Crunchy-Postgres-Exporter

Overview

This document provides a step-by-step overview of set-up Crunchy-Postgres-Exporter using Podman, Crunchy Exporter, PostgreSQL, Prometheus and Grafana on an Ubuntu 20.04 system.

Environment details

- Ubuntu version 20.04
- RAM 4 GB

List of tools and technologies

- Podman
- PostgreSQL
- Grafana
- Prometheus

Definition of tools

- **Podman :** Podman is an open-source containerization tool that works on Linux operating systems. It allows you to create, manage, and run containers, and it offers many features that are compatible with docker commands.

- **PostgreSQL :** PostgreSQL is an open-source relational database management system used for data storage and retrieval. It offers advanced features and scalability, making it a popular open-source RDBMS.

- **Grafana :** Grafana is an open-source tool used for data visualization and monitoring. It allows data to be visually represented in the form of graphs, charts, making data analysis and performance monitoring easier.

- **Prometheus :** Prometheus is an open-source monitoring and alerting system used for system and application performance monitoring. Its primary purpose is data collection, storage, querying, and alerting, allowing the detection of real-time insights and performance issues.

Command for the setup or configuration

Step 1. Install podman.

```
sudo sh -c "echo 'deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_$(lsb_relea
se -rs)/ /' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list"
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ sudo sh -c "echo 'deb https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_$(
lsb_release -rs)/ /' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list"
```

```
wget
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_$(lsb_relea
se -rs)/Release.key
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ wget https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_$(lsb_release -rs)/
Release.key
--2023-10-25 16:21:41--  https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_20.04/Release.key
Resolving download.opensuse.org (download.opensuse.org)... 195.135.221.134, 2001:67c:2178:8::13
Connecting to download.opensuse.org (download.opensuse.org)|195.135.221.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1093 (1.1K) [application/x-download]
Saving to: ‘Release.key’

Release.key                          100%[==============================================================================================
===============>]   1.07K  --.-KB/s    in 0s

2023-10-25 16:21:43 (60.6 MB/s) - ‘Release.key’ saved [1093/1093]
```

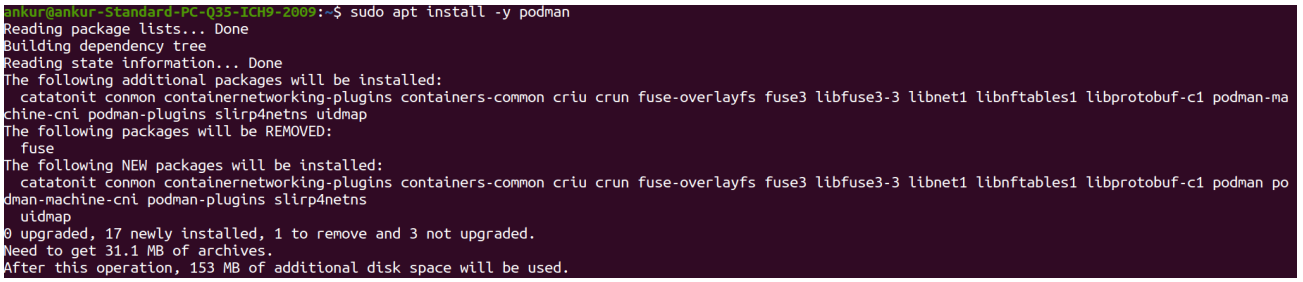
```
sudo apt-key add - < Release.key
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ sudo apt-key add - < Release.key
OK
```

```
sudo apt update
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_20.04  InRelease [1,642 B]
Get:6 https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_20.04  Packages [15.0 kB]
Fetched 16.6 kB in 5s (3,603 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

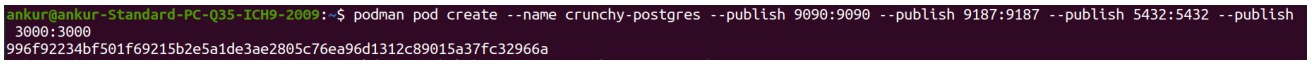
```
sudo apt install -y podman
```




```
podman --version
```

Step 2. Create Pod name crunchy-postgres for all the 4 containers.

```
podman pod create --name crunchy-postgres --publish 9090:9090 --publish 9187:9187 --publish 5432:5432 --publish 3000:3000
```

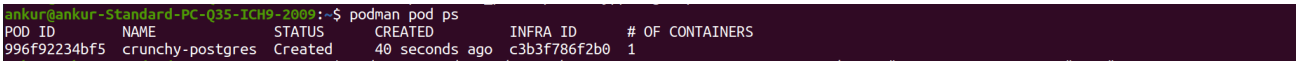


- **`podman pod create`**: This command is used to create a new pod, which is a group of containers that share the same network namespace. Containers within a pod can communicate with each other using the loopback interface .
- **`--name crunchy-postgres`**: the pod will be named "crunchy-postgres".
- **`--publish 9090:9090`**: This flag maps port 9090 from the host to port 9090 within the pod. Port 9090 is commonly used for services like Prometheus.
- **`--publish 9187:9187`**: This flag maps port 9187 from the host to port 9187 within the pod. Port 9187 is the default port for the PostgreSQL Exporter, which exposes PostgreSQL performance metrics.
- **`--publish 5432:5432`**: This flag maps port 5432 from the host to port 5432 within the pod. Port 5432 is the default port for PostgreSQL database connections.

- **`--publish 3000:3000`**: This flag maps port 3000 from the host to port 3000 within the pod. Port 3000 is commonly used for Grafana.

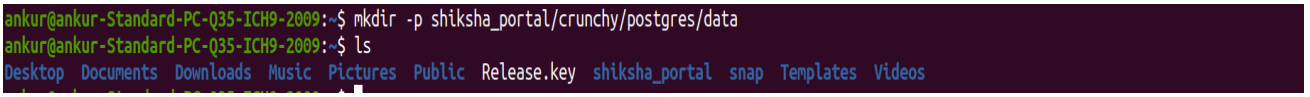
Step 3. Check the created pod status.

