

all setup in live database i need to loki also in log not in localhost live ip this is postgresql ip 192.168.6.208:5432 all core set up in ubuntu

ChatGPT said:

ChatGPT

To set up **Grafana** with a live **PostgreSQL** database on IP `192.168.6.208:5432` and integrate it with **Loki** for logging, along with Prometheus for monitoring, follow the comprehensive steps below. This setup assumes an Ubuntu environment.

1. Install and Configure PostgreSQL Exporter

To monitor the live PostgreSQL database, use the **PostgreSQL Exporter**.

Install PostgreSQL Exporter:

bash

Copy code

wget

```
https://github.com/prometheus-community/postgres_exporter/releases/download/v0.13.0/postgres_exporter-0.13.0.linux-amd64.tar.gz
tar -xvzf postgres_exporter-0.13.0.linux-amd64.tar.gz
mv postgres_exporter /usr/local/bin/
```

1.

Set Up Data Source for Exporter: Create a PostgreSQL user with read-only permissions:

sql

Copy code

```
CREATE USER exporter WITH PASSWORD 'exporter_password';
GRANT CONNECT ON DATABASE your_database TO exporter;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO exporter;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO exporter;
```

2.

Run PostgreSQL Exporter:

bash

Copy code

postgres_exporter

```
--data-source-name="postgresql://exporter:exporter_password@192.168.6.208:5432/your_database" &
```

3. The exporter will run on port 9187 by default.
-

2. Install Prometheus

Download and Install Prometheus:

bash

Copy code

wget

```
https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz
tar -xvzf prometheus-2.47.0.linux-amd64.tar.gz
mv prometheus-2.47.0.linux-amd64 /usr/local/bin/prometheus
```

- 1.

Configure Prometheus: Edit the configuration file `prometheus.yml`:

yaml

Copy code

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'postgres'
    static_configs:
      - targets: ['192.168.6.208:9187']
```

- 2.

Start Prometheus:

bash

Copy code

```
./prometheus --config.file=prometheus.yml &
```

3. Prometheus will run on port 9090 by default.
-

3. Install Loki for Centralized Logging

Install Loki:

bash

Copy code

```
wget
```

```
https://github.com/grafana/loki/releases/download/v2.9.1/loki-linux-amd64.zip
```

```
unzip loki-linux-amd64.zip
```

```
chmod +x loki-linux-amd64
```

```
mv loki-linux-amd64 /usr/local/bin/loki
```

1.

Configure Loki: Create a configuration file `loki-config.yaml`:

yaml

Copy code

```
server:
```

```
  http_listen_port: 3100
```

```
ingester:
```

```
  wal:
```

```
    enabled: true
```

```
  chunk_idle_period: 5m
```

```
  max_chunk_age: 1h
```

```
  chunk_target_size: 1572864
```

```
schema_config:
```

```
  configs:
```

```
    - from: 2022-05-01
```

```
      store: boltdb-shipper
```

```
      object_store: filesystem
```

```
      schema: v11
```

```
      index:
```

```
        prefix: index_
```

```
        period: 24h
```

```
storage_config:
```

```
  boltdb_shipper:
```

```
    active_index_directory: /tmp/loki/boltdb-shipper-active
```

```
cache_location: /tmp/loki/boltdb-shipper-cache
shared_store: filesystem
```

```
filesystem:
  directory: /tmp/loki/chunks
```

```
limits_config:
  retention_period: 30d
```

```
chunk_store_config:
  max_look_back_period: 0s
```

2.

Run Loki:

bash

Copy code

```
loki --config.file=loki-config.yaml &
```

3. Loki will run on port **3100**.

4. Configure Grafana

Install Grafana:

bash

Copy code

```
sudo apt-get install grafana
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

1.

2. **Access Grafana:**

- Open: **http://<your-live-ip>:3000**.
- Default credentials: **admin / admin**.

3. **Add Data Sources:**

- **PostgreSQL:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **PostgreSQL** and configure:
 - Host: **192.168.6.208:5432**
 - Database: **your_database**

- User: `exporter`
 - Password: `exporter_password`
 - SSL: **Disable** (if no SSL is used).
 - Test and save.
 - **Prometheus:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Prometheus**.
 - URL: `http://<your-live-ip>:9090`.
 - Test and save.
 - **Loki:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Loki**.
 - URL: `http://<your-live-ip>:3100`.
 - Test and save.
4. **Create Dashboards:**
- For PostgreSQL Logs:

Use queries like:

sql

Copy code

```
SELECT timestamp AS time, status, message
FROM logs
ORDER BY timestamp DESC;
```

-
- For Prometheus Metrics:

Query examples:

promql

Copy code

```
pg_stat_database_connections{datname="your_database"}
```

-
- For Loki Logs:
 - Use `{job="your_job"}` to filter logs.

