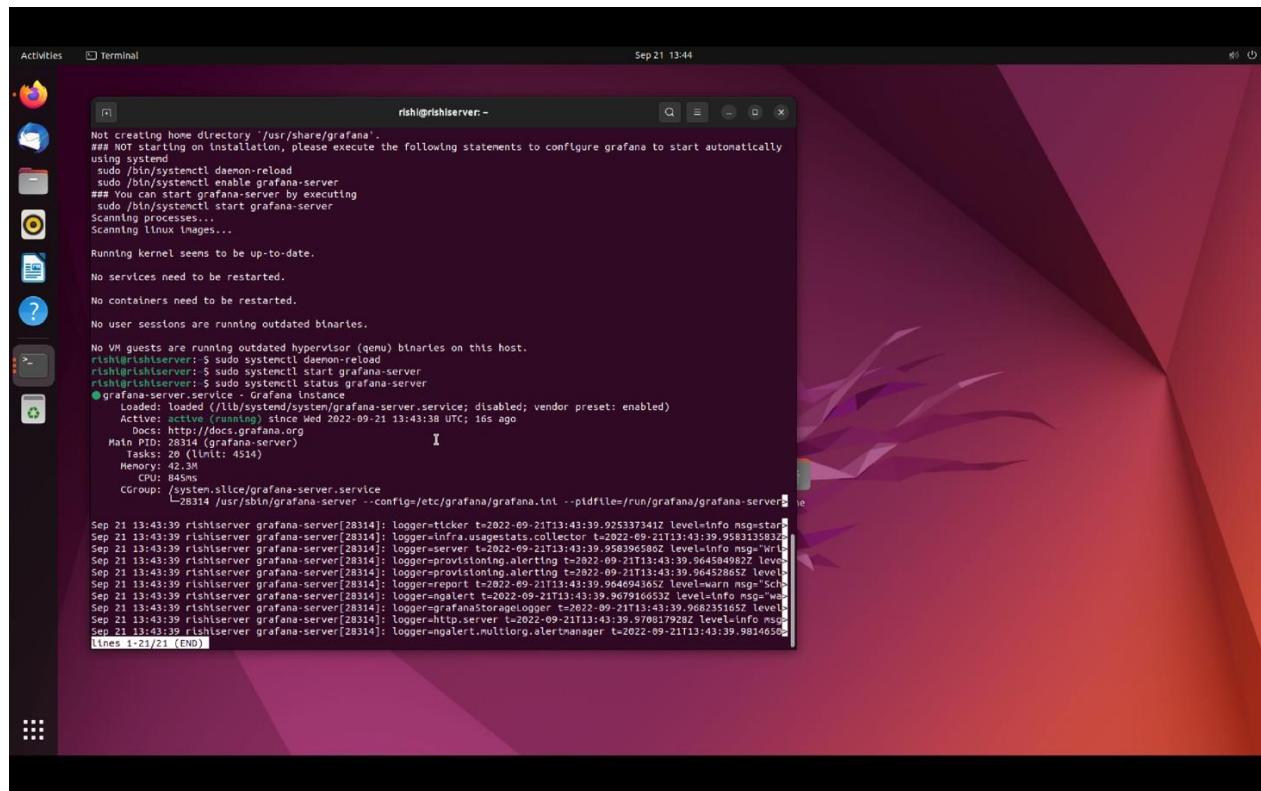
- Update repository and Install Grafana

```
sudo apt-get update
sudo apt-get install grafana-enterprise
```