```
podman pod ps
```



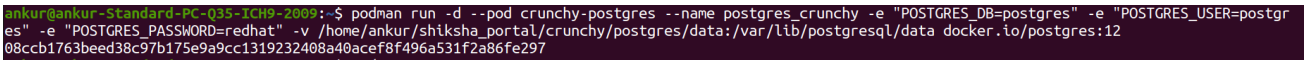
Step 4. Create a directory for mounting the persistent storage for the postgres data.

```
mkdir -p shiksha_portal/crunchy/postgres/data
```



Step 5. Create a postgres container in this pod which was created with the name crunchy-postgres.

```
podman run -d --pod crunchy-postgres --name postgres_crunchy -e
"POSTGRES_DB=postgres" -e "POSTGRES_USER=postgres" -e
"POSTGRES_PASSWORD=redhat" -v
/home/ankur/shiksha_portal/crunchy/postgres/data:/var/lib/postgresql/data
docker.io/postgres:12
```



- **`podman run`**: This command is used to run a new container.

- **`-d`**: This flag indicates that the container should run in detached mode (in the background).
- **`--pod crunchy-postgres`**: This flag specifies that the container should be part of the existing "crunchy-postgres" pod.

- **`--name postgres_crunchy`**: This flag assigns the name "postgres_crunchy" to the container.
- **`-e "POSTGRES_DB=postgres"`**: This flag sets the environment variable **`POSTGRES_DB`** within the container to "postgres",
- **`-e "POSTGRES_USER=postgres"`**: This flag sets the environment variable **`POSTGRES_USER`** within the container to "postgres",It indicates the username.
- **`-e "POSTGRES_PASSWORD=redhat"`**: This flag sets the environment variable **`POSTGRES_PASSWORD`** within the container to "redhat", which is the password for the PostgreSQL user.
- **`-v/home/ankur/shiksha_portal/crunchy/postgres/data:/var/lib/postgresql/data`**: This allows you to persist the PostgreSQL data outside the container, ensuring that the data is retained even if the container is removed.
- **`docker.io/postgres:12`**: This specifies the Docker image to use for the container. The container will be based on the "postgres:12" image from Docker Hub.

Step 6. Check the container status.

```
podman ps
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman ps
CONTAINER ID  IMAGE                           COMMAND     CREATED             STATUS                 PORTS
                                              NAMES
c3b3f786f2b0  k8s.gcr.io/pause:3.5                         2 minutes ago       Up About a minute ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp, 0.0.
0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  996f92234bf5-infra
08ccb1763bee  docker.io/library/postgres:12  postgres    About a minute ago  Up About a minute ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp, 0.0.
0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  postgres_crunchy
```

Step 7. Do changes in postgresql.conf file.

We will need to modify your postgresql.conf configuration file to tell PostgreSQL to load shared libraries.

Change directory to postgres then run sudo su for root privilege and change directory to data.

```
cd shiksha_portal/cruchy/postgres/
```

```
sudo su
```

```
cd data/
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ cd shiksha_portal/crunchy/postgres/
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/crunchy/postgres$ sudo su
root@ankur-Standard-PC-Q35-ICH9-2009:/home/ankur/shiksha_portal/crunchy/postgres# cd data/
root@ankur-Standard-PC-Q35-ICH9-2009:/home/ankur/shiksha_portal/crunchy/postgres/data# ls
base          pg_dynshmem    pg_logical    pg_replslot   pg_stat      pg_tblspc    pg_wal              postgresql.conf
global        pg_hba.conf    pg_multixact  pg_serial     pg_stat_tmp  pg_twophase  pg_xact             postmaster.opts
pg_commit_ts  pg_ident.conf  pg_notify     pg_snapshots  pg_subtrans  PG_VERSION   postgresql.auto.conf  postmaster.pid
```

```
home/ankur/shiksha_portal/crunchy/postgres/data# echo
"shared_preload_libraries = 'pg_stat_statements,auto_explain'" >>
postgresql.conf
```

```
root@ankur-Standard-PC-Q35-ICH9-2009:/home/ankur/shiksha_portal/crunchy/postgres/data# echo "shared_preload_libraries = 'pg_stat_statements,auto_explain
'" >> postgresql.conf
```

Step 8. Install vim.

