Monitoring Demo

Agenda:

1. Install Prometheus/ Node Exporter
2. Pointing to a domain name
3. Reverse proxy with NGINX
4. Add SSL to Prometheus Reverse Proxy
5. Add basic authentication to the Prometheus User Interface
6. Scrape Target Basics
7. Add rules
8. Install Prometheus AlertManager
9. Install Grafana

Prerequisites:

All commands were tested on a **Ubuntu 22.04 LTS**. If you chose to use another distribution, you might need to check appropriate commands to get the same result.

1. Have access to an unrestricted Ubuntu server. I got mine from https://www.digitalocean.com but you can use other providers as long as they don't have extra firewall configuration that may interfere with this demo
2. You will need to have root access
3. To be able to do the DNS configuration part, you also need to have a domain purchased in advance.

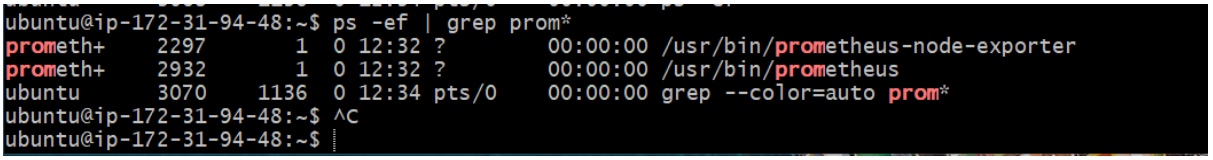
1. **Install Prometheus/ Node Exporter through the package manager from Ubuntu**

First, we are gonna update packages and install prometheus:

```
sudo apt update
sudo apt install prometheus
```

This install will set up a Prometheus process and also a Node Exporter process.

```
ubuntu@ip-172-31-94-48:~$ ps -ef | grep prom*
prometh+       2297            1    0  12:32  ?              00:00:00
/usr/bin/prometheus-node-exporter
prometh+       2932            1    0  12:32  ?              00:00:00
/usr/bin/prometheus
```



NOTE: Prometheus will listen on port 9090 and node exporter on port 9100.

To check those services are actually running:

```
sudo service prometheus status
```

```
ubuntu@ip-172-31-94-48:~$ sudo service prometheus status
● prometheus.service - Monitoring system and time series database
     Loaded: loaded (/usr/lib/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-01 12:32:25 UTC; 12min ago
       Docs: https://prometheus.io/docs/introduction/overview/
             man:prometheus(1)
   Main PID: 2932 (prometheus)
      Tasks: 6 (limit: 1130)
     Memory: 60.3M (peak: 60.6M)
        CPU: 860ms
     CGroup: /system.slice/prometheus.service
             └─2932 /usr/bin/prometheus

Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.493Z caller=head.go:676 level=info compo
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.493Z caller=head.go:684 level=info compo
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.495Z caller=head.go:755 level=info compo
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.495Z caller=head.go:792 level=info compo
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.498Z caller=main.go:1025 level=info fs_t
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.498Z caller=main.go:1028 level=info msg=
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.498Z caller=main.go:1209 level=info msg=
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.503Z caller=main.go:1246 level=info msg=
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.503Z caller=main.go:989 level=info msg="
Sep 01 12:32:26 ip-172-31-94-48 prometheus[2932]: ts=2024-09-01T12:32:26.503Z caller=manager.go:999 level=info co
lines 1-22/22 (END)
```

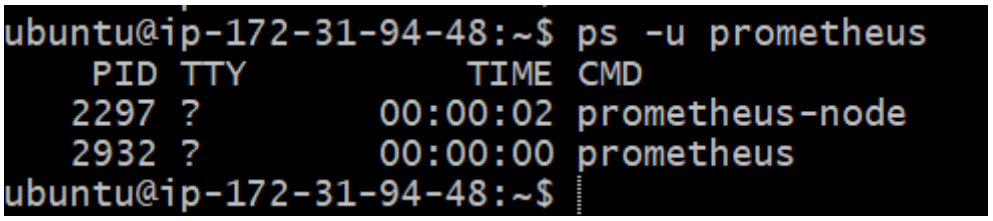
Similar to get node exporter status:

```
sudo service prometheus-node-exporter status
```

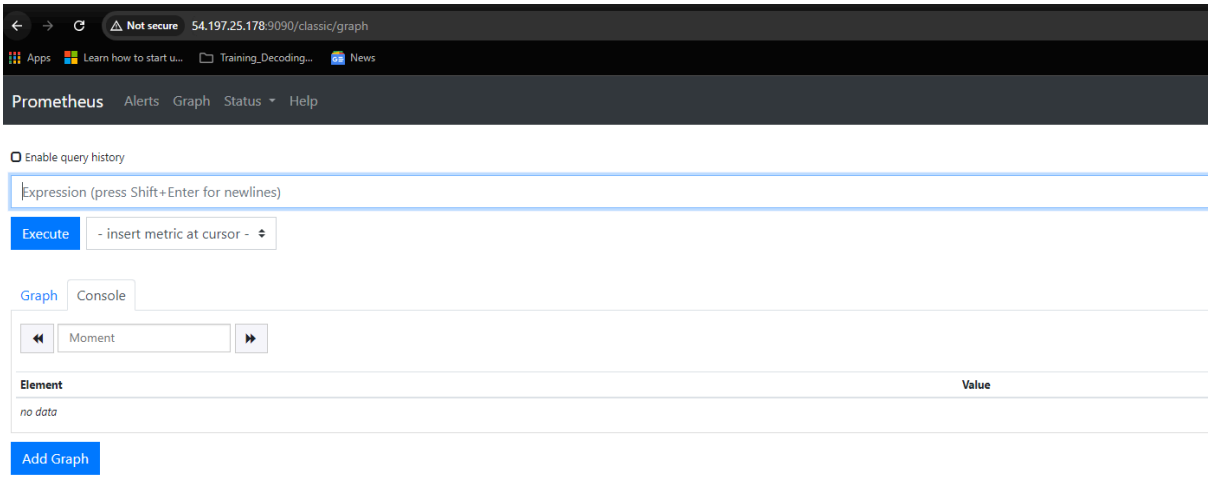
```
ubuntu@ip-172-31-94-48:~$ sudo service prometheus-node-exporter status
● prometheus-node-exporter.service - Prometheus exporter for machine metrics
     Loaded: loaded (/usr/lib/systemd/system/prometheus-node-exporter.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-01 12:32:20 UTC; 13min ago
       Docs: https://github.com/prometheus/node_exporter
   Main PID: 2297 (prometheus-node)
      Tasks: 5 (limit: 1130)
     Memory: 10.4M (peak: 11.7M)
        CPU: 2.581s
     CGroup: /system.slice/prometheus-node-exporter.service
             └─2297 /usr/bin/prometheus-node-exporter

Sep 01 12:32:20 ip-172-31-94-48 prometheus-node-exporter[2297]: ts=2024-09-01T12:32:20.485Z caller=node_exporte
Sep 01 12:32:20 ip-172-31-94-48 prometheus-node-exporter[2297]: ts=2024-09-01T12:32:20.485Z caller=node_exporte
Sep 01 12:32:20 ip-172-31-94-48 prometheus-node-exporter[2297]: ts=2024-09-01T12:32:20.485Z caller=node_exporte
```

We also get a user called prometheus, which is running 2 processes:

