**Assignment:Guacamole**

**Link Followed:** [**https://guacamole.apache.org/doc/gug/guacamole-docker.html**](https://guacamole.apache.org/doc/gug/guacamole-docker.html)

**->Create file guacamole**

**yogendra@yogendra:~/Documents$ vim guacamole**

**->Paste the given below given code in the file and save it.**

#!/bin/bash

podman pod create --name guacamole --publish 9080:8080 --publish 4822:4822

##Create a postgres container, give it the desired environment variables, attach it to the created pod

podman run -dt \

--pod guacamole \

--name guacamole-postgres \

-e POSTGRES\_DB=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e PGDATA=/var/lib/postgresql/data/pgdata \

-v /home/yogendra/Documents/pod/guacamole/postgres:/var/lib/postgresql/data \

postgres

podman run -dt \

--pod guacamole \

--name guacamole-guacd \

guacamole/guacd

podman run -dt \

--pod guacamole \

--name guacamole-app \

-e POSTGRES\_HOSTNAME=localhost \

-e POSTGRES\_DATABASE=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e GUACD\_HOSTNAME=localhost \

guacamole/guacamole

**yogendra@yogendra:~/Documents$ ls**

guacamole pod podman

**yogendra@yogendra:~/Documents$ cat guacamole**

#!/bin/bash

podman pod create --name guacamole --publish 9080:8080 --publish 4822:4822

##Create a postgres container, give it the desired environment variables, attach it to the created pod

podman run -dt \

--pod guacamole \

--name guacamole-postgres \

-e POSTGRES\_DB=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e PGDATA=/var/lib/postgresql/data/pgdata \

-v /home/yogendra/Documents/pod/guacamole/postgres:/var/lib/postgresql/data \

postgres

podman run -dt \

--pod guacamole \

--name guacamole-guacd \

guacamole/guacd

podman run -dt \

--pod guacamole \

--name guacamole-app \

-e POSTGRES\_HOSTNAME=localhost \

-e POSTGRES\_DATABASE=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e GUACD\_HOSTNAME=localhost \

guacamole/guacamole

**yogendra@yogendra:~/Documents$ ls**

guacamole pod podman

**yogendra@yogendra:~/Documents$ chmod 755 guacamole**

**yogendra@yogendra:~/Documents$ ./guacamole**

de349bf91aa2e62092245a20efaa70fd198464971aca976de99ae17e0eaa98fc

Trying to pull docker.io/library/postgres:latest...

Getting image source signatures

Copying blob 9c68de39d930 done

Copying blob 16b62ca80c8f done

Copying blob 02d7a77348fd done

Copying blob fbd795da1fe1 done

Copying blob a603fa5e3b41 done

Copying blob 2e441a95082c done

Copying blob 1c97f440fe14 done

Copying blob 87a3f78bc5d1 done

Copying blob bea1513c492d done

Copying blob 264b18cba666 done

Copying blob ed48cace97fa done

Copying blob e3c377e275ff done

Copying blob 86fa351e30cb done

Copying config 68f5d950dc done

Writing manifest to image destination

Storing signatures

138f3da83aec1383305df00aef7f9e6fc4bae668b8269e875a2d6690e791d88f

Trying to pull docker.io/guacamole/guacd:latest...

Getting image source signatures

Copying blob 026a71c97118 done

Copying blob 3cba2dc4f14f done

Copying blob c9121539b0a1 done

Copying blob 32820e52a00e done

Copying blob 8d943e899767 done

Copying blob a2fa12e67d99 done

Copying blob 5f0cd8de6719 done

Copying config 5edb9d05bf done

Writing manifest to image destination

Storing signatures

747a3ecee03b4044a537313be71d3520cc6d7029daf78d10669ade7dcbeaf16e

Trying to pull docker.io/guacamole/guacamole:latest...

Getting image source signatures

Copying blob e96e057aae67 done

Copying blob 191e5a2019b6 done

Copying blob c297198544b2 done

Copying blob 22cd2b580a64 done

Copying blob 4ced2591451d done

Copying blob d83c5460c007 done

Copying blob d770395f8fe4 done

Copying blob 5951b2698b9e done

Copying blob 36f25ef8cc20 done

Copying blob f783c379a474 done

Copying blob 1ce3d5c677d1 done

Copying config 035c15d672 done

Writing manifest to image destination

Storing signatures

fac266a02ff7877b9a9cd02d75c0d2c54b6afdb47b2dc2640f43b2476171b921

**yogendra@yogendra:~/Documents$ podman ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

91525ce16ea1 k8s.gcr.io/pause:3.5 3 minutes ago Up 2 minutes ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:9080->8080/tcp de349bf91aa2-infra

138f3da83aec docker.io/library/postgres:latest postgres 2 minutes ago Up 2 minutes ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:9080->8080/tcp guacamole-postgres

747a3ecee03b docker.io/guacamole/guacd:latest /bin/sh -c /usr/l... About a minute ago Up About a minute ago (healthy) 0.0.0.0:4822->4822/tcp, 0.0.0.0:9080->8080/tcp guacamole-guacd

fac266a02ff7 docker.io/guacamole/guacamole:latest /opt/guacamole/bi... 26 seconds ago Up 26 seconds ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:9080->8080/tcp guacamole-app

**yogendra@yogendra:~/Documents$ podman pod ps**

POD ID NAME STATUS CREATED INFRA ID # OF CONTAINERS

de349bf91aa2 guacamole Running 4 minutes ago 91525ce16ea1 4

yogendra@yogendra:~/Documents$

**yogendra@yogendra:~/Documents$ podman exec -it guacamole-app bash**

**guacamole@guacamole:/opt/guacamole$ /opt/guacamole/bin/initdb.sh --postgres > /tmp/initdb.sql**

guacamole@guacamole:/opt/guacamole$ **exit**

exit

**yogendra@yogendra:~/Documents$ podman cp guacamole-app:/tmp/initdb.sql .**

**yogendra@yogendra:~/Documents$ podman cp initdb.sql guacamole-postgres:/tmp**

**yogendra@yogendra:~/Documents$ podman exec -it guacamole-postgres bash**

**root@guacamole:/# psql -h 127.0.0.1 -U guacamole\_user -d guacamole\_db < /tmp/initdb.sql**

CREATE TYPE

CREATE TYPE

CREATE TYPE

CREATE TYPE

CREATE TYPE

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE TABLE

CREATE INDEX

INSERT 0 1

INSERT 0 1

INSERT 0 6

INSERT 0 3

**root@guacamole:/# exit**

exit

**yogendra@yogendra:~/Documents$ sudo vim /etc/ssh/sshd\_config**

[sudo] password for yogendra:

-> Write the below given lines in the file and save it.

**HostKeyAlgorithms +ssh-rsa**

**->Now restart sshd service and check it's status:**

**yogendra@yogendra:~/Documents$ systemctl restart sshd**

**yogendra@yogendra:~/Documents$ systemctl status sshd**

\u25cf ssh.service - OpenBSD Secure Shell server

Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2022-12-02 20:21:16 IST; 11s ago

Docs: man:sshd(8)

man:sshd\_config(5)

Process: 13855 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)

Main PID: 13856 (sshd)

Tasks: 1 (limit: 2339)

Memory: 1.7M

CPU: 91ms

CGroup: /system.slice/ssh.service

\u2514\u250013856 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 02 20:21:16 yogendra systemd[1]: Starting OpenBSD Secure Shell server...

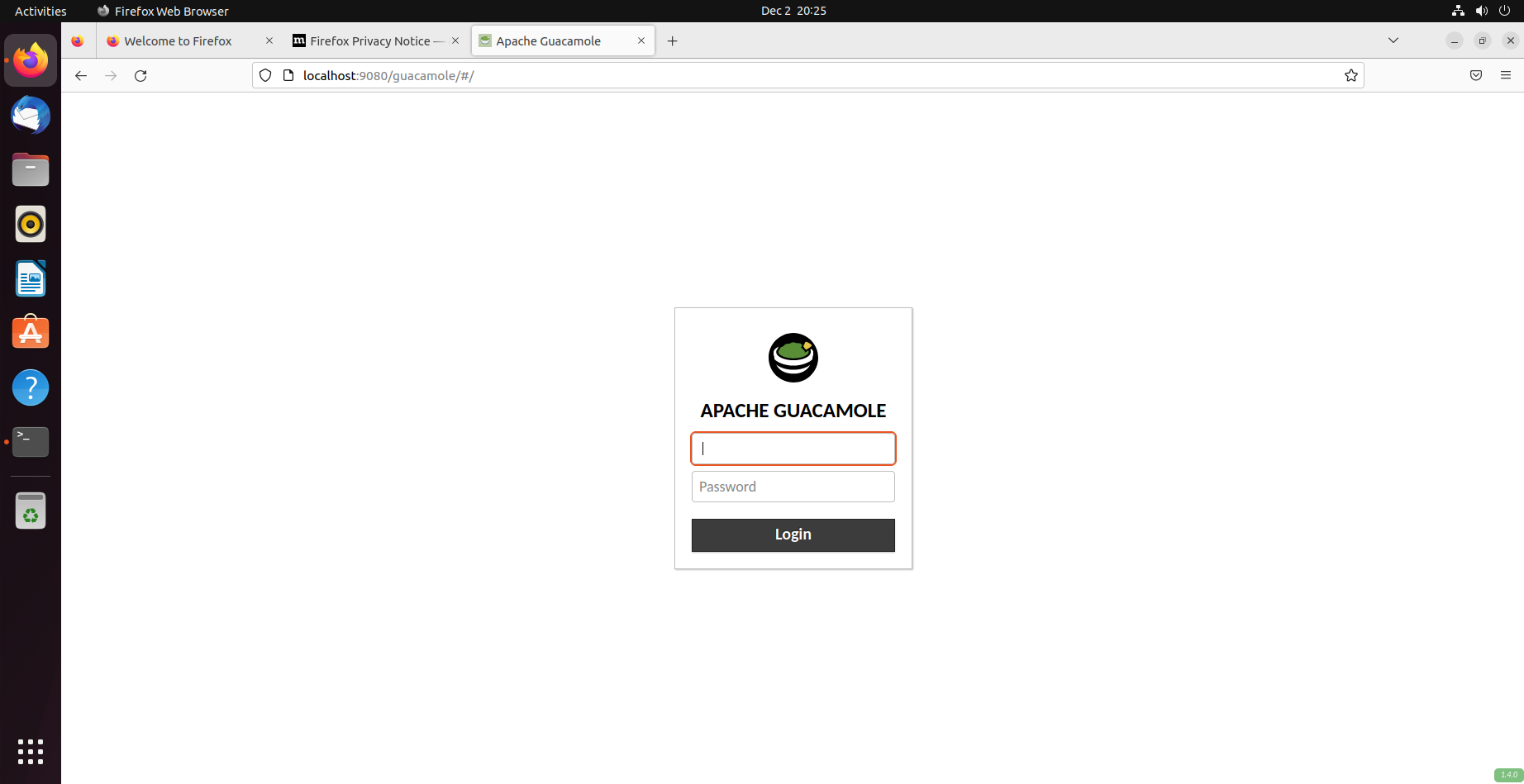
Dec 02 20:21:16 yogendra sshd[13856]: Server listening on 0.0.0.0 port 22.

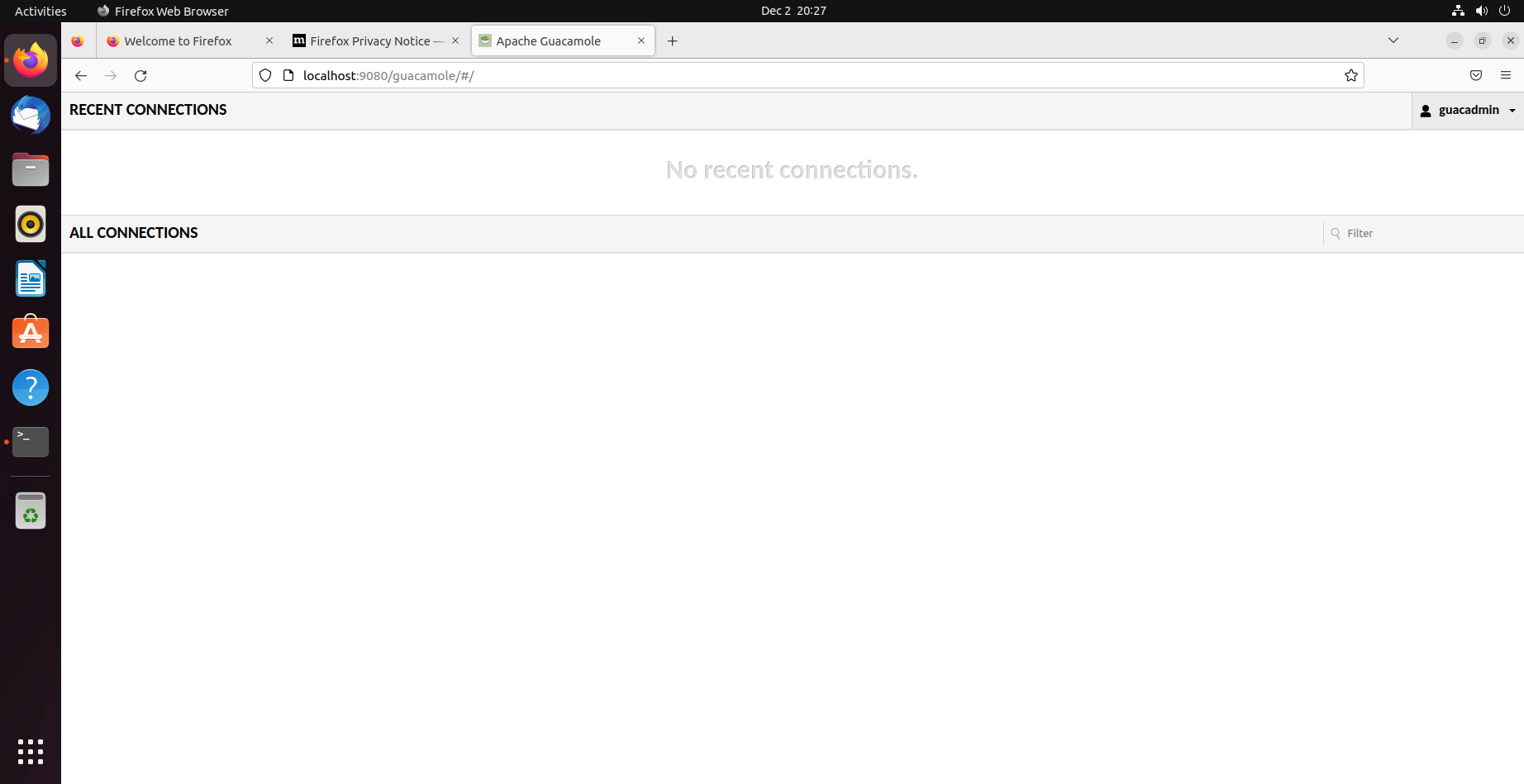
Dec 02 20:21:16 yogendra sshd[13856]: Server listening on :: port 22.

