

To set up PostgreSQL logs from your database PC (IP: 192.168.6.208) and forward them to Grafana Loki installed on your PC (IP: 192.168.6.108), follow these step-by-step instructions:

Setup on Your PC (192.168.6.108): Install and Configure Loki, Grafana, and Promtail

Step 1: Install Grafana

Import GPG Key and Add Grafana Repository:

bash

Copy code

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb  
stable main"
```

1.

Update Repositories and Install Grafana:

bash

Copy code

```
sudo apt update  
sudo apt install grafana
```

2.

Start and Enable Grafana:

bash

Copy code

```
sudo systemctl start grafana-server  
sudo systemctl enable grafana-server
```

3.

4. **Access Grafana UI:** Open <http://192.168.6.108:3000> in a browser.

- Default username: [admin](#)
- Default password: [admin](#) (change or skip when prompted).

Step 2: Install and Configure Loki

Download and Install Loki Binary:

bash

Copy code

```
curl -s https://api.github.com/repos/grafana/loki/releases/latest |  
grep browser_download_url | grep loki-linux-amd64.zip | cut -d '"' -f  
4 | wget -i -  
sudo apt install unzip -y  
unzip loki-linux-amd64.zip  
sudo mv loki-linux-amd64 /usr/local/bin/loki
```

- 1.
2. **Configure Loki:**

Create the required directories:

bash

Copy code

```
sudo mkdir -p /data/loki
```

○

Download and edit the Loki config:

bash

Copy code

```
sudo wget -O /etc/loki-local-config.yaml
```

```
https://raw.githubusercontent.com/grafana/loki/main/cmd/loki/loki-local-config.yaml
```

```
sudo nano /etc/loki-local-config.yaml
```

○

Update the config:

yaml

Copy code

```
auth_enabled: false
```

```
server:
```

```
  http_listen_port: 3100
```

```
common:
```

```
  path_prefix: /data/loki
```

```
  storage:
```

```
    filesystem:
```

```
      chunks_directory: /data/loki/chunks
```

```
      rules_directory: /data/loki/rules
```

```
schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h
```

○

Create a Systemd Service for Loki:

bash

Copy code

```
sudo nano /etc/systemd/system/loki.service
```

Add:

ini

Copy code

```
[Unit]
```

```
Description=Loki service
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/local/bin/loki -config.file /etc/loki-local-config.yaml
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Reload, start, and enable Loki:

bash

Copy code

```
sudo systemctl daemon-reload
```

```
sudo systemctl start loki.service
```

```
sudo systemctl enable loki.service
```

3.

Step 3: Install Promtail

Download and Install Promtail Binary:

bash

Copy code

```
curl -s https://api.github.com/repos/grafana/loki/releases/latest |  
grep browser_download_url | grep promtail-linux-amd64.zip | cut -d '"'  
-f 4 | wget -i -  
unzip promtail-linux-amd64.zip  
sudo mv promtail-linux-amd64 /usr/local/bin/promtail
```

1.

2. Configure Promtail:

Create the Promtail config:

bash

Copy code

```
sudo nano /etc/promtail-local-config.yaml
```

○

Add:

yaml

Copy code

server:

```
  http_listen_port: 9080
```

positions:

```
  filename: /data/loki/positions.yaml
```

clients:

```
- url: http://192.168.6.108:3100/loki/api/v1/push
```

scrape_configs:

```
- job_name: postgres-logs
```

```
  static_configs:
```

```
    - targets:
```

```
      - 192.168.6.208 # PostgreSQL server IP
```

```
    labels:
```

```
      job: postgres
```

```
      __path__: /var/log/postgresql/*.log
```

○

Create a Systemd Service for Promtail:

bash

Copy code

```
sudo nano /etc/systemd/system/promtail.service
```

Add:

ini

Copy code

```
[Unit]
```

```
Description=Promtail service
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/local/bin/promtail -config.file
```

```
/etc/promtail-local-config.yaml
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Reload, start, and enable Promtail:

bash

Copy code

