intelligaia

Prometheus-Grafana Installation and Setup

By
Ashutosh Gupta
DevOps Engineer
DevOps/Networking Department
Intelligaia Technologies

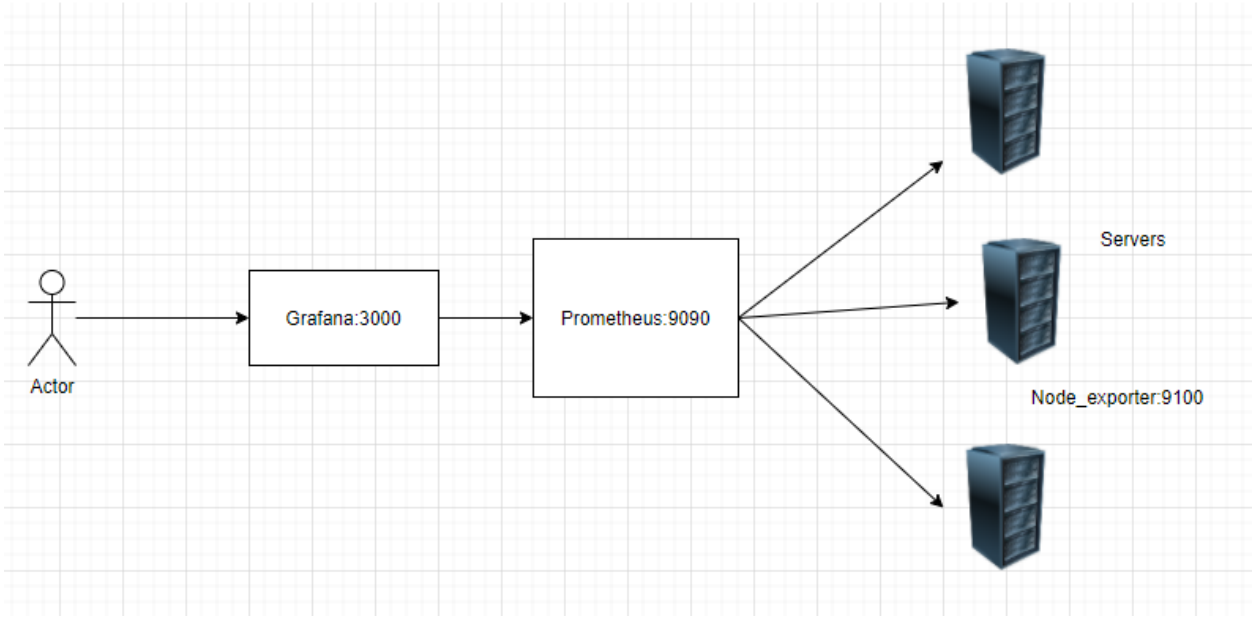
	Author	Version	Change Reference
23/08/2023	Ashutosh Gupta	1.0	Creation of document – Draft

Table Of contents

1. Introduction:

Prometheus

- prometheus is time series metrics based monitoring tools
- node exporter is the agent of the prometheus which provides the system state/cpu/ram/disk
- Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company. To emphasize this, and to clarify the project's governance structure, Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.
- Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.



Features

Prometheus's main features are:

- a multi-dimensional data model with time series data identified by metric name and key/value pairs
- PromQL, a flexible query language to leverage this dimensionality
- no reliance on distributed storage; single server nodes are autonomous
- time series collection happens via a pull model over HTTP
- pushing time series is supported via an intermediary gateway
- targets are discovered via service discovery or static configuration
- multiple modes of graphing and dashboarding support

Grafana

- The open-source platform for monitoring and observability
- Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share dashboards with your team and foster a data-driven culture:
- Visualizations: Fast and flexible client side graphs with a multitude of options. Panel plugins offer many different ways to visualize metrics and logs.
- Dynamic Dashboards: Create dynamic & reusable dashboards with template variables that appear as dropdowns at the top of the dashboard.
- Explore Metrics: Explore your data through ad-hoc queries and dynamic drilldown. Split view and compare different time ranges, queries and data sources side by side.
- Explore Logs: Experience the magic of switching from metrics to logs with preserved label filters. Quickly search through all your logs or streaming them live.
- Alerting: Visually define alert rules for your most important metrics. Grafana will continuously evaluate and send notifications to systems like Slack, PagerDuty, VictorOps, OpsGenie.
- Mixed Data Sources: Mix different data sources in the same graph! You can specify a data source on a per-query basis. This works for even custom datasources.

2. Prerequisites:

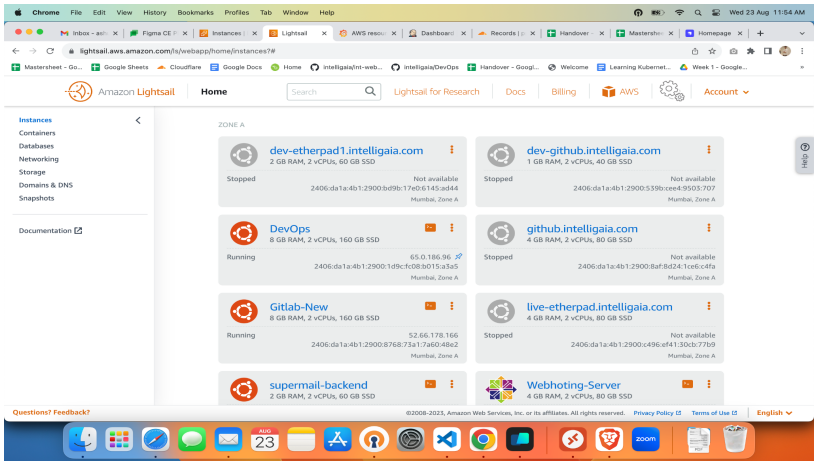
- Linux Server (Ubuntu 20.04 LTS)

  We have create lightsail instance with the name DevOps in Mumbai regions with configuration

  RAM 8G

  2vCPUs

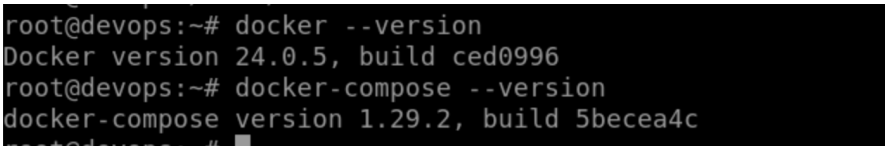
  160GB SSD storage



- Here we have selected Ubuntu 20.04 LTS

```
root@devops:/home/ubuntu# cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
```

- Docker and Docker-compose installed on ubuntu 20.04



We can use below link to install docker and docker-compose on ubuntu

```
Unset
https://docs.docker.com/desktop/install/windows-install/

https://docs.docker.com/compose/install/
```

- you will need prometheus , grafana and node_exporter
- prometheus runs behind the grafana, grafana is used for visualization and prometheus is the data collector which uses node_exporter agent
- We need to install node_exporter on each and every node which needs to be monitored.
- For installing node_exporter we can use below link

```
Unset
https://prometheus.io/docs/guides/node-exporter/
```