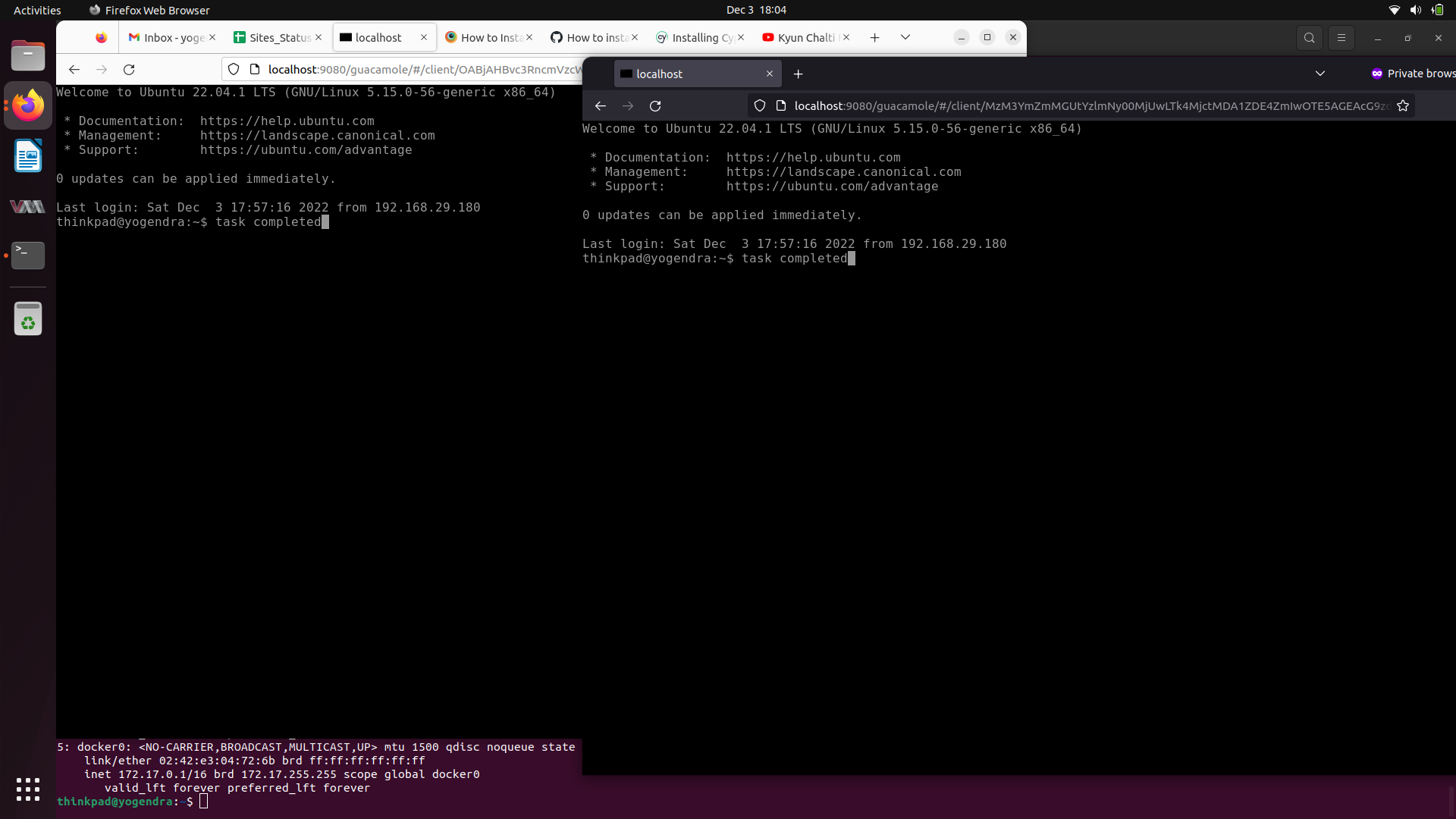
Dec 02 20:21:16 yogendra systemd[1]: Started OpenBSD Secure Shell server.

yogendra@yogendra:~/Documents$

**After this open Browser and write localhost:9080/guacamole in the search bar.**







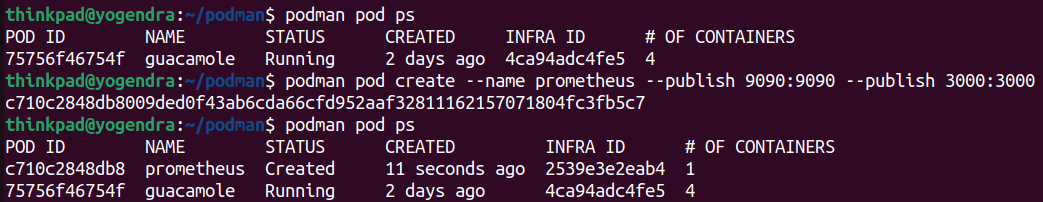
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Task:-Setup Postgres Exporter, Process Exporter and JMX Exporter using Podman and Save the metrics in Prometheus and plot the graph in Grafana.**

| **WORK BREAKDOWN STRUCTURE** | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Sr. No.** | **Sub-tasks** | **Assigned by** | **Technology** | **Status** | **Start Date** | **Planned End Date** | **Actual End Date** | **Priority** | **Remark** |
| Setup Postgres Exporter, Process Exporter and JMX Exporter using Podman and Save the metrics in Prometheus and plot the graph in Grafana. | 1 | Create JMX Exporter | Varad Sir | Linux | Open | 5-12-2022 | 7-12-2022 |  | P1 |  |
| 2 | Create Process Exporter | Varad Sir | Linux | Open | 5-12-2022 | 7-12-2022 |  | P1 |  |
| 3 | Create Postgres Exporter | Varad Sir | Linux | Close | 5-12-2022 | 7-12-2022 | 7-12-2022 | P1 |  |
| 4 | Create Prometheus container | Varad Sir | Linux | Close | 6-12-2022 | 7-12-2022 | 6-12-2022 | P1 |  |
| 5 | Create Grafana container | Varad Sir | Linux | Close | 6-12-2022 | 7-12-2022 | 6-12-2022 | P1 |  |

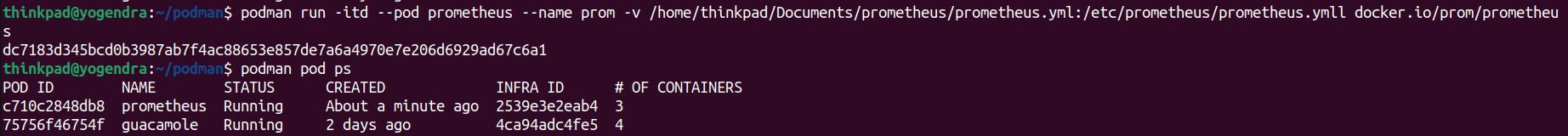
**Create Pod for Prometheus and Garafana**

**->thinkpad@yogendra:~/podman$ podman pod create --name prometheus --publish 9090:9090 --publish 3000:3000**

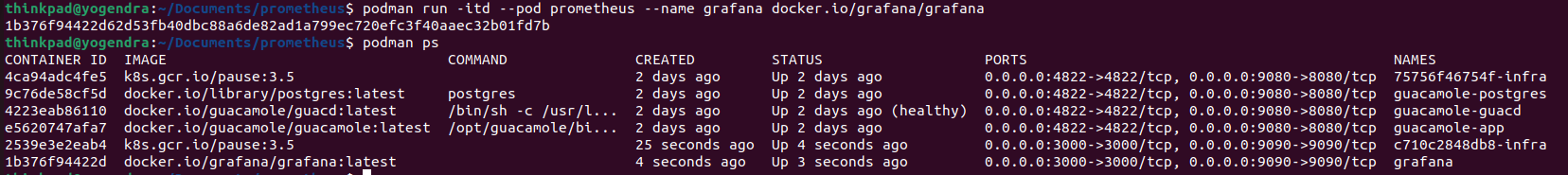
****

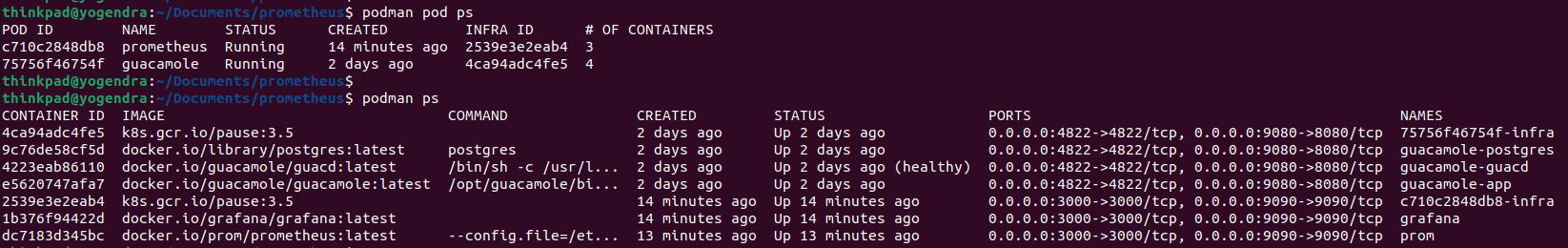
**Now create prometheus and Grafana container:**

**->thinkpad@yogendra:~/podman$ podman run -itd --pod prometheus --name prom -v /home/thinkpad/Documents/prometheus/prometheus.yml:/etc/prometheus/prometheus.ymll docker.io/prom/prometheus**

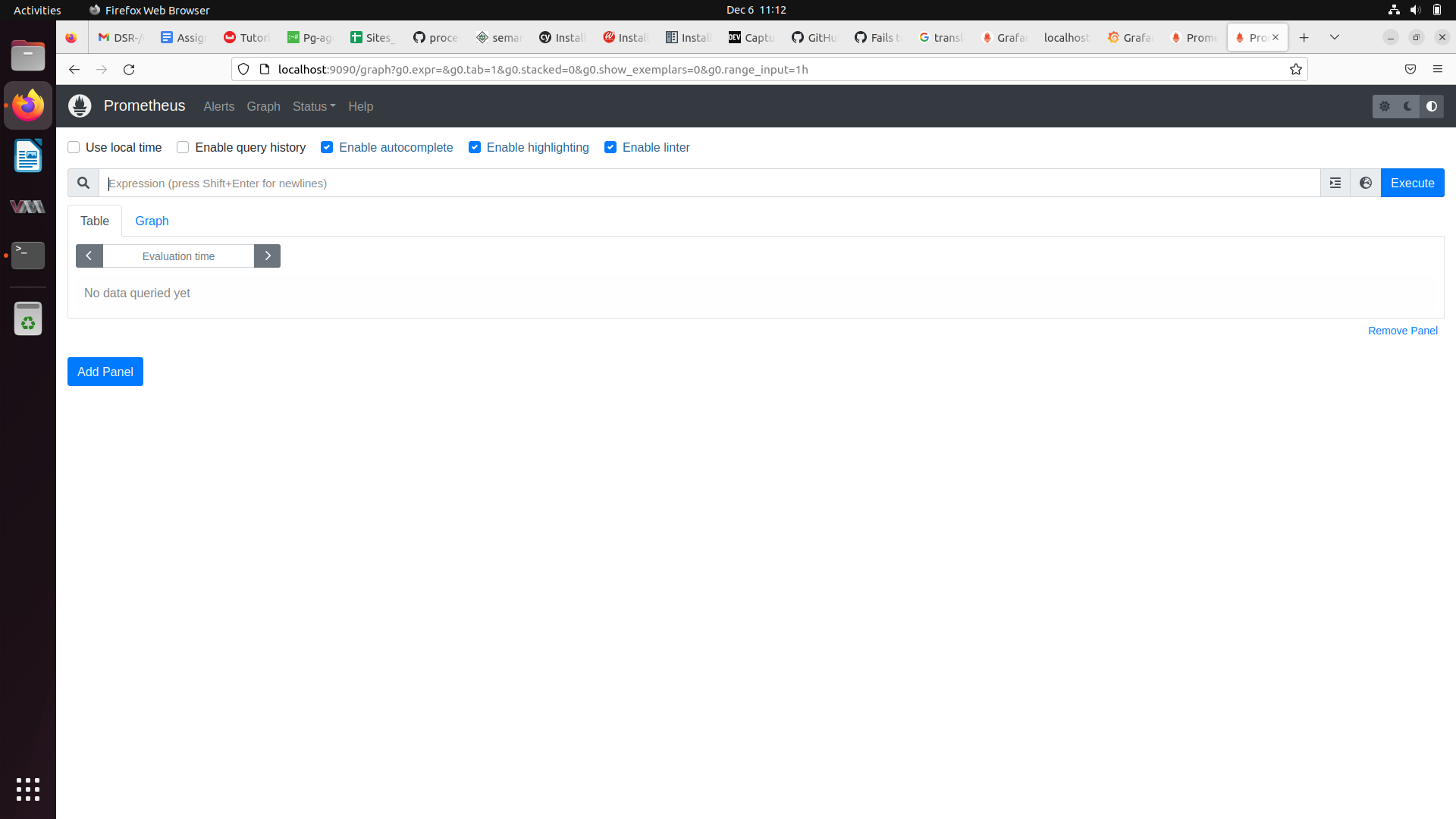
****

**->thinkpad@yogendra:~/Documents/prometheus$ podman run -itd --pod prometheus --name grafana docker.io/grafana/grafana**

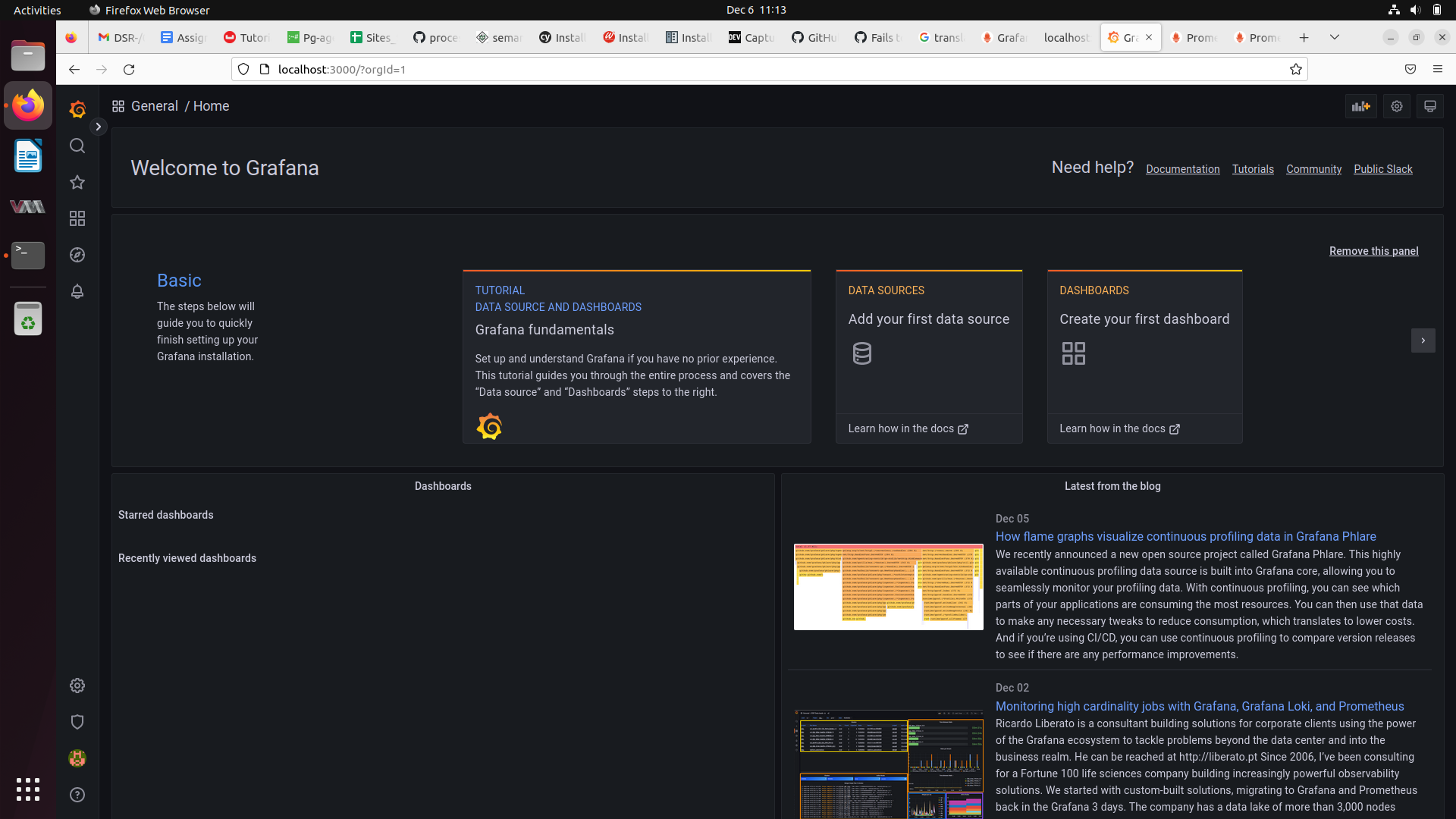
****

****

**Now Open Browser and write localhost:9090 ins search bar to open the Prometheus**

****

**To open grafana write localhost:3000 in search bar:**

****

**To Setup exporters:**

**Create file and paste the below given content and save the file:-**

#!/bin/bash

podman pod create --name guacamole --publish 9080:8080 --publish 4822:4822 --publish 9256:9256 --publish 9187:9187 --publish 5432:5432 --publish 5556:5556

##Create a postgres container, give it the desired environment variables, attach it to the created pod

podman run -dt \

--pod guacamole \

--name guacamole-postgres \

-e POSTGRES\_DB=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e PGDATA=/var/lib/postgresql/data/pgdata \

-v /home/yogendra/Documents/pod/guacamole/postgres:/var/lib/postgresql/data \

postgres

podman run -dt \

--pod guacamole \

--name guacamole-guacd \

guacamole/guacd

podman run -dt \

--pod guacamole \

--name guacamole-app \

-e POSTGRES\_HOSTNAME=localhost \

-e POSTGRES\_DATABASE=guacamole\_db \

-e POSTGRES\_USER=guacamole\_user \

-e POSTGRES\_PASSWORD=redhat \

-e GUACD\_HOSTNAME=localhost \

guacamole/guacamole

podman run -itd \

--pod guacamole \

--name pgsql-exporter \

-e DATA\_SOURCE\_NAME="postgresql://guacamole\_user:redhat@127.0.0.1:5432/guacamole\_db?sslmode=disable" \

quay.io/prometheuscommunity/postgres-exporter

podman run -itd \

--pod guacamole \

--name process-exporter \

--privileged \

-v /proc:/host/proc \

docker.io/ncabatoff/process-exporter

podman run -itd \

--pod guacamole \

--name jmx-exporter \

docker.io/bitnami/jmx-exporter:latest

**Note:** before executing the script create the prometheus.yml at any location and paste that location in the guacamole-postgres script command for persistent volume.

**->Make file executable:**

**yogendra@yogendra:~/Documents$ chmod 755 guacamole**

**-> Execute the file:**

**yogendra@yogendra:~/Documents$ ./guacamole**

55997203b6112f0c4d6b7a193ef7c6ddafd439fd53aa032638cf5fa345068213

a3c9ef13ff34c66bff9b3ab70d11b23e64ba80c1c0fdf3fa93b755ba0b9eb79c

ee26e2d66fff6ad3cf35a15d8866527d1f085dd460125046ff22ed556666049c

a33793142d12b2e0515074e38448ec0f6e7b8d319adc1463c4dd632c07c82022

4463af3de4480e8850c8fd893a35ac08aa71be3302778ec277bc9bf6176b9a0a

Trying to pull docker.io/ncabatoff/process-exporter:latest...

