using the both in prometheu in node_exporter and postgresql_exporter is right or wrong

ChatGPT said:
ChatGPT

Using both **`node_exporter`** and **`postgresql_exporter`** in Prometheus is absolutely correct and is a common practice. Here's why:

1. **`node_exporter`**:
   - Collects metrics about the system, such as CPU usage, memory usage, disk I/O, network statistics, etc.
   - Provides an overview of the health and performance of the underlying host.
2. **`postgresql_exporter`**:
   - Collects database-specific metrics, such as query performance, connection counts, buffer usage, and database size.
   - Focuses on the health and performance of the PostgreSQL database.

Why Use Both?

- They serve **different purposes**:
  - `node_exporter` monitors the server/host level metrics.
  - `postgresql_exporter` monitors PostgreSQL-specific metrics.
- Combining them provides **comprehensive monitoring** for both the hardware and the database application.

So, it is not only **right** but also a **best practice** to use them together for holistic monitoring.

4o

You said:
then you creating me confirgation file in prometheus.yml

ChatGPT said:
ChatGPT

Here's a basic configuration file for `prometheus.yml` that includes both **`node_exporter`** and **`postgresql_exporter`** targets:

yaml
Copy code
```yaml
global:
  scrape_interval: 15s # How often to scrape targets by default.
  evaluation_interval: 15s # How often to evaluate rules by default.
```

```
scrape_configs:
  # Scrape configuration for Node Exporter
  - job_name: 'node_exporter'
    static_configs:
      - targets:
          - 'localhost:9100' # Replace with your Node Exporter target

  # Scrape configuration for PostgreSQL Exporter
  - job_name: 'postgresql_exporter'
    static_configs:
      - targets:
          - 'localhost:9187' # Replace with your PostgreSQL Exporter
target
```

Steps to Use:

1. **Adjust Targets**:
    - Replace `localhost` with the IP address or hostname of the servers where your exporters are running.
    - Ensure the ports (`9100` for `node_exporter` and `9187` for `postgresql_exporter`) match the actual configuration.
2. **Restart Prometheus**:
    - Save the file as `prometheus.yml`.

Restart Prometheus to apply the changes:
bash
Copy code

```
systemctl restart prometheus
```

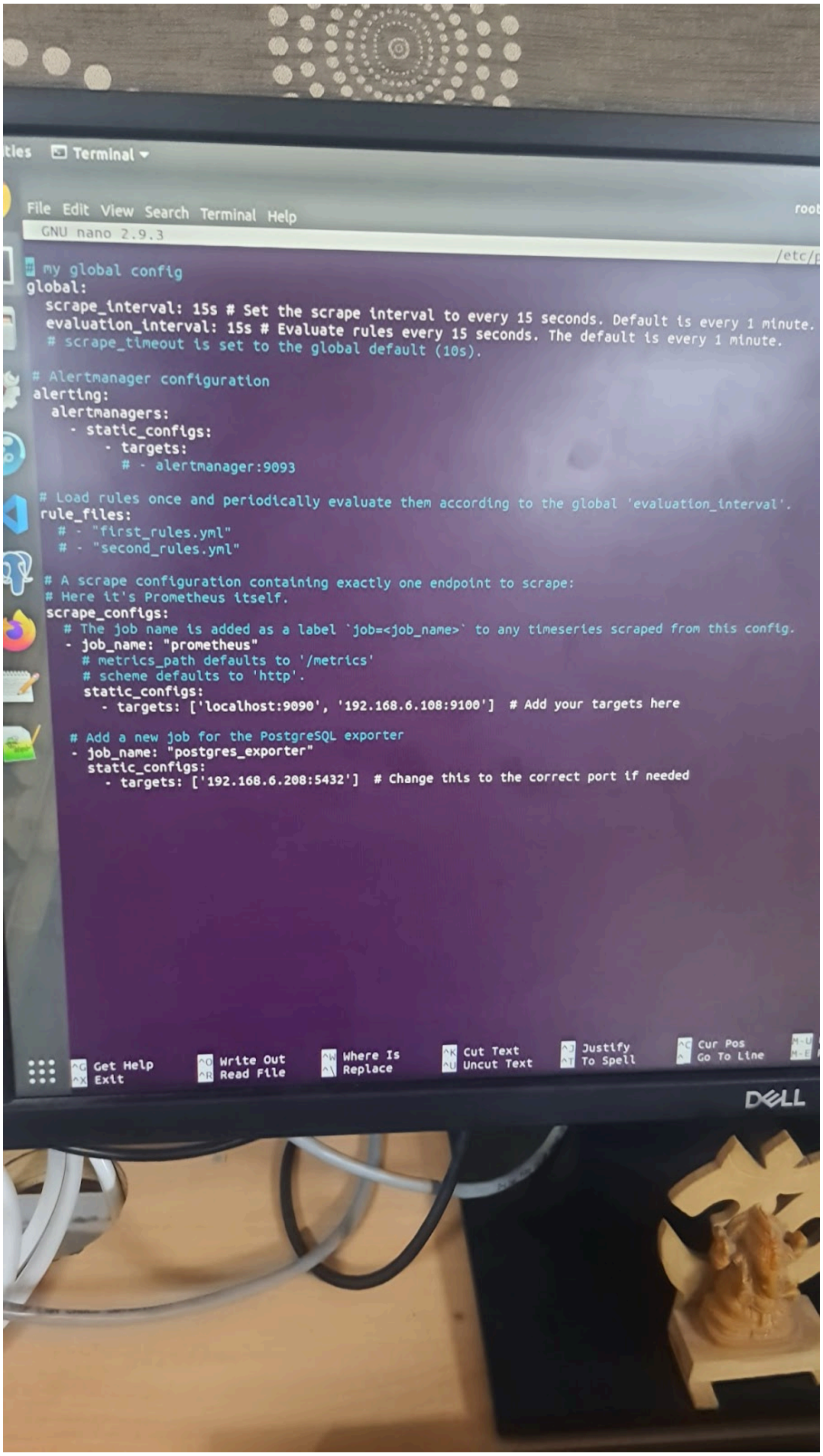
    ○

3. **Verify Targets**:
    - Access Prometheus' web UI (usually at `http://localhost:9090`).
    - Go to **Status > Targets** to confirm both exporters are being scraped successfully.

Let me know if you need further customization!

4o

You said:



```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['localhost:9090', '192.168.6.108:9100']  # Add your targets here

  # Add a new job for the PostgreSQL exporter
  - job_name: "postgres_exporter"
    static_configs:
      - targets: ['192.168.6.208:5432']  # Change this to the correct port if needed
```

I am using this configration but not Runing in postgresql IP 193.168.6.208 5432 creating more problem creating good prometheus.yml file

ChatGPT said:

ChatGPT

From your shared configuration, it seems the `postgresql_exporter` configuration may have issues in targeting the correct IP and port.

Fixed Configuration for Prometheus

Here is a corrected and optimized `prometheus.yml` configuration for your setup:

yaml
Copy code

```
global:
  scrape_interval: 15s # Default scrape interval.
  evaluation_interval: 15s # Default evaluation interval.

alerting:
  alertmanagers:
    - static_configs:
        - targets: ['alertmanager:9093'] # Update with your
Alertmanager's address if needed.

scrape_configs:
  # Scrape Prometheus itself
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  # Scrape Node Exporter
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['192.168.6.108:9100'] # Replace with your Node
Exporter IP and port.

  # Scrape PostgreSQL Exporter
  - job_name: 'postgresql_exporter'
    static_configs:
      - targets: ['192.168.6.208:9187'] # PostgreSQL Exporter's IP and
default port.
```

Key Fixes:

1. **Correct Target Port for PostgreSQL Exporter**:
    - The `postgresql_exporter` listens on port `9187` by default, not `5432` (which is used by PostgreSQL itself). Ensure you configure the PostgreSQL exporter to connect to the database and expose metrics on `9187`.
2. **Exporter Configuration**:
    - Verify that your `postgresql_exporter` is running and configured correctly to connect to PostgreSQL on `192.168.6.208:5432`.