```
sudo apt install vim
```

```
root@ankur-Standard-PC-Q35-ICH9-2009:/home/ankur/shiksha_portal/crunchy/postgres/data# sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 7,116 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim-runtime all 2:8.1.2269-1ubuntu5.18 [5,875 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim amd64 2:8.1.2269-1ubuntu5.18 [1,242 kB]
Fetched 7,116 kB in 5s (1,474 kB/s)
Selecting previously unselected package vim-runtime.
(Reading database ... 182837 files and directories currently installed.)
```

"We can also make these configuration changes in the postgresql.conf file. Search for '**shared_preload_libraries**' and remove the '#' symbol to uncomment it and add following line "**pg_stat_statements,auto_explain**"

```
vim postgresql.conf
```

```
# - Shared Library Preloading -

shared_preload_libraries = 'pg_stat_statements,auto_explain'    # (change requires restart)
#local_preload_libraries = ''
#session_preload_libraries = ''
#jit_provider = 'llvmjit'               # JIT library to use

# - Other Defaults -
```

Step 9. Crunchy-postgres-exporter image.

Download tar file from the given link. and run the given command on the same path.
Link:- https://drive.google.com/file/d/1pTXscjFxxsv1g0NFMCDofeRD-vf_I06M/view

```
podman load -i crunchy.tar
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ cd Desktop/
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/Desktop$ ls
crunchy.tar
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/Desktop$ podman load -i crunchy.tar
Getting image source signatures
Copying blob b3885f64f7e3 skipped: already exists
Copying blob ad5362eb8973 skipped: already exists
Copying blob ae62f8792f69 skipped: already exists
Copying blob c26e09c93862 skipped: already exists
Copying blob 962a9dd994b2 skipped: already exists
Copying blob 7af0eb8c2f76 skipped: already exists
Copying blob 17eb7f871815 skipped: already exists
Copying blob 4adb4ecc245f skipped: already exists
Copying blob d033a1278b47 skipped: already exists
Copying blob 77c44a6b944a skipped: already exists
Copying blob 7d7e1c4979ec skipped: already exists
Copying blob 71d1943e024b skipped: already exists
Copying config c9110d4443 done
Writing manifest to image destination
Storing signatures
Loaded image(s): sha256:c9110d4443a66c8cd6047ea82562858efa97a68396b3889c323dd4adbfe43406
```

```
podman images
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/Desktop$ podman images
REPOSITORY                  TAG         IMAGE ID      CREATED       SIZE
<none>                      <none>      c9110d4443a6  7 weeks ago   228 MB
docker.io/library/postgres  12          8a6e58b2e819  2 months ago  412 MB
k8s.gcr.io/pause            3.5         ed210e3e4a5b  2 years ago   690 kB
```

Then run the container with the image ID and use your container's image ID.

```
podman run -itd --pod crunchy-postgres --name crunchy -e
EXPORTER_PG_PASSWORD=redhat c9110d4443a6
```

- **podman run:** This is the basic command for running containers with Podman.
- **-itd:** These are flags used with the podman run command:
- **-i:** This option makes the container interactive, allowing you to interact with it through the terminal.

- **-t:** This option allocates a pseudo-TTY, which is often used in conjunction with -i to provide a terminal interface.
- **-d:** This option runs the container in detached mode, meaning it runs in the background, and you can continue using your terminal for other tasks.
- **--pod crunchy-postgres:** This specifies the name of the pod to which the container should be added.
- **--name crunchy:** This assigns a name to the container, in this case, "crunchy."
- **-e EXPORTER_PG_PASSWORD=redhat:** This option sets an environment variable within the container. In this case, it sets the environment variable EXPORTER_PG_PASSWORD with the value redhat.
- **c9110d4443a6:** This is the container image identifier, typically an image ID.

Again check container status.

```
podman ps
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/Desktop$ cd ..
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman run -itd --pod crunchy-postgres --name crunchy -e EXPORTER_PG_PASSWORD=redhat c9110d4443a6
40577073b06e2d25837fa72a200731df1ad6b75d10fe06a621a22bdac43e65f3
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman ps
CONTAINER ID  IMAGE                           COMMAND               CREATED         STATUS             PORTS
                                              NAMES
c3b3f786f2b0  k8s.gcr.io/pause:3.5                                  26 minutes ago  Up 25 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp, 0.
0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  996f92234bf5-infra
08ccb1763bee  docker.io/library/postgres:12   postgres              25 minutes ago  Up 25 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp, 0.
0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  postgres_crunchy
40577073b06e  c9110d4443a6                    /opt/cpm/bin/star...  6 minutes ago   Up 6 minutes ago   0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp, 0.
0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  crunchy
```

Step 10. Copy `setup.sql` from Container to Host:

Copy the **setup.sql** file from the **/opt/cpm/conf/pg12/** directory within the Crunchy container to your current working directory on the host machine..

```
podman cp crunchy:/opt/cpm/conf/pg12/setup.sql .
```

- **podman cp:** This is a command used to copy files between your host system and a container using Podman.
- **crunchy:/opt/cpm/conf/pg12/setup.sql:** This part specifies the source path of the file inside the container you want to copy.

- **(.):** This indicates the destination path on your host system where the file will be copied.

NOTE:- Don't miss the (.) which is taking place after setup.sql because it represents a copy in your current directory.

Step 11. Remove the Test Container.

Remove the **crunchy** container.

```
podman rm -f crunchy
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman rm -f crunchy
40577073b06e2d25837fa72a200731df1ad6b75d10fe06a621a22bdac43e65f3
```

Here, we will copy the **setup.sql** file from the host to inside the PostgreSQL container in the **/var/lib/postgresql** directory.