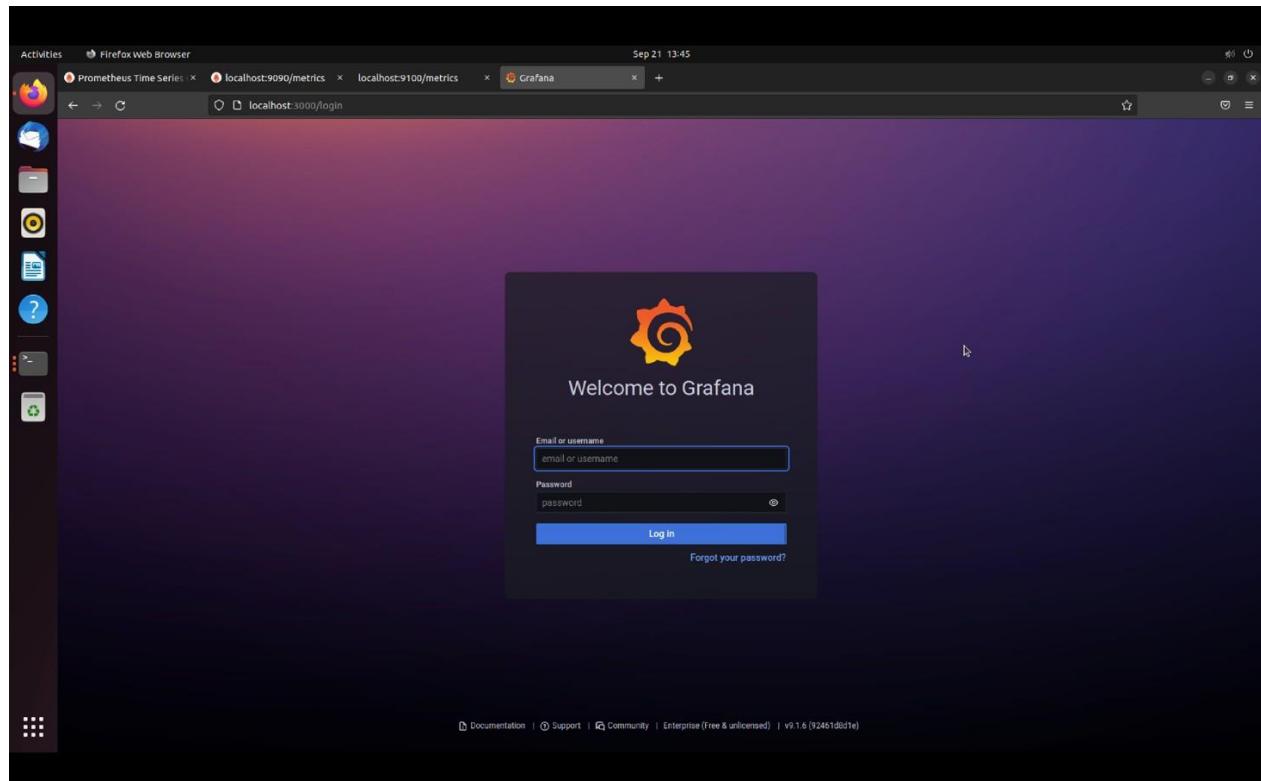


- Start the Grafana Server –

```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl status grafana-
server
```