3. Installation

- We have installed prometheus and grafana using docker-compose
- We have created directory inside **/opt** which is **/opt/prom-graf** then create separate folder for prometheus and grafana
- We have created **prom-grafa-compose.yaml** inside **/opt/prom-graf**
- cat **prom-grafa-compose.yaml**

Unset

```
version: '3.8'
services:
 prometheus:
  image: prom/prometheus:v2.28.0
  restart: unless-stopped
  container_name: prometheus
  volumes:
   - ./prometheus.yml:/etc/prometheus/prometheus.yml
   - /opt/prom-graf/prometheus-grafana-nfs/prometheus:/prometheus
  ports:
   - 9090:9090
  command: ["--config.file=/etc/prometheus/prometheus.yml",
"--storage.tsdb.path=/prometheus",
"--web.console.libraries=/etc/prometheus/console_libraries",
"--web.console.templates=/etc/prometheus/consoles"]

 grafana:
  image: grafana/grafana:8.2.3
  container_name: grafana
  restart: unless-stopped
  user: "0"
  environment:
   TZ: "Asia/Kolkata"
   GF_INSTALL_PLUGINS:
"grafana-clock-panel,grafana-simple-json-datasource,simpod-json-datasource,marc
usolsson-json-datasource"
  volumes:
   - ./defaults.ini:/etc/grafana/grafana.ini
   - ./ldap.toml:/usr/share/grafana/conf/ldap.toml
   - /opt/prom-graf/prometheus-grafana-nfs/grafana:/var/lib/grafana
  ports:
   - 3000:3000

 blackbox:
```

```
  image: prom/blackbox-exporter:v0.15.1
  privileged: true
  restart: unless-stopped
  volumes:
   - ./blackbox.yml:/etc/blackboxexporter/config.yml
  ports:
   - '9115:9115'
  command: ["--config.file=/etc/blackboxexporter/config.yml"]

 pushgateway:
  image: prom/pushgateway
  container_name: pushgateway
  restart: unless-stopped
  ports:
   - "9091:9091"
  command:
   - '--web.listen-address=:9091'
   - '--push.disable-consistency-check'
   - '--persistence.interval=5m'
```

- We need to create **prometheus.yml** file inside the **/opt/prom-graf**

```
Unset
global:
 scrape_interval:   15s
 evaluation_interval: 15s

scrape_configs:
   - job_name: 'local-prometheus'
     static_configs:
     - targets: ['localhost:9090']

   - job_name: node
     static_configs:
     - targets: ['localhost:9100']

   - job_name: 'lp-openvpn-1'
     static_configs:
     - targets: ['43.204.0.79:9100']


   - job_name: 'dev-etherpad.intelligaia.com'
     static_configs:
```

```
  - targets: ['3.6.9.108:9100']

- job_name: 'dev-github.intelligaia.com'
  static_configs:
  - targets: ['13.127.115.120:9100']

- job_name: 'exportcomments.io-1'
  static_configs:
  - targets: ['15.206.99.140:9100']

- job_name: 'github.intelligaia.com-1'
  static_configs:
  - targets: ['13.233.13.224:9100']

- job_name: 'live-etherpad.intelligaia.com(New)'
  static_configs:
  - targets: ['3.109.128.148:9100']

- job_name: 'jenkins-worker-1'
  static_configs:
  - targets: ['43.205.232.119:9100']

- job_name: 'neeraj-sir-1'
  static_configs:
  - targets: ['43.205.43.56:9100']

- job_name: 'Spacetantra'
  static_configs:
  - targets: ['3.109.228.117:9100']

- job_name: 'ap-haproxy-1'
  static_configs:
  - targets: ['3.111.190.147:9100']

- job_name: 'dev-scrumboard.intelligaia.com'
  static_configs:
  - targets: ['3.111.82.180:9100']

- job_name: 'live-scrumboard.intelligaia.com'
  static_configs:
  - targets: ['65.0.166.68:9100']

- job_name: 'tuthastu.com'
  static_configs:
  - targets: ['3.111.66.77:9100']
```

```
- job_name: 'teza.tuthastu.com'
 static_configs:
 - targets: ['13.234.141.192:9100']

- job_name: 'intelligaia.com'
 static_configs:
 - targets: ['13.126.72.99:9100']

- job_name: 'dev.exportcomments.io'
 static_configs:
 - targets: ['43.204.53.156:9100']

- job_name: 'Littlesketchers.com'
 static_configs:
 - targets: ['13.235.90.41:9100']

- job_name: 'dev.aclis.io'
 static_configs:
 - targets: ['172.31.40.33:9100']

- job_name: 'aclis.io'
 static_configs:
 - targets: ['18.197.159.84:9100']

- job_name: 'calendar.intelligaia.com'
 static_configs:
 - targets: ['43.205.28.106:9100']

- job_name: 'dev-calendar.intelligaia.com'
 static_configs:
 - targets: ['15.207.205.179:9100']

- job_name: 'colorshift.intelligaia.com'
 static_configs:
 - targets: ['43.204.89.182:9100']

# - job_name: 'backend-api.intelligaia.com'
#  static_configs:
#  - targets: ['172.31.5.164:9100']

# - job_name: 'superemail.intelligaia.com'
#  static_configs:
#  - targets: ['172.31.11.39:9100']


#- job_name: 'test.intelligaia.com'
```

```
# static_configs:
# - targets: ['172.31.26.213:9100']

- job_name: 'pushgateway'
 honor_labels: true
 static_configs:
  - targets: ['172.26.2.229:9091']

- job_name: 'blackbox'
 metrics_path: /probe
 params:
  module: [http_2xx] # Look for a HTTP 200 response.
 static_configs:
  - targets:
   - https://intelligaia.com
   - https://exportcomments.io
   - https://dev.exportcomments.io
   - https://dev-etherpad.intelligaia.com
   - https://grafana.intelligaia.com
   - https://jenkins.intelligaia.com/login?from=%2F
   - https://sonarqube.intelligaia.com
   - https://spacetantra.com
   - https://littlesketchers.com
   - https://figma.com
   - https://infina.projectsjunction.com
   - https://dev-github.intelligaia.com
   - https://github.intelligaia.com/
   - https://live-etherpad.intelligaia.com/
   - https://payroll.intelligaia.com/
   - https://dev-scrumboard.intelligaia.com
   - https://scrumboard.intelligaia.com
   - https://tuthastu.com
   - https://admin.tuthastu.com
   - https://teza.tuthastu.com
   - https://teza-admin.tuthastu.com
   - https://greendotexpeditions.com
   - https://gennextsmartschool.com
   - https://projecthost.us
   - https://baljitphotography.com
   - https://littlesketchers.com
   - https://bhumivardaan.com
   - https://cds.ind.in
   - https://intelligaia.com
   - https://rajivkaul.com
```

```
    - https://cheenakaul.com
    - https://roopinder.com
    - https://gianigurditsingh.com
    - https://feedback.intelligaia.com
    - https://admin.intelligaia.com
    - https://sendgrid.intelligaia.com
    - https://api.exportcomments.io
    - https://admin.spacetantra.com
    - https://dev-api.exportcomments.io
    - https://intelligaia.com
    - https://dev.aclis.io
    - https://aclis.io
    - https://dev-etherpad-api.intelligaia.com
    - https://ebooks.intelligaia.com
    - https://hpe-grommet.intelligaia.com
    - https://zoom-freshsales.intelligaia.com
    - https://live-etherpad-api.intelligaia.com
    - https://api-scrumboard.intelligaia.com
    - https://timer-scrumboard.intelligaia.com
    - https://api-dev-scrumboard.intelligaia.com
    - https://timer-dev-scrumboard.intelligaia.com
    - https://colorshift.intelligaia.com
    - https://dev-calendar.intelligaia.com
    - https://calendar.intelligaia.com
    - https://backend-api.intelligaia.com
    - https://superemail.intelligaia.com
    - https://superemail-api-v1.intelligaia.com

  relabel_configs:
   - source_labels: [__address__]
     target_label: __param_target
   - source_labels: [__param_target]
     target_label: instance
   - target_label: __address__
     replacement: 172.26.2.229:9115
```