Getting image source signatures

Copying blob 2719841c7194 done

Copying config 6847317f2c done

Writing manifest to image destination

Storing signatures

0cf148fcf562ee027c6b94afafccae1104e0cfe305d9ea47eeb9f97567c3c1aa

Trying to pull docker.io/bitnami/jmx-exporter:latest...

Getting image source signatures

Copying blob 5f033fc38509 done

Copying blob 9b0425129f68 done

Copying config acdcb52ce5 done

Writing manifest to image destination

Storing signatures

f468b82bacb236c7ecfe28dd9b89f451e55932fe2888397ee2fc250889d81f74

****

**yogendra@yogendra:~/Documents$ podman ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

ccfbc6f79962 k8s.gcr.io/pause:3.5 About a minute ago Up About a minute ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp 55997203b611-infra

a3c9ef13ff34 docker.io/library/postgres:latest postgres About a minute ago Up About a minute ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-postgres

ee26e2d66fff docker.io/guacamole/guacd:latest /bin/sh -c /usr/l... About a minute ago Up About a minute ago (healthy) 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-guacd

a33793142d12 docker.io/guacamole/guacamole:latest /opt/guacamole/bi... About a minute ago Up About a minute ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-app

4463af3de448 quay.io/prometheuscommunity/postgres-exporter:latest About a minute ago Up About a minute ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp pgsql-exporter

0cf148fcf562 docker.io/ncabatoff/process-exporter:latest About a minute ago Up About a minute ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp process-exporter

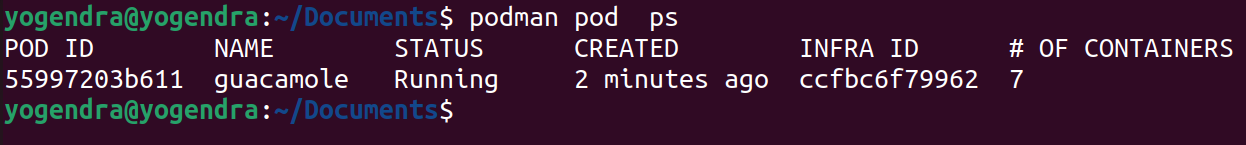
f468b82bacb2 docker.io/bitnami/jmx-exporter:latest 5556 example\_conf... 5 seconds ago Up 4 seconds ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp jmx-exporter

**yogendra@yogendra:~/Documents$ podman pod ps**

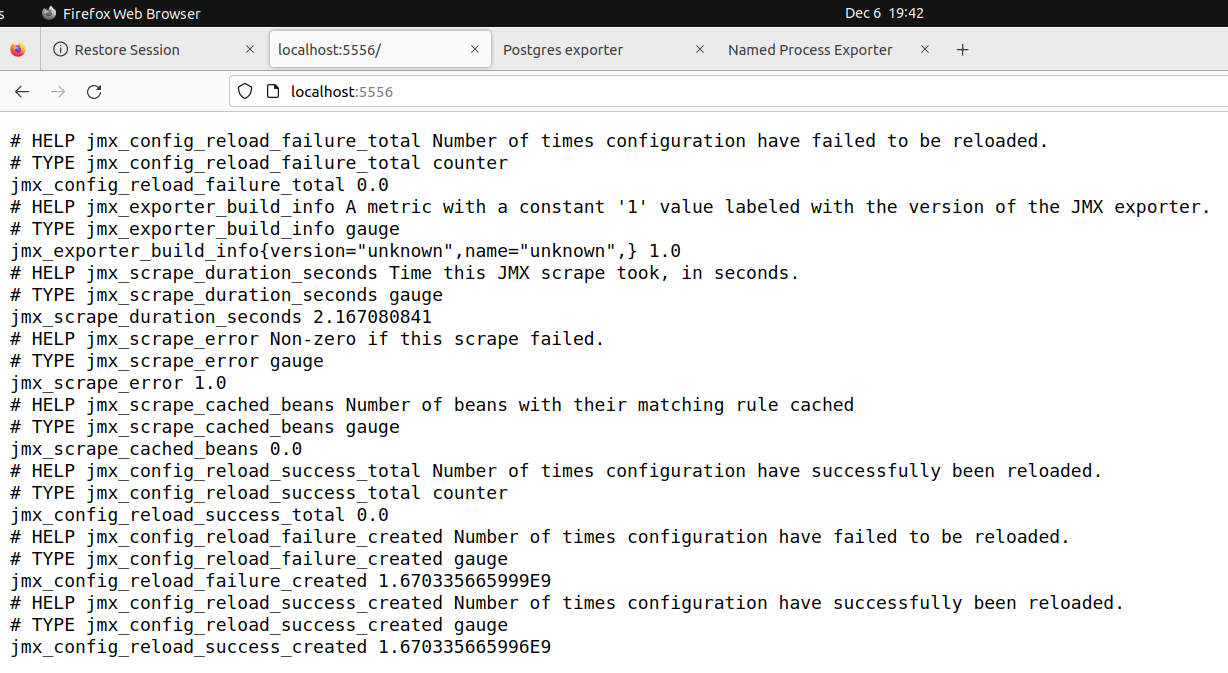
POD ID NAME STATUS CREATED INFRA ID # OF CONTAINERS

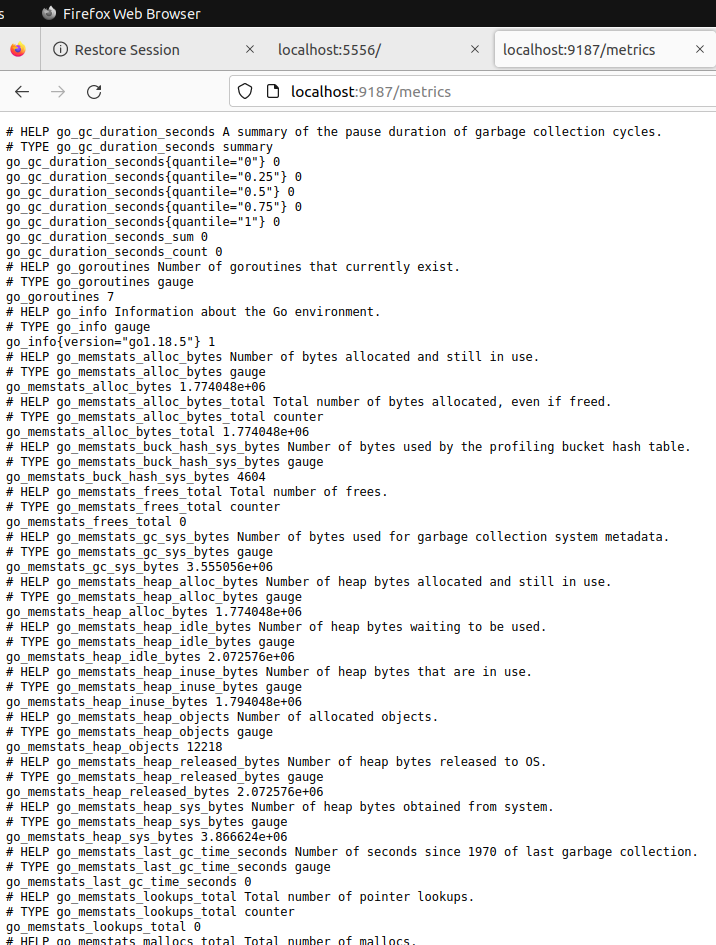
55997203b611 guacamole Running 2 minutes ago ccfbc6f79962 7

yogendra@yogendra:~/Documents$

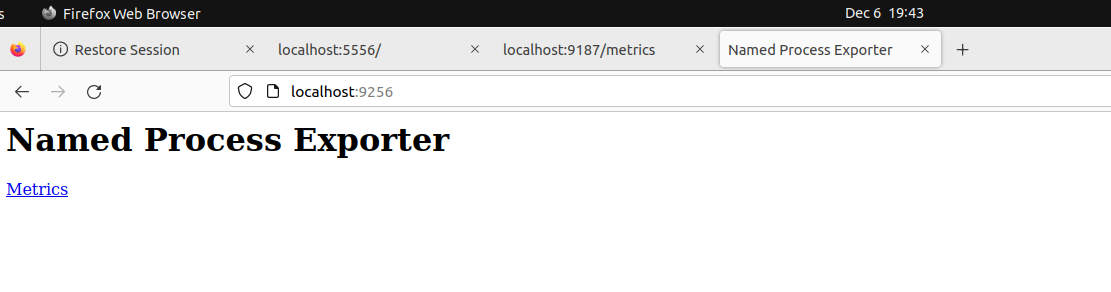
****

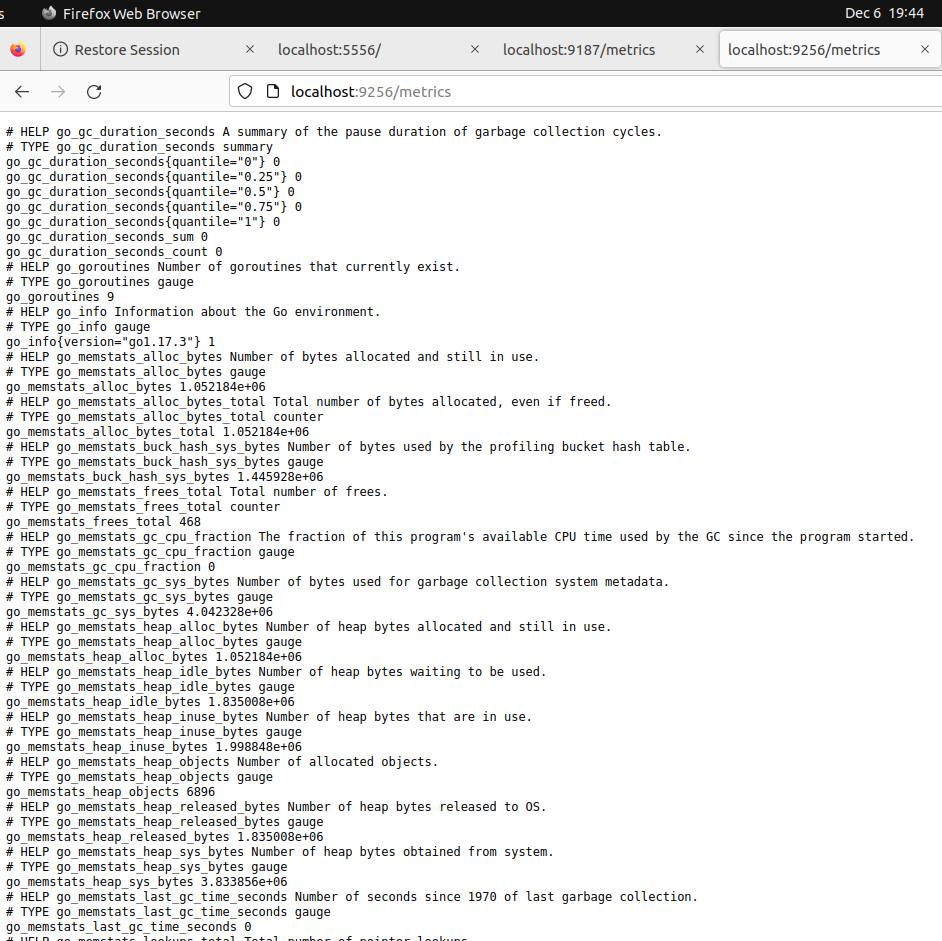
**Open browser and write localhost:5556 in search bar for jmx exporter metrics**

****

**For Postgres Exporter write localhost:9187 in search bar:**

**Process Exporter:**

****

****

**======================================================================**

# **HTTP response status codes**

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

1. Informational responses (100–199)
2. Successful responses (200–299)
3. Redirection messages (300–399)
4. Client error responses (400–499)
5. Server error responses (500–599)

## **Information responses**

**100 Continue**

This interim response indicates that the client should continue the request or ignore the response if the request is already finished.

**101 Switching Protocols**

This code is sent in response to an Upgrade request header from the client and indicates the protocol the server is switching to.

**102 Processing (WebDAV)**

This code indicates that the server has received and is processing the request, but no response is available yet.

**103 Early Hints**

This status code is primarily intended to be used with the Link header, letting the user agent start preloading resources while the server prepares a response.

## **Successful responses**

**200 OK**

The request succeeded. The result meaning of "success" depends on the HTTP method:

* GET: The resource has been fetched and transmitted in the message body.
* HEAD: The representation headers are included in the response without any message body.
* PUT or POST: The resource describing the result of the action is transmitted in the message body.
* TRACE: The message body contains the request message as received by the server.

**201 Created**

The request succeeded, and a new resource was created as a result. This is typically the response sent after POST requests, or some PUT requests.

**202 Accepted**

The request has been received but not yet acted upon. It is noncommittal, since there is no way in HTTP to later send an asynchronous response indicating the outcome of the request. It is intended for cases where another process or server handles the request, or for batch processing.

**203 Non-Authoritative Information**

This response code means the returned metadata is not exactly the same as is available from the origin server, but is collected from a local or a third-party copy. This is mostly used for mirrors or backups of another resource. Except for that specific case, the 200 OK response is preferred to this status.

**204 No Content**

There is no content to send for this request, but the headers may be useful. The user agent may update its cached headers for this resource with the new ones.

**205 Reset Content**

Tells the user agent to reset the document which sent this request.

**206 Partial Content**

This response code is used when the Range header is sent from the client to request only part of a resource.

**207 Multi-Status (WebDAV)**

Conveys information about multiple resources, for situations where multiple status codes might be appropriate.

**208 Already Reported (WebDAV)**

Used inside a <dav:propstat> response element to avoid repeatedly enumerating the internal members of multiple bindings to the same collection.

**226 IM Used (HTTP Delta encoding)**

The server has fulfilled a GET request for the resource, and the response is a representation of the result of one or more instance-manipulations applied to the current instance.

## **Redirection messages**

**300 Multiple Choices**

The request has more than one possible response. The user agent or user should choose one of them. (There is no standardised way of choosing one of the responses, but HTML links to the possibilities are recommended so the user can pick.)

**301 Moved Permanently**

The URL of the requested resource has been changed permanently. The new URL is given in the response.

**302 Found**

This response code means that the URI of requested resource has been changed *temporarily*. Further changes in the URI might be made in the future. Therefore, this same URI should be used by the client in future requests.

**303 See Other**

The server sent this response to direct the client to get the requested resource at another URI with a GET request.

**304 Not Modified**

This is used for caching purposes. It tells the client that the response has not been modified, so the client can continue to use the same cached version of the response.

**305 Use Proxy Deprecated**

Defined in a previous version of the HTTP specification to indicate that a requested response must be accessed by a proxy. It has been deprecated due to security concerns regarding in-band configuration of a proxy.

**306 unused**

This response code is no longer used; it is just reserved. It was used in a previous version of the HTTP/1.1 specification.

**307 Temporary Redirect**

The server sends this response to direct the client to get the requested resource at another URI with the same method that was used in the prior request. This has the same semantics as the 302 Found HTTP response code, with the exception that the user agent *must not* change the HTTP method used: if a POST was used in the first request, a POST must be used in the second request.

**308 Permanent Redirect**

This means that the resource is now permanently located at another URI, specified by the Location: HTTP Response header. This has the same semantics as the 301 Moved Permanently HTTP response code, with the exception that the user agent *must not* change the HTTP method used: if a POST was used in the first request, a POST must be used in the second request.

## **Client error responses**

**400 Bad Request**

The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

**401 Unauthorised**

Although the HTTP standard specifies "unauthorised", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

**402 Payment Required Experimental**

This response code is reserved for future use. The initial aim for creating this code was using it for digital payment systems, however this status code is used very rarely and no standard convention exists.

**403 Forbidden**

The client does not have access rights to the content; that is, it is unauthorised, so the server is refusing to give the requested resource. Unlike 401 Unauthorised, the client's identity is known to the server.

**404 Not Found**

The server can not find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 Forbidden to hide the existence of a resource from an unauthorised client. This response code is probably the most well known due to its frequent occurrence on the web.

**405 Method Not Allowed**

The request method is known by the server but is not supported by the target resource. For example, an API may not allow calling DELETE to remove a resource.

**406 Not Acceptable**

This response is sent when the web server, after performing server-driven content negotiation, doesn't find any content that conforms to the criteria given by the user agent.

**407 Proxy Authentication Required**

This is similar to 401 Unauthorised but authentication is needed to be done by a proxy.

**408 Request Timeout**

This response is sent on an idle connection by some servers, even without any previous request by the client. It means that the server would like to shut down this unused connection. This response is used much more since some browsers, like Chrome, Firefox 27+, or IE9, use HTTP pre-connection mechanisms to speed up surfing. Also note that some servers merely shut down the connection without sending this message.

**409 Conflict**

This response is sent when a request conflicts with the current state of the server.

**410 Gone**

This response is sent when the requested content has been permanently deleted from server, with no forwarding address. Clients are expected to remove their caches and links to the resource. The HTTP specification intends this status code to be used for "limited-time, promotional services". APIs should not feel compelled to indicate resources that have been deleted with this status code.

**411 Length Required**

Server rejected the request because the Content-Length header field is not defined and the server requires it.

**412 Precondition Failed**

The client has indicated preconditions in its headers which the server does not meet.

**413 Payload Too Large**

Request entity is larger than limits defined by server. The server might close the connection or return an Retry-After header field.

**414 URI Too Long**

The URI requested by the client is longer than the server is willing to interpret.

**415 Unsupported Media Type**

The media format of the requested data is not supported by the server, so the server is rejecting the request.

