This is the **PoC dashboard** I created to monitor our pipeline performance

**Set UP Grafana using docker**

<https://grafana.com/docs/plugins/grafana-gitlab-datasource/latest/>

**Other Commands**

ss -pnltue | grep postgresql // To see Postgres is running where and port#

OR

Use

sudo netstat -vaultnp | grep postgres

sudo -i -u postgres

psql

\conninfo

To connect directly:

psql -h localhost -U postgres -d postgres

OR with testdbuser

psql -h localhost -U testdbuser -d testdb

\l // List of databases

drop database testdb; / /DROP a DATABSE

\du \\ To see ALL ROLES and users

GRANT ALL PRIVILEGES ON DATABASE testdb TO testdbuser; //example

DROP USER IF EXISTS testdbuser; // drop or delete a user.

createuser –interactive //to create a new user and role , first go to postgres psql

sudo -i -u postgres

postgres@prashants-test3:~$ psql

psql (16.1 (Ubuntu 16.1-1.pgdg22.04+1))

Type "help" for help.

postgres=# \l

**List of databases**

Name | Owner | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules | Access privileges

-----------+----------+----------+-----------------+---------+---------+------------+-----------+-----------------------

postgres | postgres | UTF8 | libc | C.UTF-8 | C.UTF-8 | | |

template0 | postgres | UTF8 | libc | C.UTF-8 | C.UTF-8 | | | =c/postgres +

| | | | | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | libc | C.UTF-8 | C.UTF-8 | | | =c/postgres +

| | | | | | | | postgres=CTc/postgres

(3 rows)

**To uninstall Postgres**

sudo apt-get --purge remove postgresql postgresql-\*

Also list pacakges using dpkg -l | grep postgres and then remove all

sudo apt *remove* postgresql postgresql-13 postgresql-14 postgresql-16 postgresql-client postgresql-client-16 postgresql-client-common postgresql-common

**Queries used:**

SELECT table\_name  
  FROM information\_schema.tables  
WHERE table\_schema='public'  
   AND table\_name LIKE '%pipeline%'  
   AND table\_type='BASE TABLE';

**All Working QUERIES**

SELECT

CASE

WHEN (SELECT traversal\_ids[2] FROM namespaces WHERE id = $projectid) IS NULL THEN

CONCAT(

COALESCE((SELECT name FROM namespaces WHERE id = (SELECT traversal\_ids[1] FROM namespaces WHERE id = $projectid)), ''),

'/',

COALESCE((SELECT name FROM namespaces WHERE id = (SELECT traversal\_ids[1] FROM namespaces WHERE id = $projectid)), '')

)

ELSE

CONCAT(

COALESCE((SELECT name FROM namespaces WHERE id = (SELECT traversal\_ids[1] FROM namespaces WHERE id = $projectid)), ''),

'/',

COALESCE((SELECT name FROM namespaces WHERE id = (SELECT traversal\_ids[2] FROM namespaces WHERE id = $projectid)), '')

)

END AS concatenated\_name

FROM namespaces

WHERE id = $projectid;

**Learning Queries**

SELECT CURRENT\_DATE, CURRENT\_TIME, CURRENT\_TIMESTAMP;

current\_date | current\_time | current\_timestamp

--------------+--------------------+-------------------------------

2023-12-12 | 07:37:12.486072+00 | 2023-12-12 07:37:12.486072+00

SELECT AGE('1990-05-15'::DATE, '2023-01-15'::DATE) AS age\_in\_years;

age\_in\_years

-------------------

-32 years -8 mons

Current timestamp SELECT NOW();

Convert string to timestamp SELECT '2023-01-15 12:30:00'::TIMESTAMP;

timestamp

---------------------

2023-01-15 12:30:00

SELECT CURRENT\_TIMESTAMP + INTERVAL '1 day' AS tomorrow,

CURRENT\_TIMESTAMP - INTERVAL '1 week' AS last\_week;

SELECT ('2023-02-15'::DATE - '2023-01-15'::DATE) AS days\_difference;

days\_difference

-----------------

31

SELECT TO\_CHAR('2023-01-15'::DATE, 'Day') AS weekday\_name;

weekday\_name

--------------

Sunday

SELECT '2023-01-15 12:00:00'::TIMESTAMPTZ AT TIME ZONE 'UTC' AS utc\_time;

utc\_time

---------------------

2023-01-15 12:00:00

**Use the SUM() window function to calculate the running total of a time-series dataset**

SELECT time, wins,

SUM(wins) OVER (ORDER BY time) AS running\_total

FROM tourneys;

time | wins | running\_total

----------------------------+------+---------------

2023-12-11 08:08:25.532248 | 7 | 7

2023-12-11 08:14:46.609398 | 9 | 16

2023-12-11 08:15:12.116748 | 10 | 26

2023-12-11 08:15:34.405358 | 3 | 29

SELECT EXTRACT(HOUR FROM '2023-01-15 08:45:00'::TIMESTAMP) AS hour,

EXTRACT(MINUTE FROM '2023-01-15 08:45:00'::TIMESTAMP) AS minute;

hour | minute

------+--------

8 | 45

SELECT EXTRACT(YEAR FROM CURRENT\_DATE) as year\_result;

year\_result

-------------

2023

Conditional Date format

SELECT

CASE

WHEN EXTRACT(MONTH FROM CURRENT\_DATE) = 12 THEN TO\_CHAR(CURRENT\_DATE, 'YYYY-MM-DD')