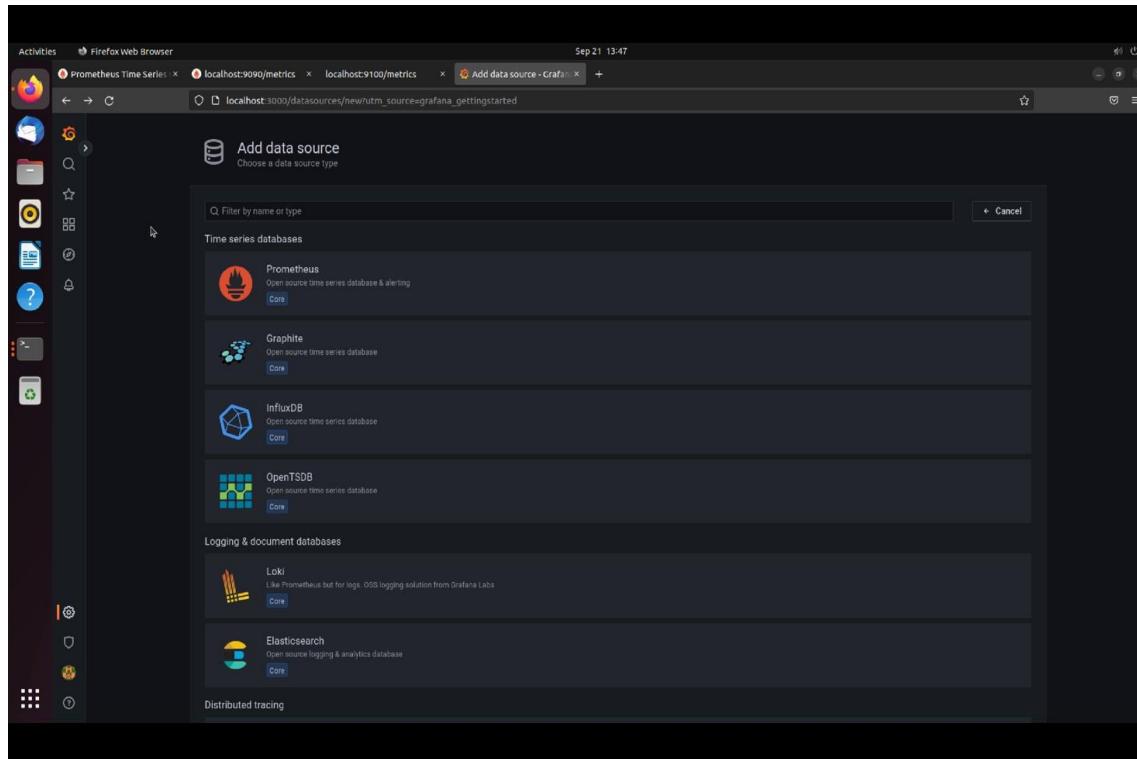


- Access the Grafana Dashboard using <http://localhost:3000/login>

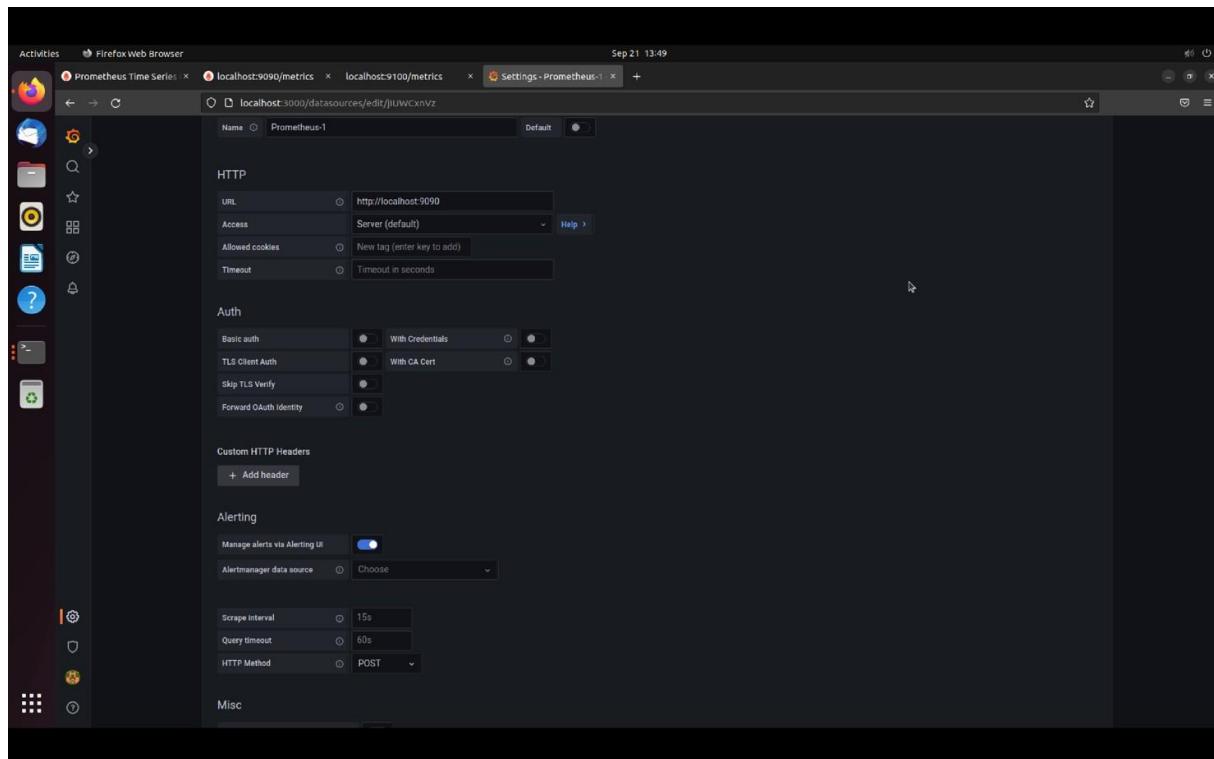


## Step 6: Setting up Grafana