- With prometheus.yml we need to create blackbox.yml
- Cat blackbox.yml

Unset

```
modules:
 http_2xx:
  prober: http
  timeout: 5s
  http:
   method: GET
```

- We need to create and mount all the files and folders listed in docker-compose file
- For Grafana we need to create defaults.ini with few configurational changes and ldap.toml which has active directory configuration

- Cat ldap.toml

Unset

```
# To troubleshoot and get more log info enable ldap debug logging in grafana.ini
# [log]
#filters = ldap:debug

[[servers]]
# Ldap server host (specify multiple hosts space separated)
host = "dc1.ad.intelligaia.com"
# Default port is 389 or 636 if use_ssl = true
port = 389
# Set to true if LDAP server should use an encrypted TLS connection (either with
STARTTLS or LDAPS)
use_ssl = false
# If set to true, use LDAP with STARTTLS instead of LDAPS
start_tls = false
# set to true if you want to skip ssl cert validation
ssl_skip_verify = false
# set to the path to your root CA certificate or leave unset to use system defaults
# root_ca_cert = "/path/to/certificate.crt"
# Authentication against LDAP servers requiring client certificates
# client_cert = "/path/to/client.crt"
# client_key = "/path/to/client.key"

# Search user bind dn
#bind_dn = "cn=admin,dc=grafana,dc=org"
bind_dn = "CN=grafana-svc,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
#bind_dn = "grafana-svc@ad.intelligaia.com"
```

```
# Search user bind password
# If the password contains # or ; you have to wrap it with triple quotes. Ex
"""#password;"""
bind_password = 'vibgyor@158'

# User search filter, for example "(cn=%s)" or "(sAMAccountName=%s)" or "(uid=%s)"
#search_filter = "(cn=%s)"
search_filter = "(sAMAccountName=%s)"

# An array of base dns to search through
#search_base_dns = ["dc=grafana,dc=org"]
search_base_dns = ["DC=ad,DC=intelligaia,DC=com"]

## For Posix or LDAP setups that does not support member_of attribute you can define the
below settings
## Please check grafana LDAP docs for examples
# group_search_filter = "(&(objectClass=posixGroup)(memberUid=%s))"
# group_search_base_dns = ["ou=groups,dc=grafana,dc=org"]
group_search_base_dns = ["OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"]
# group_search_filter_user_attribute = "uid"

# Specify names of the ldap attributes your ldap uses
[servers.attributes]
#name = "Name"
#surname = "Surname"
#username = "sAMAccountName"
#member_of = "MemberOf"
#email =  "EmailAddress"

name = "givenName"
surname = "sn"
username = "sAMAccountName"
member_of = "memberOf"
email =  "mail"

# Map ldap groups to grafana org roles
[[servers.group_mappings]]
#group_dn = "cn=admins,ou=groups,dc=grafana,dc=org"
group_dn = "CN=grafana-admin,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Admin"
# To make user an instance admin  (Grafana Admin) uncomment line below
#grafana_admin = true
# The Grafana organization database id, optional, if left out the default org (id 1)
will be used
```

```
#org_id = 1

[[servers.group_mappings]]
#group_dn = "cn=users,ou=groups,dc=grafana,dc=org"
group_dn = "CN=grafana-editor,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Editor"

[[servers.group_mappings]]
# If you want to match all (or no ldap groups) then you can use wildcard
group_dn = "CN=grafana-viewer,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Viewer"
```

- You need to add necessary parameters such as

```
host = "dc1.ad.intelligaia.com"
port = 389
bind_dn = "CN=grafana-svc,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
bind_password = 'vibgyor@158'
search_filter = "(sAMAccountName=%s)"
search_base_dns = ["DC=ad,DC=intelligaia,DC=com"]
[servers.attributes]
name = "givenName"
surname = "sn"
username = "sAMAccountName"
member_of = "memberOf"
email =  "mail"
group_dn = "CN=grafana-admin,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Admin"

group_dn = "CN=grafana-editor,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Editor"

group_dn = "CN=grafana-viewer,OU=ServiceAccounts,DC=ad,DC=intelligaia,DC=com"
org_role = "Viewer"
```

- Once you’re ready with all the mount directory and files then we can run below command to begin setup

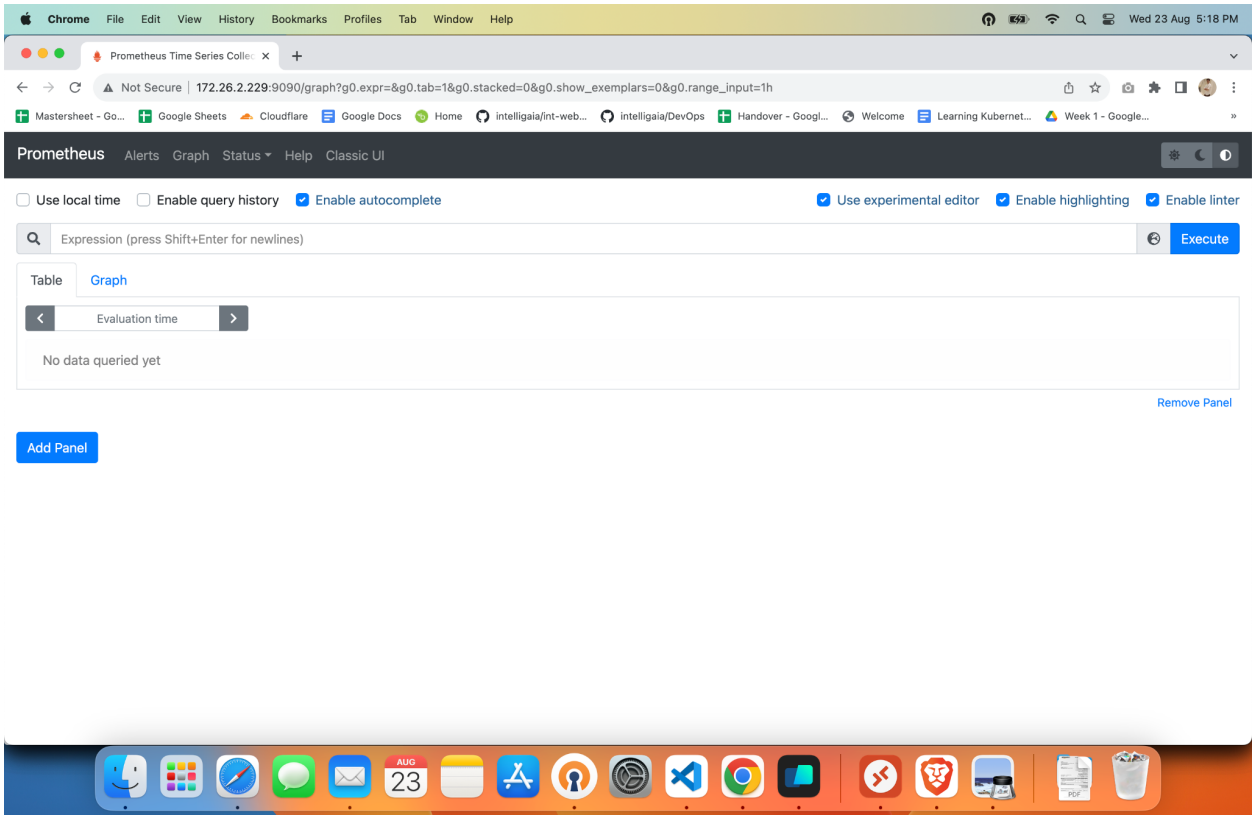
```
Unset
docker-compose -f prom-grafa-compose.yaml up -d
```