ELSE TO\_CHAR(CURRENT\_DATE, 'DD/MM/YYYY')

END AS formatted\_date;

formatted\_date

----------------

2023-12-12

Retrieve records from a specific date range:

SELECT \*

FROM your\_table

WHERE date\_column BETWEEN '2023-01-01' AND '2023-01-31';

SELECT

SUBSTRING\_INDEX('2023-06-29 05:30:00', ' ', 1) AS date\_part,

SUBSTRING\_INDEX('2023-06-29 05:30:00', ' ', -1) AS time\_part;

+------------+-----------+

| date\_part | time\_part |

+------------+-----------+

| 2023-06-29 | 05:30:00 |

DATE\_FORMAT('2023-06-29 05:30:00', '%Y') AS year,

DATE\_FORMAT('2023-06-29 05:30:00', '%m') AS month,

DATE\_FORMAT('2023-06-29 05:30:00', '%d') AS day;

+------+-------+------+

| year | month | day |

+------+-------+------+

| 2023 | 06 | 29 |

To modify your PostgreSQL SQL query to group the data by months using the **$\_\_timeGroupAlias** macro in Grafana, you should adjust the time grouping interval to **'1 mon'** (1 month). Here's the modified query:

SELECT $\_\_timeGroupAlias("time", '1 mon', 0) AS time\_group,

SUM(wins) AS total\_wins,

MAX(size) AS max\_size

FROM tourneys

WHERE $\_\_timeFilter("time") GROUP BY time\_group ORDER BY time\_group;

-- SELECT UNNEST(STRING\_TO\_ARRAY(

-- REGEXP\_REPLACE(your\_column, '[^0-9,]', '', 'g'), ','))::INT AS result

-- FROM cte\_output;

--SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(CONCAT(column1, ',', column2), ',', 1), ',', -1) AS result

--FROM your\_table;

-- SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(CONCAT(column1, ',', column2), ',', 1), ',', -1) AS result

-- FROM your\_table;

-- SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(CONCAT(column1, ',', column2), ',', 1), ',', -1) AS result

-- FROM your\_table;

-- Declare and set a variable

DECLARE

my\_variable INT := 0;

BEGIN

-- Run your query and save the result into the variable

SELECT DISTINCT name INTO my\_variable

FROM projects

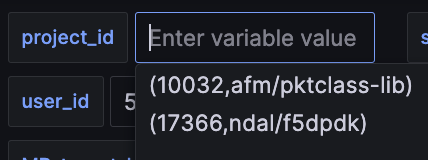
LIMIT 1;

-- Print the value of the variable

RAISE NOTICE 'my\_variable: %', my\_variable;

**Use of Datalinks:**

Example of how to use data links



Project\_id has project# and project path

Extract project path from project\_id and then use it as dynamic path in data links:

SPLIT\_PART(SPLIT\_PART('$project\_id', ',', 2), ')', 1) AS project\_path -- For dynamic project path link to MR

**Installing pgBouncer**

<https://www.youtube.com/watch?v=ddKm7a7xOpk>

<https://www.youtube.com/watch?v=XkyP2h6YOj4&t=990s>

existing repo with PgBouncer example: https://gitswarm.f5net.com/mbip/tools/pipeline-status#install-and-configure-pgbouncer

<https://pgdash.io/blog/pgbouncer-connection-pool.html>

sudo apt update

sudo apt install pgbouncer -y

systemctl status pgbouncer

sudo apt install postgresql-client ## required to test out the things.

Edit do settings for pgbouncer.ini and userlist.txt

**NOTE**: after changes to pgbouncer.ini restart the pgbouncer service

File userlist.txt contents

"readonlyuser" "4bU\*\*\*\*\*\*\*\*\*PXI"

File pgbouncer.ini contents

[databases]

pgbouncer\_db = host=gitlabxxxxxxx.com port=5432 dbname=gitlabhq\_production auth\_user=readonlyuser

listen\_addr = \*

listen\_port = 6432

auth\_type = md5

auth\_file = /etc/pgbouncer/userlist.txt

pool\_mode = transaction

ignore\_startup\_parameters = extra\_float\_digits

max\_client\_conn = 100

default\_pool\_size = 20

min\_pool\_size = 0

**Ensure Service is UP and Running:**

systemctl enable **pgbounce**r.service

systemctl status **pgbounce**r.service

sudo journalctl -xeu **pgbounce**r.service

systemctl restart **pgbounce**r.service

 systemctl status **pgbounce**r.service

**Reference:**

For Queries example in Postgres see : <https://www.digitalocean.com/community/tutorials/introduction-to-queries-postgresql>

<https://phoenixnap.com/kb/postgres-create-user>

<https://linuxhint.com/install-and-setup-postgresql-database-ubuntu-22-04/>

<https://cloudinfrastructureservices.co.uk/how-to-install-postgresql-on-ubuntu-22-04-server/>

<https://itslinuxfoss.com/install-setup-postgresql-database-ubuntu-22-04/#google_vignette>

<https://www.prisma.io/dataguide/postgresql/connecting-to-postgresql-databases>

<https://www.digitalocean.com/community/tutorials/introduction-to-queries-postgresql>