Dashboard

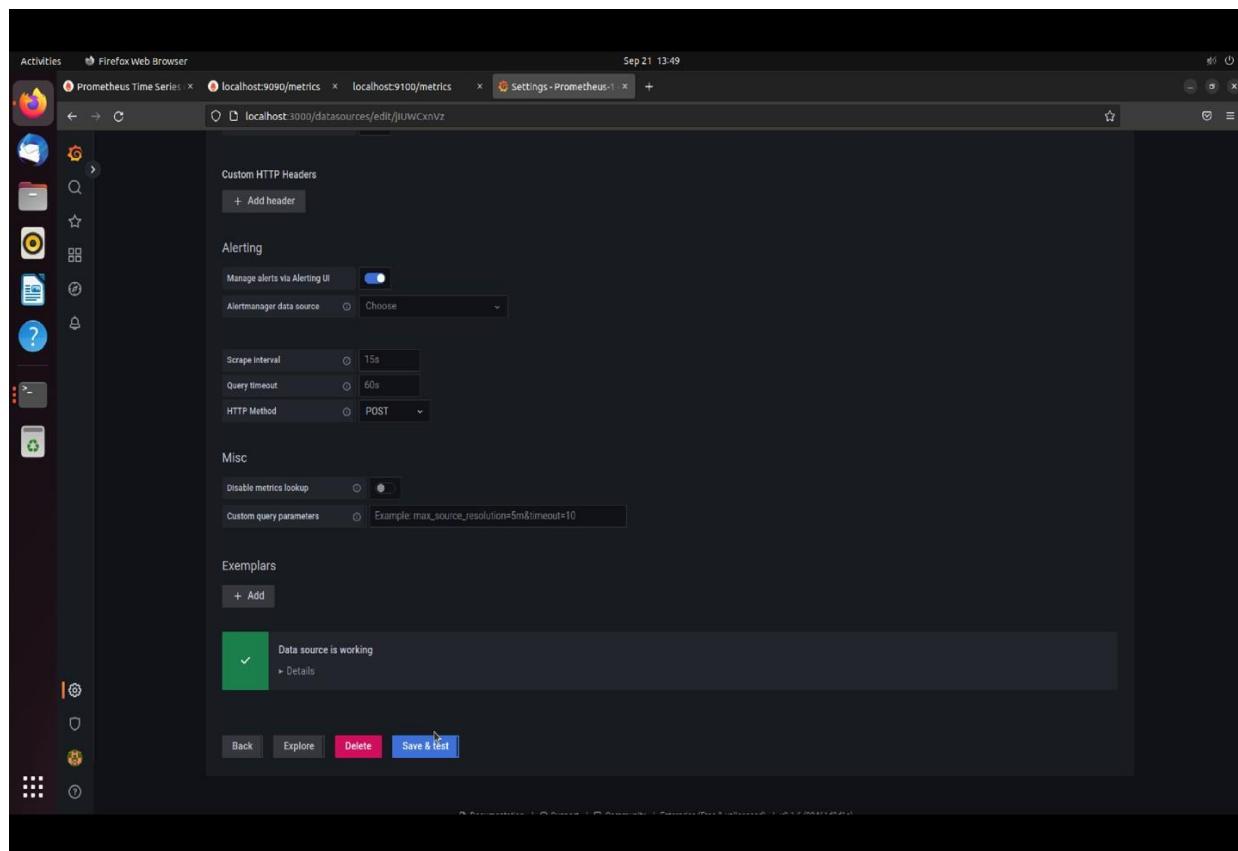
- Goto your Grafana Dashboard and click on Gear(Settings Icon)->Data Sources
- Select the Prometheus as preferred data source –