```
podman cp /home/ankur/setup.sql postgres_crunchy:/var/lib/postgresql
```

- **podman cp:** This is a command used to copy files between your host system and a container using Podman.
- **/home/ankur/setup.sql:** This is the source path on your local system, specifying the file you want to copy.
- **postgres_crunchy:** This is the name of the destination container where you want to copy the file. Containers are often given names for easy reference.
- **/var/lib/postgresql:** This part specifies the destination path within the container. It means you want to copy the file to the /var/lib/postgresql directory inside the "postgres_crunchy" container.

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman cp /home/ankur/setup.sql postgres_crunchy:/var/lib/postgresql
```

Step 12. Go inside the postgres container & Push setup.sql in postgres database.

```
podman exec -it postgres_crunchy bash
```

```
root@crunchy-postgres:/# ls

root@crunchy-postgres:/# cd var/lib/postgresql/

root@crunchy-postgres:/var/lib/postgresql# ls

root@crunchy-postgres:/var/lib/postgresql# psql -h 127.0.0.1 -U postgres -d
template1 < setup.sql
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman exec -it postgres_crunchy bash
root@crunchy-postgres:/#
root@crunchy-postgres:/#
root@crunchy-postgres:/#
root@crunchy-postgres:/#
root@crunchy-postgres:/# ls
bin  boot  dev  docker-entrypoint-initdb.d  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@crunchy-postgres:/#
root@crunchy-postgres:/#
root@crunchy-postgres:/#
root@crunchy-postgres:/# cd var/lib/postgresql/
root@crunchy-postgres:/var/lib/postgresql# ls
data  setup.sql
root@crunchy-postgres:/var/lib/postgresql# psql -h 127.0.0.1 -U postgres -d template1 < setup.sql
DO
GRANT ROLE
GRANT ROLE
ALTER ROLE
ALTER ROLE
CREATE SCHEMA
NOTICE:  table "pgbackrest_info" does not exist, skipping
DROP TABLE
CREATE TABLE
ALTER TABLE
NOTICE:  function monitor.pgbackrest_info() does not exist, skipping
DROP FUNCTION
NOTICE:  function monitor.pgbackrest_info(pg_catalog.int4) does not exist, skipping
DROP FUNCTION
CREATE FUNCTION
```

This command is used to execute SQL commands from the '**setup.sql**' file within a PostgreSQL database. It facilitates making changes to the database schema, data, or settings as specified in the SQL file.

Step 13. Create an extension.

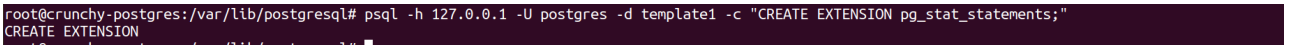
```
root@crunchy-postgres:/var/lib/postgresql# psql -h 127.0.0.1 -U postgres -d
template1 -c "CREATE EXTENSION pg_stat_statements;"
```

- **/var/lib/postgresql#:** This part of the command appears to indicate the directory or path from which the command is being executed. It's not part of the actual PostgreSQL command.
- **psql:** This is the PostgreSQL command-line client, used to interact with PostgreSQL databases.
- **-h 127.0.0.1:** This specifies the hostname or IP address of the PostgreSQL server you want to connect to. In this case, it's connecting to the local machine (127.0.0.1).

- **-U postgres:** This option specifies the username ("postgres") that you want to use to connect to the database.

- **-d template1:** This specifies the name of the database ("template1") to which you want to connect.

- **-c "CREATE EXTENSION pg_stat_statements;":** This part is the SQL command that you want to execute within the specified database. In this case, it's creating a PostgreSQL extension called "pg_stat_statements."



This command is used to enable the '**pg_stat_statements**' extension in the PostgreSQL database. It enables the collection of statistics about executed SQL statements for performance analysis.

Step 14. Here, we will log in to the PostgreSQL database, create a password for the user 'ccp_monitoring,' and also create a database named 'ankur.'

```
root@crunchy-postgres:/var/lib/postgresql# psql -h 127.0.0.1 -U postgres -d
postgres


postgres=# \password ccp_monitoring
Enter new password for user "ccp_monitoring":      (I am giving here redhat)
Enter it again:
postgres=# create database ankur;
```

In the PostgreSQL database, to exit, use the command '**\q**' and then press '**Enter**.'

- **psql:** This is the command-line utility for interacting with PostgreSQL databases. It allows you to run SQL queries, manage databases, and perform various database-related tasks.

- **-h 127.0.0.1**: This flag specifies the host where the PostgreSQL database is running.

- **-U postgres**: This flag specifies the username to use when connecting to the database. In this case, you're connecting as the PostgreSQL superuser "postgres."

- **`-d postgres`**: This flag specifies the name of the database to connect to. In this case, you're connecting to a database named "postgres."

```
root@crunchy-postgres:/var/lib/postgresql# psql -h 127.0.0.1 -U postgres -d postgres
psql (12.16 (Debian 12.16-1.pgdg120+1))
Type "help" for help.

postgres=# \password ccp_monitoring
Enter new password for user "ccp_monitoring":
Enter it again:
postgres=# create database ankur;
CREATE DATABASE
postgres=# \q
root@crunchy-postgres:/var/lib/postgresql# exit
exit
```

Step 15. Now create a crunchy-postgres-exporter container.

```
podman run -itd --pod crunchy-postgres --name crunchy -e
EXPORTER_PG_PASSWORD=redhat -e EXPORTER_PG_HOST=127.0.0.1 -e
EXPORTER_PG_USER=ccp_monitoring -e
DATA_SOURCE_NAME=postgresql://ccp_monitoring:redhat@127.0.0.1:5432/ankur?sslmo
de=disable c9110d4443a6
```