Example command to start the exporter:
bash
Copy code

```
DATA_SOURCE_NAME="postgresql://user:password@192.168.6.208:5432/dbname
" ./postgres_exporter
```

    ○

3. **Firewall and Connectivity**:
    - Ensure port `9187` on the server running the PostgreSQL exporter is open and accessible to Prometheus.
4. **Verify Services**:

Confirm that both Node Exporter (`192.168.6.108:9100`) and PostgreSQL Exporter (`192.168.6.208:9187`) are reachable:
bash
Copy code

```
curl http://192.168.6.108:9100/metrics
curl http://192.168.6.208:9187/metrics
```

    ○

Apply and Restart Prometheus:

1. Save the configuration as `/etc/prometheus/prometheus.yml`.

Restart Prometheus:
bash
Copy code

```
systemctl restart prometheus
```

2.

Check logs for errors:
bash

Copy code

```
journalctl -u prometheus
```

3.

Let me know if this resolves your issue or if you need additional help!

4o

You said:

the prometheu , localhost:9090' , 192.168.6.108:9100 this two ip is run beasuase in this is loccalhost but this is 192.168.6.208:9187 in live postgresql in database is not up in prometheus in target what is problme

ChatGPT said:

ChatGPT

The problem appears to be with the configuration of the PostgreSQL Exporter (`192.168.6.208:9187`) or its connectivity to Prometheus. Let's troubleshoot and resolve the issue systematically.

Steps to Diagnose and Fix the Problem

1. Verify PostgreSQL Exporter is Running

- On the server `192.168.6.208`, ensure the PostgreSQL Exporter is running.

Use the following command to check if the process is active:
bash
Copy code

```
ps aux | grep postgres_exporter
```

-

If not running, start the exporter with the correct connection string:
bash
Copy code

```
DATA_SOURCE_NAME="postgresql://user:password@localhost:5432/dbname"
./postgres_exporter
```

- Replace `user`, `password`, and `dbname` with your actual PostgreSQL credentials.

---

2. Verify Exporter Connectivity

From the Prometheus server, test if you can reach the PostgreSQL Exporter's `/metrics` endpoint:

bash

Copy code

```
curl http://192.168.6.208:9187/metrics
```

-
- If this fails:

Firewall Issue: Ensure port `9187` is open on `192.168.6.208`.

bash

Copy code

```
sudo ufw allow 9187
```

    ○

Network Issue: Check connectivity between Prometheus and the PostgreSQL Exporter using `ping` or `telnet`:

bash

Copy code

```
ping 192.168.6.208
telnet 192.168.6.208 9187
```

    ○

---

3. Check PostgreSQL Exporter Logs

On the PostgreSQL Exporter server (`192.168.6.208`), review logs to ensure it can connect to the PostgreSQL database.

bash

Copy code

```
./postgres_exporter --log.level=debug
```

-
- Common issues include:
    - **Authentication Issues**: Ensure the PostgreSQL user and password are correct.
    - **Database Connection Issues**: Ensure PostgreSQL is running and accessible at `localhost:5432`.

---

4. Correct Prometheus Configuration

Update the `prometheus.yml` file to correctly target the PostgreSQL Exporter:
yaml
Copy code

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

-

Save the file and restart Prometheus:
bash
Copy code

```
systemctl restart prometheus
```

-

---

5. Check Prometheus Targets

- Open the Prometheus web UI at `http://localhost:9090`.
- Go to **Status > Targets** and check the status of the `postgresql_exporter` target:
  - **Up**: The target is being scraped successfully.
  - **Down**: Check the error message in the "Last Error" column for more details.

---

6. Common Problems and Fixes

- **Wrong Port in Exporter**:
  - The default port for `postgresql_exporter` is `9187`. Ensure you configured it to use this port.
- **Wrong DATA_SOURCE_NAME**:

Use the correct PostgreSQL connection string in `DATA_SOURCE_NAME`.
bash
Copy code

```
export
DATA_SOURCE_NAME="postgresql://username:password@192.168.6.208:5432/db
name"
```

  -
- **Exporter Firewall**:

Allow traffic on port `9187` on the server running the PostgreSQL Exporter:
bash
Copy code

```
sudo iptables -A INPUT -p tcp --dport 9187 -j ACCEPT
```

- ○

Summary

The likely issue is with:

1. The PostgreSQL Exporter not running or misconfigured.
2. Firewall or connectivity issues between Prometheus and the Exporter.

If you complete the above steps and still face issues, let me know the exact error from Prometheus **Targets** or the Exporter logs for further assistance.

4o

You said:

this commend curl localhost:9187 is using in terminal showing in postgresql_exporter data in terminal but not up in target any time down then i am add promethues in data sorece how to solving this problem

ChatGPT said:

ChatGPT

If the command `curl localhost:9187` is successfully showing the metrics from `postgresql_exporter`, but Prometheus is still marking the target as **DOWN**, it usually indicates a configuration or network issue. Let's resolve it step-by-step:

Steps to Solve the Issue

1. Verify Prometheus Configuration

Check your `prometheus.yml` file and ensure it correctly targets the PostgreSQL Exporter:
yaml
Copy code

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

-
- Common mistakes to avoid:
    - Ensure the IP (`192.168.6.208`) and port (`9187`) are correct.
    - There should be no trailing slashes (`/`) in the target.

---

2. Restart Prometheus

After updating the configuration file, restart Prometheus:

bash

Copy code

```
systemctl restart prometheus
```

-

Check Prometheus logs to see if it reports any issues:

bash

Copy code

```
journalctl -u prometheus
```

-

---

3. Verify Connectivity from Prometheus to Exporter

On the Prometheus server, run the following command:

bash

Copy code

```
curl http://192.168.6.208:9187/metrics
```

-
- If this fails:

Firewall Issue: Open port `9187` on the PostgreSQL Exporter server (`192.168.6.208`):

bash

Copy code

```
sudo ufw allow 9187
```

    -

Network Issue: Ensure Prometheus can reach the target:

bash

Copy code

```
ping 192.168.6.208
telnet 192.168.6.208 9187
```

- ○

---

4. Verify Target in Prometheus UI

- Open the Prometheus web interface at `http://localhost:9090`.
- Go to **Status > Targets**.
  - If the target is **DOWN**, click on the target to view the error message.

---

5. Check PostgreSQL Exporter Logs

On the PostgreSQL Exporter server, run the exporter in debug mode:
bash
Copy code

```
./postgres_exporter --log.level=debug
```

-
- Look for any errors indicating issues with the PostgreSQL connection or the metrics endpoint.

---

6. Ensure Correct `DATA_SOURCE_NAME`

Confirm the `DATA_SOURCE_NAME` environment variable is correctly set for the PostgreSQL Exporter:
bash
Copy code

```
export
DATA_SOURCE_NAME="postgresql://username:password@localhost:5432/dbname
"
```

-
- Replace `username`, `password`, and `dbname` with valid credentials.
- Restart the PostgreSQL Exporter with this command.

---

7. Add Prometheus as a Data Source

- If you're using Grafana or another monitoring tool:
  - In Grafana, add Prometheus as a data source:
    1. Go to **Configuration > Data Sources**.
    2. Click **Add data source**.

3. Select **Prometheus**.
4. Set the URL to `http://localhost:9090` or your Prometheus server's address.
5. Save and Test.

Common Problems and Fixes

1. **Exporter is Listening Locally**:

If the exporter is only listening on `localhost`, Prometheus on another machine cannot access it. Fix this by starting the exporter to listen on all interfaces:
bash
Copy code

```
./postgres_exporter --web.listen-address="0.0.0.0:9187"
```

   ○

2. **Exporter Data but Wrong Port in Prometheus**:
   - Ensure Prometheus is configured to scrape `9187`, not `5432` (PostgreSQL's default port).
3. **TLS/HTTP Protocol Mismatch**:
   - If your exporter uses HTTPS, ensure Prometheus is configured to use `https://` in the target.

Summary

- Ensure the exporter is reachable on `192.168.6.208:9187` from the Prometheus server.
- Update `prometheus.yml` to target the correct IP and port.
- Restart Prometheus and check its logs for errors.
- Verify the target's status in the Prometheus UI.

Let me know what you observe in the **Targets** page or Prometheus logs, and I can guide further!