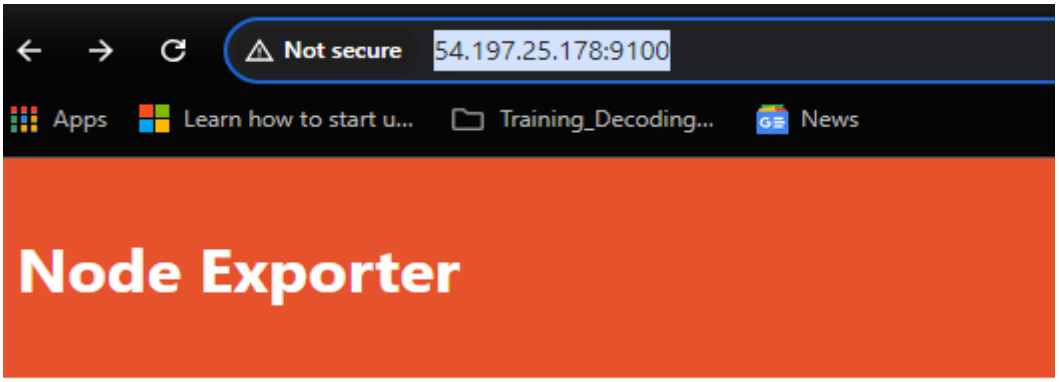
```
ps -u prometheus
```



If we don't have any FW to configure, we can already use the machine public IP + the prometheus port to get a first look at our prometheus instance:



Or at our Node Exporter instance: http://54.197.25.178:9100/



Prometheus Node Exporter

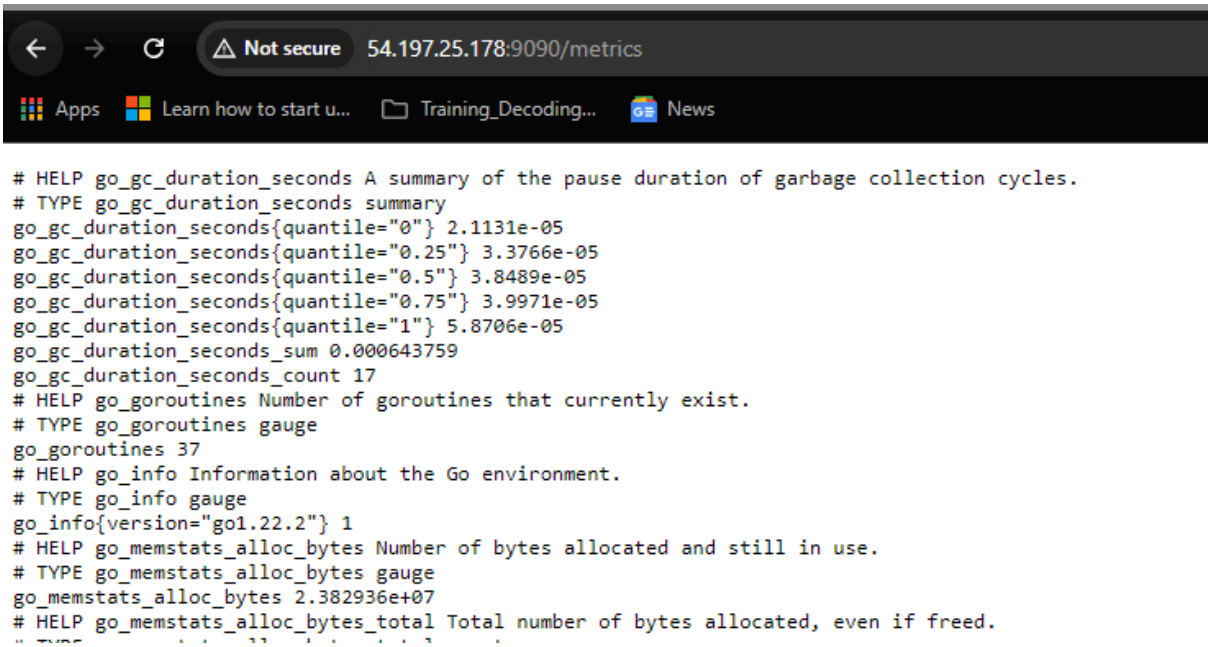
Version: (version=1.7.0, branch=debian/sid, revision=1.7.0-1ubuntu0.1)

- Metrics

Prometheus will query this 9100/metrics endpoint at intervals and we will be able to view these metrics in prometheus as time series data;

Prometheus also has a metrics endpoints, and will be reading data from both of them:

http://54.197.25.178:9090/metrics



At this moment Prometheus is accessible from the internet, we can do a little tweaks to lock it down:

1. give it an SSL certificate
2. a domain name
3. setup basic authentication so you can't access it without an username & password;

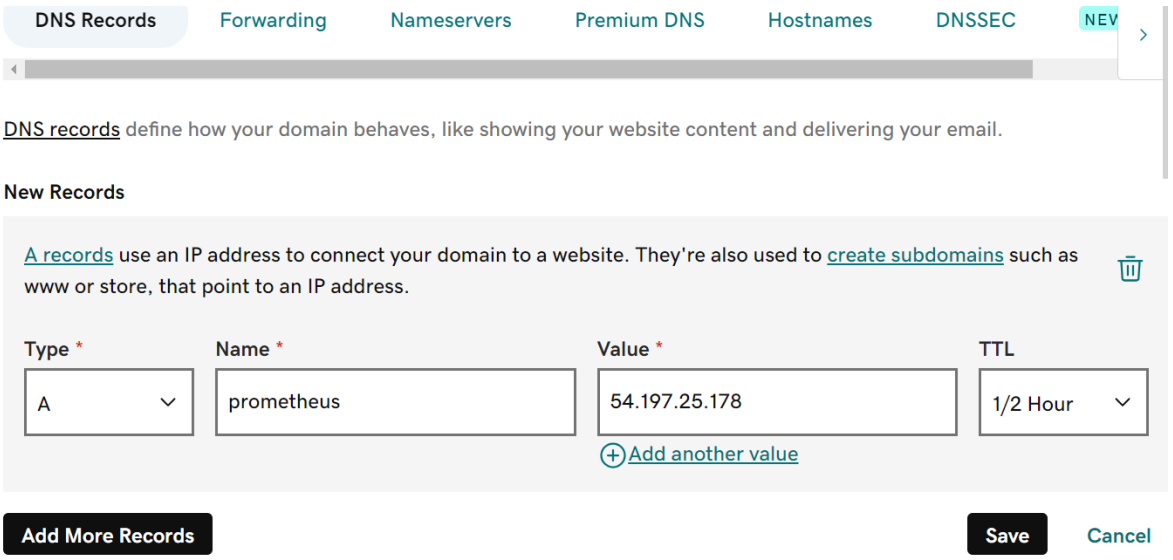
2. Pointing to a domain name

Current config: We can access Prometheus instance using our instance IP Example: http://54.197.25.178:9090/