**416 Range Not Satisfiable**

The range specified by the Range header field in the request cannot be fulfilled. It's possible that the range is outside the size of the target URI's data.

**417 Expectation Failed**

This response code means the expectation indicated by the Expect request header field cannot be met by the server.

**418 I'm a teapot**

The server refuses the attempt to brew coffee with a teapot.

**421 Misdirected Request**

The request was directed at a server that is not able to produce a response. This can be sent by a server that is not configured to produce responses for the combination of scheme and authority that are included in the request URI.

**422 Unprocessable Entity (WebDAV)**

The request was well-formed but was unable to be followed due to semantic errors.

**423 Locked (WebDAV)**

The resource that is being accessed is locked.

**424 Failed Dependency (WebDAV)**

The request failed due to failure of a previous request.

**425 Too Early Experimental**

Indicates that the server is unwilling to risk processing a request that might be replayed.

**426 Upgrade Required**

The server refuses to perform the request using the current protocol but might be willing to do so after the client upgrades to a different protocol. The server sends an Upgrade header in a 426 response to indicate the required protocol(s).

**428 Precondition Required**

The origin server requires the request to be conditional. This response is intended to prevent the 'lost update' problem, where a client GETs a resource's state, modifies it and PUTs it back to the server, when meanwhile a third party has modified the state on the server, leading to a conflict.

**429 Too Many Requests**

The user has sent too many requests in a given amount of time ("rate limiting").

**431 Request Header Fields Too Large**

The server is unwilling to process the request because its header fields are too large. The request may be resubmitted after reducing the size of the request header fields.

**451 Unavailable For Legal Reasons**

The user agent requested a resource that cannot legally be provided, such as a web page censored by a government.

## **Server error responses**

**500 Internal Server Error**

The server has encountered a situation it does not know how to handle.

**501 Not Implemented**

The request method is not supported by the server and cannot be handled. The only methods that servers are required to support (and therefore that must not return this code) are GET and HEAD.

**502 Bad Gateway**

This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.

**503 Service Unavailable**

The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded. Note that together with this response, a user-friendly page explaining the problem should be sent. This response should be used for temporary conditions and the Retry-After HTTP header should, if possible, contain the estimated time before the recovery of the service. The webmaster must also take care about the caching-related headers that are sent along with this response, as these temporary condition responses should usually not be cached.

**504 Gateway Timeout**

This error response is given when the server is acting as a gateway and cannot get a response in time.

**505 HTTP Version Not Supported**

The HTTP version used in the request is not supported by the server.

**506 Variant Also Negotiates**

The server has an internal configuration error: the chosen variant resource is configured to engage in transparent content negotiation itself, and is therefore not a proper end point in the negotiation process.

**507 Insufficient Storage (WebDAV)**

The method could not be performed on the resource because the server is unable to store the representation needed to successfully complete the request.

**508 Loop Detected (WebDAV)**

The server detected an infinite loop while processing the request.

**510 Not Extended**

Further extensions to the request are required for the server to fulfill it.

**511 Network Authentication Required**

Indicates that the client needs to authenticate to gain network access.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Commands:**

**Sed**

**sed**- stream editor for filtering and transforming text

**SYNOPSIS**

sed [OPTION]... {script-only-if-no-other-script} [input-file]...

**DESCRIPTION**

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and is consequently more efficient. But it is sed's ability to filter text in a pipeline which particularly distinguishes it from other types of editors.

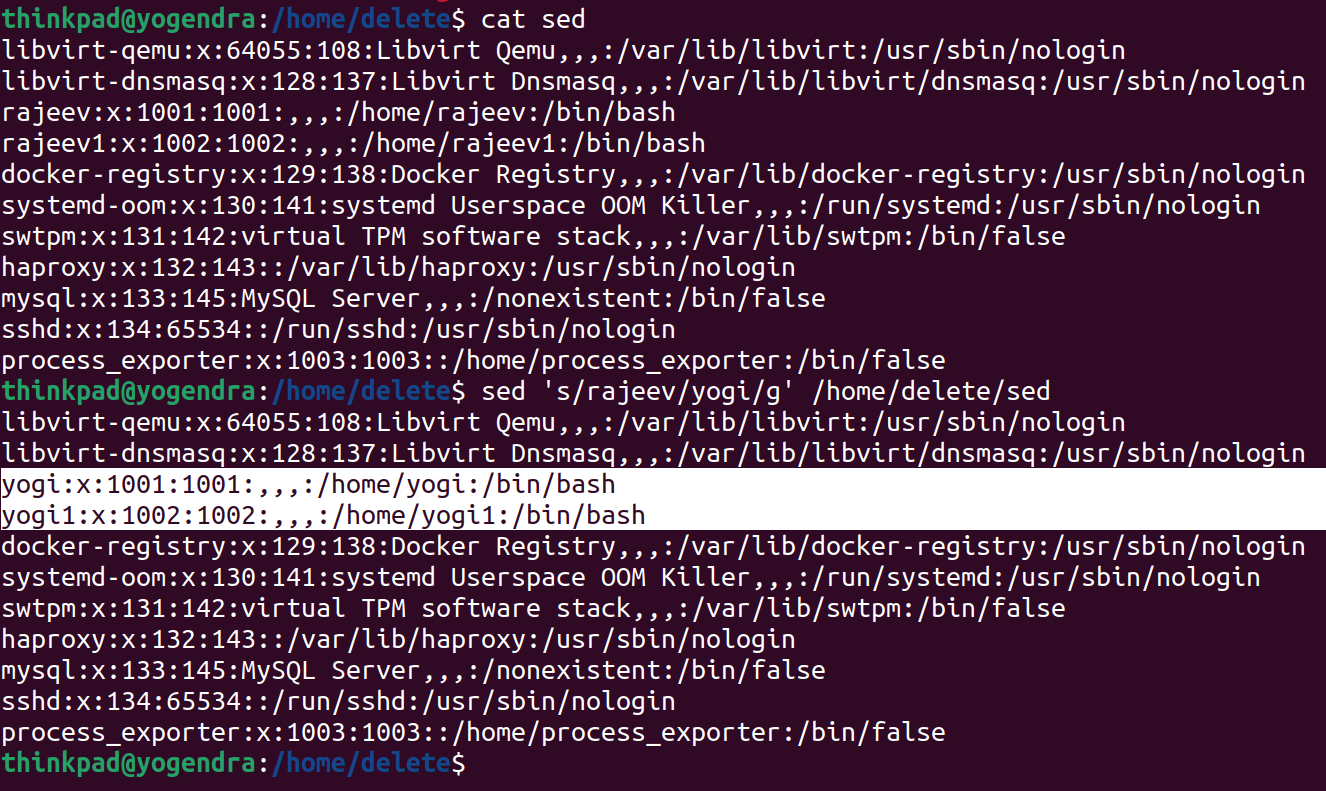
**Options:-**

-n, --quiet, --silent --> suppress automatic printing of pattern space

--debug –> annotate program execution

-e script, --expression=script –> add the script to the commands to be executed

-f script-file, --file=script-file –> add contents of script-file to command to be executed

--follow-symlinks –> follow symlinks when processing in place

—------------------------------------------------------------------------------------------------------------------------

**Awk**

**Awk –>** gawk - pattern scanning and processing language

**SYNOPSIS**

gawk [ POSIX or GNU style options ] -f program-file [ -- ] file ...

gawk [ POSIX or GNU style options ] [ -- ] program-text file ...

-i[SUFFIX], --in-place[=SUFFIX]

**DESCRIPTION**

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that match with the specified patterns and then perform the associated actions.

Awk is abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

**WHAT CAN WE DO WITH AWK?**

**1. AWK Operations:**

(a) Scans a file line by line

(b) Splits each input line into fields

(c) Compares input line/fields to pattern

(d) Performs action(s) on matched lines

**2. Useful For:**

(a) Transform data files

(b) Produce formatted reports

**3. Programming Constructs:**

(a) Format output lines

(b) Arithmetic and string operations

(c) Conditionals and loops

**Syntax:**

awk options 'selection \_criteria {action }' input-file > output-file

**Options:**

-f program-file : Reads the AWK program source from the file

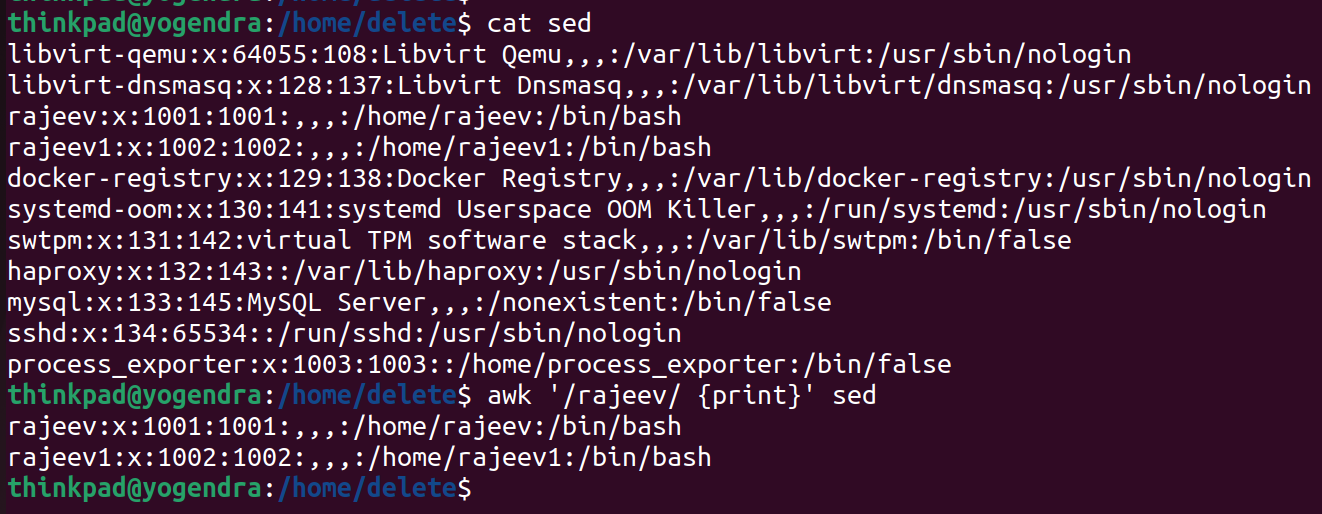
program-file, instead of from the

first command line argument.

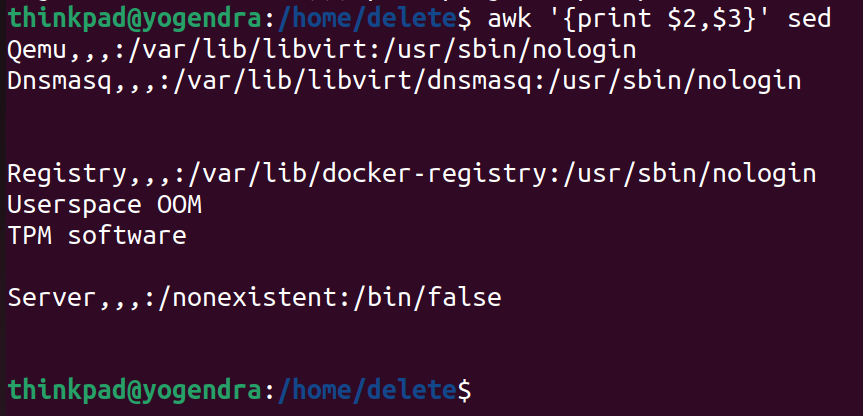
-F fs : Use fs for the input field separator

**Example:**

Print rajeev from the file sed using awk command



Print 2nd and 3rd column from file sed



**Tar**

The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

**Syntax:**

tar [options] [archive-file] [file or directory to be archived]

**Options:**

**-c :** Creates Archive

**-x :** Extract the archive

**-f :** creates archive with given filename

**-t :** displays or lists files in archived file

**-u :** archives and adds to an existing archive file

**-v :** Displays Verbose Information

**-A :** Concatenates the archive files

**-z :** zip, tells tar command that creates tar file using gzip

**-j :** filter archive tar file using tbzip

**-W :** Verify a archive file

**-r :** update or add file or directory in already existed .tar file

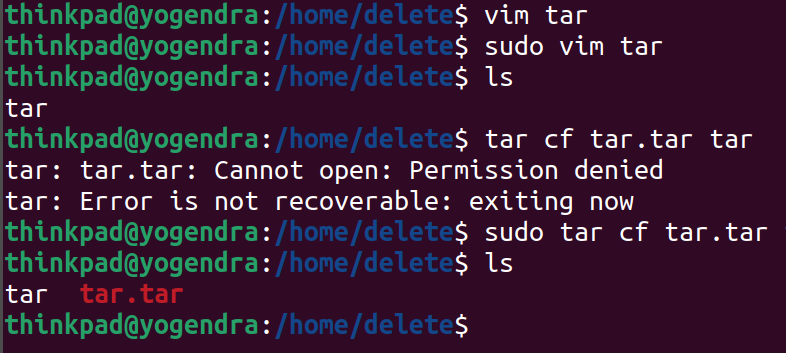
**What is an Archive file?**

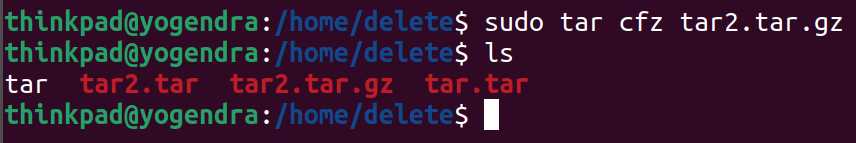
An Archive file is a file that is composed of one or more files along with metadata. Archive files are used to collect multiple data files together into a single file for easier portability and storage, or simply to compress files to use less storage space.

**Example:**

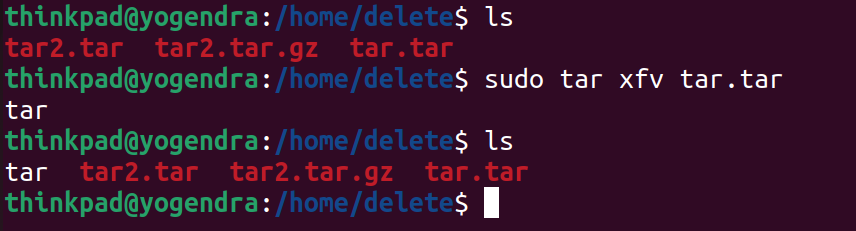
To create tar file options are c (create) and f (file)

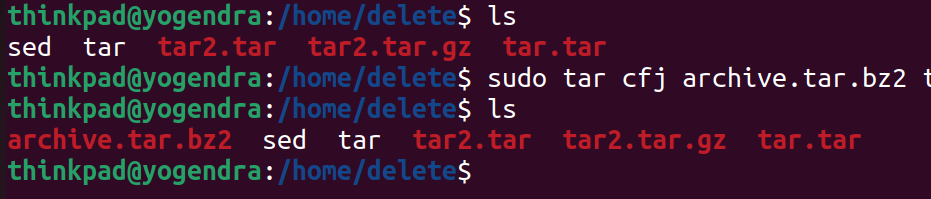
gzip compression- options used z option

s



To decompress a tar archive, options used x (extract) and f (file)



For bzip2 we use option j

—------------------------------------------------------------------------------------------------------------------------

## **What is HTTP?**

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

**Http methods :**

For HTTP/1.1, the set of common methods are defined below. This set can be expanded based on the requirements. The name of these methods is case sensitive, and they must be used in uppercase.

## **Method and Description**

**i) GET**

This method retrieves information from the given server using a given URI. GET request can retrieve the data. It can not apply other effects on the data.

**ii) HEAD**

This method is the same as the GET method. It is used to transfer the status line and header section only.

**iii) POST**

The POST request sends the data to the server. For example, file upload, customer information, etc. using the HTML forms.

**iv) PUT**

The PUT method is used to replace all the current representations of the target resource with the uploaded content.

**v) DELETE**

The DELETE method is used to remove all the current representations of the target resource, which is given by URI.

**vi) CONNECT**

This method establishes a tunnel to the server, which is identified by a given URI.

**vii) OPTIONS**

This method describes the options of communication for the target resource.

**The GET Method:**

GET is used to request data from a specified resource.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

Some notes on GET requests:

* GET requests can be cached
* GET requests remain in the browser history
* GET requests can be bookmarked
* GET requests should never be used when dealing with sensitive data
* GET requests have length restrictions
* GET requests are only used to request data (not modify)

## **The POST Method:**

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

Some notes on POST requests:

* POST requests are never cached
* POST requests do not remain in the browser history
* POST requests cannot be bookmarked
* POST requests have no restrictions on data length

## 

## **The PUT Method:**

## Put is used to send data to a server to create/update a resource.

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly has side effects of creating the same resource multiple times.

## **The HEAD Method:**

HEAD is almost identical to GET, but without the response body.

In other words, if GET /users returns a list of users, then HEAD /users will make the same request but will not return the list of users.

HEAD requests are useful for checking what a GET request will return before actually making a GET request - like before downloading a large file or response body.

## **The DELETE Method:**

The DELETE method deletes the specified resource.

## **The PATCH Method:**

The PATCH method is used to apply partial modifications to a resource.