- **`podman run`**: This command is used to run a new container.
- **`-itd`**: These flags are used together for interactive (console input/output enabled), detached (background) mode.
- **`--pod crunchy-postgres`**: This flag specifies that the container should be part of the existing "crunchy-postgres" pod.
- **`--name crunchy`**: This flag assigns the name "crunchy" to the container.
- **`-e EXPORTER_PG_PASSWORD=redhat`**: This flag sets the environment variable **`EXPORTER_PG_PASSWORD`** within the container to "redhat". This likely represents the password required for PostgreSQL Exporter to connect to the PostgreSQL instance.
- **`-e EXPORTER_PG_HOST=127.0.0.1`**: This flag sets the environment variable **`EXPORTER_PG_HOST`** within the container to "127.0.0.1", indicating the host where the PostgreSQL database is located.
- **`-e EXPORTER_PG_USER=ccp_monitoring`**: This flag sets the environment variable **`EXPORTER_PG_USER`** within the container to "ccp_monitoring", which is likely the username used by the PostgreSQL Exporter to connect to the PostgreSQL instance.
- **`-e DATA_SOURCE_NAME=...`**: This flag sets the **`DATA_SOURCE_NAME`** environment variable. It specifies the connection details for the PostgreSQL Exporter to use when connecting to the PostgreSQL database. The provided URL includes the username,

password, host, port, database name, and SSL mode settings.

- **c9110d4443a6**: This represents the ID of the container image that you want to run as a container.

Podman images.

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman images
REPOSITORY                  TAG         IMAGE ID      CREATED       SIZE
<none>                      <none>      c9110d4443a6  7 weeks ago   228 MB
docker.io/library/postgres  12          8a6e58b2e819  2 months ago  412 MB
k8s.gcr.io/pause            3.5         ed210e3e4a5b  2 years ago   690 kB
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ podman run -itd --pod crunchy-postgres --name crunchy -e EXPORTER_PG_PASSWORD=redhat -e EXPORTER_PG_HOST=127.0.
0.1 -e EXPORTER_PG_USER=ccp_monitoring -e DATA_SOURCE_NAME=postgresql://ccp_monitoring:redhat@127.0.0.1:5432/ankur?sslmode=disable c9110d4443a6
52971cd16184a350fc69591905a79b0155b4b45a1ec3f42a4cc3eba00992201a
```

Step 16. Install curl.

```
sudo apt install curl
```

- **curl:** is a command-line tool to transfer data to or from a server.

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ sudo apt install curl
[sudo] password for ankur:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 161 kB of archives.
After this operation, 413 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.20 [161 kB]
Fetched 161 kB in 2s (99.4 kB/s)
Selecting previously unselected package curl.
(Reading database ... 184673 files and directories currently installed.)
Preparing to unpack .../curl_7.68.0-1ubuntu2.20_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.20) ...
Setting up curl (7.68.0-1ubuntu2.20) ...
Processing triggers for man-db (2.9.1-1) ...
```

Check metrics.

```
curl localhost:9187/metrics | grep query
```

.

- **grep query**: This part of the command uses the **grep** command to search for lines in the input that contain the word "query". This is used to filter the metrics output to only show lines related to queries

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ curl localhost:9187/metrics | grep query
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0# HELP ccp_connection_stats_max_blocked_query_time Length of time in secon
ds of the longest running query that has been blocked by a heavyweight lock
# TYPE ccp_connection_stats_max_blocked_query_time gauge
ccp_connection_stats_max_blocked_query_time{server="127.0.0.1:5432"} 0
# HELP ccp_connection_stats_max_query_time Length of time in seconds of the longest running query
# TYPE ccp_connection_stats_max_query_time gauge
ccp_connection_stats_max_query_time{server="127.0.0.1:5432"} 0.00379
ccp_pg_stat_statements_top_max_exec_time_ms{dbname="ankur",query="SELECT checkpoints_timed, checkpoints_re",queryid="6465064219195468563",role="ccp_moni
toring",server="127.0.0.1:5432"} 10.251735
ccp_pg_stat_statements_top_max_exec_time_ms{dbname="postgres",query="create database ankur",queryid="4599619917575973624",role="postgres",server="127.0.
0.1:5432"} 243.186625
ccp_pg_stat_statements_top_max_exec_time_ms{dbname="template1",query="COMMENT ON COLUMN monitor.pg_settings_ch",queryid="-2316124150447318303",role="pos
tgres",server="127.0.0.1:5432"} 4.313338
ccp_pg_stat_statements_top_max_exec_time_ms{dbname="template1",query="CREATE EXTENSION pg_stat_statements",queryid="2643756286501377769",role="postgres"
,server="127.0.0.1:5432"} 45.21845
ccp_pg_stat_statements_top_max_exec_time_ms{dbname="template1",query="CREATE FUNCTION monitor.pg_settings_chec",queryid="-5580726982229046996",role="pos
```

Now, for the Prometheus container, we need to create the 'prometheus_crunchy' directory and the 'prometheus.yml' file inside the directory. Change directory to shiksha portal and create file.