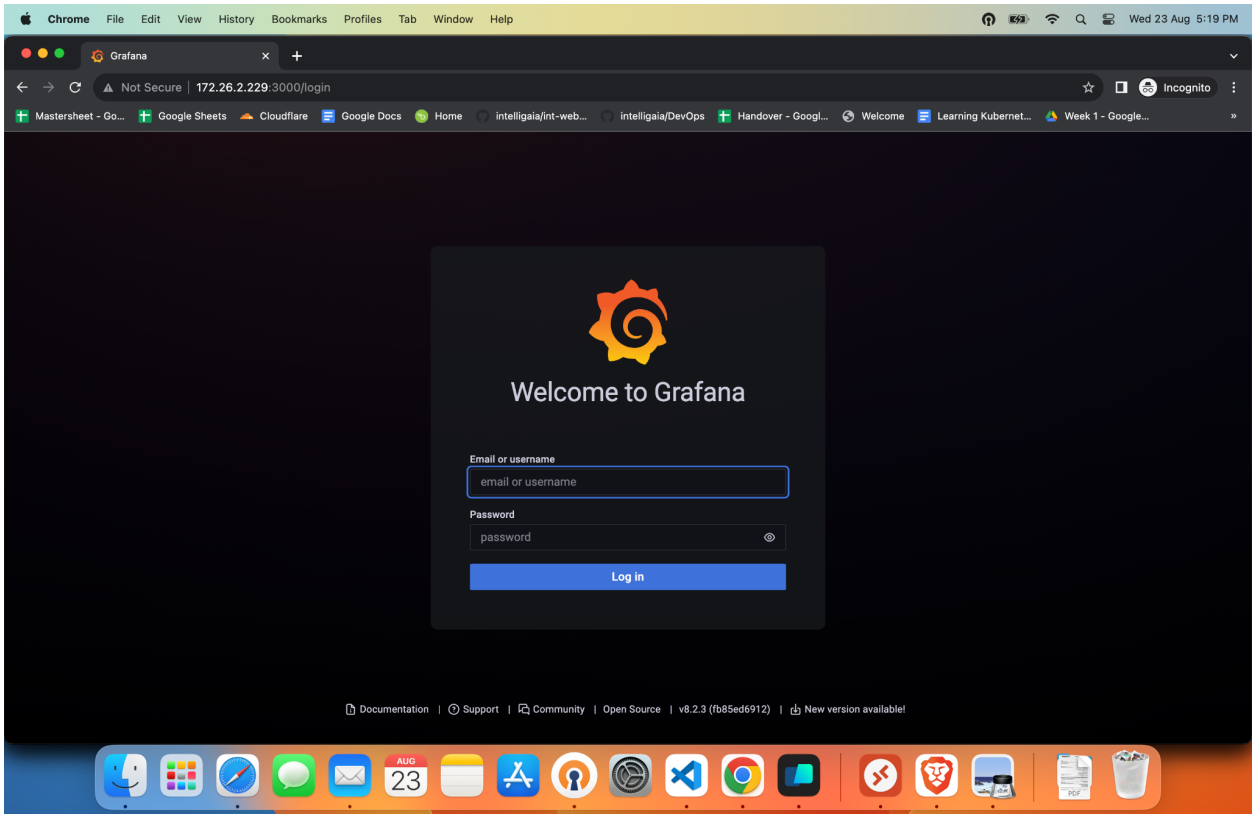
You can use docker ps to check all the containers are in starting state

```
CONTAINER ID   IMAGE                          COMMAND                  CREATED      STATUS      PORTS
                                                                                                 NAMES
971b578e873d   prom/prometheus:v2.28.0        "/bin/prometheus --c…"   6 days ago   Up 6 days   0.0.0.0:9090->9090/tcp, :::9090->9090/tcp
                                                                                                 prometheus
d7010b84c889   prom/blackbox-exporter:v0.15.1 "/bin/blackbox_expor…"   6 days ago   Up 6 days   0.0.0.0:9115->9115/tcp, :::9115->9115/tcp
                                                                                                 prom-graf_blackbox_1
43f46211f281   grafana/grafana:8.2.3          "/run.sh"                6 days ago   Up 6 days   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
                                                                                                 grafana
fe9d456f1a25   prom/pushgateway               "/bin/pushgateway --…"   6 days ago   Up 6 days   0.0.0.0:9091->9091/tcp, :::9091->9091/tcp
                                                                                                 pushgateway
```

You might want to expose the necessary ports to access the applications, Make sure you do not expose prometheus to the internet, it's just need communication with grafana.

Prometheus runs on port 9090
You can go into **Status > Targets** to see all configuration of prometheus.yml





- We need to install node_exporter on the worker nodes which needs to be monitored
- install the node_exporter on the client server follow below steps

```
Unset
wget https://github.com/prometheus/node_exporter/releases/download/v1.0.0-rc.0/node_exporter-1.0.0-rc.0.linux-amd64.tar.gz

tar -xzf node_exporter-1.0.0-rc.0.linux-amd64.tar.gz

mv node_exporter-1.0.0-rc.0.linux-amd64/node_exporter /usr/local/bin/
```