## **The OPTIONS Method:**

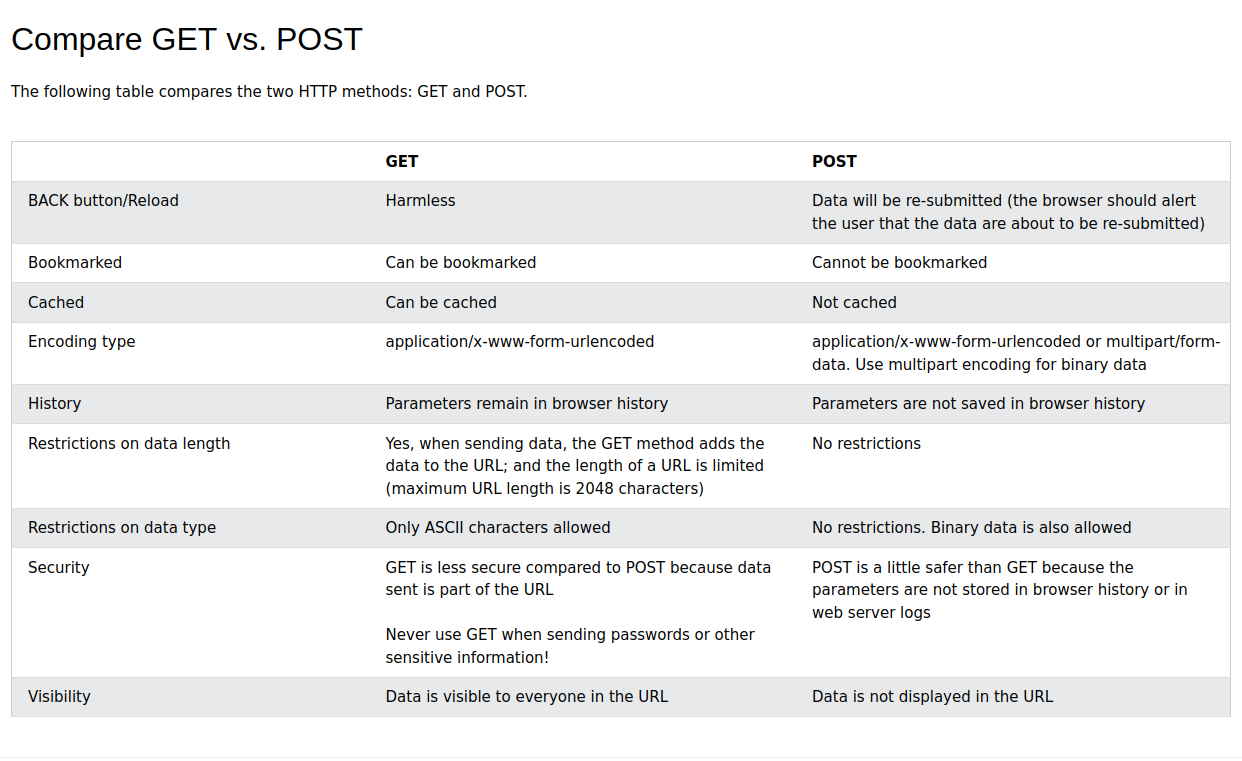
The OPTIONS method describes the communication options for the target resource.

## **The CONNECT Method:**

The CONNECT method is used to start two-way communications (a tunnel) with the requested resource.

## **The TRACE Method:**

The TRACE method is used to perform a message loop-back test that tests the path for the target resource (useful for debugging purposes).



—------------------------------------------------------------------------------------------------------------------------

**http version:**

The versions of the HTTP protocol (or "HTTP versions") commonly used on the Internet are HTTP/1.0, which is an earlier protocol including fewer functions, and HTTP/1.1, which is a later protocol including more functions. The client and server might use different versions of the HTTP protocol.

HTTP has four versions : -

* HTTP/0.9
* HTTP/1.0
* HTTP/1.1
* HTTP/2.0

Today the version in common use is HTTP/1.1 and the future will be HTTP/2.0.

—----------------------------------------------------------------------------

**http response:**

An HTTP response is made by a server to a client. The aim of the response is to provide the client with the resource it requested, or inform the client that the action it requested has been carried out; or else to inform the client that an error occurred in processing its request.

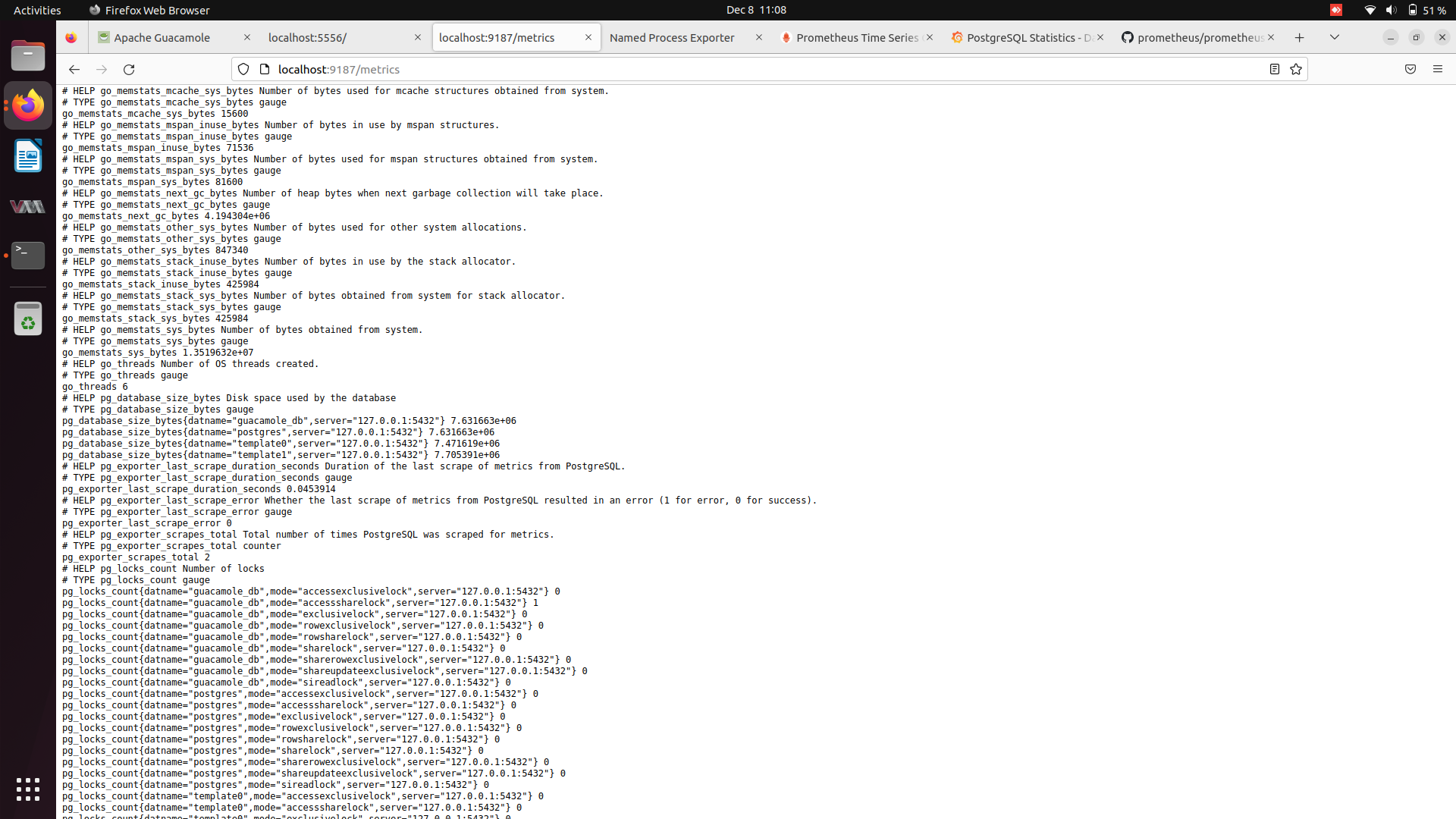
HTTP response contains:

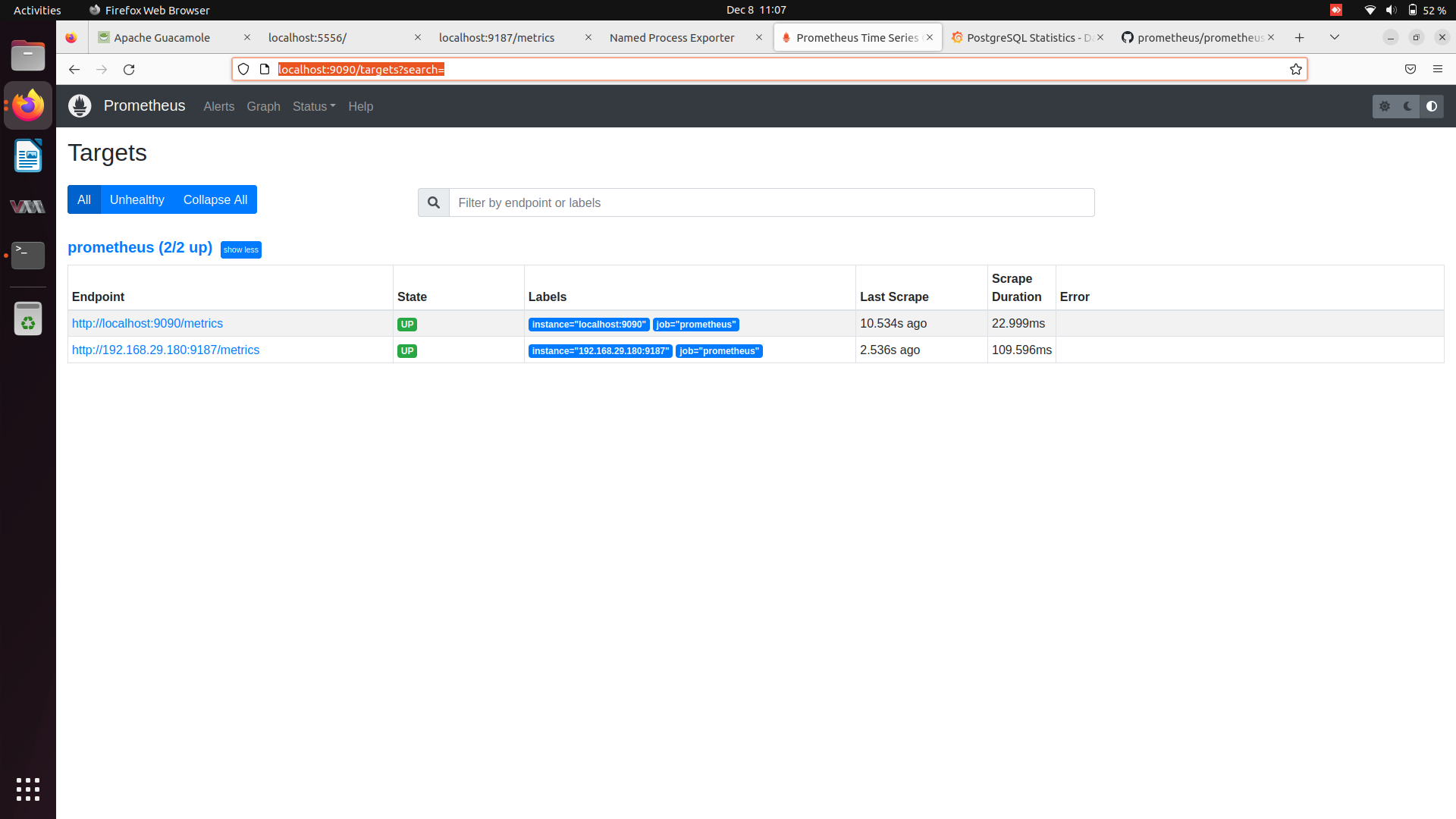
* A status line.
* A series of HTTP headers, or header fields.
* A message body, which is usually needed.

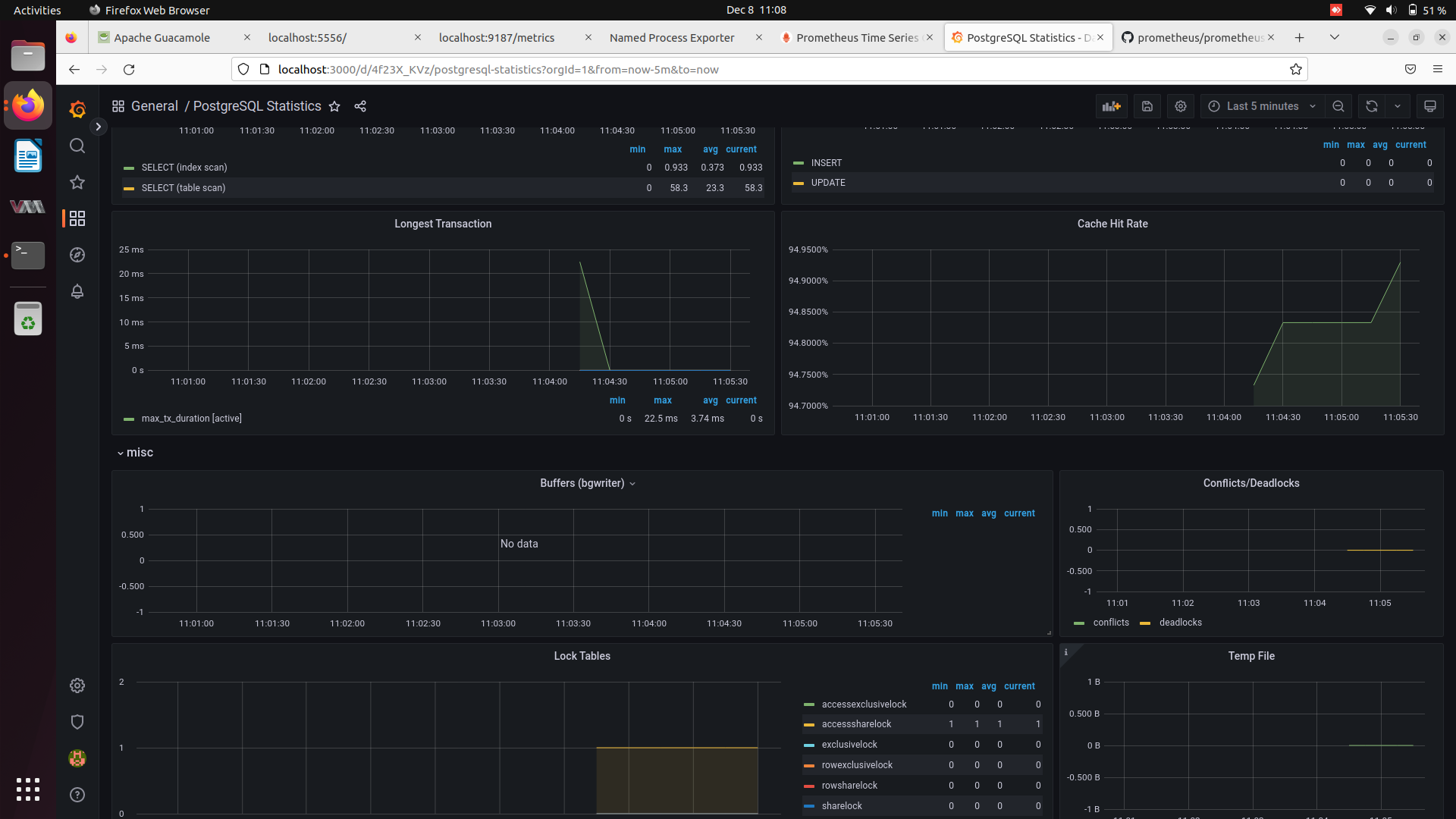
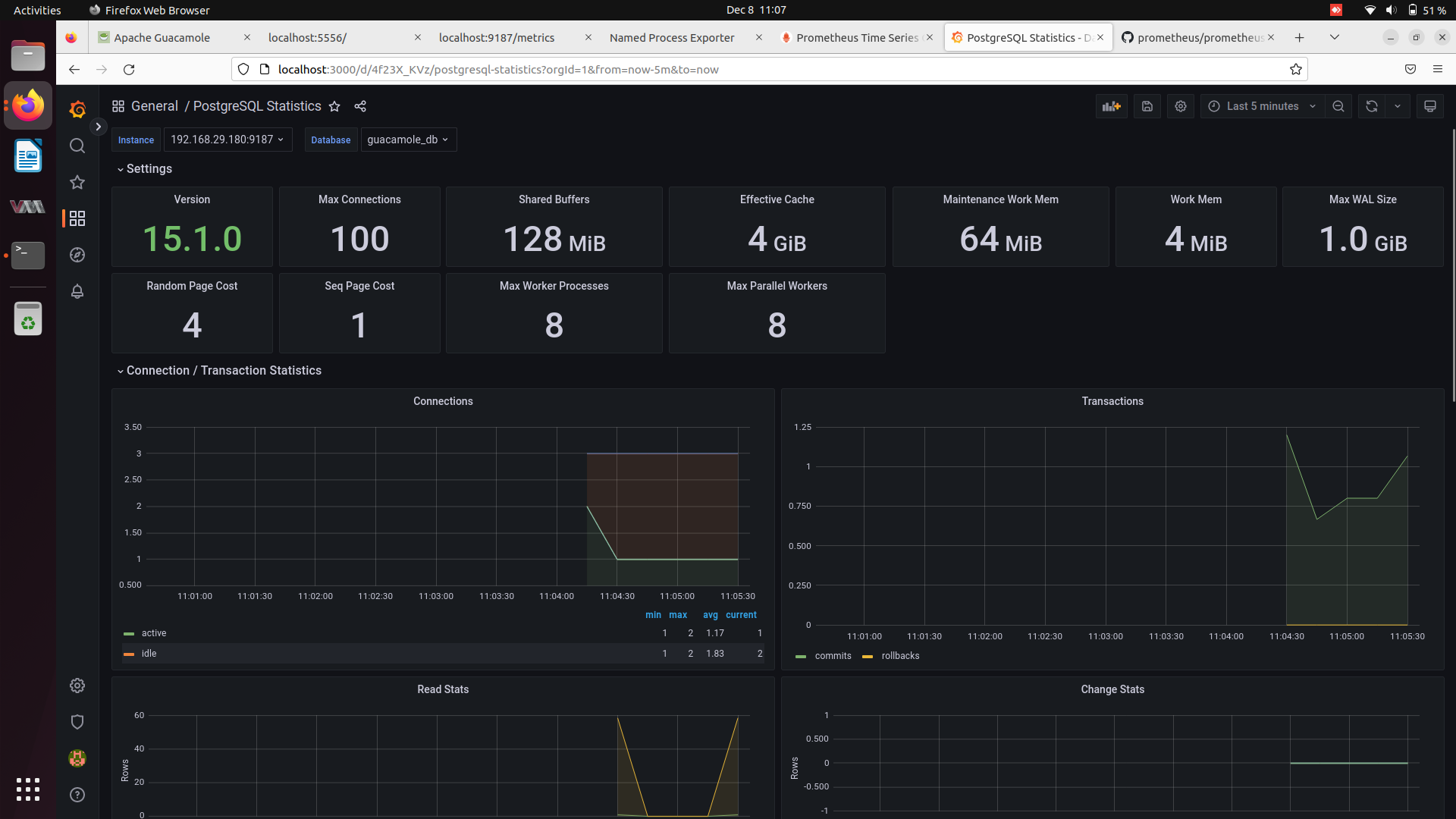
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Update [07-12-2022]

I have added the metrics in the prometheus and visualised it in the grafana.