```
:~$ cd shiksha_portal/
:~/shiksha_portal$ mkdir prometheus_crunchy
:~/shiksha_portal$ ls
:~/shiksha_portal$ cd prometheus_crunchy/
:~/shiksha_portal/prometheus_crunchy$ touch prometheus.yml
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~$ cd shiksha_portal/
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal$ ls
crunchy
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal$ mkdir prometheus_crunchy
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal$ ls
crunchy  prometheus_crunchy
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal$ cd prometheus_crunchy/
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ ls
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ touch prometheus.yml
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ ls
prometheus.yml
```

Step 17. Create a prometheus container.

Please check the path name **/home/ankur/shiksha_portal/prometheus_crunchy.**

```
podman run -itd --pod crunchy-postgres --name prometheus_crunchy -v
/home/ankur/shiksha_portal/prometheus_crunchy/prometheus.yml:/etc/prometheus/pr
ometheus.yml docker.io/prom/prometheus
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ podman run -itd --pod crunchy-postgres --name prometheus_crunchy -v /home/ankur/shiksha_portal/prometheus_crunchy/prometheus.yml:/etc/prometheus/prometheus.yml docker.io/prom/prometheus
Trying to pull docker.io/prom/prometheus:latest...
Getting image source signatures
Copying blob 07e931646da3 done
Copying blob 04d12a144c6b done
Copying blob 9a5555141b82 done
Copying blob ce5cd6bb9a56 done
Copying blob 2abcce694348 done
Copying blob f8e22ccd8ad8 done
Copying blob fe7efe91c254 done
Copying blob 08be134b67e3 done
Copying blob a836be3b6eeb done
Copying blob fadcaad2cfd9 done
Copying blob 8215e06ac3f8 done
Copying blob fd09517fa2d8 done
Copying config 22010d1e55 done
Writing manifest to image destination
Storing signatures
97ea611052d09261c57c1247248c438b04ac47c1ef760fe5e0fd8907f54baab7
```

Check all container status

```
podman ps
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ podman ps
CONTAINER ID  IMAGE                              COMMAND               CREATED         STATUS             PORTS
                                                  NAMES
c3b3f786f2b0  k8s.gcr.io/pause:3.5                                     2 hours ago     Up 2 hours ago     0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
 0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  996f92234bf5-infra
08ccb1763bee  docker.io/library/postgres:12      postgres              2 hours ago     Up 2 hours ago     0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
 0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  postgres_crunchy
52971cd16184  c9110d4443a6                       /opt/cpm/bin/star...  13 minutes ago  Up 13 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
 0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  crunchy
97ea611052d0  docker.io/prom/prometheus:latest   --config.file=/et...  2 minutes ago   Up 2 minutes ago   0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
 0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  prometheus_crunchy
```

Step 18. Configure the prometheus.yml file.

Check IP

```
hostman -I
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ hostname -I
192.168.122.23
```

```
vim Prometheus.yml
```

To insert the given content into the file, use 'i', add the content, then press 'Esc' followed by ':wq!' to save and exit.

```
/shiksha_portal/prometheus_crunchy$ cat prometheus.yml
```

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

```
scrape_configs:
  - job_name: 'prometheus'
      static_configs:
      - targets: ['192.168.122.23:9090']
      - targets: ['192.168.122.23:9187']
```

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ cat prometheus.yml
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['192.168.122.23:9090']
      - targets: ['192.168.122.23:9187']
```

Set the target in the prometheus.yml file to get metrics in prometheus and hit the browser (after restarting the prometheus container).

```
:~/shiksha_portal/prometheus_crunchy$ podman restart prometheus_crunchy
```

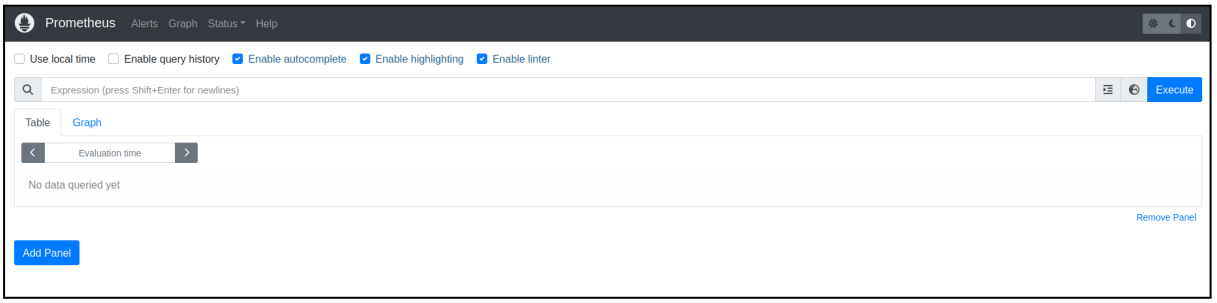
```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ podman restart prometheus_crunchy
97ea611052d09261c57c1247248c438b04ac47c1ef760fe5e0fd8907f54baab7
```