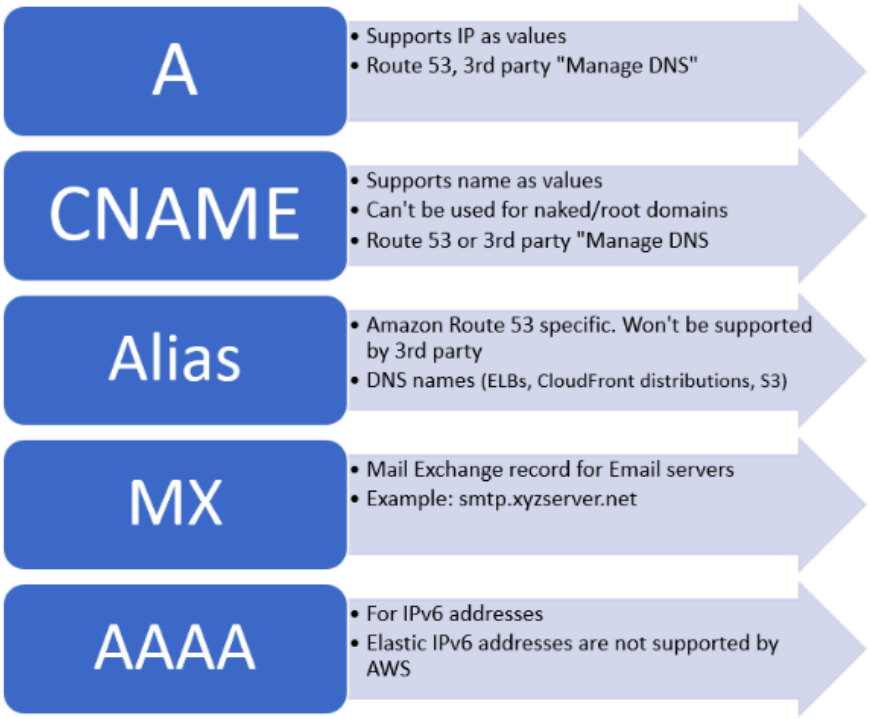
Target config: access it using DNS. Example: https://prometheus.domain name

Prerequisites: have a domain purchased;

Go to your domain purchased prom your preferred provider and Add a A tag domain:

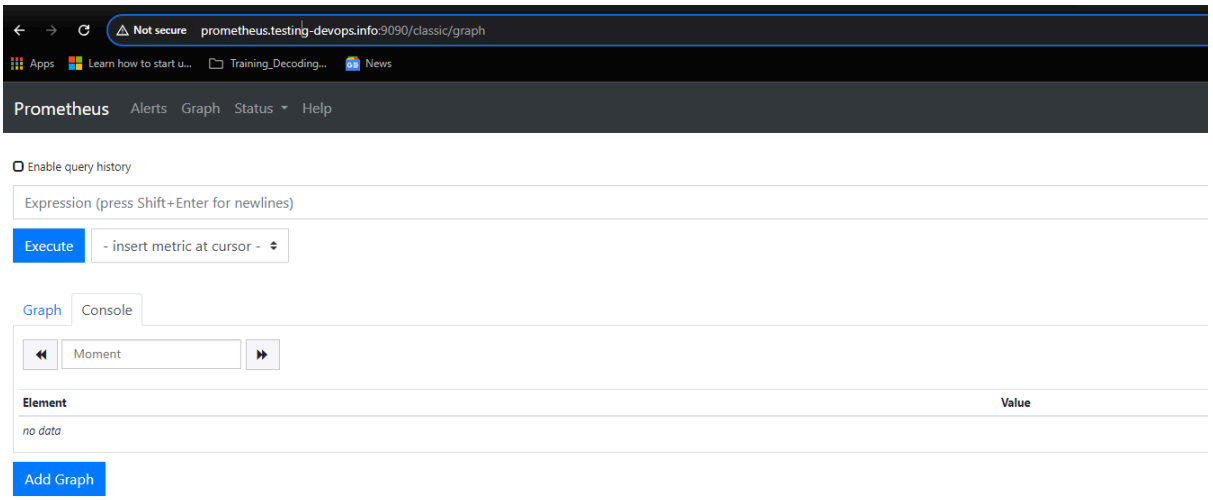


Definitions:



Note: Most DNS updates take effect within an hour, but could take up to 48 hours to update globally.

We need to check: http://prometheus.testing-devops.info:9090/ after a while;

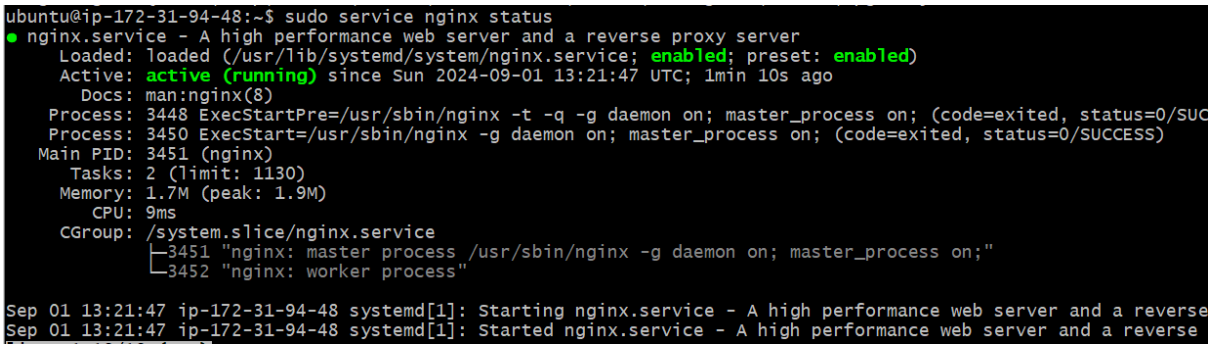


Next step is set it up under reverse proxy so we don't have to use the port any longer and give it an SSL connection;

3. Reverse proxy with NGINX

First, install nginx and check its status:

```
sudo apt install nginx
sudo service nginx status
```

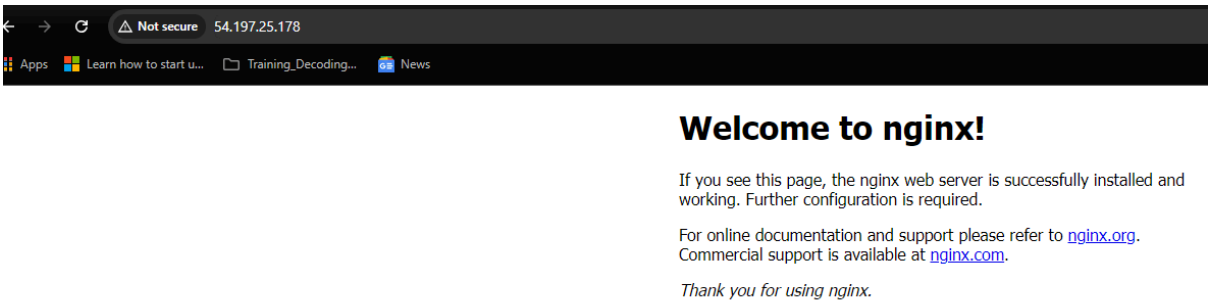


cd to NGINX folder:

```
cd /etc/nginx/sites-enabled
```

In here we will get the default page we get when we install nginx:
lrwxrwxrwx 1 root root 34 Sep 1 13:21 **default -> /etc/nginx/sites-available/default**

First screen available based on machine IP:



Note: it listens on port 80 by default so we don't have to type it;
We will create a new config. Switch to root with sudo -i in case you were not from beginning:

```
sudo vi prometheus
```

and then insert our Prometheus server name:

```
server {
   listen 80;
   listen [::]:80;
   server_name  YOUR-DOMAIN-NAME;  => prometheus.testing-devops.info;
   location / {
      proxy_pass         http://localhost:9090/;
   }
}
```