```
sudo systemctl daemon-reload
```

```
sudo systemctl start promtail.service
```

```
sudo systemctl enable promtail.service
```

3.

Setup on Database PC (192.168.6.208): Configure PostgreSQL for Logging

1. Enable Logging in PostgreSQL:

Open the PostgreSQL configuration file:

bash

Copy code

```
sudo nano /etc/postgresql/12/main/postgresql.conf
```

○

Update these parameters:

ini

Copy code

```
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_statement = 'all'
log_duration = on
```

○

Restart PostgreSQL:

bash

Copy code

```
sudo systemctl restart postgresql
```

○

2. **Share the Logs Directory via SSH:** Ensure Promtail can access `/var/log/postgresql` via the `192.168.6.108` machine. Use `rsync` or `scp` if direct access isn't possible.

Step 4: Configure Loki Data Source in Grafana

1. Open Grafana UI: <http://192.168.6.108:3000>.
2. Navigate to **Connections** → **Data Sources** → **Add Data Source**.
3. Select **Loki**, set the URL as <http://192.168.6.108:3100>, and click **Save & Test**.

Step 5: Query Logs in Grafana

1. Navigate to **Explore** in Grafana.
2. Select your **Loki** data source.

Enter a query to filter logs, e.g.:

logql

Copy code

```
{job="postgres"}
```

3.

You should now see the PostgreSQL logs from `192.168.6.208` visualized on your Grafana dashboard.

40

You said:

this setup i am see database loges in my pc 108

ChatGPT said:

ChatGPT

To set up a system where PostgreSQL logs from a remote database server (IP: 192.168.6.208) are forwarded to Loki and visualized in Grafana on your local PC (IP: 192.168.6.108), follow the steps below:

On the Database Server (192.168.6.208)

1. Enable PostgreSQL Logging

Ensure PostgreSQL logging is enabled on the database server by modifying its configuration file:

bash

Copy code

```
sudo vim /etc/postgresql/<version>/main/postgresql.conf
```

- Set the following parameters:

conf

Copy code

```
logging_collector = on
log_directory = 'pg_log'
log_filename = 'postgresql-%Y-%m-%d.log'
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
```

- Restart PostgreSQL to apply the changes:

bash

Copy code

```
sudo systemctl restart postgresql
```

2. Install and Configure Promtail

Download and install Promtail:

bash

Copy code

```
curl -s https://api.github.com/repos/grafana/loki/releases/latest |  
grep browser_download_url | grep promtail-linux-amd64.zip | cut -d '"'  
-f 4 | wget -i -  
sudo apt install unzip -y  
unzip promtail-linux-amd64.zip  
sudo mv promtail-linux-amd64 /usr/local/bin/promtail
```

1.

Create a configuration file for Promtail:

bash

Copy code

```
sudo vim /etc/promtail-local-config.yaml
```

Add the following content:

yaml

Copy code

```
server:  
  http_listen_port: 9080  
  grpc_listen_port: 0  
  
positions:  
  filename: /var/log/promtail-positions.yaml  
  
clients:  
  - url: http://192.168.6.108:3100/loki/api/v1/push  
  
scrape_configs:  
  - job_name: postgresql_logs  
    static_configs:  
      - targets:  
          - localhost  
        labels:  
          job: postgresql  
          host: "192.168.6.208"  
          __path__: /var/lib/postgresql/<version>/pg_log/*.log
```

2. Replace <version> with the PostgreSQL version (e.g., 13, 14).

Create a systemd service for Promtail:

bash

Copy code

```
sudo vim /etc/systemd/system/promtail.service
```

Add the following content:

ini

Copy code

```
[Unit]
```

```
Description=Promtail service
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/local/bin/promtail -config.file
```

```
/etc/promtail-local-config.yaml
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```

3.

Start and enable Promtail:

bash

Copy code

```
sudo systemctl daemon-reload
```

```
sudo systemctl start promtail
```

```
sudo systemctl enable promtail
```

4.