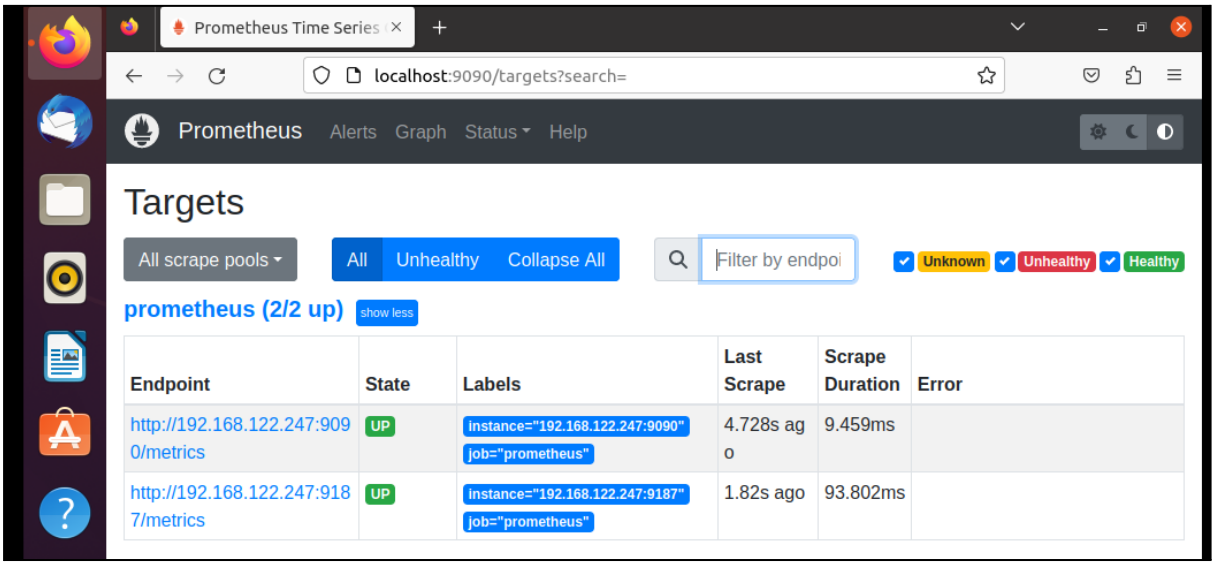
http://192.168.122.23:9090/

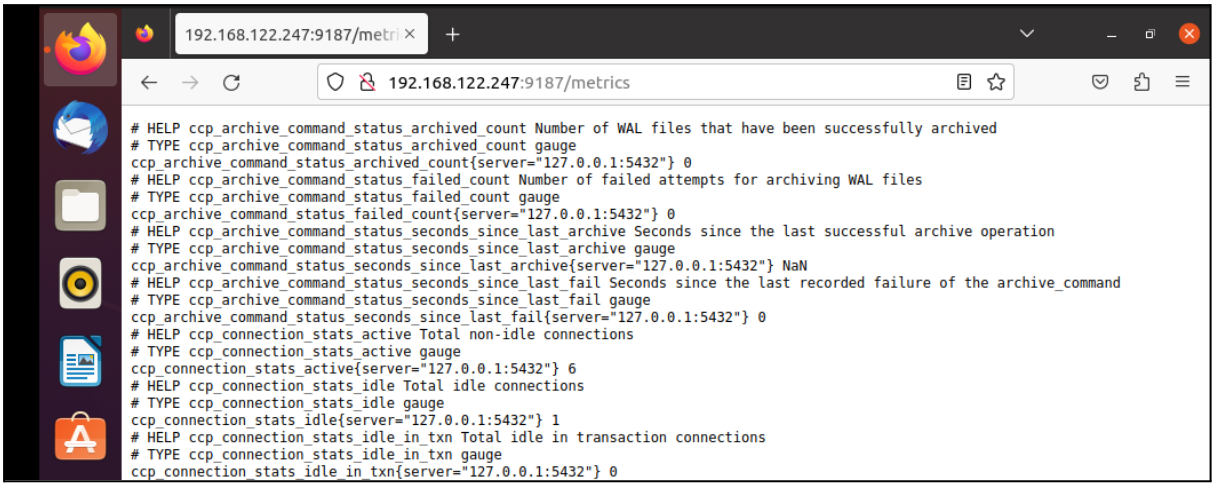
Or

http://localhost:9090/

Open Prometheus in the browser by using the IP or 'localhost' and then click 'Status' > 'Targets' > 'Status' to confirm that all statuses are up.







Step 19. Create a grafana container for Visualisation of the metrics data.

```
podman run -itd --pod crunchy-postgres --name grafana_crunchy
docker.io/grafana/grafana
```

- **podman run:** This is the command to run a container using Podman.
- **-itd:** These are options:

-i: Keep STDIN open even if not attached (i.e., interactive mode).
-t: Allocate a pseudo-TTY (i.e., terminal).
-d: Run the container in the background (detached mode).

- **--pod crunchy-postgres:** This specifies that the container should be part of the "crunchy-postgres" pod.

- **--name grafana_crunchy:** This assigns the name "grafana_crunchy" to the running container.

- **docker.io/grafana/grafana:** This is the image you are using to create the container. It specifies the image's location, which is "docker.io" in this case, and the image name "grafana/grafana."

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ podman run -itd --pod crunchy-postgres --name grafana_crunchy docker.io/grafa
na/grafana
Trying to pull docker.io/grafana/grafana:latest...
Getting image source signatures
Copying blob 7264a8db6415 done
Copying blob 3cf7ed17dad5 done
Copying blob 5aeff27f6208 done
Copying blob 00fbaa4ff278 done
Copying blob b82f661477d7 done
Copying blob 85d88afc7ec6 done
Copying blob 1712a81b1c5b done
Copying blob 2309246adff5 done
Copying blob 4a6fe4caa3c7 done
Copying blob 6309adfe5521 done
Copying config 2fbe6143d3 done
Writing manifest to image destination
Storing signatures
a5379ada4905d02973f0e646419ef282fa90041caaf4168f6923f218b86fa66c
```

Now check all container status.