Create service for node_exporter

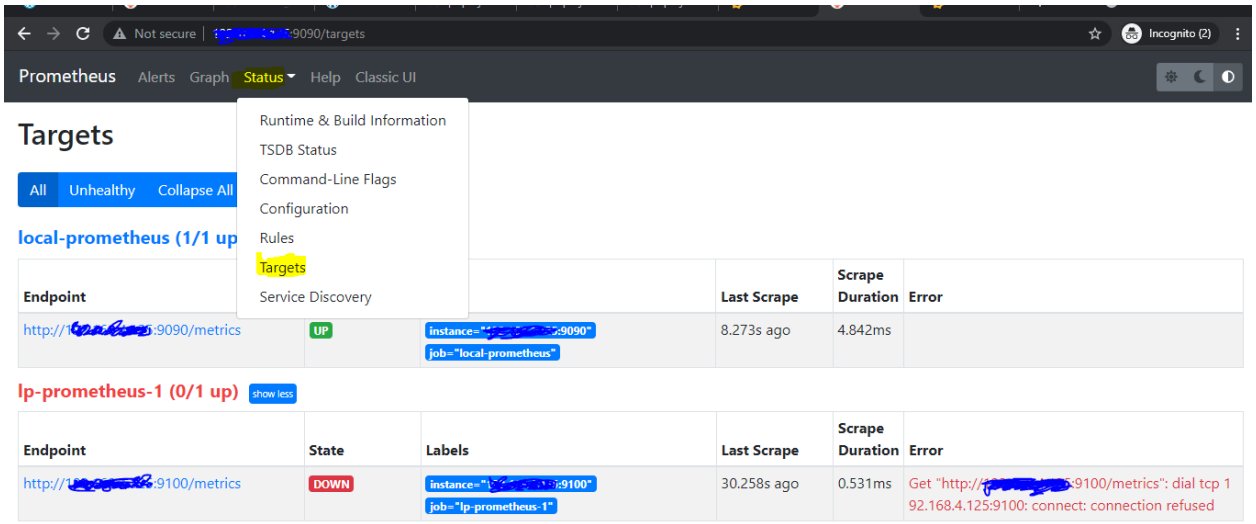
- use below command to do so

```
Unset
echo '[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=root
Group=root
Type=simple
Restart=on-failure
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
' >/etc/systemd/system/node_exporter.service
```

- Enable and start node_exporter service

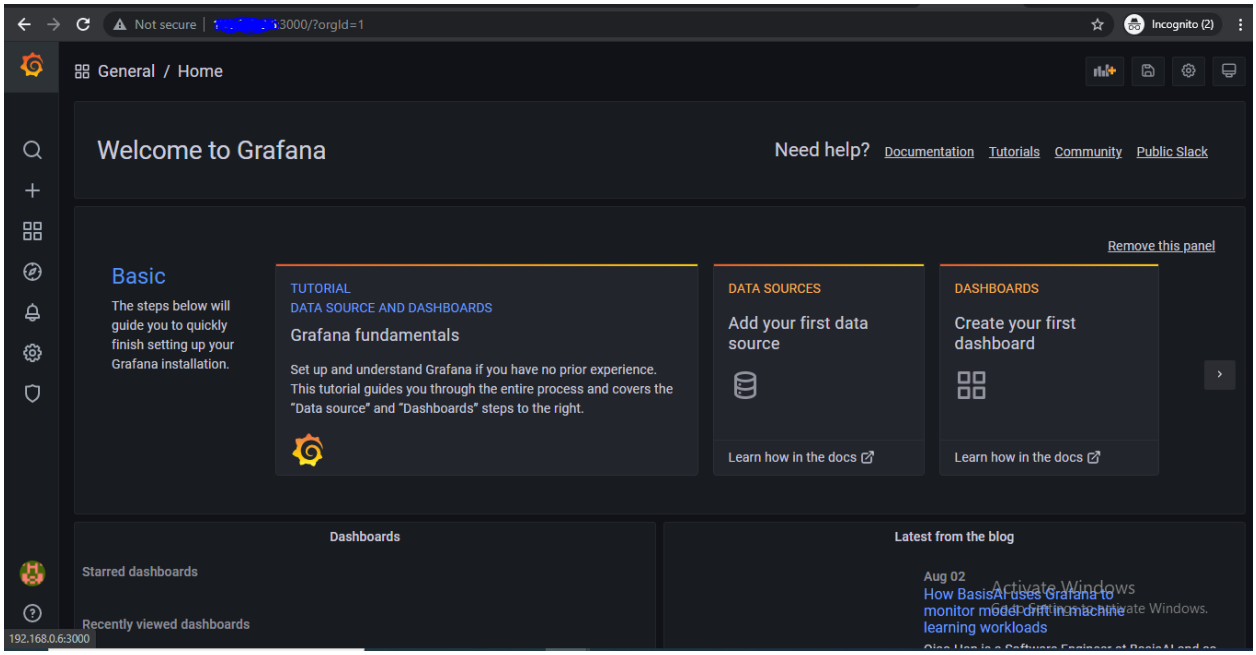
```
Unset
systemctl daemon-reload
systemctl enable node_exporter
systemctl start node_exporter
systemctl status node_exporter
```