5. Set Up Alerts in Grafana

1. **Navigate to Alerts:**
 - Create alerts based on Prometheus or PostgreSQL metrics.

For example, alert when PostgreSQL connections exceed a threshold:

promql

Copy code

```
pg_stat_database_connections{datname="your_database"} > 50
```

○

2. Set Notification Channels:

- Configure email, Slack, or other integrations.

6. Finalize Security

1. Secure Grafana:

- Enable HTTPS for Grafana.
- Configure strong passwords for all accounts.

2. Firewall Rules:

- Open only necessary ports (5432, 9090, 3100, 3000).

Use `ufw`:

bash

Copy code

```
sudo ufw allow 5432/tcp
```

```
sudo ufw allow 9090/tcp
```

```
sudo ufw allow 3100/tcp
```

```
sudo ufw allow 3000/tcp
```

○

Let me know if further details are needed!

4o

You said:

my live database in another pc then you give me proper setup

ChatGPT said:

ChatGPT

Here's the step-by-step guide for setting up **Grafana**, **Loki**, **Prometheus**, and connecting to a live PostgreSQL database hosted on another PC with IP `192.168.6.208:5432`. This guide ensures a complete setup for monitoring, logging, and visualization.

1. Prerequisites

- Ensure the database server (192.168.6.208) allows incoming connections:

Edit the PostgreSQL configuration file (`postgresql.conf`):

conf

Copy code

```
listen_addresses = '*'
```

○

Add the client IP of your Grafana/Prometheus server in `pg_hba.conf`:

conf

Copy code

```
host all all <Grafana_Server_IP>/32 md5
```

○

Restart PostgreSQL:

bash

Copy code

```
sudo systemctl restart postgresql
```

○

Open the PostgreSQL port (`5432`) on the database server:

bash

Copy code

```
sudo ufw allow 5432/tcp
```

-

2. Install and Configure PostgreSQL Exporter

To monitor the PostgreSQL database from Grafana via Prometheus:

Download PostgreSQL Exporter: On the server hosting Prometheus:

bash

Copy code

```
wget
```

```
https://github.com/prometheus-community/postgres_exporter/releases/download/v0.13.0/postgres_exporter-0.13.0.linux-amd64.tar.gz
```

```
tar -xvzf postgres_exporter-0.13.0.linux-amd64.tar.gz
sudo mv postgres_exporter /usr/local/bin/
```

1.

Create a Read-Only User in PostgreSQL: On the database server (192.168.6.208):

sql

Copy code

```
CREATE USER exporter WITH PASSWORD 'exporter_password';
GRANT CONNECT ON DATABASE your_database TO exporter;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO exporter;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO
exporter;
```

2.

Run PostgreSQL Exporter: On the Prometheus server:

bash

Copy code

```
postgres_exporter
--data-source-name="postgresql://exporter:exporter_password@192.168.6.
208:5432/your_database" &
```

3. The exporter runs on port 9187 by default.

3. Install and Configure Prometheus

Install Prometheus:

bash

Copy code

wget

```
https://github.com/prometheus/prometheus/releases/download/v2.47.0/pro
metheus-2.47.0.linux-amd64.tar.gz
tar -xvzf prometheus-2.47.0.linux-amd64.tar.gz
sudo mv prometheus-2.47.0.linux-amd64 /usr/local/bin/prometheus
```

1.

Edit Prometheus Configuration: Create or edit `prometheus.yml`:

yaml

Copy code

```
global:
```



```
scrape_interval: 15s
```

```
scrape_configs:
```

```
- job_name: 'postgres'
  static_configs:
    - targets: ['192.168.6.208:9187']
```

2.

Start Prometheus:

bash

Copy code

```
prometheus --config.file=prometheus.yml &
```

3. Prometheus runs on port 9090.

4. Install and Configure Loki

Install Loki:

bash

Copy code

```
wget
```

```
https://github.com/grafana/loki/releases/download/v2.9.1/loki-linux-amd64.zip
```

```
unzip loki-linux-amd64.zip
```

```
chmod +x loki-linux-amd64
```

```
sudo mv loki-linux-amd64 /usr/local/bin/loki
```

1.