- Enter the hostname or IP address of the prometheus server

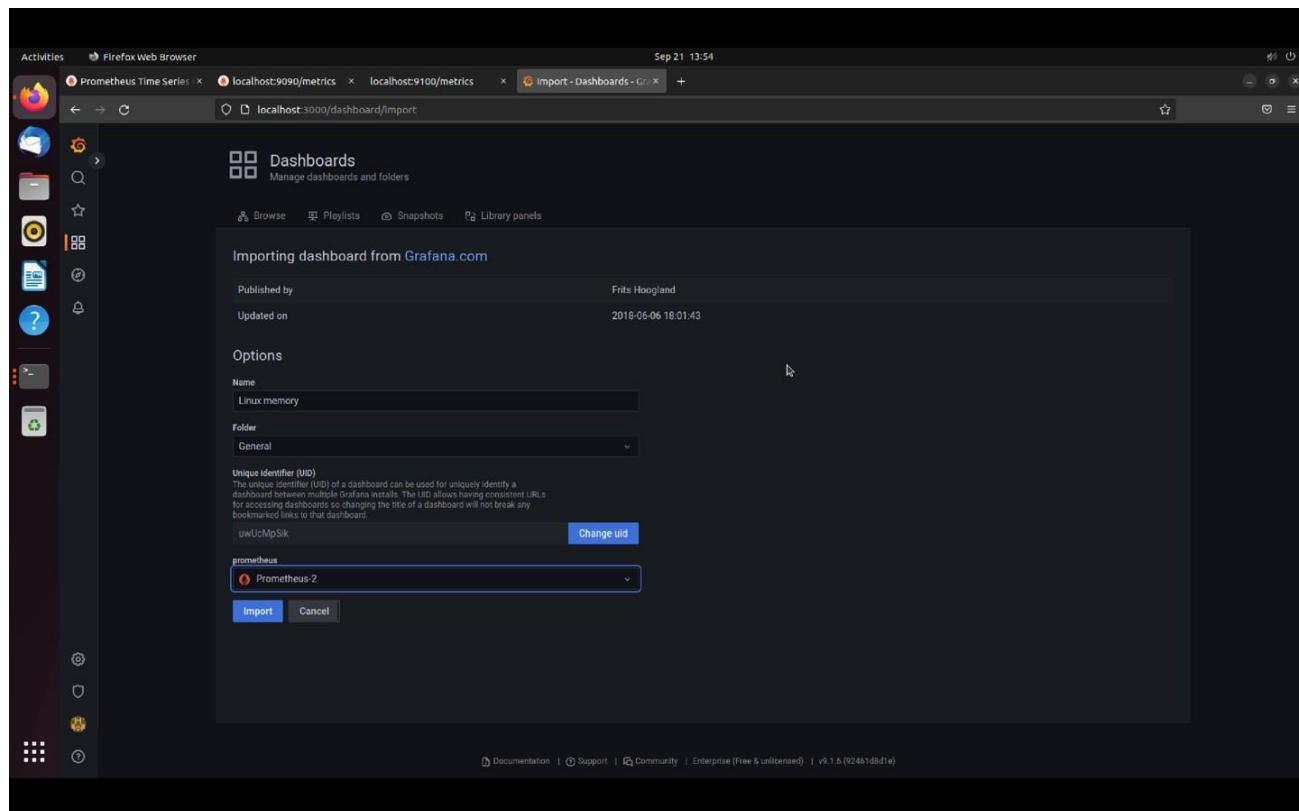


- Save and test data source

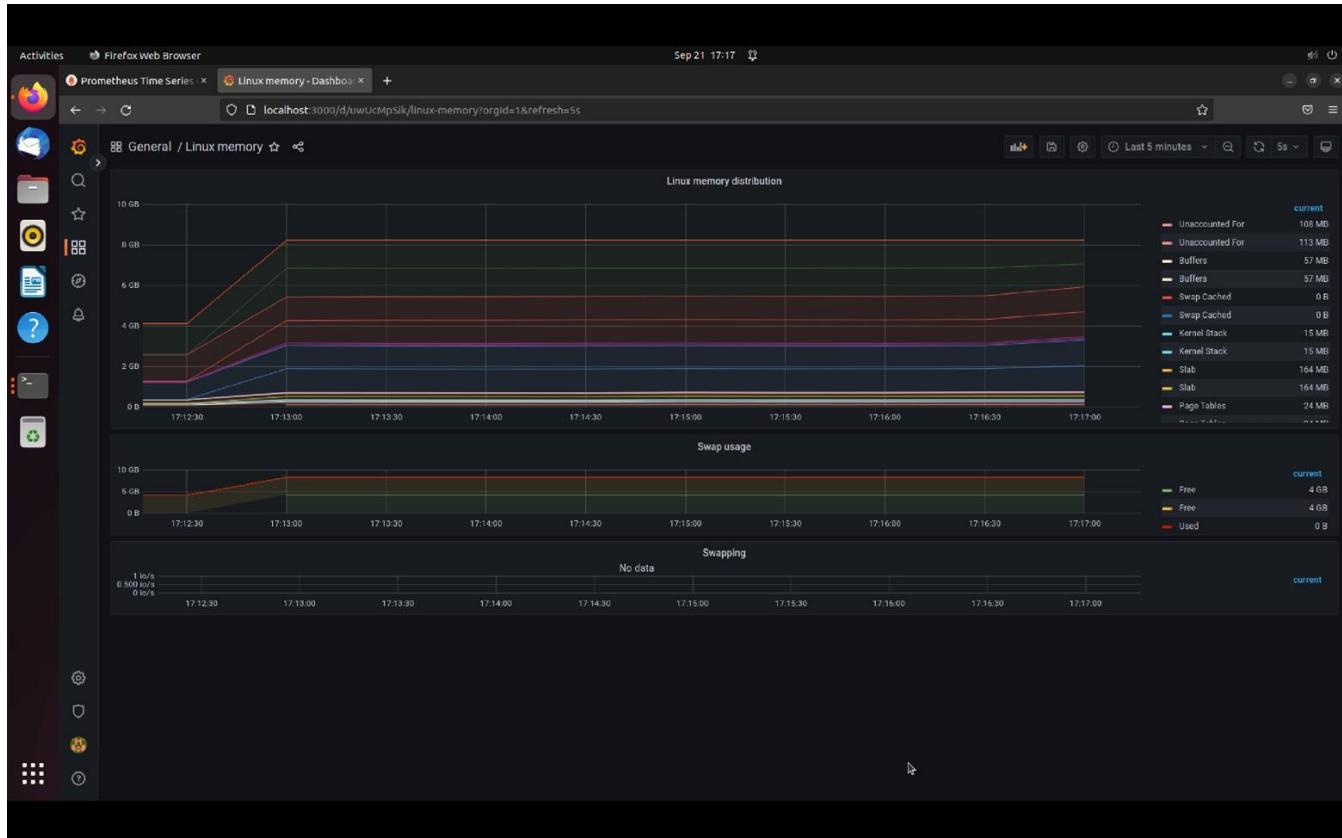


## Step 7: Import Grafana Dashboard from Grafana Labs

- Goto Grafana Dashboard and search for Linux Memory
- Click on the result and then copy the board ID .e.g 2747
- Goto Import and enter the board ID .e.g. 2747



- Click load and then select the data source as Prometheus
- Finally Click on Import
- You should be able to see the memory dashboard



## Conclusion:

First, I've installed the Hadoop framework and created an application using the MapReduce programming model. Then used Prometheus and Grafana to monitor, visualize and analyse the performance of resource utilization.