=======================================================

**Update of date 08-12-2022**

I have tried to create the jmx exporter using sscaling/jmx-prometheus-exporter image from docker hub but it was not showing the metrics in the prometheus.

Link: <https://hub.docker.com/r/sscaling/jmx-prometheus-exporter>

Steps:

**thinkpad@yogendra:~/Documents$ sudo docker search jmx**

NAME DESCRIPTION STARS OFFICIAL AUTOMATED

sscaling/jmx-prometheus-exporter A docker image containing a released version… 12 [OK]

bitnami/jmx-exporter 6

jmxtrans/jmxtrans Monitor JVM via JMX 3 [OK]

logzio/jmx2graphite JMX to Graphite every x seconds in one comma… 1

newrelic/k8s-nri-jmx New Relic Infrastructure JMX Integration (ht… 1

opsguruhub/jmx-web-test 1

sematext/jmxinit 0

loyaltyone/jmxtrans JMX Monitoring Tool 0 [OK]

scrapinghub/jmx\_exporter A process for exposing JMX Beans via HTTP fo… 0 [OK]

banzaicloud/jmx\_exporter 0

koarch/jmx-exporter Arm64 Image for Jmx Exporter based on https:… 0

banzaicloud/jmx-javaagent 0

jmxtrans/testing Images for automated testing 0

gradiant/jmxproxy Jmxproxy project 0 [OK]

dwpdigital/jmx-exporter A process for exposing JMX Beans via HTTP fo… 0

aristanetworks/jmx-exporter prometheus jmx exporter standalone http serv… 0

webgames/jmx-stats jmx-stats 0 [OK]

flokkr/jmxpromo-sidecar 0

ruoyuchen/jmxexporters jmx 0 [OK]

ykorzikowski/jmx-prometheus-exporter jmx exporter for prometheus 0

reasland/jmx-prometheus-exporter Prometheus HTTP JMX Exporter 0

kunickiaj/jmxtrans 0

pebbletech/jmx-cloudwatch Forwards JMX->CloudWatch, highly customized … 0

strimzi/jmxtrans 0

saagie/jmx-prometheus-javaagent 0

**thinkpad@yogendra:~/Documents$ sudo docker pull sscaling/jmx-prometheus-exporter**

Using default tag: latest

latest: Pulling from sscaling/jmx-prometheus-exporter

169185f82c45: Pull complete

ca14bef7a00d: Pull complete

626030083f7e: Pull complete

6a27f86a22f8: Pull complete

ed646a24af12: Pull complete

9e501562c1eb: Pull complete

7dfb5bba2f08: Pull complete

5b4f1f5bdfe9: Pull complete

Digest: sha256:48e3bd31f132146751e449429a3e8a260b141f2acd587878fc50b3eda4466c5d

Status: Downloaded newer image for sscaling/jmx-prometheus-exporter:latest

docker.io/sscaling/jmx-prometheus-exporter:latest

**thinkpad@yogendra:~/Documents$ sudo docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

ghcr.io/coroot/coroot-pg-agent latest 74b1ede52a21 2 months ago 136MB

sscaling/jmx-prometheus-exporter latest 46303433ca79 3 years ago 131MB

**thinkpad@yogendra:~/Documents$ sudo podman run -itd --name jmx -p "5556:5556" -v /home/thinkpad/Documents/config.yml:/opt/jmx\_exporter/config.yml 46303433ca79**

Error: short-name "46303433ca79" did not resolve to an alias and no unqualified-search registries are defined in "/etc/containers/registries.conf"

**thinkpad@yogendra:~/Documents$ sudo podman run -itd --name jmx -p "5556:5556" -v /home/thinkpad/Documents/config.yml:/opt/jmx\_exporter/config.yml sscaling/jmx-prometheus-exporter**

Error: short-name "sscaling/jmx-prometheus-exporter" did not resolve to an alias and no unqualified-search registries are defined in "/etc/containers/registries.conf"

**thinkpad@yogendra:~/Documents$ sudo podman run -itd --name jmx -p "5557:5557" -v /home/thinkpad/Documents/config.yml:/opt/jmx\_exporter/config.yml docker.io/sscaling/jmx-prometheus-exporter**

Error: error creating container storage: the container name "jmx" is already in use by "2537d2a1a3b7372bdf638927d49ce82b988c8264c04e09e7cb73fc602efc8d86". You have to remove that container to be able to reuse that name.: that name is already in use

thinkpad@yogendra:~/Documents$ podman rm -f jmx

Error: no container with name or ID "jmx" found: no such container

thinkpad@yogendra:~/Documents$ podman ps -a |grep jmx

3e39cb433d31 docker.io/bitnami/jmx-exporter:latest 5556 example\_conf... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp jmx-exporter

thinkpad@yogendra:~/Documents$ sudo podman ps -a |grep jmx

2537d2a1a3b7 docker.io/sscaling/jmx-prometheus-exporter:latest usr/local/bin/dum... About a minute ago Created 0.0.0.0:5556->5556/tcp jmx

**thinkpad@yogendra:~/Documents$ sudo podman rm -f jmx**

2537d2a1a3b7372bdf638927d49ce82b988c8264c04e09e7cb73fc602efc8d86

**thinkpad@yogendra:~/Documents$ podman run -itd --name jmx -p "5557:5557" -v /home/thinkpad/Documents/config.yml:/opt/jmx\_exporter/config.yml docker.io/sscaling/jmx-prometheus-exporter**

22170b0eba449afc82466b851dfc8d04a710e31c2ac352ceb62a5199bc05a61d

**thinkpad@yogendra:~/Documents$ podman ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f53e26a8d363 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp 0691168b2050-infra

33c33201d1a7 docker.io/library/postgres:latest postgres 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-postgres

a8ef8d7754b8 docker.io/guacamole/guacd:latest /bin/sh -c /usr/l... 8 hours ago Up 8 hours ago (healthy) 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-guacd

a2980deb94d2 docker.io/guacamole/guacamole:latest /opt/guacamole/bi... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-app

00ac7fbb49a6 docker.io/ncabatoff/process-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp process-exporter

3e39cb433d31 docker.io/bitnami/jmx-exporter:latest 5556 example\_conf... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp jmx-exporter

a387a4d1548b quay.io/prometheuscommunity/postgres-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp pgsql-exporter

a627e5d86b51 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp 5c3e988e556f-infra

0b1d6e291798 docker.io/prom/prometheus:latest --config.file=/et... 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp prom

5e8ea1092474 docker.io/grafana/grafana:latest 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp grafana

22170b0eba44 docker.io/sscaling/jmx-prometheus-exporter:latest usr/local/bin/dum... 11 seconds ago Up 11 seconds ago 0.0.0.0:5557->5557/tcp jmx

thinkpad@yogendra:~/Documents$ ls

config.yml guacamole initdb.sql pod podman prometheus script

thinkpad@yogendra:~/Documents$ podman exec -it prom sh

/prometheus $ vi /etc/prometheus/prometheus.yml

/prometheus $

thinkpad@yogendra:~/Documents$ podman restart prom

0b1d6e291798bbc71ca0c3472d12f8236ac1a6ca32805d1f9776c82b3df751eb

thinkpad@yogendra:~/Documents$ podman restart prom

0b1d6e291798bbc71ca0c3472d12f8236ac1a6ca32805d1f9776c82b3df751eb

**thinkpad@yogendra:~/Documents$ podman ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f53e26a8d363 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp 0691168b2050-infra

33c33201d1a7 docker.io/library/postgres:latest postgres 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-postgres

a8ef8d7754b8 docker.io/guacamole/guacd:latest /bin/sh -c /usr/l... 8 hours ago Up 8 hours ago (healthy) 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-guacd

a2980deb94d2 docker.io/guacamole/guacamole:latest /opt/guacamole/bi... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-app

00ac7fbb49a6 docker.io/ncabatoff/process-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp process-exporter

3e39cb433d31 docker.io/bitnami/jmx-exporter:latest 5556 example\_conf... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp jmx-exporter

a387a4d1548b quay.io/prometheuscommunity/postgres-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp pgsql-exporter

a627e5d86b51 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp 5c3e988e556f-infra

5e8ea1092474 docker.io/grafana/grafana:latest 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp grafana

22170b0eba44 docker.io/sscaling/jmx-prometheus-exporter:latest usr/local/bin/dum... 3 minutes ago Up 3 minutes ago 0.0.0.0:5557->5557/tcp jmx

thinkpad@yogendra:~/Documents$ podman ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f53e26a8d363 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp 0691168b2050-infra

33c33201d1a7 docker.io/library/postgres:latest postgres 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-postgres

a8ef8d7754b8 docker.io/guacamole/guacd:latest /bin/sh -c /usr/l... 8 hours ago Up 8 hours ago (healthy) 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-guacd

a2980deb94d2 docker.io/guacamole/guacamole:latest /opt/guacamole/bi... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp guacamole-app

00ac7fbb49a6 docker.io/ncabatoff/process-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp process-exporter

3e39cb433d31 docker.io/bitnami/jmx-exporter:latest 5556 example\_conf... 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp jmx-exporter

a387a4d1548b quay.io/prometheuscommunity/postgres-exporter:latest 8 hours ago Up 8 hours ago 0.0.0.0:4822->4822/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:5556->5556/tcp, 0.0.0.0:9080->8080/tcp, 0.0.0.0:9187->9187/tcp, 0.0.0.0:9256->9256/tcp pgsql-exporter

a627e5d86b51 k8s.gcr.io/pause:3.5 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp 5c3e988e556f-infra

0b1d6e291798 docker.io/prom/prometheus:latest --config.file=/et... 8 hours ago Exited (2) About a minute ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp prom

5e8ea1092474 docker.io/grafana/grafana:latest 8 hours ago Up 8 hours ago 0.0.0.0:3000->3000/tcp, 0.0.0.0:9090->9090/tcp grafana

22170b0eba44 docker.io/sscaling/jmx-prometheus-exporter:latest usr/local/bin/dum... 3 minutes ago Up 3 minutes ago 0.0.0.0:5557->5557/tcp jmx

**thinkpad@yogendra:~/Documents$ podman exec -it prom sh**

Error: can only create exec sessions on running containers: container state improper

**thinkpad@yogendra:~/Documents$ podman rm -f prom**

0b1d6e291798bbc71ca0c3472d12f8236ac1a6ca32805d1f9776c82b3df751eb

**thinkpad@yogendra:~/Documents$ podman run -itd --pod prometheus --name prom -v /home/thinkpad/Documents/prometheus/prometheus.yml:/etc/prometheus/prometheus.ymll docker.io/prom/prometheus**

9b3fb259dd96efa1a1b8a8f53d7b0e8d73391a1e4879c902620336afa5fb7ab5

**thinkpad@yogendra:~/Documents$ vim /home/thinkpad/Documents/prometheus/prometheus.yml**

**thinkpad@yogendra:~/Documents$ podman restart prom**

9b3fb259dd96efa1a1b8a8f53d7b0e8d73391a1e4879c902620336afa5fb7ab5

thinkpad@yogendra:~/Documents$ podman exec -it prom sh

**/prometheus $ vi /etc/prometheus/prometheus.yml**

**thinkpad@yogendra:~/Documents$ podman restart prom**

9b3fb259dd96efa1a1b8a8f53d7b0e8d73391a1e4879c902620336afa5fb7ab5

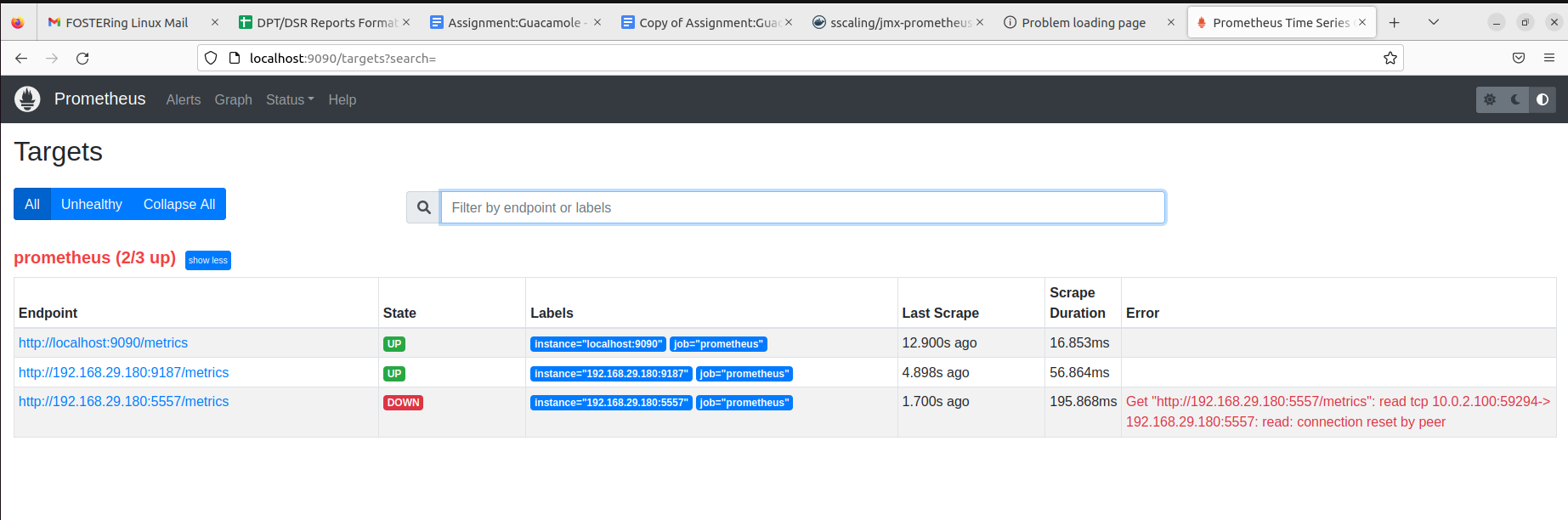
**thinkpad@yogendra:~/Documents$ podman restart prom**