4o

You said:

without prometheus i am add loki in loges in postgresql data base then i am add data sourece in loki in grafana

ChatGPT said:

ChatGPT

To directly use **Loki** for logging PostgreSQL logs and visualizing them in **Grafana**, follow these steps:

Steps to Set Up Loki for PostgreSQL Logs

1. Install and Configure Loki

Install Loki on your server using Docker, system packages, or binaries. For Docker:
bash
Copy code

```
docker run -d --name loki -p 3100:3100 grafana/loki:latest
```

-

Create a configuration file (`loki-config.yml`) if needed:
yaml
Copy code

```
server:
  http_listen_port: 3100

positions:
  filename: /tmp/positions.yaml

scrape_configs:
  - job_name: 'postgresql_logs'
    static_configs:
      - targets:
          - localhost
        labels:
          job: postgresql_logs
          __path__: /path/to/postgresql/logs/*.log
```

-
- Replace `/path/to/postgresql/logs/*.log` with the path to your PostgreSQL log files.

---

2. Enable PostgreSQL Logging

Modify `postgresql.conf` to enable logging:
conf
Copy code
```
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql.log'
log_statement = 'all'
log_min_duration_statement = 0
```

-

Restart PostgreSQL to apply changes:
bash
Copy code
```
systemctl restart postgresql
```

-

---

3. Install Promtail

- Install Promtail to push logs from the PostgreSQL log files to Loki.

Example Docker command:
bash
Copy code
```
docker run -d --name promtail -v /path/to/config:/etc/promtail -v
/path/to/logs:/var/log/postgresql grafana/promtail:latest
```

-

Promtail configuration example (`promtail-config.yml`):
yaml
Copy code
```
server:
  http_listen_port: 9080

clients:
  - url: http://localhost:3100/loki/api/v1/push

positions:
  filename: /tmp/positions.yaml
```

```
scrape_configs:
  - job_name: postgresql_logs
    static_configs:
      - targets:
          - localhost
        labels:
          job: postgresql_logs
          __path__: /var/log/postgresql/postgresql.log
```

-

---

4. Add Loki as a Data Source in Grafana

- Open Grafana and go to **Configuration > Data Sources**.
- Add a new data source:
  - **Type**: Loki.
  - **URL**: `http://<loki_server_ip>:3100`.
  - **Access**: Server (default).
- Click **Save & Test**.

---

5. Create Dashboards in Grafana

- In Grafana, create a new panel:
  - Select the **Loki** data source.

Use the following query to filter PostgreSQL logs:
loki
Copy code

```
{job="postgresql_logs"}
```

  -
- Customize the query further to filter by log levels or specific messages.

Troubleshooting Tips

1. **Logs Not Visible in Loki**:

    - Ensure the paths in `promtail-config.yml` are correct and match the location of PostgreSQL logs.

Check Promtail logs for errors:
bash
Copy code
```
docker logs promtail
```

    -
2. **No Data in Grafana**:

Verify Loki is reachable from Grafana using `curl`:
bash
Copy code
```
curl http://<loki_server_ip>:3100/ready
```

    -
    - Ensure the data source is correctly configured in Grafana.
3. **Debug Promtail Logs**:
    - Add the `--log.level=debug` flag when starting Promtail to debug issues.

Summary

1. Set up Loki to receive logs.
2. Enable PostgreSQL logging and configure Promtail to push logs to Loki.
3. Add Loki as a data source in Grafana and query logs using `{job="postgresql_logs"}`.

Let me know if you face any specific issue during this setup!

4o

You said:
i am using in live database ip is 192.168.6.208:5432 then you creating me this ip in loki and promtail proper setup

ChatGPT said:
ChatGPT

To configure **Loki** and **Promtail** to collect PostgreSQL logs from a live database server with the IP `192.168.6.208:5432`, follow these steps:

---

1. Enable PostgreSQL Logging on the Live Server

You need to ensure the PostgreSQL logs are being written to files on the live database server (`192.168.6.208`).

SSH into the live database server:
bash
Copy code

```
ssh user@192.168.6.208
```

1.

Edit the `postgresql.conf` file to enable logging:
bash
Copy code

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

Make the following changes:
conf
Copy code

```
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql.log'
log_statement = 'all'
log_min_duration_statement = 0
```

2.

Restart PostgreSQL to apply changes:
bash
Copy code

```
sudo systemctl restart postgresql
```

3. Logs will now be stored in `/var/log/postgresql/postgresql.log`.

---

2. Install and Configure Loki

Loki can run on the same machine as Grafana or any other machine. For this example, we assume Loki is running locally.

Run Loki using Docker (on your monitoring server):
bash

Copy code

```
docker run -d --name loki -p 3100:3100 grafana/loki:latest
```

1.

Ensure Loki is running:
bash
Copy code

```
curl http://localhost:3100/ready
```

2.

---

3. Install and Configure Promtail

Promtail will read PostgreSQL logs from the live database server and send them to Loki.

3.1 Install Promtail on the Monitoring Server

Use Docker to run Promtail:

bash
Copy code

```
docker run -d --name promtail -v /path/to/config:/etc/promtail -v
/var/log/postgresql:/var/log/postgresql grafana/promtail:latest
```

3.2 Configure Promtail

Create a Promtail configuration file (`promtail-config.yml`):

yaml
Copy code

```
server:
  http_listen_port: 9080

clients:
  - url: http://localhost:3100/loki/api/v1/push

positions:
  filename: /tmp/positions.yaml

scrape_configs:
```

```
- job_name: postgresql_logs
  static_configs:
    - targets:
        - 192.168.6.208  # IP of your PostgreSQL server
      labels:
        job: postgresql_logs
        __path__: /var/log/postgresql/postgresql.log
```

Replace `/var/log/postgresql/postgresql.log` with the actual path to the log file on the live database server.

---

4. Add Loki as a Data Source in Grafana

1. Go to **Configuration > Data Sources** in Grafana.
2. Add a new data source:
    - **Type**: Loki.
    - **URL**: `http://<loki_server_ip>:3100` (e.g., `http://localhost:3100` if Loki runs locally).
    - Click **Save & Test**.

---

5. Create a Dashboard in Grafana

1. In Grafana, create a new panel.
2. Select the Loki data source.

Use the following query to view logs:
loki
Copy code

```
{job="postgresql_logs"}
```

3.
4. Apply filters to display only the logs you are interested in.

Troubleshooting

1. **Promtail Not Sending Logs**:

Check Promtail logs:
bash
Copy code

```
docker logs promtail
```

- ○
- ○ Ensure the log path in `promtail-config.yml` is correct.

2. **No Logs in Grafana**:

Verify Loki is running and accessible:
bash
Copy code

```
curl http://localhost:3100/ready
```

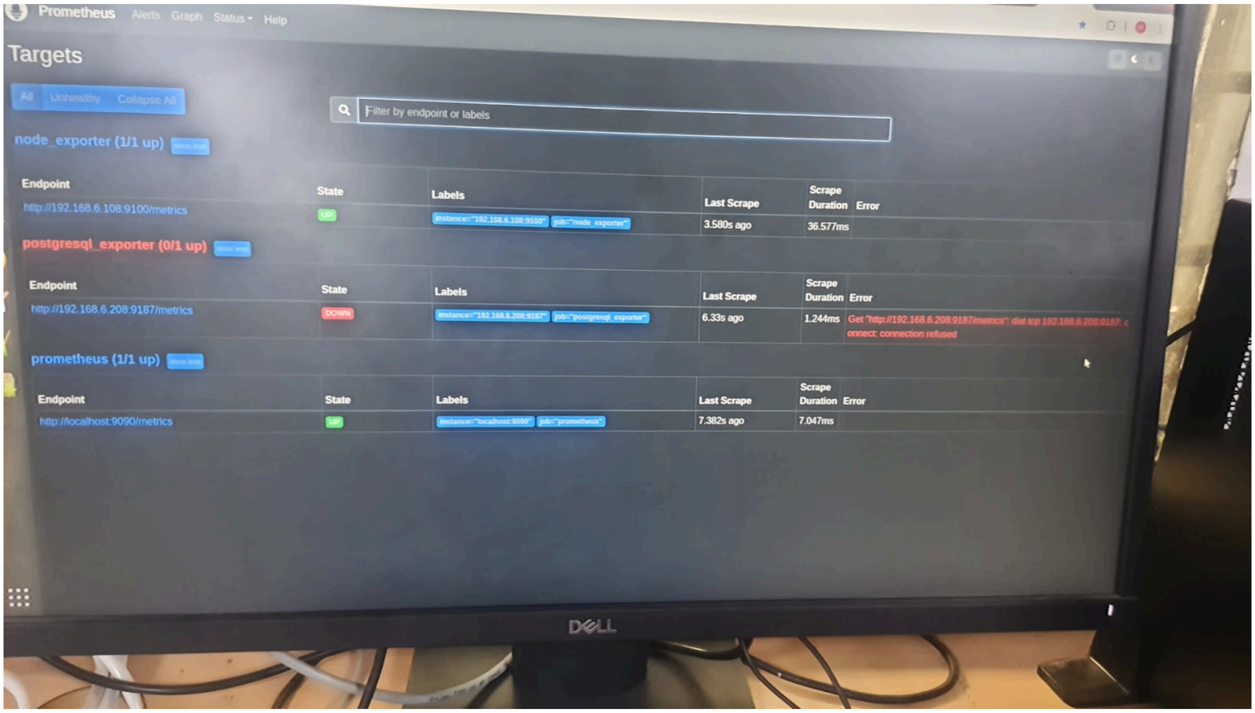
- ○

3. **Database Logs Not Appearing**:
   - Ensure PostgreSQL is logging correctly to the file path `/var/log/postgresql/postgresql.log`.
   - Check file permissions.