Create Loki Configuration: Create loki-config.yaml:

yaml

Copy code

```
server:
```

```
  http_listen_port: 3100
```

```
ingester:
```

```
  wal:
```

```
    enabled: true
```

```
  chunk_idle_period: 5m
```

```
max_chunk_age: 1h
chunk_target_size: 1572864
```

schema_config:

```
configs:
  - from: 2022-05-01
    store: boltdb-shipper
    object_store: filesystem
    schema: v11
    index:
      prefix: index_
      period: 24h
```

storage_config:

```
boltdb_shipper:
  active_index_directory: /var/lib/loki/boltdb-shipper-active
  cache_location: /var/lib/loki/boltdb-shipper-cache
  shared_store: filesystem
```

```
filesystem:
  directory: /var/lib/loki/chunks
```

limits_config:

```
retention_period: 30d
```

2.

Run Loki:

bash

Copy code

```
loki --config.file=loki-config.yaml &
```

3. Loki runs on port 3100.

5. Install and Configure Grafana

Install Grafana:

bash

Copy code

```
sudo apt-get update
```

```
sudo apt-get install grafana
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

- 1.
2. **Access Grafana:** Open `http://<Grafana_Server_IP>:3000` in your browser.
Default credentials: `admin / admin`.
3. **Add Data Sources:**
 - **PostgreSQL:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **PostgreSQL**.
 - Fill in the details:
 - **Host:** `192.168.6.208:5432`
 - **Database:** `your_database`
 - **User:** `exporter`
 - **Password:** `exporter_password`
 - **SSL Mode:** `disable` (if no SSL).
 - Test and save.
 - **Prometheus:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Prometheus**.
 - URL: `http://<Prometheus_Server_IP>:9090`.
 - Test and save.
 - **Loki:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Loki**.
 - URL: `http://<Grafana_Server_IP>:3100`.
 - Test and save.
4. **Create Dashboards:**

PostgreSQL Logs:

sql

Copy code

```
SELECT timestamp AS time, status, message
FROM logs
ORDER BY timestamp DESC;
```

○

Prometheus Metrics:

promql

Copy code

```
pg_stat_database_connections{datname="your_database"}
```

○

6. Secure the Setup

Firewall Rules: On the database server:

bash

Copy code

```
sudo ufw allow from <Grafana_Server_IP> to any port 5432
```

On the Grafana server:

bash

Copy code

```
sudo ufw allow 3000/tcp
```

```
sudo ufw allow 3100/tcp
```

```
sudo ufw allow 9090/tcp
```

- 1.
2. **Enable HTTPS for Grafana:** Configure SSL certificates for secure access to Grafana.

With this setup, Grafana will monitor metrics from Prometheus, logs from Loki, and query live PostgreSQL data from the external database (192.168.6.208:5432). Let me know if further clarification is needed!

4o

You said:

You set by set in introction which setup in my pc live database means another pc

ChatGPT said:

ChatGPT

Here's a clear **step-by-step guide** to set up monitoring and logging for a **live PostgreSQL database** hosted on another PC (192.168.6.208) using **Grafana**, **Prometheus**, and **Loki**. This guide assumes you are setting up everything on your local PC (Ubuntu) while connecting to the live database on the other PC.

Step 1: Verify Connectivity to the Live PostgreSQL Database

Test Database Connectivity: From your local PC, run:

bash

Copy code

```
psql -h 192.168.6.208 -U your_username -d your_database
```

1. Replace `your_username` and `your_database` with your PostgreSQL credentials.

If the connection fails, ensure:

- PostgreSQL on 192.168.6.208 is configured to accept remote connections.

`pg_hba.conf` on the database server includes your local PC's IP:

conf

Copy code

```
host all all <your_local_ip>/32 md5
```

-
- PostgreSQL is listening on all interfaces (`listen_addresses = '*'` in `postgresql.conf`).

Port `5432` is open on the database server's firewall:

bash

Copy code

```
sudo ufw allow 5432/tcp
```

-

Step 2: Install PostgreSQL Exporter on Your PC

PostgreSQL Exporter will gather metrics from the live database.

Download and Install Exporter:

bash

Copy code

```
wget
```

```
https://github.com/prometheus-community/postgres\_exporter/releases/download/v0.13.0/postgres\_exporter-0.13.0.linux-amd64.tar.gz
```

```
tar -xvzf postgres_exporter-0.13.0.linux-amd64.tar.gz
```

```
sudo mv postgres_exporter /usr/local/bin/
```

- 1.

Create a Monitoring User on the Database Server: On the database server (192.168.6.208):

sql

Copy code