- Make sure you open the port 9100 from the worker node in security group

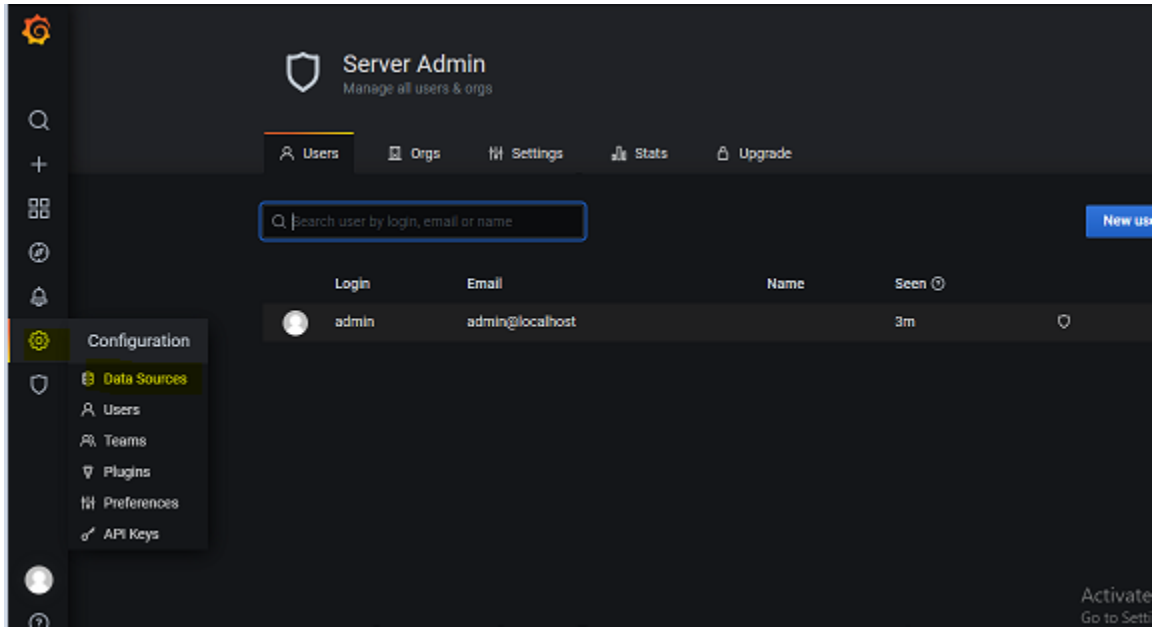


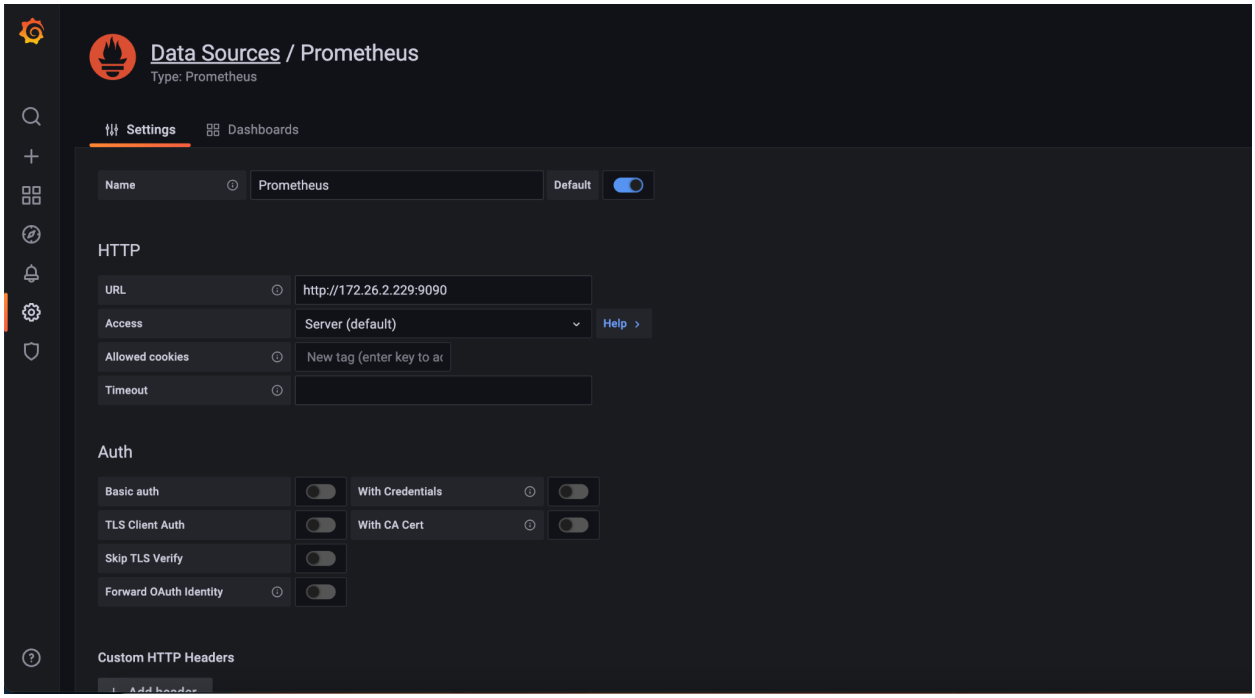
Grafana Dashboard setup

- login first time by using default password admin admin it will ask you to change the password
- But here we have configured ldap so we can use active directory password
- After login we can see this page, We need to configured the datasource and then we need to configure the dashboard