Let me know if you encounter issues during the setup!

4o

You said:



Targets

node_exporter (1/1 up)

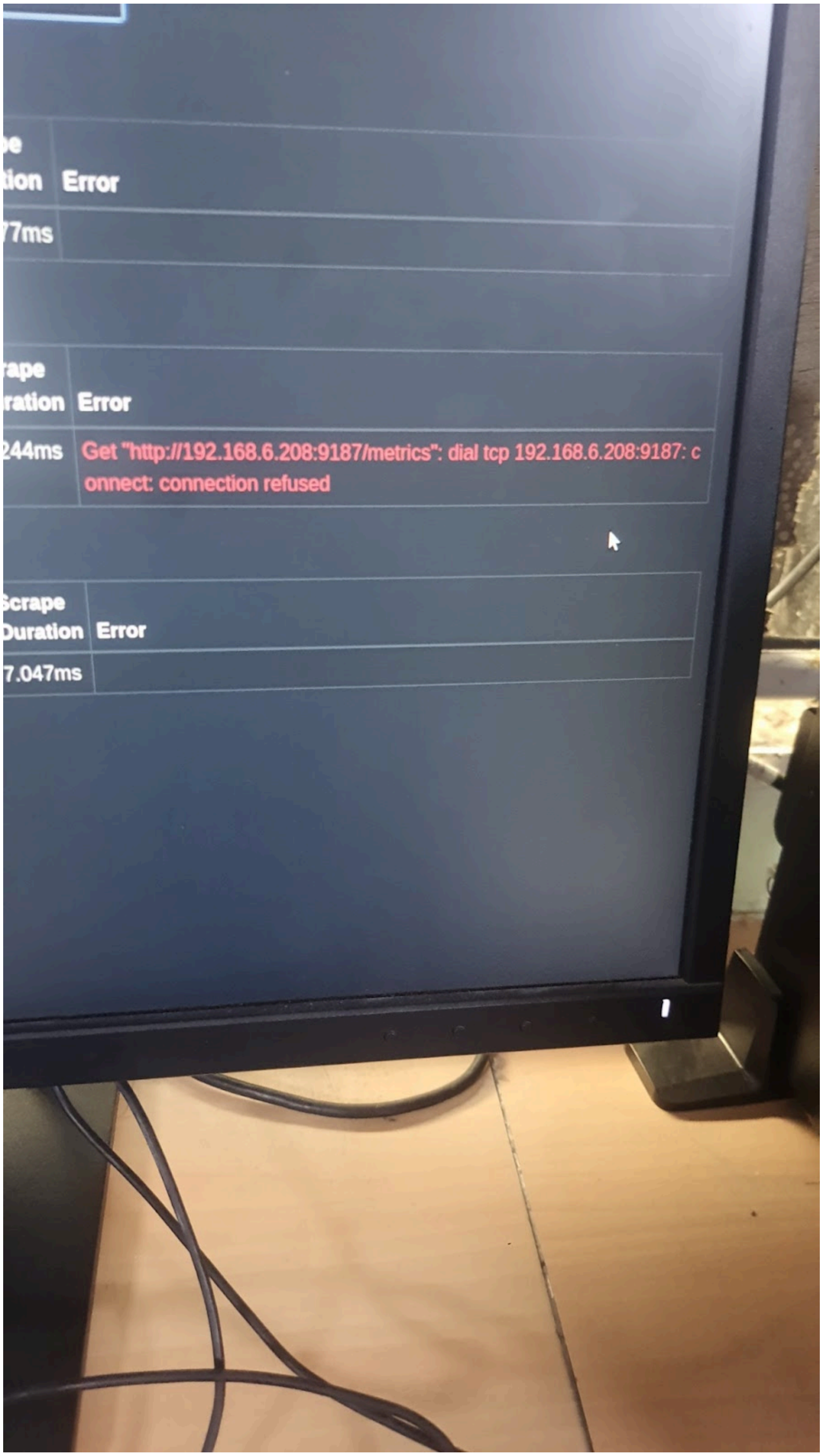
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.6.108:9100/metrics	UP	instance="192.168.6.108:9100" job="node_exporter"	3.580s ago	36.577ms	

postgresql_exporter (0/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.6.208:9187/metrics	DOWN	instance="192.168.6.208:9187" job="postgresql_exporter"	6.33s ago	1.244ms	Get "http://192.168.6.208:9187/metrics": dial tcp 192.168.6.208:9187: connect: connection refused

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	7.382s ago	7.047ms	



e tion	Error
7ms	

rape ration	Error
244ms	Get "http://192.168.6.208:9187/metrics": dial tcp 192.168.6.208:9187: connect: connection refused

crape Duration	Error
7.047ms	

this is error problem is down this IP 192.168.6.208 9187

ChatGPT said:

ChatGPT

The error message **"connect: connection refused"** indicates that Prometheus is unable to connect to the `postgresql_exporter` on IP `192.168.6.208` and port `9187`. This could be due to several reasons. Let's troubleshoot:

---

1. Check if `postgresql_exporter` is running

- On the server `192.168.6.208`, verify that `postgresql_exporter` is running and listening on port `9187`.

Use the following command:
bash
Copy code

```
netstat -tuln | grep 9187
```

-
- If no process is listening, restart the `postgresql_exporter` service.

---

2. Verify Firewall Rules

- Ensure the firewall on `192.168.6.208` allows traffic on port `9187`.

If using `iptables` or `firewalld`, add a rule to allow traffic:
bash
Copy code

```
sudo firewall-cmd --add-port=9187/tcp --permanent
sudo firewall-cmd --reload
```

-

Alternatively, disable the firewall temporarily for testing:
bash
Copy code

```
sudo systemctl stop firewalld
```

-

---

3. Check PostgreSQL Exporter Configuration

- Ensure `postgresql_exporter` is configured correctly to connect to your database.

The `DATA_SOURCE_NAME` environment variable must have the correct database connection string:
php
Copy code

```
postgres://<user>:<password>@192.168.6.208:5432/<database>?sslmode=disable
```

-
- Restart the exporter with the correct configuration.

---

4. Check Network Connectivity

From the Prometheus server, test if you can reach `192.168.6.208:9187`:
bash
Copy code

```
curl http://192.168.6.208:9187/metrics
```

-
- If the connection fails, check:
  - Network routes between Prometheus and `192.168.6.208`.
  - Any intermediate firewalls or security groups.