```
ankur@ankur-Standard-PC-Q35-ICH9-2009:~/shiksha_portal/prometheus_crunchy$ podman ps
CONTAINER ID  IMAGE                           COMMAND               CREATED        STATUS            PORTS
                                              NAMES
c3b3f786f2b0  k8s.gcr.io/pause:3.5                                  4 hours ago    Up 59 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  996f92234bf5-infra
08ccb1763bee  docker.io/library/postgres:12    postgres              4 hours ago    Up 31 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  postgres_crunchy
52971cd16184  c9110d4443a6                      /opt/cpm/bin/star...  2 hours ago    Up 34 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  crunchy
97ea611052d0  docker.io/prom/prometheus:latest  --config.file=/et...  2 hours ago    Up 29 minutes ago  0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  prometheus_crunchy
a5379ada4905  docker.io/grafana/grafana:latest                       7 minutes ago  Up 7 minutes ago   0.0.0.0:3000->3000/tcp, 0.0.0.0:5432->5432/tcp,
0.0.0.0:9090->9090/tcp, 0.0.0.0:9187->9187/tcp  grafana_crunchy
```

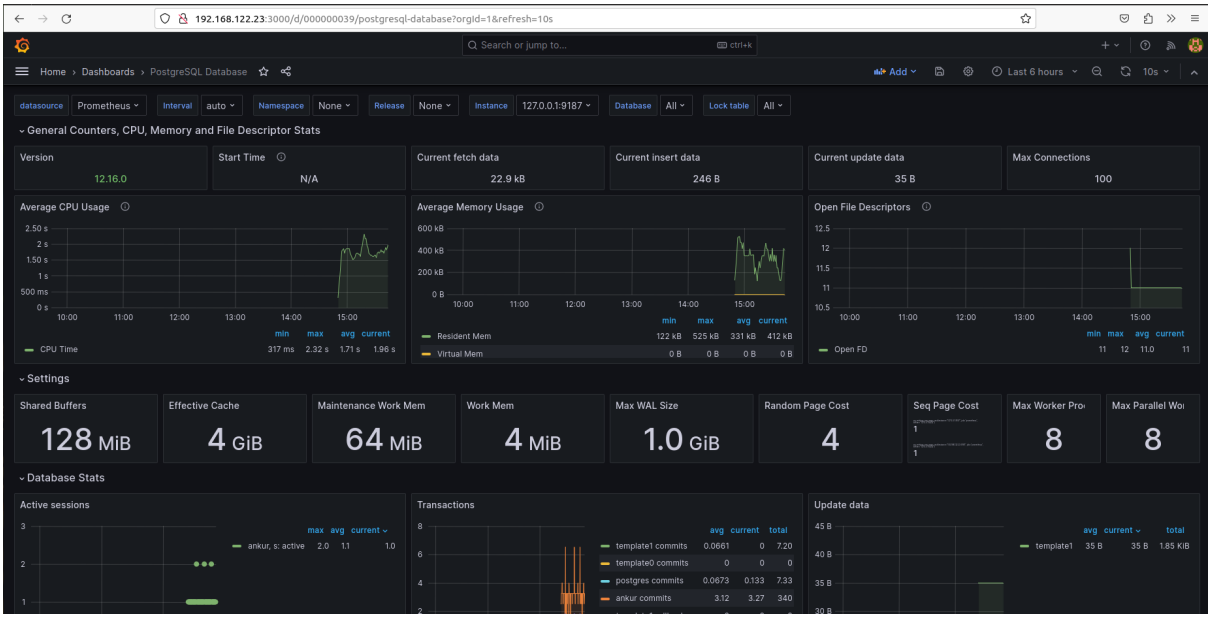
Visit http://localhost:3000/

Login to Grafana using the default ID and password (admin), and then change the password.

Click on **"Data source" > Select "Prometheus" > Paste the Prometheus URL > Save and Test.**

Next, click on **"Create dashboard" > "Import dashboard" > Enter ID 9628 > Choose "DS Prometheus" as the data source we created.**

Select **"Prometheus"** as the data source and import the dashboard with **ID 9628.**


192.168.122.23:3000/d/000000039/postgresql-database?orgId=1&refresh=10s
Search or jump to...
Home › Dashboards › PostgreSQL Database
Add
Last 6 hours
10s
datasource
Prometheus
Interval
auto
Namespace
None
Release
None
Instance
127.0.0.1:9187
Database
All
Lock table
All
General Counters, CPU, Memory and File Descriptor Stats
Version
12.16.0
Start Time
N/A
Current fetch data
22.9 kB
Current insert data
246 B
Current update data
35 B
Max Connections
100
Average CPU Usage
CPU Time
317 ms
2.32 s
1.71 s
1.96 s
Average Memory Usage
Resident Mem
122 kB
525 kB
331 kB
412 kB
Virtual Mem
0 B
0 B
0 B
0 B
Open File Descriptors
Open FD
11
12
11.0
11
Settings
Shared Buffers
128 MiB
Effective Cache
4 GiB
Maintenance Work Mem
64 MiB
Work Mem
4 MiB
Max WAL Size
1.0 GiB
Random Page Cost
4
Seq Page Cost
Max Worker Pro
8
Max Parallel Wo
8
Database Stats
Active sessions
ankur, s: active
2.0
1.1
1.0
Transactions
template1 commits
0.0661
0
7.20
template0 commits
0
0
0
postgres commits
0.0673
0.133
7.33
ankur commits
3.12
3.27
340
Update data
template1
35 B
35 B
1.85 KiB