```
CREATE USER exporter WITH PASSWORD 'exporter_password';
```

```
GRANT CONNECT ON DATABASE your_database TO exporter;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO exporter;  
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO  
exporter;
```

2.

Run PostgreSQL Exporter on Your PC:

bash

Copy code

```
postgres_exporter  
--data-source-name="postgresql://exporter:exporter_password@192.168.6.  
208:5432/your_database" &
```

3. The exporter runs on port **9187** by default.

Step 3: Install and Configure Prometheus

Download and Install Prometheus:

bash

Copy code

```
wget  
https://github.com/prometheus/prometheus/releases/download/v2.47.0/pro  
metheus-2.47.0.linux-amd64.tar.gz  
tar -xvzf prometheus-2.47.0.linux-amd64.tar.gz  
sudo mv prometheus-2.47.0.linux-amd64 /usr/local/bin/prometheus
```

1.

Create Prometheus Configuration: Edit `prometheus.yml` to include the PostgreSQL

Exporter:

yaml

Copy code

```
global:  
  scrape_interval: 15s  
  
scrape_configs:  
  - job_name: 'postgres'  
    static_configs:  
      - targets: ['localhost:9187']
```

2.

Start Prometheus:

bash

Copy code

```
prometheus --config.file=prometheus.yml &
```

3. Prometheus runs on port 9090.

Step 4: Install and Configure Loki

Loki will collect logs for visualization in Grafana.

Download and Install Loki:

bash

Copy code

```
wget
https://github.com/grafana/loki/releases/download/v2.9.1/loki-linux-amd64.zip
unzip loki-linux-amd64.zip
chmod +x loki-linux-amd64
sudo mv loki-linux-amd64 /usr/local/bin/loki
```

1.

Create Loki Configuration: Create loki-config.yaml:

yaml

Copy code

```
server:
  http_listen_port: 3100

ingester:
  wal:
    enabled: true
  chunk_idle_period: 5m
  max_chunk_age: 1h
  chunk_target_size: 1572864

schema_config:
  configs:
```

```
- from: 2022-05-01
  store: boltdb-shipper
  object_store: filesystem
  schema: v11
  index:
    prefix: index_
    period: 24h
```

```
storage_config:
  boltdb_shipper:
    active_index_directory: /var/lib/loki/boltdb-shipper-active
    cache_location: /var/lib/loki/boltdb-shipper-cache
    shared_store: filesystem
```

```
filesystem:
  directory: /var/lib/loki/chunks
```

```
limits_config:
  retention_period: 30d
```

2.

Run Loki:

bash

Copy code

```
loki --config.file=loki-config.yaml &
```

3. Loki runs on port 3100.

Step 5: Install and Configure Grafana

Install Grafana:

bash

Copy code

```
sudo apt-get update
sudo apt-get install grafana
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

1.

2. **Access Grafana:** Open `http://<your_local_pc_ip>:3000` in a browser.
Default credentials: `admin / admin`.
 3. **Add Data Sources:**
 - **PostgreSQL:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **PostgreSQL** and configure:
 - Host: `192.168.6.208:5432`
 - Database: `your_database`
 - User: `exporter`
 - Password: `exporter_password`
 - SSL Mode: Disable (if no SSL is configured).
 - Test and save.
 - **Prometheus:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Prometheus**.
 - URL: `http://localhost:9090`.
 - Test and save.
 - **Loki:**
 - Navigate to **Configuration > Data Sources > Add data source**.
 - Select **Loki**.
 - URL: `http://localhost:3100`.
 - Test and save.
-

Step 6: Create Dashboards in Grafana

PostgreSQL Metrics: Use Prometheus metrics like:

`promql`

Copy code

```
pg_stat_database_connections{datname="your_database"}
```

1.

Logs via Loki: Query logs with:

`plaintext`

Copy code

```
{job="postgresql"}
```

2.

3. **Combine Metrics and Logs:** Create panels combining metrics from Prometheus and logs from Loki for better insights.
-

Step 7: Secure the Setup

Firewall Rules: On your local PC:

bash

Copy code

```
sudo ufw allow 3000/tcp # Grafana
sudo ufw allow 3100/tcp # Loki
sudo ufw allow 9090/tcp # Prometheus
```

- 1.
2. **Enable HTTPS for Grafana:** Configure SSL for secure web access.

This setup integrates your live PostgreSQL database, logs, and metrics for monitoring and visualization in Grafana. Let me know if you need further clarification!