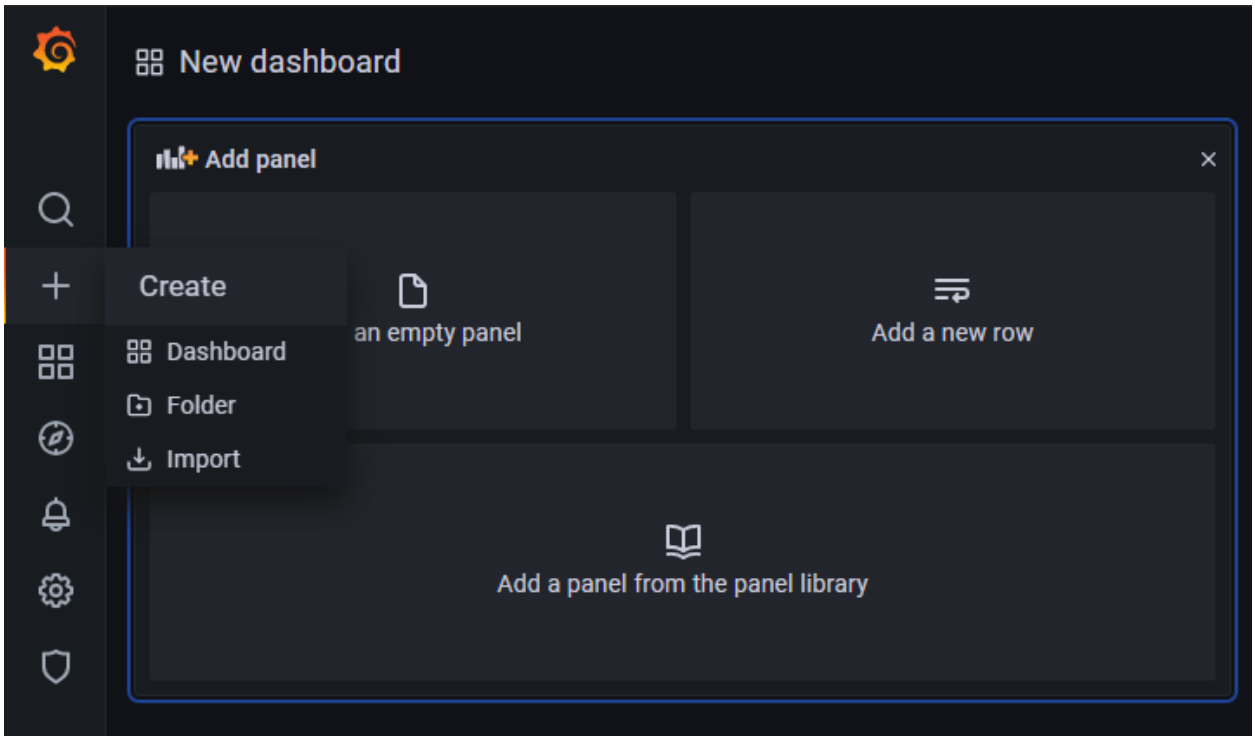


- Click on setting and datasource

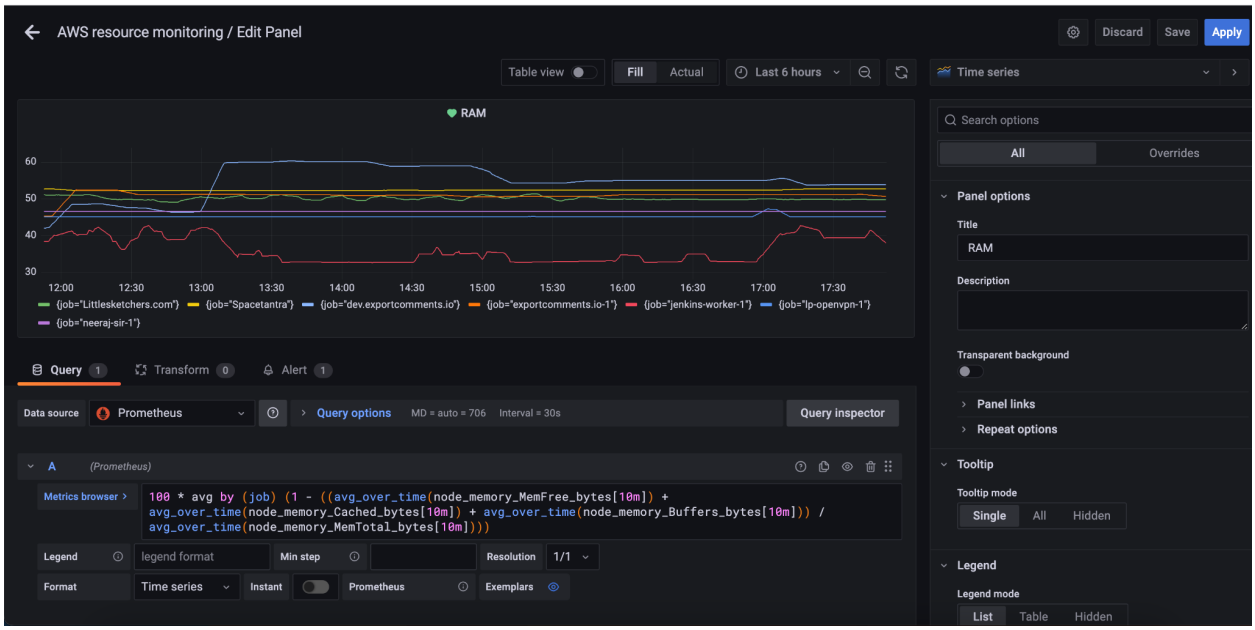




- Click on save and test it will show successful
- Make sure you open the port 9090 so that grafana can detect and communicate.
- Now we can proceed to create dashboard



You can create new panel inside the dashboard



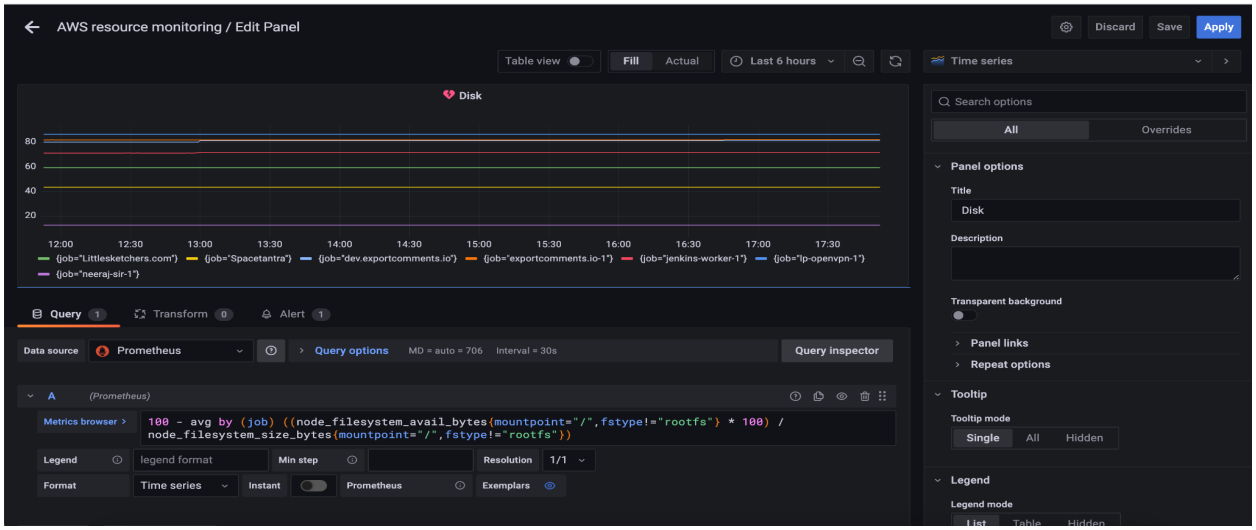
- You can use below query to monitor RAM

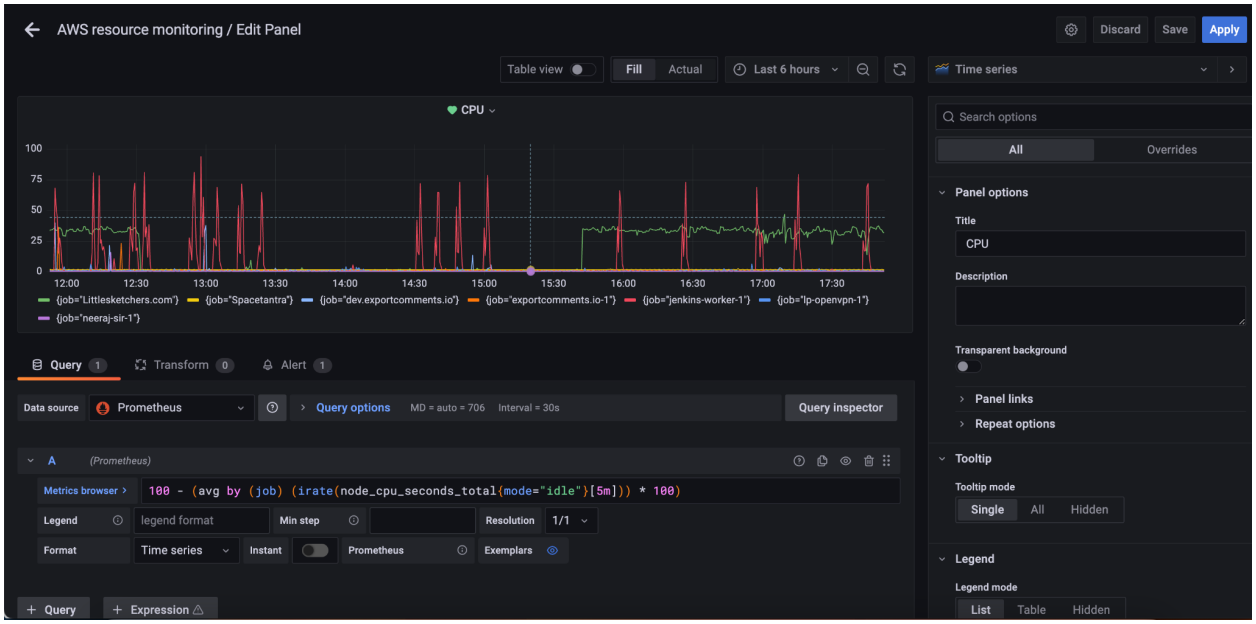
```
Unset
100 * avg by (job) (1 -
((avg_over_time(node_memory_MemFree_bytes[10m]) +
avg_over_time(node_memory_Cached_bytes[10m]) +
avg_over_time(node_memory_Buffers_bytes[10m])) /
avg_over_time(node_memory_MemTotal_bytes[10m])))
```

- We can Title the panel RAM and then we can begin adding new panels
- We can use below queries to the respective panels and save with the respective names.