9b3fb259dd96efa1a1b8a8f53d7b0e8d73391a1e4879c902620336afa5fb7ab5

**thinkpad@yogendra:~/Documents$ hostname -I**

192.168.29.180 192.168.122.1 172.17.0.1 10.88.0.1 2405:201:4019:6026:2195:7f50:f998:9263 2405:201:4019:6026:c8ae:6ad8:8145:398d

thinkpad@yogendra:~/Documents$

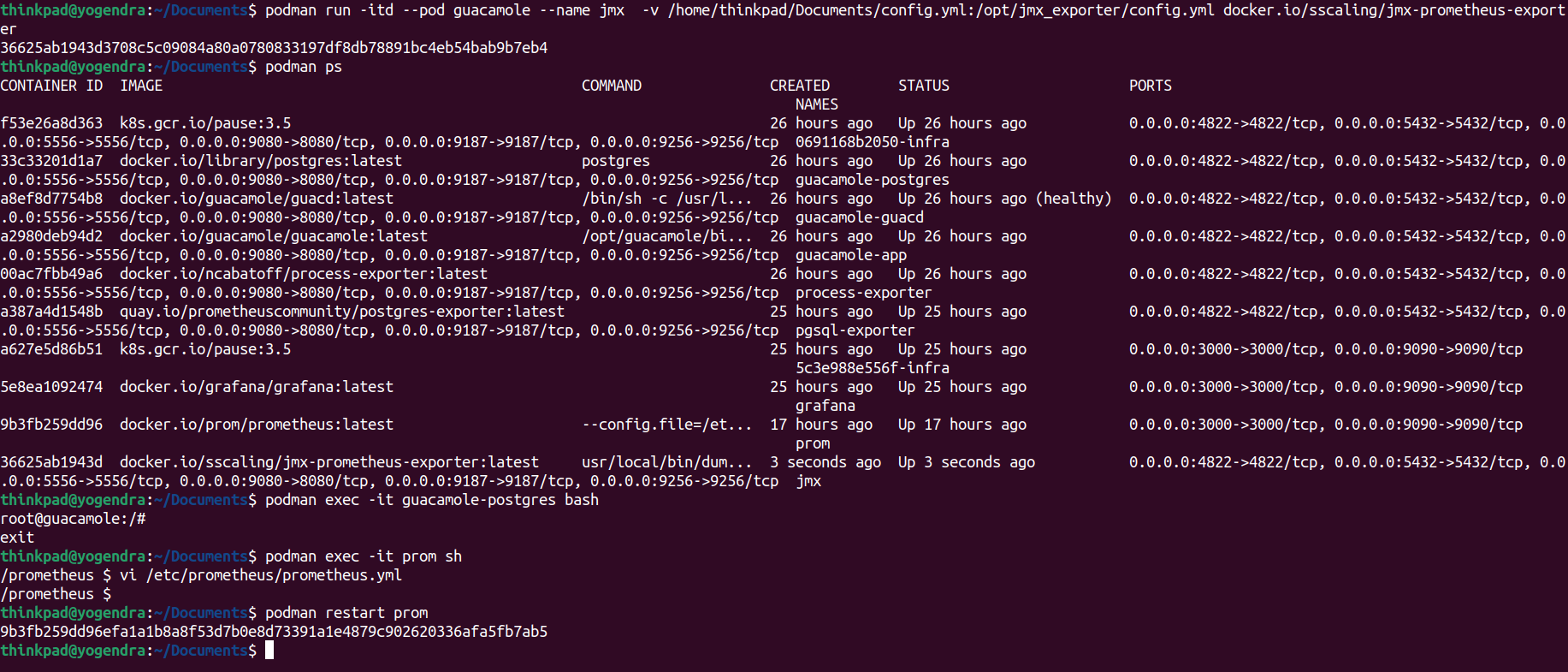


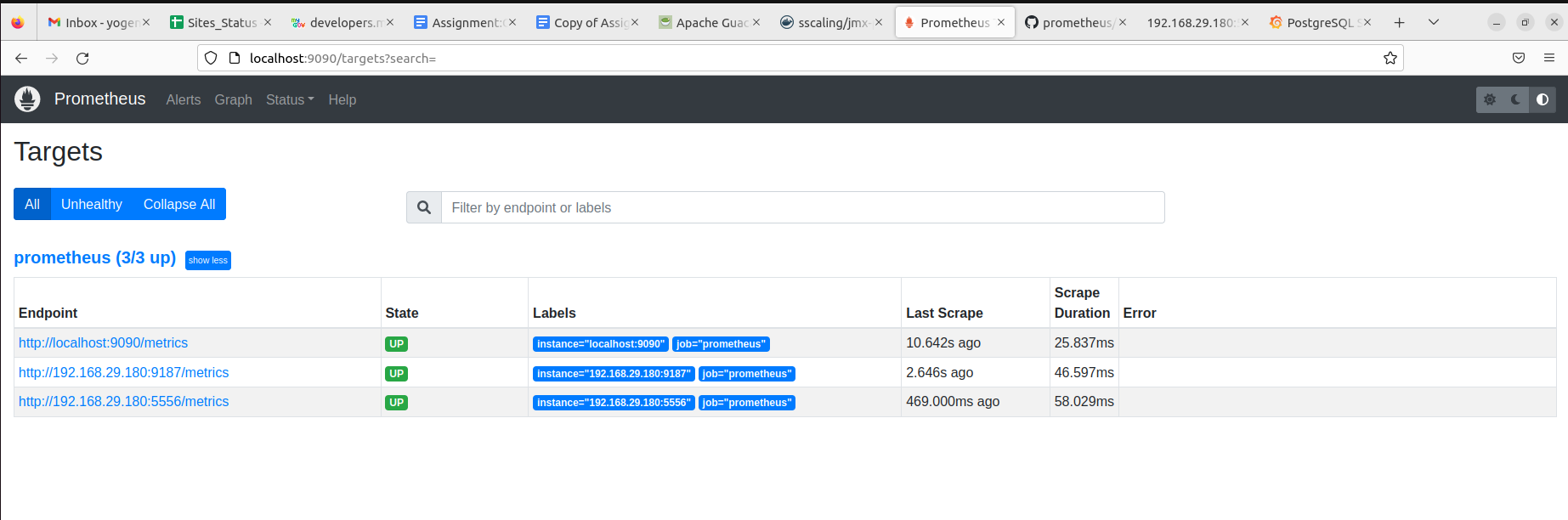
—---------------------------------------------------------------------------------------------

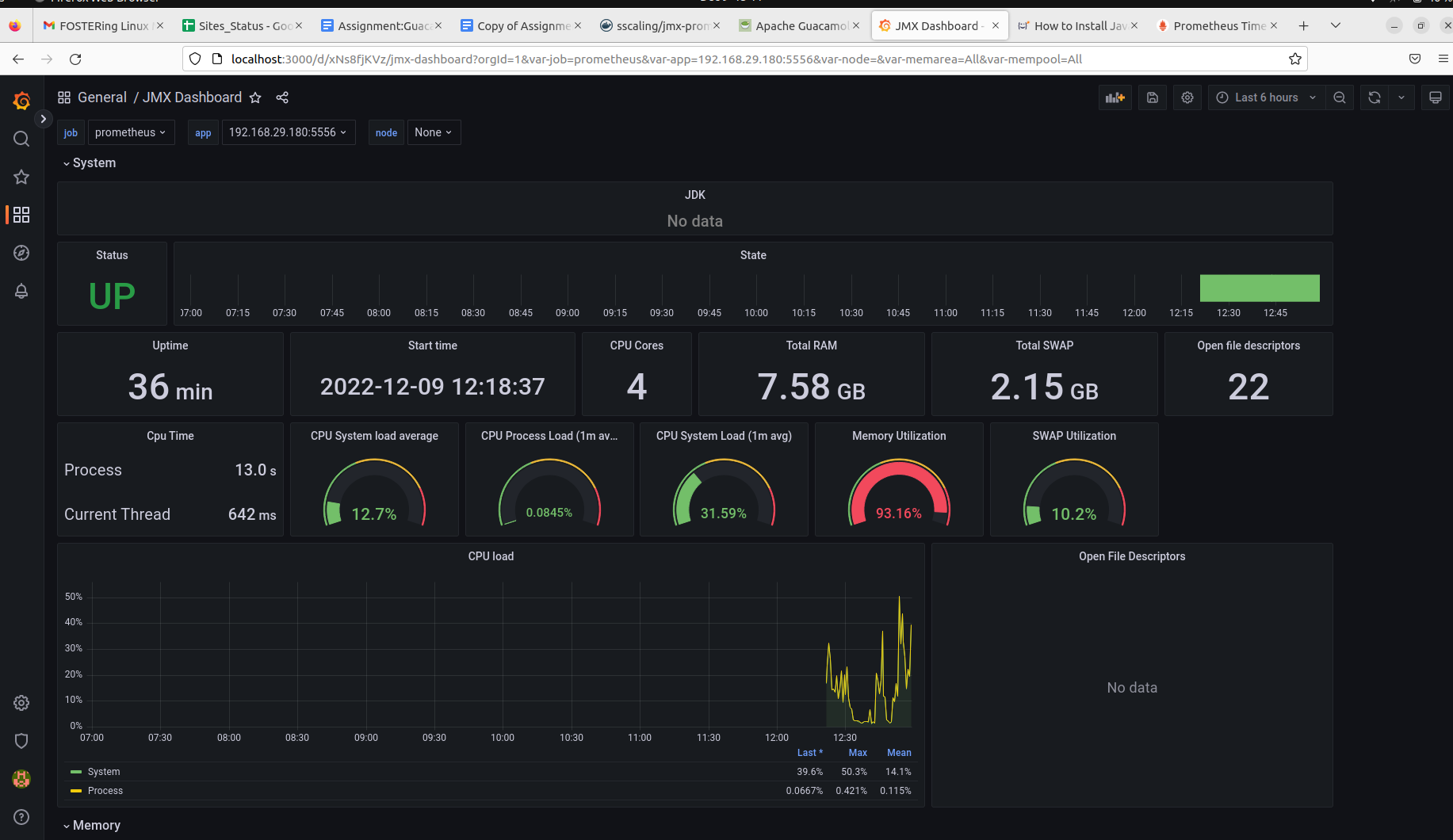
Update of 09-12-2022

I have run the below given command to setup the jmx container

thinkpad@yogendra:~/Documents$ podman run -itd --pod guacamole --name jmx -v /home/thinkpad/Documents/config.yml:/opt/jmx\_exporter/config.yml docker.io/sscaling/jmx-prometheus-exporter







—---------------------------------------------------------------------------------------------