So now that we added server name, it is going to forward that default location into 9090, the default port that prometheus listens to;

```
ubuntu@ip-172-31-94-48:/etc/nginx/sites-enabled$ cat prometheus
server {
    listen 80;
    listen [::]:80;
    server_name  prometheus.testing-devops.info;

    location / {
        proxy_pass          http://localhost:9090/;
    }
}

ubuntu@ip-172-31-94-48:/etc/nginx/sites-enabled$
```

We can also test the new configuration has no errors:

```
nginx -t
```

```
root@ip-172-31-94-48:/etc/nginx/sites-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ip-172-31-94-48:/etc/nginx/sites-enabled#
```

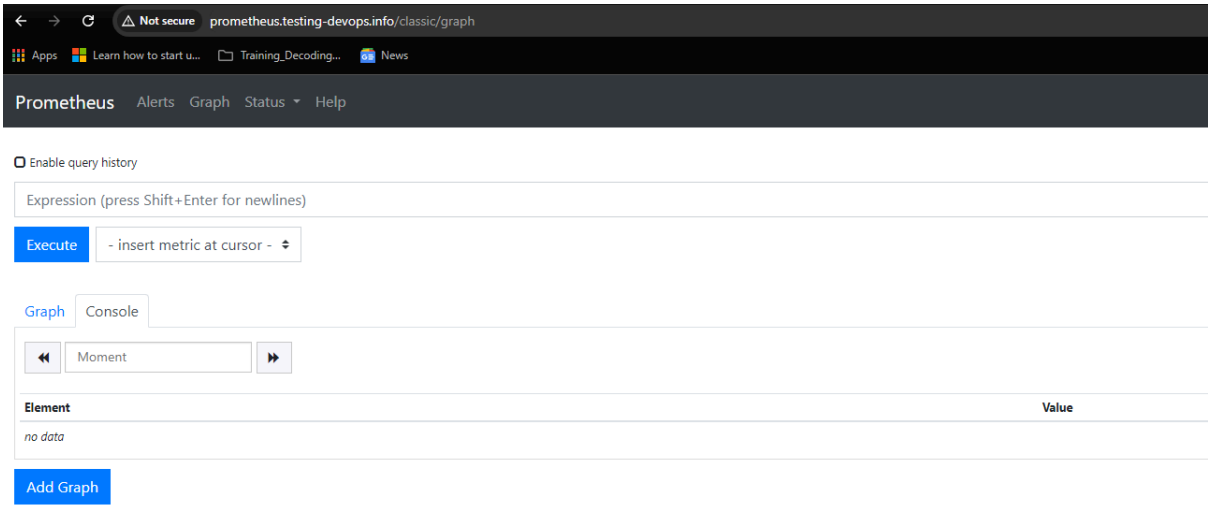
Next, restart nginx to load recent updates:

```
root@ip-172-31-94-48:/etc/nginx/sites-enabled# sudo service nginx restart

root@ip-172-31-94-48:/etc/nginx/sites-enabled# sudo service nginx status
```

```
root@ip-172-31-94-48:/etc/nginx/sites-enabled# sudo service nginx status
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-01 13:45:49 UTC; 9s ago
       Docs: man:nginx(8)
    Process: 3663 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3666 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3667 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.7M (peak: 2.4M)
        CPU: 10ms
     CGroup: /system.slice/nginx.service
             ├─3667 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3668 "nginx: worker process"

Sep 01 13:45:49 ip-172-31-94-48 systemd[1]: Starting nginx.service - A high performance web server and a reverse prox>
Sep 01 13:45:49 ip-172-31-94-48 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy>

root@ip-172-31-94-48:/etc/nginx/sites-enabled#
```

Now if we check the url in the browser, it will direct us to prometheus, with no need to use the port:

http://prometheus.testing-devops.info/classic/graph



Next we will also set up SSL, as at the moment our connection is not secure:

4. Add SSL to Prometheus Reverse Proxy

1 way to get a certificate from: https://certbot.eff.org/instructions?ws=nginx&os=snap

```
sudo snap install --classic certbot
```

```
root@prometheus:/etc/nginx/sites-enabled# sudo snap install --classic certbot
2024-09-01T14:44:54Z INFO Waiting for automatic snapd restart...
certbot 2.11.0 from Certbot Project (certbot-eff✓) installed
root@prometheus:/etc/nginx/sites-enabled#
```

```
sudo certbot --nginx
```

Answer to provided questions:

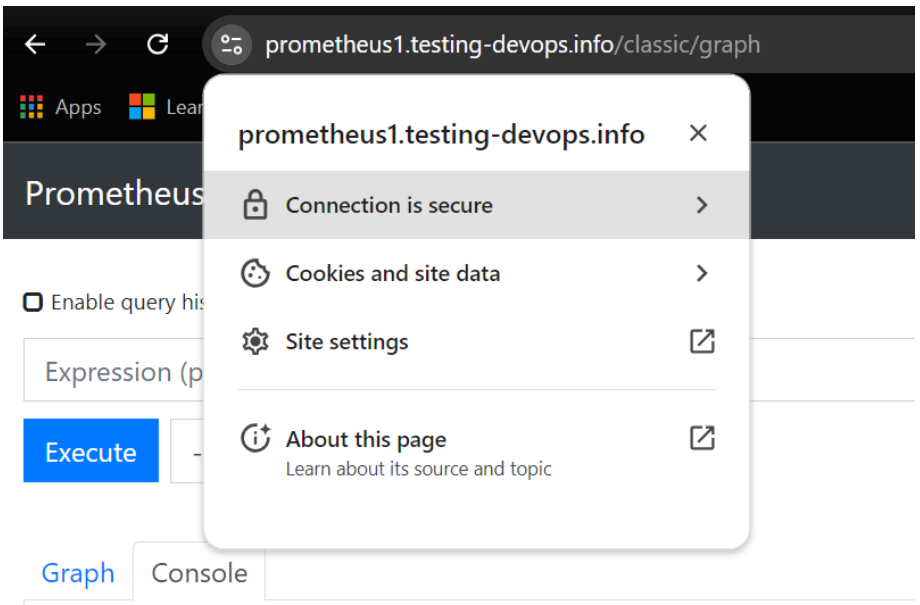
- Enter email address: enter your email address
- Accept terms & conditions: Y
- Campaigns: N
- Select the domain: option 1

It will scan the nginx config and find the website:

```
Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: prometheus1.testing-devops.info
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1
Requesting a certificate for prometheus1.testing-devops.info

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/prometheus1.testing-devops.info/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/prometheus1.testing-devops.info/privkey.pem
This certificate expires on 2024-11-30.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for prometheus1.testing-devops.info to /etc/nginx/sites-enabled/prometheus
Congratulations! You have successfully enabled HTTPS on https://prometheus1.testing-devops.info

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
root@prometheus:/etc/nginx/sites-enabled#
```

After installing the SSL cert, the https page will work:

https://prometheus1.testing-devops.info/classic/graph



Another thing, if we type in http, now it will FW it to https automatically;

Also, if we check out the prometheus config file from sites-enabled, it will have more information added to support SSL configuration, extra config added automatically by certbot;