---

5. Prometheus Configuration

Ensure the Prometheus configuration (`prometheus.yml`) has the correct target for `postgresql_exporter`:
yaml
Copy code

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

-

Reload Prometheus after updating the configuration:
bash

Copy code

```
curl -X POST http://localhost:9090/-/reload
```

-

---

6. Check Logs

Review the logs of `postgresql_exporter` to identify any errors:

bash

Copy code

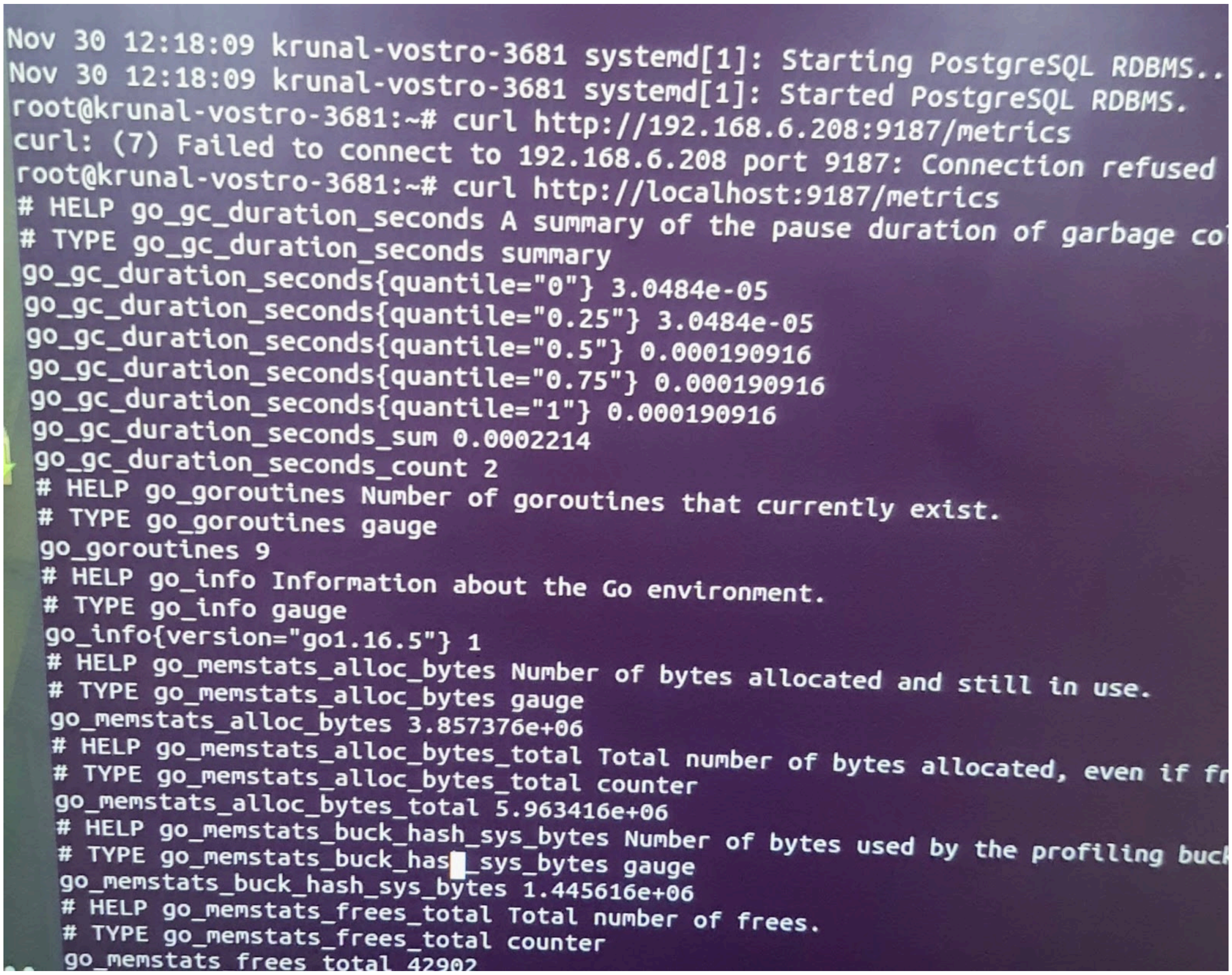
```
journalctl -u postgresql_exporter
```

-
- Look for issues such as database connection failures or misconfiguration.

After following these steps, the target in Prometheus should show **UP**. If the problem persists, let me know where you get stuck.

4o

You said:



```
Nov 30 12:18:09 krunal-vostro-3681 systemd[1]: Starting PostgreSQL RDBMS..
Nov 30 12:18:09 krunal-vostro-3681 systemd[1]: Started PostgreSQL RDBMS.
root@krunal-vostro-3681:~# curl http://192.168.6.208:9187/metrics
curl: (7) Failed to connect to 192.168.6.208 port 9187: Connection refused
root@krunal-vostro-3681:~# curl http://localhost:9187/metrics
# HELP go_gc_duration_seconds A summary of the pause duration of garbage co
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.0484e-05
go_gc_duration_seconds{quantile="0.25"} 3.0484e-05
go_gc_duration_seconds{quantile="0.5"} 0.000190916
go_gc_duration_seconds{quantile="0.75"} 0.000190916
go_gc_duration_seconds{quantile="1"} 0.000190916
go_gc_duration_seconds_sum 0.0002214
go_gc_duration_seconds_count 2
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.16.5"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 3.857376e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if fr
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 5.963416e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling buck
# TYPE go_memstats_buck_has█_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.445616e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 42902
```

this is localhost/9187 is run in terminal but failed in 192.168.6.208 is not run what problem how to solving this problem

ChatGPT said:

ChatGPT

The issue is that the `postgresql_exporter` is successfully running and accessible on `localhost:9187` but is not accessible from the external network on `192.168.6.208:9187`. This is likely due to one of the following reasons:

---

1. Exporter Binding to localhost

- By default, the `postgresql_exporter` might be binding only to `127.0.0.1` (localhost), preventing external access.
- **Solution**: Check and modify the `postgresql_exporter` configuration to bind to all network interfaces (`0.0.0.0`) or the specific IP `192.168.6.208`.

If running as a service, check the service file:
bash
Copy code

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

Ensure the `ExecStart` line specifies:
bash
Copy code

```
ExecStart=/path/to/postgresql_exporter
--web.listen-address="0.0.0.0:9187"
```

    ○

Reload the service after modification:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgresql_exporter
```

    ○

---

2. Firewall or Security Rules

- A firewall on the machine or in the network may block access to port `9187`.
- **Solution**:

Check and open port `9187` in the firewall:
bash
Copy code

```
sudo firewall-cmd --add-port=9187/tcp --permanent
sudo firewall-cmd --reload
```

    ○

Test access from another machine:
bash
Copy code

```
curl http://192.168.6.208:9187/metrics
```

    ○

---

3. Network Interface Configuration

- The server might not be configured to allow connections on the `192.168.6.208` interface.
- **Solution**:

Check the network interfaces on the machine:
bash
Copy code

```
ip addr
```

  -
  - Ensure the IP `192.168.6.208` is active and properly assigned.

---

4. SELinux (If Enabled)

- If SELinux is enabled, it might block the `postgresql_exporter` from binding to external IPs.
- **Solution**:

Check SELinux status:
bash
Copy code

```
sestatus
```

  -

Temporarily disable SELinux for testing:
bash
Copy code

```
sudo setenforce 0
```

  -
  - If this resolves the issue, create a policy to allow the `postgresql_exporter` to bind externally.

---

5. Prometheus Access

- If you are trying to access the exporter from Prometheus, ensure Prometheus can reach `192.168.6.208:9187` over the network.

Test the connectivity from the Prometheus server:

bash

Copy code

```
curl http://192.168.6.208:9187/metrics
```

-

Follow these steps, and the issue should be resolved. Let me know if you encounter any challenges!

4o

You said:

this is error creating this IP 192.168.6.208 9187. Navigation Menu Code Issues Pull requests Breadcrumbsgrafan_pro /README.md Latest commit Gauravgp77 Gauravgp77 1 minute ago History 86 lines (72 loc) · 7.19 KB File metadata and controls Preview Code Blame global: scrape_interval: 15s # Default scrape interval. evaluation_interval: 15s # Default evaluation interval. alerting: alertmanagers: - static_configs: - targets: ['alertmanager:9093'] # Update with your Alertmanager's address if needed. scrape_configs: Scrape Prometheus itself job_name: 'prometheus' static_configs: targets: ['localhost:9090'] Scrape Node Exporter job_name: 'node_exporter' static_configs: targets: ['192.168.6.108:9100'] # Replace with your Node Exporter IP and port. oot@krunal-vostro-3681:~# journalctl -u postgres_exporter.service -- Logs begin at Thu 2023-03-02 18:28:05 IST, end at Mon 2024-12-02 14:41:34 IST. -- Nov 29 17:04:34 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: ts=2024-11-29T11:34:38.557Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: level=info ts=2024-11-29T11:34:38.568Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: ts=2024-11-29T11:46:45.506Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: level=info ts=2024-11-29T11:46:45.531Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: # Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: Change Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: localhost Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the