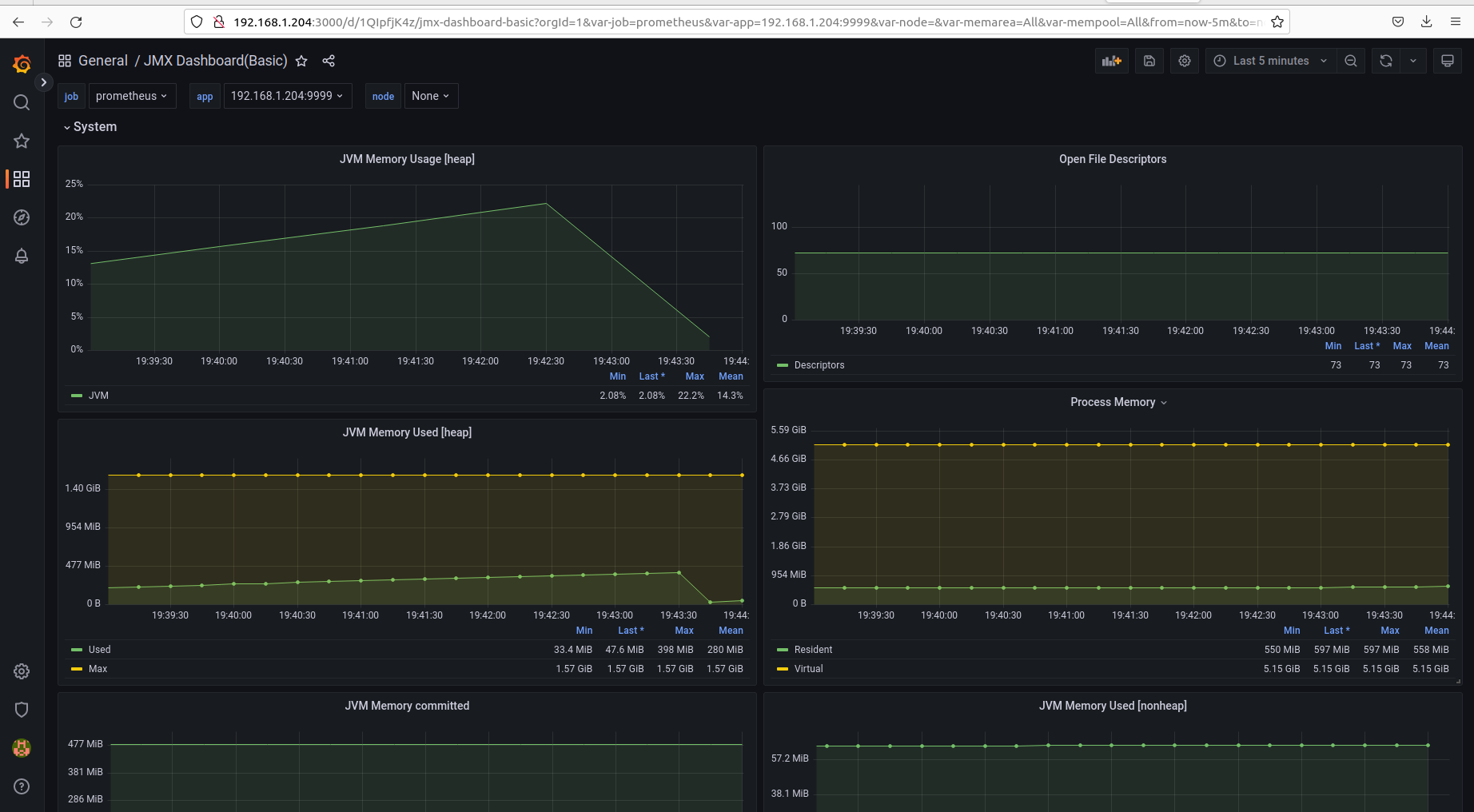
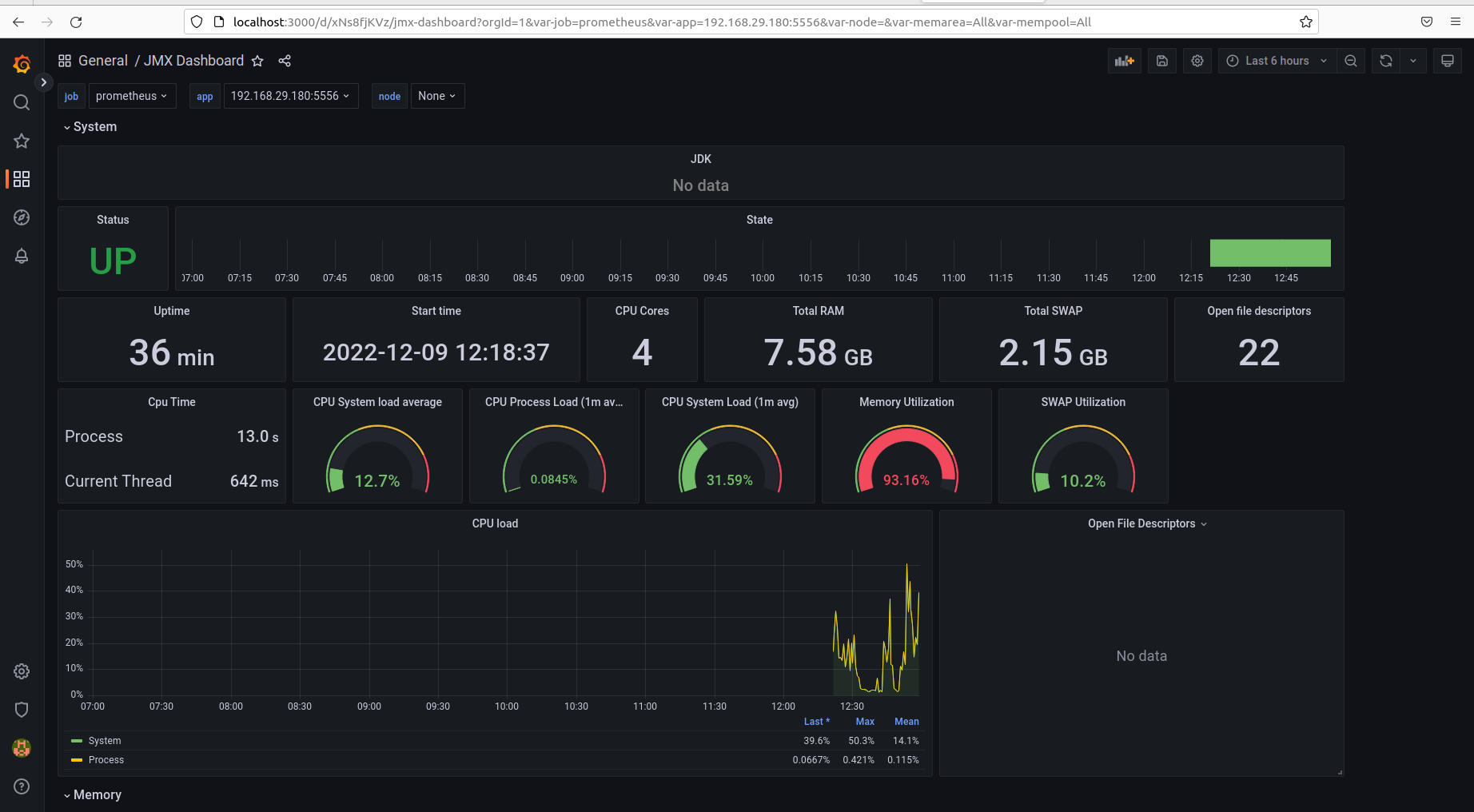
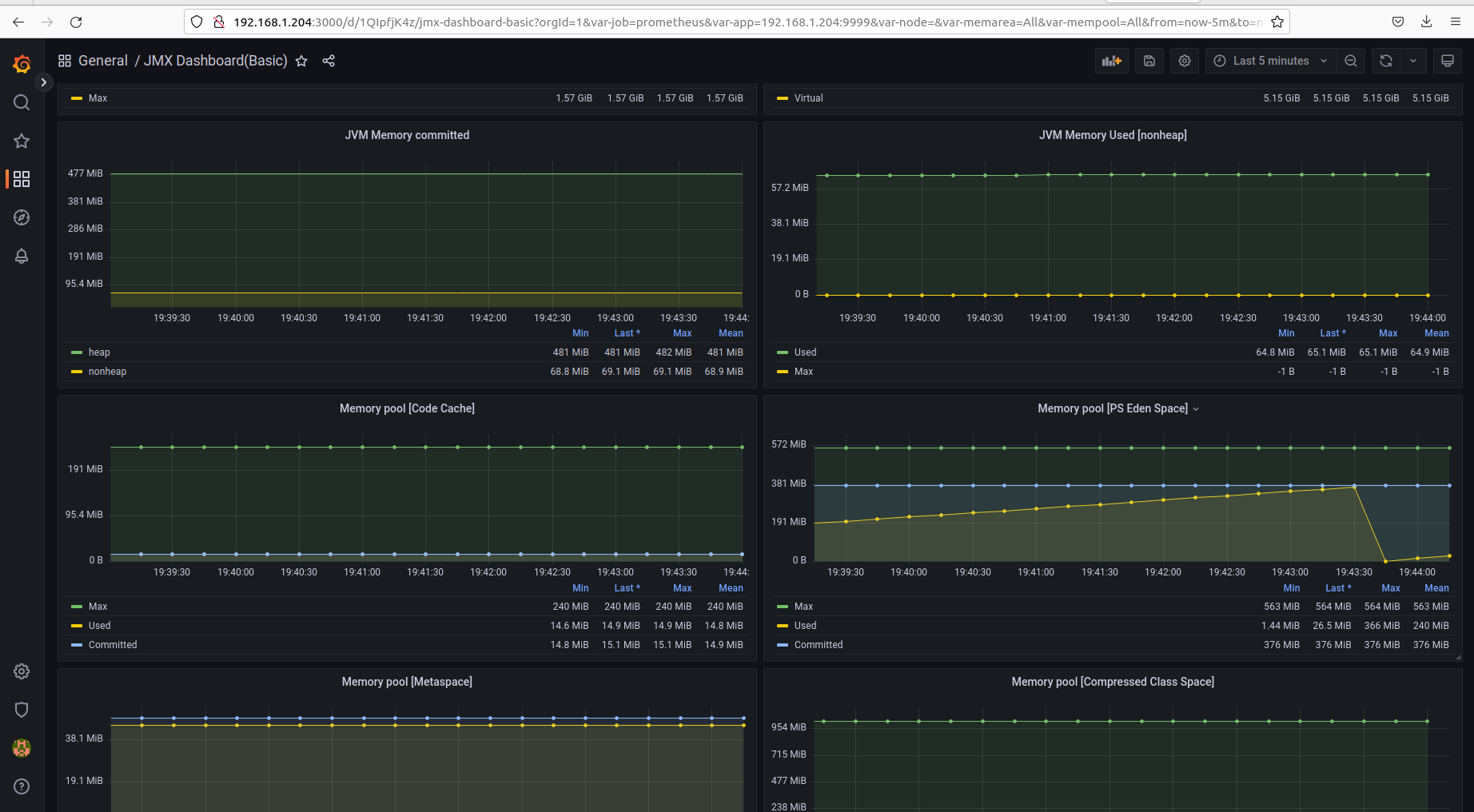
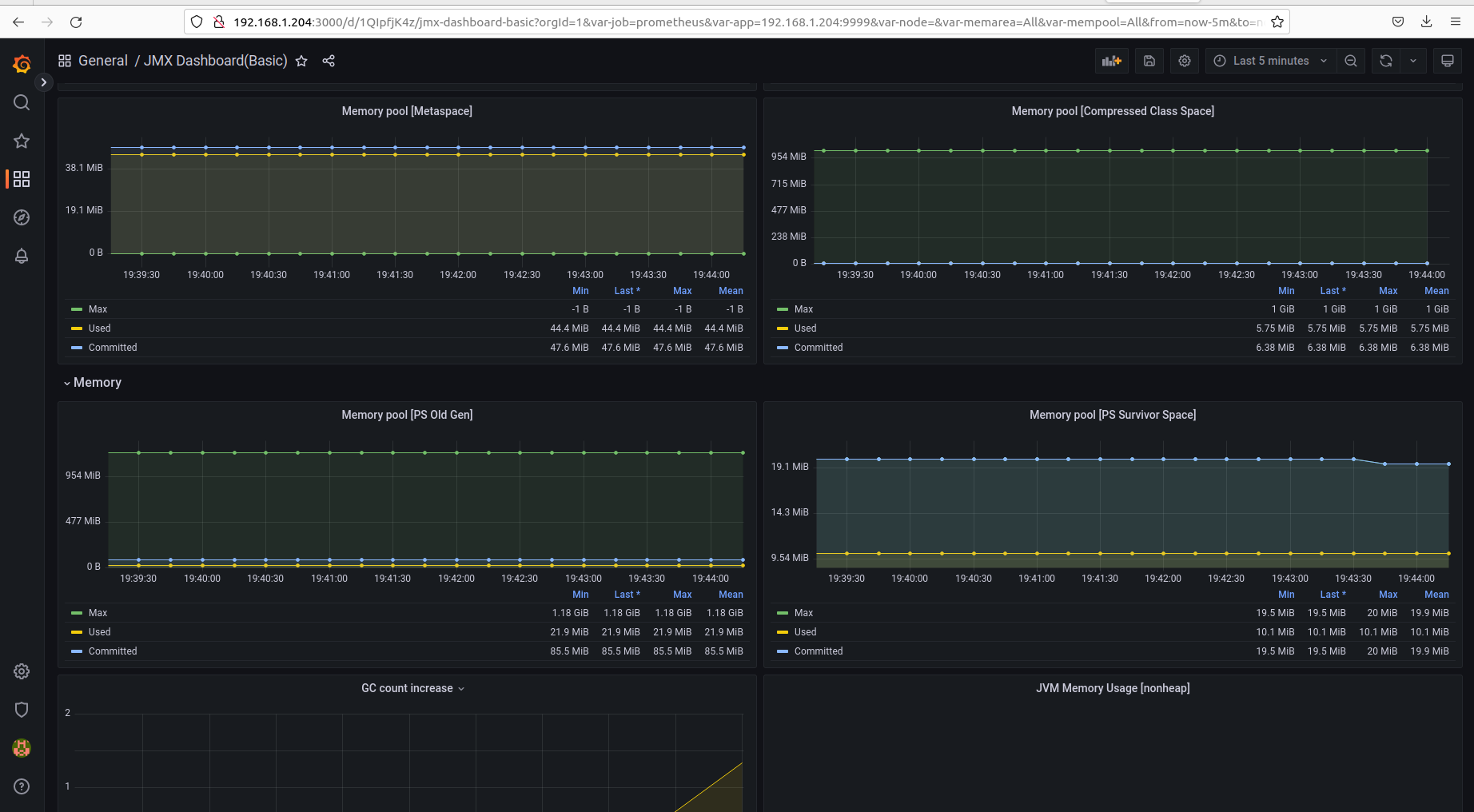
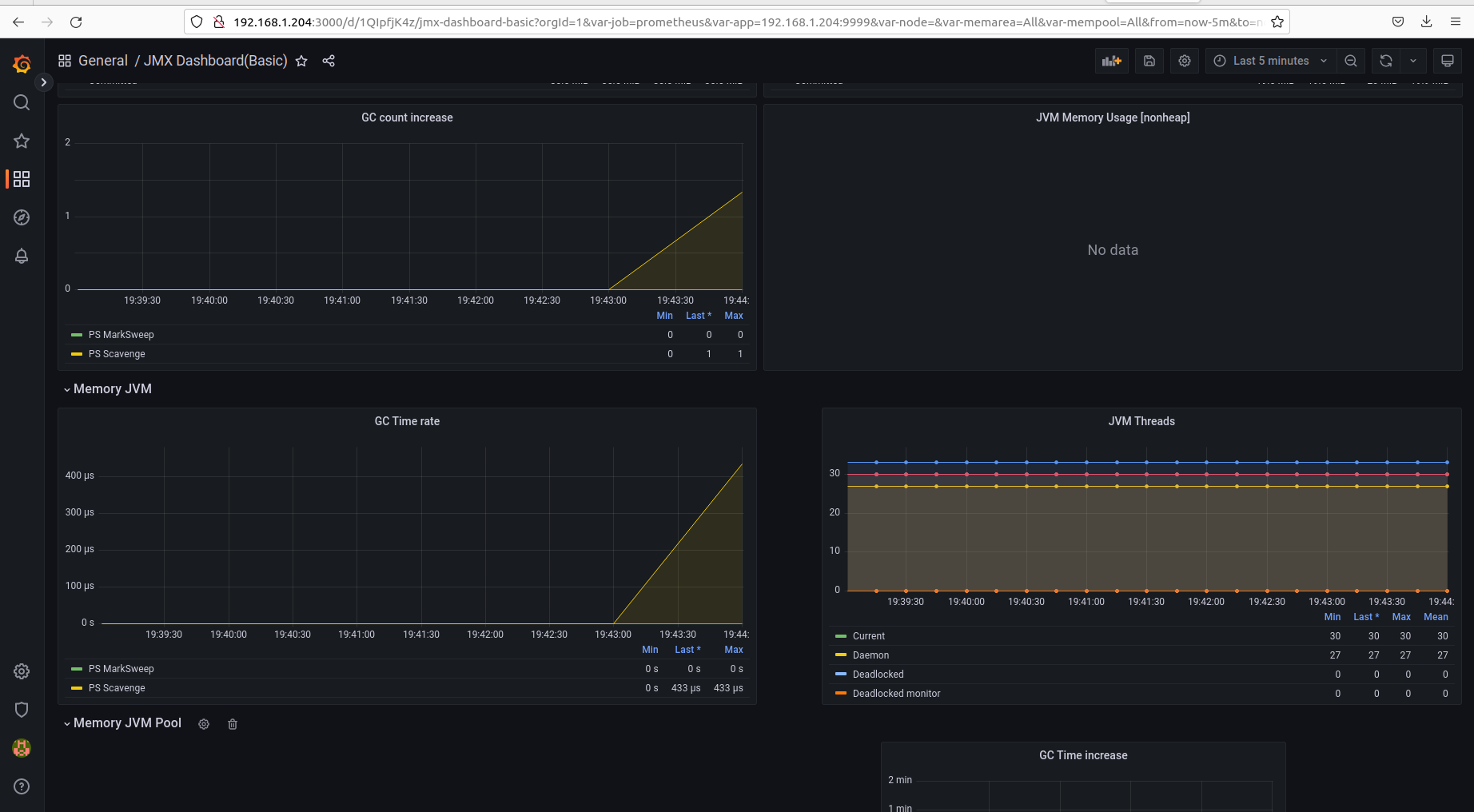
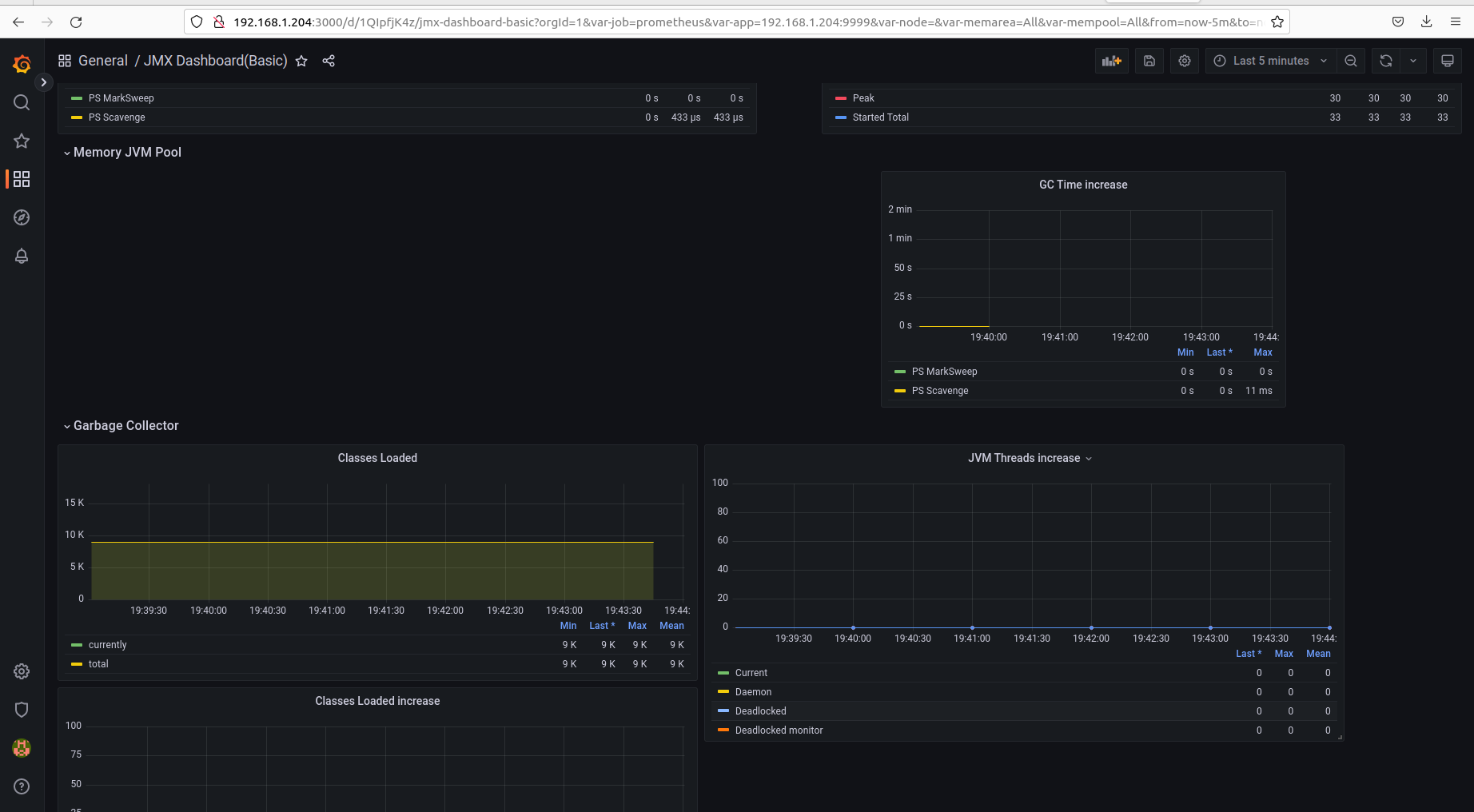
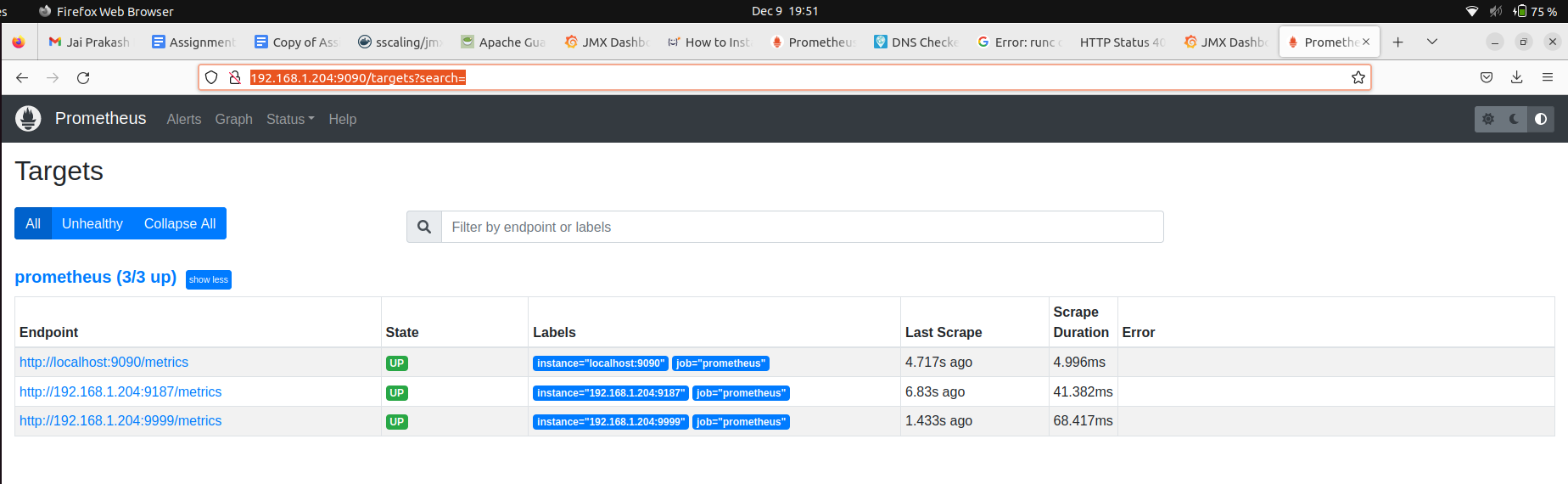
Update of 10-12-2022

I have followed example given by [Harsh Choudhary](mailto:harsh.choudhary@fosteringlinux.com) Sir to set up the guacamole-app container with jmx agent.

**Command:**

podman run -idt --name guacamole-app -p 8080:8080 -p 9999:9999 -v **/home/tomcat/tomcat.yaml**:/usr/local/tomcat/bin/tomcat.yml -v **/home/tomcat/jmx\_prometheus\_javaagent-0.16.1.jar**:/usr/local/tomcat/bin/jmx\_prometheus\_javaagent-0.16.1.jar -v **/home/tomcat/setenv.sh**:/usr/local/tomcat/bin/setenv.sh -e POSTGRES\_HOSTNAME=localhost -e POSTGRES\_DATABASE=guacamole\_db -e POSTGRES\_USER=guacamole\_user -e POSTGRES\_PASSWORD=redhat -e GUACD\_HOSTNAME=localhost guacamole/guacamole

Metrics in Prometheus and the graph in grafana:



Thank You