On Your Local PC (192.168.6.108)

1. Install and Configure Loki

Download and install Loki:

bash

Copy code

```
curl -s https://api.github.com/repos/grafana/loki/releases/latest |  
grep browser_download_url | grep loki-linux-amd64.zip | cut -d '"' -f  
4 | wget -i -
```

```
sudo apt install unzip -y
unzip loki-linux-amd64.zip
sudo mv loki-linux-amd64 /usr/local/bin/loki
```

1.

Create a Loki configuration file:

bash

Copy code

```
sudo vim /etc/loki-local-config.yaml
```

Add the following content:

yaml

Copy code

```
auth_enabled: false
```

```
server:
```

```
  http_listen_port: 3100
```

```
  grpc_listen_port: 9095
```

```
common:
```

```
  path_prefix: /var/lib/loki
```

```
  storage:
```

```
    filesystem:
```

```
      chunks_directory: /var/lib/loki/chunks
```

```
      rules_directory: /var/lib/loki/rules
```

```
  replication_factor: 1
```

```
  ring:
```

```
    kvstore:
```

```
      store: inmemory
```

```
schema_config:
```

```
  configs:
```

```
    - from: 2020-10-24
```

```
      store: boltdb
```

```
      object_store: filesystem
```

```
      schema: v11
```

```
      index:
```

```
        prefix: index_
```

```
        period: 24h
```

```
ruler:
  alertmanager_url: ""
```

2.

Create necessary directories:

bash

Copy code

```
sudo mkdir -p /var/lib/loki/chunks
sudo mkdir -p /var/lib/loki/rules
```

3.

Create a systemd service for Loki:

bash

Copy code

```
sudo vim /etc/systemd/system/loki.service
```

Add the following content:

ini

Copy code

```
[Unit]
Description=Loki service
After=network.target

[Service]
ExecStart=/usr/local/bin/loki -config.file /etc/loki-local-config.yaml
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

4.

Start and enable Loki:

bash

Copy code

```
sudo systemctl daemon-reload
sudo systemctl start loki
sudo systemctl enable loki
```

5.

2. Install and Configure Grafana

Install Grafana:

bash

Copy code

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb  
stable main"  
sudo apt update  
sudo apt install grafana -y
```

1.

Start and enable Grafana:

bash

Copy code

```
sudo systemctl start grafana-server  
sudo systemctl enable grafana-server
```

2.

3. Access Grafana:

- Open a browser and navigate to <http://192.168.6.108:3000>.
- Default login credentials:
 - Username: [admin](#)
 - Password: [admin](#)

3. Configure Loki as a Data Source in Grafana

1. In Grafana, go to **Configuration > Data Sources > Add Data Source**.
2. Select **Loki**.
3. Enter the URL: <http://localhost:3100>.
4. Click **Save & Test**.

4. Visualize PostgreSQL Logs

1. Go to **Explore** in Grafana.
2. Select the **Loki** data source.

Run the following query to visualize PostgreSQL logs:

logql

Copy code

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

3.

Testing the Setup

- Generate some logs on the database server by interacting with PostgreSQL.
- Check Grafana to confirm that the logs are being visualized correctly.

This setup ensures that PostgreSQL logs from the remote server (192.168.6.208) are forwarded to Loki and visualized on your PC (192.168.6.108).

4o

You said:

you clear me Promtail install in database pc 208 and loki and grafan setup in 108 mean my pc i am right

ChatGPT said:

ChatGPT

Yes, you are correct! Here's the setup summary for clarity:

1. On the Database Server (192.168.6.208)

- **Install Promtail:**
Promtail will collect PostgreSQL logs from the database server.
 - Configure Promtail to point to your PC (192.168.6.108), where Loki is running.
 - Promtail reads logs from the PostgreSQL log directory (e.g., `/var/lib/postgresql/<version>/pg_log/*.log`) and sends them to Loki.

2. On Your PC (192.168.6.108)

- **Install and Configure Loki:**
Loki will receive logs from Promtail running on the database server.
- **Install and Configure Grafana:**
Grafana will connect to Loki and visualize the logs.