5. Add basic authentication to the Prometheus User Interface:

```
cd /etc/nginx

sudo apt install apache2-utils
```

To create a password file for an user called admin:

```
htpasswd -c /etc/nginx/.htpasswd admin
```

```
root@prometheus:/etc/nginx# htpasswd -c /etc/nginx/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
root@prometheus:/etc/nginx#
```

New file password created:

```
root@prometheus:/etc/nginx# ls -la
total 80
drwxr-xr-x   8 root root 4096 Sep  1 15:01 .
drwxr-xr-x 116 root root 4096 Sep  1 15:00 ..
-rw-r--r--   1 root root   44 Sep  1 15:02 .htpasswd
drwxr-xr-x   2 root root 4096 Mar 31 02:40 conf.d
```

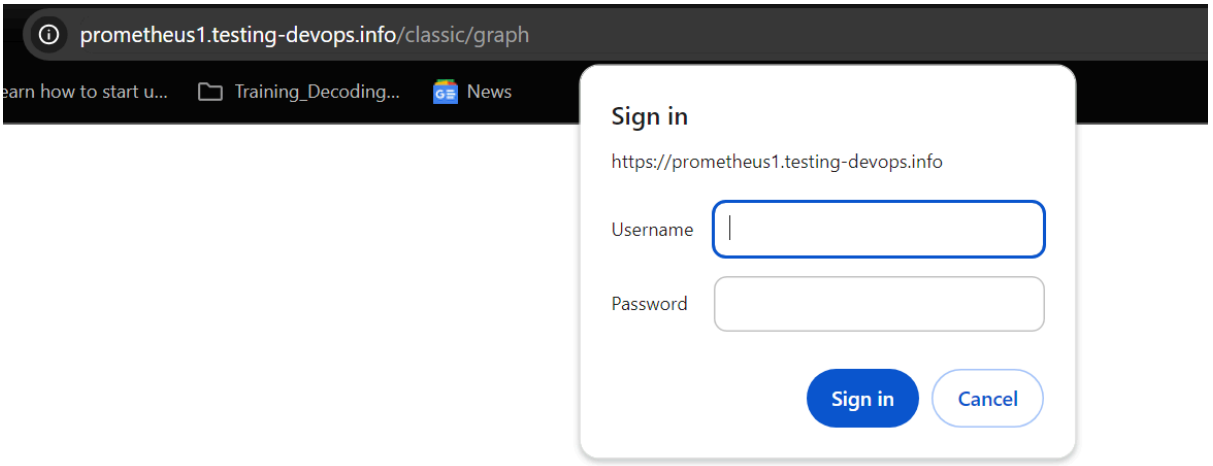
Now, in the prometheus file that we have under nginx sites-enabled path, we will add a pointer to the user file.

```
auth_basic  "Password Protected";
auth_basic_user_file  /etc/nginx/.htpasswd;
```

root@prometheus: /etc/nginx

```
 GNU nano 7.2                                  /etc/nginx/sites-enabled/prometheus *
server {
    server_name  prometheus1.testing-devops.info;
    auth_basic  "Password protected";
    auth_basic_user_file /etc/nginx/.htpasswd;
    location / {
        proxy_pass          http://localhost:9090/;
    }
```

Restart NGINX and check service status and then try to login again to Prometheus.

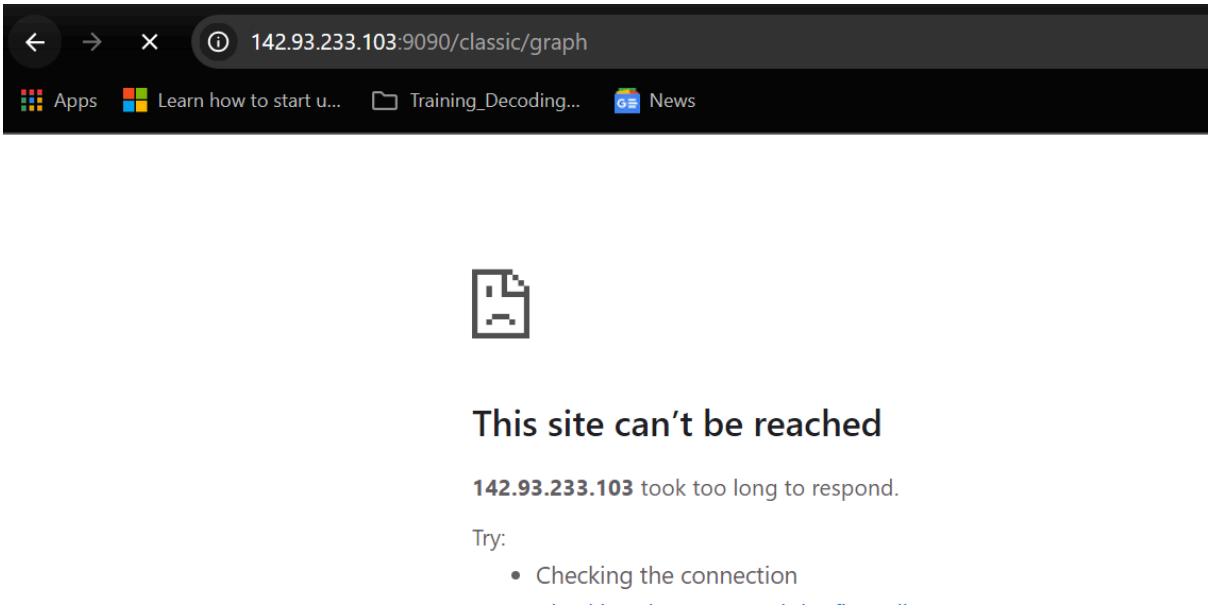


At this point anyone can still access prometheus without a password using directly the IP and port: http://142.93.233.103:9090/classic/graph

but we can block external connections using following commands:

```
iptables -A INPUT -p tcp -s localhost --dport 9090 -j ACCEPT
iptables -A INPUT -p tcp --dport 9090 -j DROP
iptables -A INPUT -p tcp -s localhost --dport 9100 -j ACCEPT
iptables -A INPUT -p tcp --dport 9100 -j DROP
iptables -L
```

```
Sep 01 15:08:14 prometheus systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
root@prometheus:/etc/nginx# iptables -A INPUT -p tcp -s localhost --dport 9090 -j ACCEPT
root@prometheus:/etc/nginx# iptables -A INPUT -p tcp --dport 9090 -j DROP
root@prometheus:/etc/nginx# iptables -A INPUT -p tcp -s localhost --dport 9100 -j ACCEPT
root@prometheus:/etc/nginx# iptables -A INPUT -p tcp --dport 9100 -j DROP
root@prometheus:/etc/nginx# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9090
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9090
DROP       tcp  --  anywhere             anywhere             tcp dpt:9090
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
DROP       tcp  --  anywhere             anywhere             tcp dpt:9100

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@prometheus:/etc/nginx#
```