```
Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
Nov 29 17:35:14 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29
17:35:14 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:35:14
krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:35:14
krunal-vostro-3681 postgres_exporter[12452]: ts=2024-11-29T12:05:14.408Z caller=log.go:168
level=info msg="Listening on address" address=:9187 Nov 29 17:35:14 krunal-vostro-3681
postgres_exporter[12452]: level=info ts=2024-11-29T12:05:14.409Z caller=tls_config.go:191
msg="TLS is disabled." http2=false Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: #
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
Change Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
localhost Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
-- Reboot -- Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov
29 18:11:52 krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.253Z
caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 18:11:52
krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.254Z caller=log.go:168
level=error msg="Error running HTTP server" err="listen tcp :9187: bind: address already in
use" Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process
exited, code=exited, status=1/FAILURE Nov 29 18:11:52 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:11:56 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:11:56 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:11:56 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:16 krunal-vostro-3681
```

```
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.529Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.530Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:16 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:16 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:32 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.188Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.189Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:32 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:32 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. -- Reboot -- Nov 29 18:27:32
krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 18:27:32
krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. lines 1-54
```

ChatGPT said:

ChatGPT

The error message **"listen tcp :9187: bind: address already in use"** indicates that another process is already using port `9187`. Here are the steps to troubleshoot and resolve this issue:

Steps to Resolve:

Identify the Process Using Port 9187: Run the following command to find the process:
bash
Copy code

```bash
sudo lsof -i :9187
```

or
bash
Copy code

```bash
sudo netstat -tuln | grep 9187
```

1.

Stop the Conflicting Process: If another instance of `postgres_exporter` or any other service is using the port, stop it:
bash
Copy code

```bash
sudo kill -9 <PID>
```

2. Replace `<PID>` with the process ID from the above command.

Check `postgres_exporter` Status: Verify that only one instance of `postgres_exporter` is running:
bash
Copy code

```
sudo systemctl status postgres_exporter.service
```

3.

Restart `postgres_exporter` Service: Restart the service to ensure it binds to the desired port:
bash
Copy code

```
sudo systemctl restart postgres_exporter.service
```

4.

Update Service Configuration (If Needed): Check the `postgres_exporter.service` file and ensure the `--web.listen-address` parameter specifies a unique port or correct IP:
bash
Copy code

```
ExecStart=/path/to/postgres_exporter --web.listen-address=:9187
```

If a specific IP is required:
bash
Copy code

```
ExecStart=/path/to/postgres_exporter
--web.listen-address=192.168.6.208:9187
```

Reload the systemd configuration:
bash
Copy code

```
sudo systemctl daemon-reload
```

5.

Check Logs: If the issue persists, check logs for further details:
bash
Copy code

```
journalctl -u postgres_exporter.service
```

6.

Avoid Future Conflicts:

- Use a unique port if multiple exporters are running.
- Ensure no other services are configured to use the same port.

Let me know if you need further assistance!

4o

You said:

this is using in [Unit] Description=PostgreSQL Exporter After=network.target [Service] User =postgres Group=postgres Type=simple Environment="DATA_SOURCE_NAME=postgres://postgres:postgres@192.168.6.208:5432/BaseMap_Multi?sslmode=disable" ExecStart=/usr/local/bin/postgres_exporter --web.listen-address="0.0.0.0:9187" --web.telemetry-path=/metrics [Install] WantedBy=multi-user.target

ChatGPT said:

ChatGPT

The error `listen tcp :9187: bind: address already in use` indicates that another process is already using port `9187` on your system, preventing the PostgreSQL Exporter from binding to that port.

Here's how to resolve the issue:

Steps to Fix

Check What is Using Port 9187 Run the following command to identify which process is using the port:
bash
Copy code
```
sudo netstat -tulnp | grep 9187
```
or
bash
Copy code
```
sudo lsof -i :9187
```

1.

Kill the Conflicting Process If a process is identified as using port `9187`, you can terminate it:
bash
Copy code
```
sudo kill -9 <PID>
```

2. Replace `<PID>` with the process ID from the previous step.

Restart the PostgreSQL Exporter Restart the PostgreSQL Exporter service:
bash

Copy code

```
sudo systemctl restart postgres_exporter.service
```

3.
4. **Change the Port (if Needed)** If port `9187` is required by another application and cannot be freed, change the PostgreSQL Exporter's listening port:

Edit the `ExecStart` line in the service file to use a different port (e.g., `9190`):
bash
Copy code

```
ExecStart=/usr/local/bin/postgres_exporter
--web.listen-address="0.0.0.0:9190" --web.telemetry-path=/metrics
```

○

Reload the systemd configuration and restart the service:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

○

Verify the Status Check the status to ensure the service is running without errors:
bash
Copy code

```
sudo systemctl status postgres_exporter.service
```

5.

Let me know if you encounter any issues!

4o

You said:

```
root@krunal-vostro-3681:# sudo netstat -tuln | grep 9187 tcp6 0 0 :::9187 :::* LISTEN root@krunal-vostro-3681:# ps -l F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD 4 S 0 15931 5811 0 80 0 - 16462 poll_s pts/0 00:00:00 sudo 4 S 0 15932 15931 0 80 0 - 16285 wait pts/0 00:00:00 su 4 S 0 15946 15932 0 80 0 - 5724 wait pts/0 00:00:01 bash 4 R 0 27390 15946 0 80 0 - 7230 - pts/0 00:00:00 ps root@krunal-vostro-3681:# sudo nano /etc/systemd/system/postgres_exporter.service root@krunal-vostro-3681:# sudo lsof -i :9187 COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME postgres_ 26883 postgres 3u IPv6 252903 0t0 TCP *:9187 (LISTEN) root@krunal-vostro-3681:# ps -lh 4 0 4787 1 20 0 16188 1968 poll_s Ss+ tty6 0:00 /sbin/agetty -o -p -- \u --noclear tty6 linux 4 0 15931 5811 20 0 65848 4488 poll_s S pts/0 0:00 sudo su - 4 0 15932 15931 20 0 65140 3840 wait S pts/0 0:00
```

su - 4 0 15946 15932 20 0 22896 5480 wait S pts/0 0:01 -su 4 0 27497 15946 20 0 28920 1560 - R+ pts/0 0:00 ps -lh root@krunal-vostro-3681:#

ChatGPT said:

ChatGPT

From your output, it appears that the process `postgres_` with PID `26883` is already bound to port `9187`. This is why the PostgreSQL Exporter is failing to start, as the port is already in use.

Next Steps:

1. Confirm and Stop the Existing Process

Run the following command to stop the process occupying port `9187`:

bash

Copy code

```
sudo kill -9 26883
```

2. Restart the PostgreSQL Exporter Service

After killing the process, restart the PostgreSQL Exporter service:

bash

Copy code

```
sudo systemctl restart postgres_exporter.service
```

3. Verify the Service Status

Check the status of the PostgreSQL Exporter to ensure it is running:

bash

Copy code

```
sudo systemctl status postgres_exporter.service
```

4. Prevent Future Conflicts

If you suspect another instance of PostgreSQL Exporter or a similar process is running inadvertently:

- Ensure no duplicate service files are present in `/etc/systemd/system/` or `/lib/systemd/system/`.

If the port conflict persists, consider changing the `--web.listen-address` in the service file to use a different port (e.g., `9190`):
bash
Copy code

```
ExecStart=/usr/local/bin/postgres_exporter
--web.listen-address="0.0.0.0:9190" --web.telemetry-path=/metrics
```

Reload systemd and restart:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

-

Let me know if you need further help!

4o

You said:
Again creating thi problem. this is error creating this IP 192.168.6.208 9187. Navigation Menu Code Issues Pull requests Breadcrumbsgrafan_pro /README.md Latest commit Gauravgp77 Gauravgp77 1 minute ago History 86 lines (72 loc) · 7.19 KB File metadata and controls Preview Code Blame global: scrape_interval: 15s # Default scrape interval. evaluation_interval: 15s # Default evaluation interval. alerting: alertmanagers: - static_configs: - targets: ['alertmanager:9093'] # Update with your Alertmanager's address if needed. scrape_configs: Scrape Prometheus itself job_name: 'prometheus' static_configs: targets: ['localhost:9090'] Scrape Node Exporter job_name: 'node_exporter' static_configs: targets: ['192.168.6.108:9100'] # Replace with your Node Exporter IP and port. oot@krunal-vostro-3681:~# journalctl -u postgres_exporter.service -- Logs begin at Thu 2023-03-02 18:28:05 IST, end at Mon 2024-12-02 14:41:34 IST. -- Nov 29 17:04:34 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: ts=2024-11-29T11:34:38.557Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: level=info ts=2024-11-29T11:34:38.568Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: ts=2024-11-29T11:46:45.506Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: level=info ts=2024-11-29T11:46:45.531Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: # Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:

```
Change Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
localhost Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to
Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the
Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:35:08 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
Nov 29 17:35:14 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29
17:35:14 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:35:14
krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:35:14
krunal-vostro-3681 postgres_exporter[12452]: ts=2024-11-29T12:05:14.408Z caller=log.go:168
level=info msg="Listening on address" address=:9187 Nov 29 17:35:14 krunal-vostro-3681
postgres_exporter[12452]: level=info ts=2024-11-29T12:05:14.409Z caller=tls_config.go:191
msg="TLS is disabled." http2=false Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: #
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
Change Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
localhost Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
-- Reboot -- Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov
29 18:11:52 krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.253Z
caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 18:11:52
krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.254Z caller=log.go:168
level=error msg="Error running HTTP server" err="listen tcp :9187: bind: address already in
use" Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process
exited, code=exited, status=1/FAILURE Nov 29 18:11:52 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:11:56 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=info
```

```
msg="Listening on address" address=:9187 Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:11:56 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:11:56 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:16 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.529Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.530Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:16 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:16 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:32 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.188Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.189Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:32 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:32 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. -- Reboot -- Nov 29 18:27:32
krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 18:27:32
krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. lines 1-54
```

ChatGPT said:

ChatGPT

The error `bind: address already in use` indicates that the port `9187` is already being used by another process. This is causing the PostgreSQL Exporter to fail when attempting to start. Here's how you can resolve the issue:

Steps to Fix the Issue

Identify the Process Using Port 9187: Run the following command to see which process is using the port:
bash
Copy code

```
sudo lsof -i :9187
```

1. This will show you the `PID` of the process.

Stop the Conflicting Process: If you confirm the process using the port is unnecessary or conflicting, stop it using:
bash

Copy code

```
sudo kill -9 <PID>
```

2. Replace `<PID>` with the actual `PID` of the process.

Modify the Port: If you cannot stop the conflicting process, you can configure the PostgreSQL Exporter to listen on a different port. Update the `postgres_exporter.service` file:
bash
Copy code

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

Add or modify the `--web.listen-address` argument to use a different port:
css
Copy code

```
ExecStart=/path/to/postgres_exporter --web.listen-address=:NEW_PORT
```

3. Replace `NEW_PORT` with an available port.

Reload the Service: After making changes to the `.service` file, reload and restart the service:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

4.

Verify: Check if the service is running without errors:
bash
Copy code

```
sudo systemctl status postgres_exporter.service
```

5.

Test Access: Ensure you can access the exporter on the new port or the resolved configuration:
bash
Copy code

```
curl http://<IP>:<PORT>/metrics
```

6.

If the issue persists, check for typos or misconfigurations in the `.service` file or related Prometheus configurations. Let me know if you need further clarification!

4o

You said:

```
Nov 29 17:35:14 krunal-vostro-3681 postgres_exporter[12452]: level=info
ts=2024-11-29T12:05:14.409Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov
29 17:59:17 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9:
Invalid environment assignment, ignoring: # Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
Change Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
localhost Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
-- Reboot -
```

ChatGPT said:

ChatGPT

The logs you provided indicate two main issues with the `postgres_exporter.service`:

1. **"Invalid environment assignment" Error**:
   This error suggests there is an issue in your `postgres_exporter.service` file. Specifically, it seems like there are invalid or malformed environment variable assignments in the `[Service]` section. For example, lines may be improperly formatted or include comments that are incorrectly parsed. Ensure every `Environment=` line is properly formatted without extra characters or comments.
   Steps to fix:

Edit the service file:
bash
Copy code

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

○

Review the `Environment=` lines. For example:
ini
Copy code

```
Environment="DATA_SOURCE_NAME=postgresql://user:password@localhost:5432/dbname"
```

- Ensure there are no stray comments or invalid syntax. Comments should start with `#` and be on a separate line.

Save the file and reload the systemd daemon:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

○

2. **"Address already in use" Error**:
   This error occurs when another process is already using port `9187`. You need to identify the process using the port and either stop it or configure `postgres_exporter` to use a different port.

4o

You said:
I am using in. 9190 Also showing this problme

ChatGPT said:
ChatGPT

If you're using port `9190` for `postgres_exporter` and still seeing the error, follow these steps to resolve it:

1. Verify the Port is Free

Check if port `9190` is already in use by another process:

bash
Copy code

```
sudo netstat -tuln | grep 9190
```

If this command shows output, another service is occupying the port. To identify the process:

bash
Copy code

```
sudo lsof -i :9190
```

You can stop the conflicting process if necessary:

bash
Copy code
```
sudo kill -9 <PID>
```

2. Check the Service File for Correct Configuration

Edit your `postgres_exporter.service` file:

bash
Copy code
```
sudo nano /etc/systemd/system/postgres_exporter.service
```

Ensure the `web.listen-address` argument specifies port `9190`:

ini
Copy code
```
[Service]
ExecStart=/usr/local/bin/postgres_exporter --web.listen-address=:9190
```

3. Reload and Restart the Service

Reload the systemd daemon to apply changes:

bash
Copy code
```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

Verify the status:

bash
Copy code
```
sudo systemctl status postgres_exporter.service
```

4. Inspect Logs for Further Issues

If the service still fails, inspect the logs for more details:

bash
Copy code

```
sudo journalctl -u postgres_exporter.service
```

5. Test Access

Open your browser or use `curl` to check if the exporter is running on port `9190`:

bash
Copy code

```
curl http://localhost:9190/metrics
```

If you still face issues, share the relevant logs from `journalctl` for further analysis.

4o

You said:
curl http://localhost:9190/metrics this is run in local in terminal but 192.168.6.208:9187 is not run in local what is problem

ChatGPT said:
ChatGPT

If `curl http://localhost:9190/metrics` works but accessing `http://192.168.6.208:9190` (not `9187` as mentioned earlier) does not work, it suggests that `postgres_exporter` is only listening on `localhost` (`127.0.0.1`) and not on the external IP address.

Here's how you can resolve the issue:

---

1. Update the `postgres_exporter` Binding Address

Ensure that the exporter is configured to bind to all network interfaces (`0.0.0.0`) instead of just `localhost`. Update the `postgres_exporter.service` file to include:

ini
Copy code

```
[Service]
ExecStart=/usr/local/bin/postgres_exporter --web.listen-address=:9190
```

The `:9190` ensures the exporter listens on all IPs, including `192.168.6.208`.

After making changes:

bash
Copy code
```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

---

2. Verify the Listening Address

Check which address the service is bound to:

bash
Copy code
```
sudo netstat -tuln | grep 9190
```

You should see:

ruby
Copy code
```
tcp   0   0 0.0.0.0:9190   0.0.0.0:*   LISTEN
```

If it says `127.0.0.1:9190`, the service is still only accessible locally.

---

3. Check Firewall Rules

If the service is listening on the correct IP, ensure your firewall allows traffic on port `9190`:

bash
Copy code
```
sudo ufw allow 9190/tcp
sudo ufw reload
```

Alternatively, if using `iptables`:

bash
Copy code

```
sudo iptables -A INPUT -p tcp --dport 9190 -j ACCEPT
```

---

4. Test External Access

From a different machine on the same network, run:

bash
Copy code
```
curl http://192.168.6.208:9190/metrics
```

If it works, your configuration is correct.

---

5. Inspect Logs

If it still doesn't work, check the logs for potential errors:

bash
Copy code
```
sudo journalctl -u postgres_exporter.service
```

If the above steps don't resolve the issue, share the output of the `netstat` and logs for deeper troubleshooting.

4o

You said:
i am using this curl http://192.168.6.208:9190/metrics is not run in how to solving this

ChatGPT said:
ChatGPT

If `curl http://192.168.6.208:9190/metrics` is not working, but `curl http://localhost:9190/metrics` works locally, it indicates that the `postgres_exporter` is not accessible via the network. Here's how to fix it:

Step 1: Verify Postgres Exporter Binding

Ensure that the `postgres_exporter` is configured to listen on all network interfaces (`0.0.0.0`) instead of just `localhost`.

Open the `postgres_exporter.service` file:
bash
Copy code
```
sudo nano /etc/systemd/system/postgres_exporter.service
```

1.

Check the `ExecStart` line. It should include `--web.listen-address=:9190`:
ini
Copy code
```
ExecStart=/usr/local/bin/postgres_exporter --web.listen-address=:9190
```

2.
    - `:9190` makes the service listen on all available network interfaces.

Save the file and reload the systemd configuration:
bash
Copy code
```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

3.