Unset

```
#CPU: 100 - (avg by (job)
(irate(node_cpu_seconds_total{mode="idle"}[5m])) *
100)

#DISK usage: 100 - avg by (job)
((node_filesystem_avail_bytes{mountpoint="/",fstype!=
"rootfs"} * 100) /
node_filesystem_size_bytes{mountpoint="/",fstype!="ro
otfs"})

#DISK read IO: avg by (job)
(irate(node_disk_read_bytes_total{device="sda"}[5m])
/ 1024 / 1024)

#DISK WRITE IO: avg by (job)
(irate(node_disk_written_bytes_total{device="sda"}[1m
]) / 1024 / 1024)
```





- For exposing the grafana dashboard on grafana.intelligaia.com we have used nginx
- Nginx can be installed using below command

```
Unset
apt update
apt install nginx -y
```

- The nginx configuration are configured at /etc/nginx/conf.d/grafana.conf and /etc/nginx/conf.d/pushgateway.conf
- Cat /etc/nginx/conf.d/grafana.conf

```
Unset
server {
  server_name grafana.intelligaia.com;
  access_log off;

  location / {

    proxy_pass    http://127.0.0.1:3000;

    proxy_set_header  Host      $host;
    proxy_set_header  X-Real-IP    $remote_addr;
    proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
```

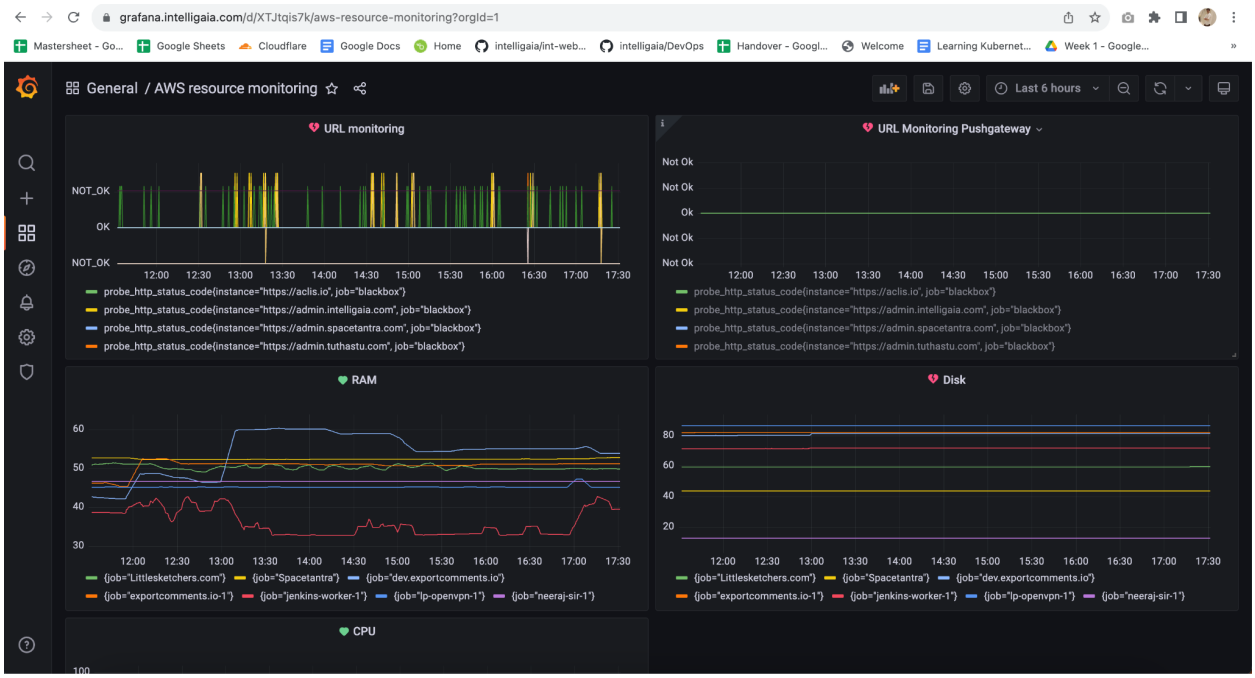
```
    proxy_set_header  X-Forwarded-Proto http;
    proxy_max_temp_file_size 0;
    proxy_connect_timeout   150;
    proxy_send_timeout    100;
    proxy_read_timeout    100;

    proxy_buffer_size     8k;
    proxy_buffers       4 32k;
    proxy_busy_buffers_size  64k;
    proxy_temp_file_write_size 64k;

  }


  listen 443 ssl; # managed by Certbot
  ssl_certificate /etc/letsencrypt/live/grafana.intelligaia.com/fullchain.pem; #
managed by Certbot
  ssl_certificate_key /etc/letsencrypt/live/grafana.intelligaia.com/privkey.pem;
# managed by Certbot
  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
  if ($host = grafana.intelligaia.com) {
    return 301 https://$host$request_uri;
  } # managed by Certbot


  server_name grafana.intelligaia.com;
  listen 80;
  return 404; # managed by Certbot


}
```

- Cat /etc/nginx/conf.d/pushgateway.conf

```
Unset
server {
  server_name pushgateway.intelligaia.com;
  access_log off;
```

```
  location / {

    proxy_pass    http://127.0.0.1:9091;

    proxy_set_header  Host      $host;
    proxy_set_header  X-Real-IP    $remote_addr;
    proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header  X-Forwarded-Proto http;
    proxy_max_temp_file_size 0;
    proxy_connect_timeout   150;
    proxy_send_timeout    100;
    proxy_read_timeout    100;

    proxy_buffer_size     8k;
    proxy_buffers       4 32k;
    proxy_busy_buffers_size  64k;
    proxy_temp_file_write_size 64k;

  }

  listen 443 ssl; # managed by Certbot
  ssl_certificate
/etc/letsencrypt/live/pushgateway.intelligaia.com/fullchain.pem; # managed by
Certbot
  ssl_certificate_key
/etc/letsencrypt/live/pushgateway.intelligaia.com/privkey.pem; # managed by
Certbot
  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server {
  if ($host = pushgateway.intelligaia.com) {
    return 301 https://$host$request_uri;
  } # managed by Certbot
  server_name pushgateway.intelligaia.com;
  listen 80;
  return 404; # managed by Certbot


}
```

Grafana web portal


grafana.intelligaia.com/d/XTJtqis7k/aws-resource-monitoring?orgId=1
Mastersheet - Go...
Google Sheets
Cloudflare
Google Docs
Home
intelligaia/int-web...
intelligaia/DevOps
Handover - Googl...
Welcome
Learning Kubernet...
Week 1 - Google...
General / AWS resource monitoring
Last 6 hours
URL monitoring
NOT_OK
OK
NOT_OK
12:00 12:30 13:00 13:30 14:00 14:30 15:00 15:30 16:00 16:30 17:00 17:30
probe_http_status_code{instance="https://aclis.io", job="blackbox"}
probe_http_status_code{instance="https://admin.intelligaia.com", job="blackbox"}
probe_http_status_code{instance="https://admin.spacetantra.com", job="blackbox"}
probe_http_status_code{instance="https://admin.tuthastu.com", job="blackbox"}
URL Monitoring Pushgateway
Not Ok
Not Ok
Ok
Not Ok
Not Ok
RAM
60
50
40
30
{job="Littlesketchers.com"} {job="Spacetantra"} {job="dev.exportcomments.io"}
{job="exportcomments.io-1"} {job="jenkins-worker-1"} {job="lp-openvpn-1"} {job="neeraj-sir-1"}
Disk
80
60
40
20
CPU
100