6. Scrape Target Basics:

We can see the metrics prometheus exposes by doing a curl to:

curl http://localhost:9090/metrics

```
root@prometheus:/etc/nginx# curl http://localhost:9090/metrics | tail -15
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 79991    0 79991    0     0  16.2M      0 --:--:-- --:--:-- --:--:-- 19.0M
prometheus_tsdb_wal_writes_failed_total 0
# HELP prometheus_web_federation_errors_total Total number of errors that occurred while sending federation responses.
# TYPE prometheus_web_federation_errors_total counter
prometheus_web_federation_errors_total 0
# HELP prometheus_web_federation_warnings_total Total number of warnings that occurred while sending federation responses.
# TYPE prometheus_web_federation_warnings_total counter
prometheus_web_federation_warnings_total 0
# HELP promhttp_metric_handler_requests_in_flight Current number of scrapes being served.
# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_in_flight 1
# HELP promhttp_metric_handler_requests_total Total number of scrapes by HTTP status code.
# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="200"} 705
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
root@prometheus:/etc/nginx#
```

We can also inspect on the server using:

config.file	/etc/prometheus/prometheus.yml

or review it from https://prometheus1.testing-devops.info/classic/config

7. Prometheus Rules

Recording Rules:

CD into prometheus folder and create a rules file:

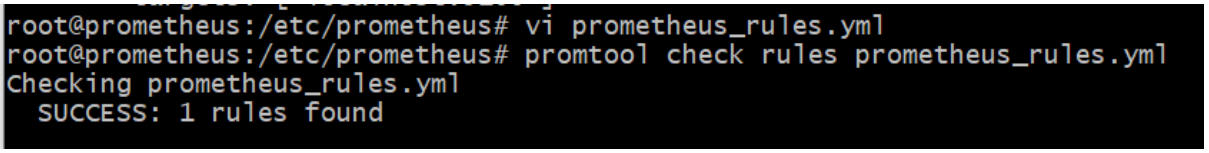
```
cd /etc/prometheus

sudo vi prometheus_rules.yml
```

Add a test expression as recording rule:

```
groups:
  - name: custom_rules
    rules:
      - record: node_memory_MemFree_percent
                              expr: 100 - (100 * node_memory_MemFree_bytes / node_memory_MemTotal_bytes)
```

And also check if the rule file has good syntax using a tool called promtool:

```
promtool check rules prometheus_rules.yml
```



If we get a success result on the check, we can add the *prometheus_rules.yml* reference to the *prometheus.yml* rule_files section.

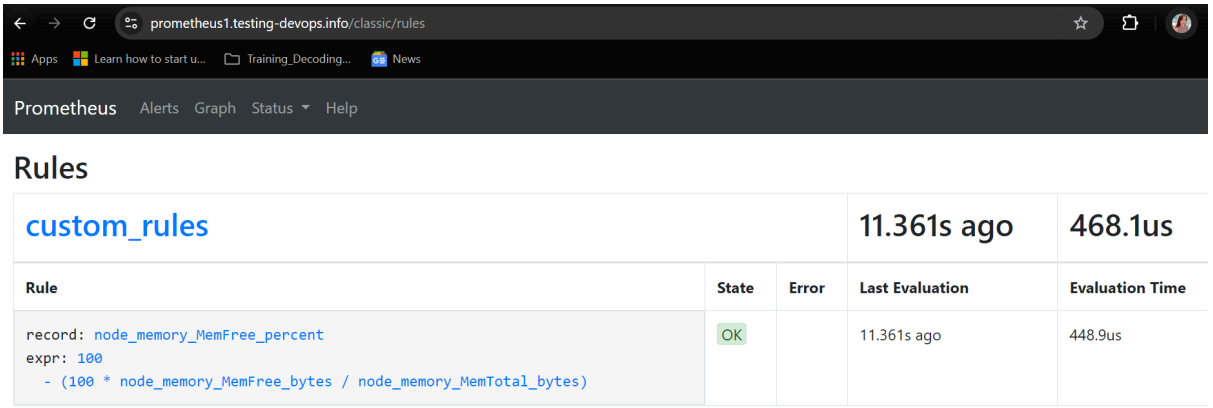
```
rule_files:

  - "prometheus_rules.yml"
```

And then we can restart Prometheus service:

```
sudo service prometheus restart
```

sudo service prometheus status

```
root@prometheus:/etc/prometheus# sudo service prometheus restart
root@prometheus:/etc/prometheus# sudo service prometheus status
● prometheus.service - Monitoring system and time series database
     Loaded: loaded (/usr/lib/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-09-05 12:23:37 UTC; 7s ago
       Docs: https://prometheus.io/docs/introduction/overview/
             man:prometheus(1)
   Main PID: 71608 (prometheus)
      Tasks: 6 (limit: 1113)
     Memory: 56.1M (peak: 56.4M)
        CPU: 416ms
     CGroup: /system.slice/prometheus.service
             └─71608 /usr/bin/prometheus

Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.212Z caller=head.go:755 level=info c
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.340Z caller=head.go:755 level=info c
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.340Z caller=head.go:755 level=info c
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.340Z caller=head.go:792 level=info c
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.343Z caller=main.go:1025 level=info
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.343Z caller=main.go:1028 level=info
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.344Z caller=main.go:1209 level=info
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.345Z caller=main.go:1246 level=info
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.345Z caller=main.go:989 level=info m
Sep 05 12:23:38 prometheus prometheus[71608]: ts=2024-09-05T12:23:38.345Z caller=manager.go:999 level=inf
```



Alerting rules:

We create a different file and we need to add the reference to it in the *rule_files* section in *prometheus.yml*

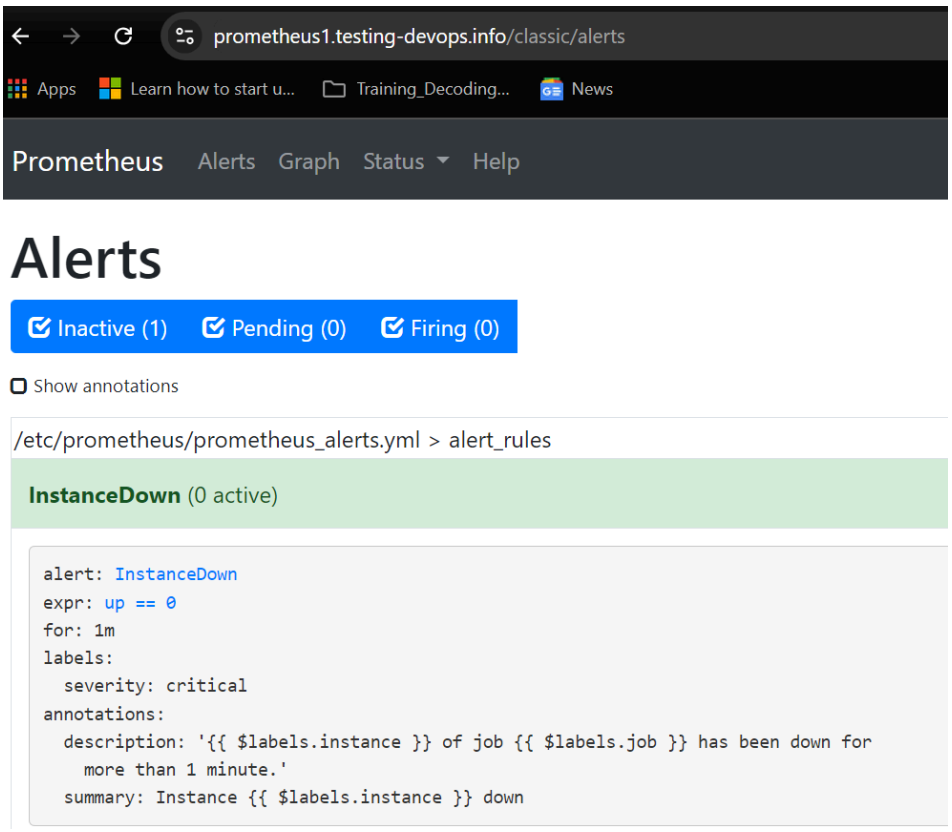
root@prometheus:/etc/prometheus# cat prometheus_alerts.yml

```
groups:

  - name: alert_rules
```

```
rules:

 - alert: InstanceDown

   expr: up == 0

   for: 1m

   labels:

     severity: critical

   annotations:

     summary: 'Instance {{ $labels.instance }} down'

       description: '{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 1 minute.'
```