This separation ensures:

- Promtail runs **on the database server** (208) to collect and forward logs.
- Loki and Grafana run **on your PC** (108) to store and display logs.

Let me know if you need further details!

40

You said:

this Create a Loki configuration file: bash Copy code sudo vim /etc/loki-local-config.yaml Add the following content: yaml Copy code auth_enabled: false server: http_listen_port: 3100 grpc_listen_port: 9095 common: path_prefix: /var/lib/loki storage: filesystem: chunks_directory: /var/lib/loki/chunks rules_directory: /var/lib/loki/rules replication_factor: 1 ring: kvstore: store: inmemory schema_config: configs: - from: 2020-10-24 store: boltdb object_store: filesystem schema: v11 index: prefix: index_period: 24h ruler: alertmanager_url: "" this loki configuration file alertmanager_url: "" mean which url puting this 108 208 which ip clear me explaine me

ChatGPT said:

ChatGPT

The `alertmanager_url` field in the Loki configuration is **only used if you want Loki to send alerts to an Alertmanager instance** (commonly part of the Prometheus ecosystem). If you're not using an Alertmanager, you can leave it as an empty string (""), as shown in your configuration.

What to do with `alertmanager_url`?

Scenario 1: No Alertmanager (default case)

If you're not setting up an Alertmanager, leave this field as "".

- Loki won't send alerts, but Grafana can still display logs.

Scenario 2: Using an Alertmanager

If you have Alertmanager running on a specific server (e.g., your PC or another server):

Replace `alertmanager_url: ""` with the **URL of the Alertmanager** in the format:

php

Copy code

`http://<alertmanager-ip>:<alertmanager-port>`

-

Example:

vbnet

Copy code

```
alertmanager_url: "http://192.168.6.108:9093"
```

○

Your Case:

If **Alertmanager is NOT set up**, leave it as:

vbnet

Copy code

```
alertmanager_url: ""
```

•

If you plan to set up Alertmanager on **your PC (108)**:

vbnet

Copy code

```
alertmanager_url: "http://192.168.6.108:9093"
```

•

If you set up Alertmanager on **the database server (208)**:

vbnet

Copy code

```
alertmanager_url: "http://192.168.6.208:9093"
```

•

Conclusion

In your setup (108 for Loki and Grafana, 208 for Promtail):

- You can leave `alertmanager_url: ""` unless you are explicitly setting up Alertmanager.

Youtube_videosn postgresql_expoter

This tutorial helps you to monitor "POSTGRES" DB with "Prometheus & Grafana".

Follow the below steps:

```
sudo cp postgres_exporter /usr/local/bin
```

```
cd /opt/postgres_exporter
```

```
sudo nano postgres_exporter.env
```

```
DATA_SOURCE_NAME="postgresql://postgres:postgres@192.168.56.106:5432/?sslmode=disable"
```

```
sudo useradd -rs /bin/false postgres
```

```
vi /etc/systemd/system/postgres_exporter.service
```

```
[Unit]
```

```
Description=Prometheus exporter for Postgresql
```

```
Wants=network-online.target
```

```
After=network-online.target
```

```
[Service]
```

```
User=postgres
```

```
Group=postgres
```

```
WorkingDirectory=/opt/postgres_exporter
```

```
EnvironmentFile=/opt/postgres_exporter/postgres_exporter.env
```

```
ExecStart=/usr/local/bin/postgres_exporter --web.listen-address=192.168.56.106:9100
```

```
--web.telemetry-path=/metrics
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl start postgres_exporter
```

```
sudo systemctl enable postgres_exporter
```

```
sudo systemctl status postgres_exporter
```

Steps to perform on Prometheus Server

```
systemctl stop prometheus
```

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

```
systemctl start prometheus
```



```
systemctl status prometheus
```

—

1} <https://www.fosstechnix.com/forward-logs-to-grafana-loki-using-promtail/>

2} https://youtu.be/5D0gPJL_K2M?si=26ZoxfomzNwsgqVx