Similar as above section, review it using promtool and restart Prometheus service.

```
root@prometheus:/etc/prometheus# vi prometheus_alerts.yml
root@prometheus:/etc/prometheus# promtool check rules prometheus_alerts.yml
Checking prometheus_alerts.yml
  SUCCESS: 1 rules found
```



8. Install Prometheus AlertManager:

```
sudo apt install prometheus-alertmanager
```

It has started a new service called prometheus-alertmanager

```
sudo service prometheus-alertmanager status
```

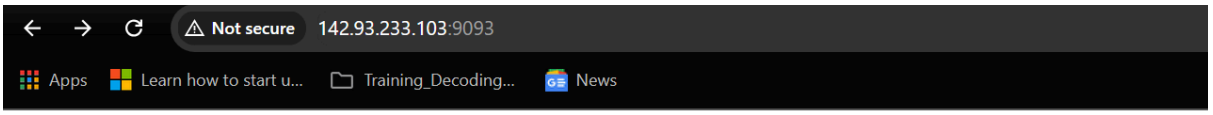
```
root@prometheus:/etc/prometheus# service prometheus-alertmanager status
● prometheus-alertmanager.service - Alertmanager for prometheus
     Loaded: loaded (/usr/lib/systemd/system/prometheus-alertmanager.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-09-05 12:54:08 UTC; 1min 12s ago
       Docs: https://prometheus.io/docs/alerting/alertmanager/
   Main PID: 72097 (prometheus-aler)
      Tasks: 7 (limit: 1113)
     Memory: 15.2M (peak: 15.4M)
        CPU: 174ms
     CGroup: /system.slice/prometheus-alertmanager.service
             └─72097 /usr/bin/prometheus-alertmanager

Sep 05 12:54:08 prometheus prometheus-alertmanager[72097]: ts=2024-09-05T12:54:08.545Z caller=main.go:246 le
Sep 05 12:54:08 prometheus prometheus-alertmanager[72097]: ts=2024-09-05T12:54:08.547Z caller=main.go:247 le
Sep 05 12:54:08 prometheus prometheus-alertmanager[72097]: ts=2024-09-05T12:54:08.567Z caller=cluster.go:186
Sep 05 12:54:08 prometheus prometheus-alertmanager[72097]: ts=2024-09-05T12:54:08.583Z caller=cluster.go:683
Sep 05 12:54:08 prometheus prometheus-alertmanager[72097]: ts=2024-09-05T12:54:08.683Z caller=coordinator.go
```

It is also managed by the user prometheus.

ps -u prometheus

```
root@prometheus:/etc/prometheus# ps -u prometheus
    PID TTY          TIME CMD
  13420 ?        00:20:45 prometheus-node
  71856 ?        00:00:01 prometheus
  72097 ?        00:00:00 prometheus-aler
```

Visit http://[your domain name or IP]:9093/



Alertmanager

The Debian package of the alertmanager does not include a web application.

Please, use the `amtool` command-line application to obtain equivalent functionality.

Alternatively, you can build and deploy the UI yourself, by placing the generated files in `/usr/share/prometheus/alertmanager/ui/`, The script located at `/usr/share/prometheus/alertmanager/generate-ui.sh` can automatically build and deply the UI.

You can still use the HTTP API, and the special handlers:

- /metrics
- /-/reload
- /-/healthy
- /-/ready

Block port 9093 for external requests:

```
iptables -A INPUT -p tcp -s localhost --dport 9093 -j ACCEPT

iptables -A INPUT -p tcp --dport 9093 -j DROP

iptables -L
```

```
root@prometheus:/etc/prometheus# iptables -A INPUT -p tcp -s localhost --dport 9093 -j ACCEPT
iptables -A INPUT -p tcp --dport 9093 -j DROP
iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9090
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9090
DROP       tcp  --  anywhere             anywhere             tcp dpt:9090
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9100
DROP       tcp  --  anywhere             anywhere             tcp dpt:9100
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9093
ACCEPT     tcp  --  localhost            anywhere             tcp dpt:9093
DROP       tcp  --  anywhere             anywhere             tcp dpt:9093

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@prometheus:/etc/prometheus# |
```

NOTE:

iptables settings will be lost in case of system reboot. You will need to reapply them manually, or install **iptables-persistent:**

```
sudo apt install iptables-persistent
```

This will save your settings into two files called,

/etc/iptables/rules.v4

/etc/iptables/rules.v6

Any changes you make to the **iptables** configuration won't be auto saved to these persistent files, so if you want to update these files with any changes, then use the commands,

iptables-save > /etc/iptables/rules.v4

iptables-save > /etc/iptables/rules.v6

Check the endpoint in the prometheus.yml is correctly set for the location of your alert manager.

```
sudo vi /etc/prometheus/prometheus.yml
```

Mine is set to the alert manager running locally on localhost:9093

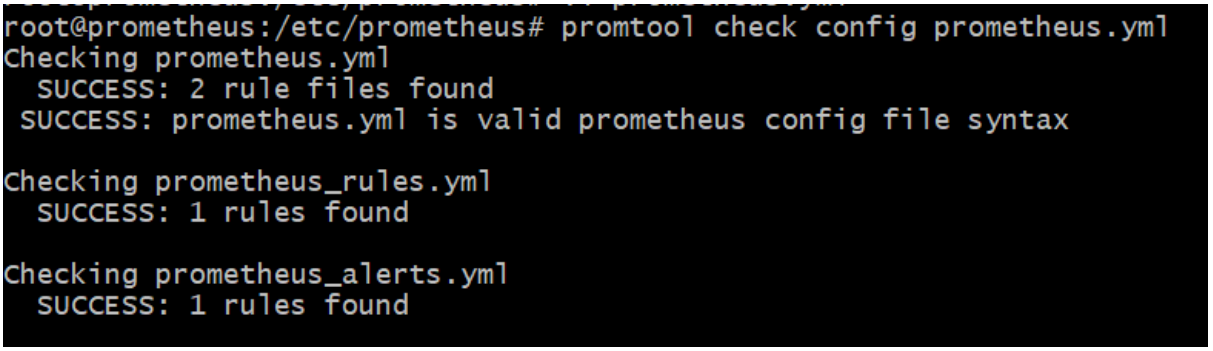
```
...

# Alertmanager configuration

alerting:

  alertmanagers:

    - static_configs:

      - targets: ['localhost:9093']
```

We can optionally also add the alert manager metrics endpoint to be scraped by Prometheus as well so that we can monitor its performance.

```
scrape_configs:

  ...

  - job_name: alertmanager

    static_configs:

      - targets: ['localhost:9093']
```

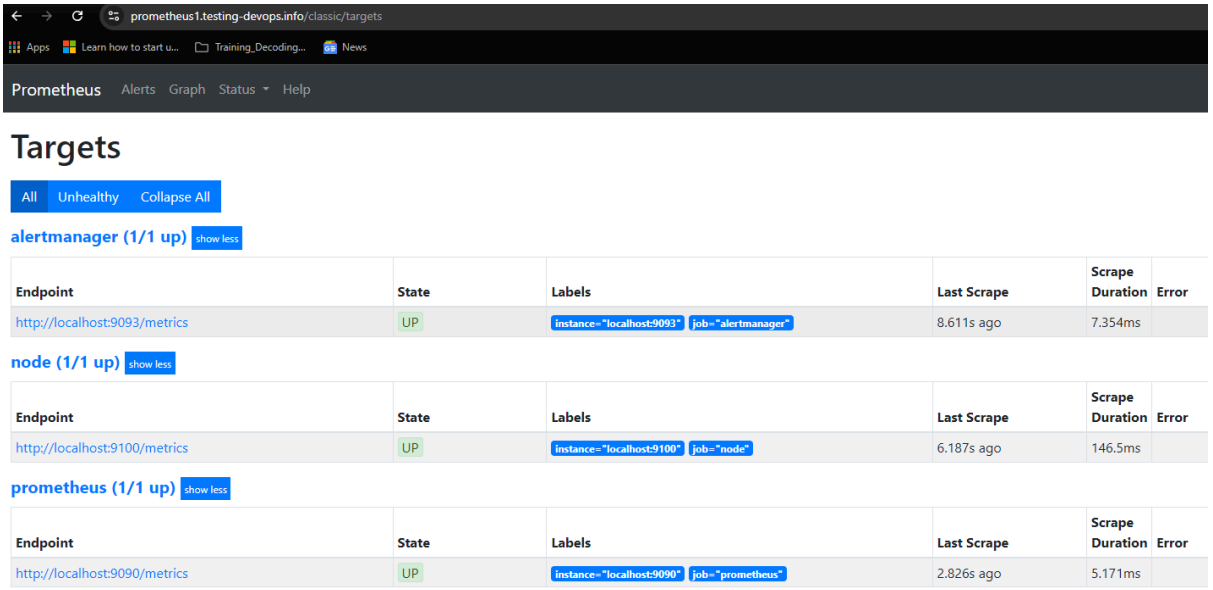
If you edit the prometheus.yml, remember to check it using promtool

```
promtool check config /etc/prometheus/prometheus.yml
```



And then restart Prometheus:

```
sudo service prometheus restart

sudo service prometheus status
```



9. Install Grafana:

Update package lists:

```
sudo apt update
```

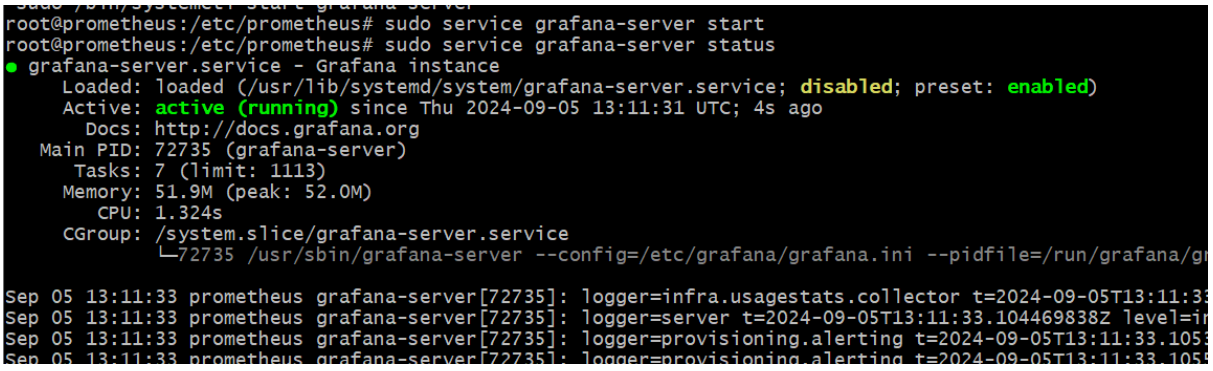
Install Grafana dependencies:

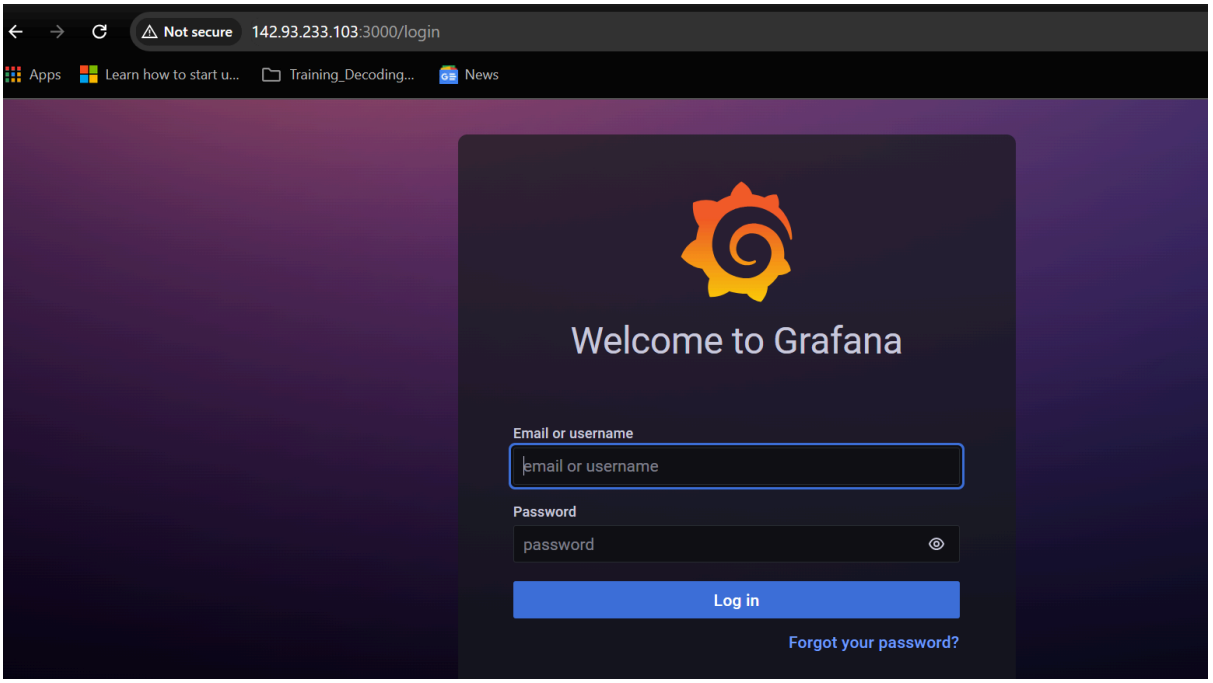
```
sudo apt-get install -y adduser libfontconfig1
```

Download the binary & run a Debian package manager:

```
wget https://dl.grafana.com/oss/release/grafana_9.3.2_amd64.deb
sudo dpkg -i grafana_9.3.2_amd64.deb
```

Start & Check status of Grafana:





Your Grafana server will be hosted at: http://[your Grafana server IP]:3000

The default Grafana login is: Username : **admin** Password : **admin**