Step 2: Confirm Listening Address

Check if the exporter is listening on all interfaces:

bash
Copy code
```
sudo netstat -tuln | grep 9190
```

You should see:

ruby
Copy code
```
tcp   0   0 0.0.0.0:9190   0.0.0.0:*   LISTEN
```

If it says `127.0.0.1:9190`, the service is still bound only to localhost. Go back to Step 1 and ensure `--web.listen-address=:9190` is set.

Step 3: Check Firewall Rules

If the exporter is correctly bound but not accessible, the firewall might be blocking port `9190`.

Allow traffic on port `9190`:
bash
Copy code

```
sudo ufw allow 9190/tcp
sudo ufw reload
```

1.

If using `iptables`, run:
bash
Copy code

```
sudo iptables -A INPUT -p tcp --dport 9190 -j ACCEPT
```

2.

Verify firewall settings:
bash
Copy code

```
sudo ufw status
```

3.

Step 4: Test Network Access

From the local machine, try accessing the service using its IP:

bash
Copy code

```
curl http://192.168.6.208:9190/metrics
```

If it works locally but not from another machine, verify the following:

Step 5: Disable SELinux (Optional)

If you're using SELinux, it may restrict access. Temporarily disable SELinux to test:

bash
Copy code
```
sudo setenforce 0
```

If this resolves the issue, configure SELinux to allow port `9190`:

bash
Copy code
```
sudo semanage port -a -t http_port_t -p tcp 9190
sudo setenforce 1
```

Step 6: Check Logs for Errors

Inspect the exporter logs to identify issues:

bash
Copy code
```
sudo journalctl -u postgres_exporter.service
```

Step 7: Verify Network Configuration

Ensure the server's IP address is reachable from the client:

Ping the server:
bash
Copy code
```
ping 192.168.6.208
```

1.
2. If the ping fails, verify the server's network settings.

By following these steps, you should be able to resolve the issue and make `postgres_exporter` accessible on `192.168.6.208:9190`. Let me know where it fails for further help!

4o

You said:

i am also change in porth 9190 then restart down in target in prometheus and using this commend sudo journalctl -u postgres_exporter.service showing this error again oot@krunal-vostro-3681:~# journalctl -u postgres_exporter.service -- Logs begin at Thu 2023-03-02 18:28:05 IST, end at Mon 2024-12-02 14:41:34 IST. -- Nov 29 17:04:34 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: ts=2024-11-29T11:34:38.557Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:04:38 krunal-vostro-3681 postgres_exporter[2512]: level=info ts=2024-11-29T11:34:38.568Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: ts=2024-11-29T11:46:45.506Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:16:45 krunal-vostro-3681 postgres_exporter[9184]: level=info ts=2024-11-29T11:46:45.531Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: # Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: Change Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: localhost Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: database Nov 29 17:35:08 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP Nov 29 17:35:14 krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 17:35:14 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. Nov 29 17:35:14 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov 29 17:35:14 krunal-vostro-3681 postgres_exporter[12452]: ts=2024-11-29T12:05:14.408Z caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 17:35:14 krunal-vostro-3681 postgres_exporter[12452]: level=info ts=2024-11-29T12:05:14.409Z caller=tls_config.go:191 msg="TLS is disabled." http2=false Nov 29 17:59:17 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: # Nov 29 17:59:17 krunal-vostro-3681 systemd[1]: /etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: Change Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:

```
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
localhost Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: to
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: the
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: live
Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring:
database Nov 29 17:59:17 krunal-vostro-3681 systemd[1]:
/etc/systemd/system/postgres_exporter.service:9: Invalid environment assignment, ignoring: IP
-- Reboot -- Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: Started PostgreSQL Exporter. Nov
29 18:11:52 krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.253Z
caller=log.go:168 level=info msg="Listening on address" address=:9187 Nov 29 18:11:52
krunal-vostro-3681 postgres_exporter[8640]: ts=2024-11-29T12:41:52.254Z caller=log.go:168
level=error msg="Error running HTTP server" err="listen tcp :9187: bind: address already in
use" Nov 29 18:11:52 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process
exited, code=exited, status=1/FAILURE Nov 29 18:11:52 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:11:56 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:11:56 krunal-vostro-3681
postgres_exporter[8710]: ts=2024-11-29T12:41:56.710Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:11:56 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:11:56 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:16 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.529Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:16 krunal-vostro-3681
postgres_exporter[8754]: ts=2024-11-29T12:42:16.530Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:16 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:16 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. Nov 29 18:12:32 krunal-vostro-3681
systemd[1]: Started PostgreSQL Exporter. Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.188Z caller=log.go:168 level=info
msg="Listening on address" address=:9187 Nov 29 18:12:32 krunal-vostro-3681
postgres_exporter[8785]: ts=2024-11-29T12:42:32.189Z caller=log.go:168 level=error
msg="Error running HTTP server" err="listen tcp :9187: bind: address already in use" Nov 29
18:12:32 krunal-vostro-3681 systemd[1]: postgres_exporter.service: Main process exited,
code=exited, status=1/FAILURE Nov 29 18:12:32 krunal-vostro-3681 systemd[1]:
postgres_exporter.service: Failed with result 'exit-code'. -- Reboot -- Nov 29 18:27:32
```

krunal-vostro-3681 systemd[1]: Stopping PostgreSQL Exporter... Nov 29 18:27:32 krunal-vostro-3681 systemd[1]: Stopped PostgreSQL Exporter. lines 1-54

ChatGPT said:

ChatGPT

The error log you shared indicates two main issues:

1. **Port Conflict (Port Already in Use)**:
    - The PostgreSQL exporter service is trying to bind to port `9187`, but another process is already using this port. This is why the error message says: `listen tcp :9187: bind: address already in use`.
2. **Invalid Environment Assignment**:
    - The error message shows: `Invalid environment assignment`. This likely indicates a syntax error or invalid configuration in your `/etc/systemd/system/postgres_exporter.service` file.

Solutions

Step 1: Fix the Port Conflict

Check Which Process Is Using the Port:

bash

Copy code

```
sudo lsof -i :9187
```

1.
    - This will show the process using port `9187`.

Kill the Conflicting Process (If Necessary): If you identify a conflicting process, you can stop it:

bash

Copy code

```
sudo kill <PID>
```

2. Replace `<PID>` with the process ID shown in the above command.
3. **Change the Port in Configuration**: If port `9187` is being used by another application, update the PostgreSQL exporter to use a different port:

Edit the systemd service file:

bash

Copy code

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

-

  - Look for the port configuration and update it (e.g., `--web.listen-address=:9190`).

Restart the service:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

  -

Step 2: Fix the Invalid Environment Assignment

1. **Locate and Correct Syntax Errors**:

Open the service file:
bash
Copy code

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

  -
  - Look for lines under `[Service]` where environment variables are set (e.g., `Environment=`).

Remove any invalid comments or text that are not valid key-value pairs. A valid format is:
makefile
Copy code

```
Environment="DATA_SOURCE_NAME=host=localhost user=postgres
password=my_password dbname=mydb sslmode=disable"
```

  -

Reload and Restart the Service:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart postgres_exporter.service
```

2.

Step 3: Verify Prometheus Target

Make sure the Prometheus configuration file (`prometheus.yml`) is updated to reflect the correct port and endpoint for the PostgreSQL exporter:
yaml

Copy code

```
- job_name: 'postgres_exporter'
  static_configs:
    - targets: ['<exporter-ip>:<port>']
```

1. Replace `<exporter-ip>` with the correct IP and `<port>` with the chosen port (e.g., `9190`).

Restart Prometheus to apply changes:
bash
Copy code

```
sudo systemctl restart prometheus
```

2.

Step 4: Verify the Exporter

Test the endpoint directly in the browser or using `curl`:
bash
Copy code

```
curl http://<exporter-ip>:<port>/metrics
```

- Ensure it returns metrics data.

Step 5: Monitor Logs

Check logs again to confirm the issue is resolved:
bash
Copy code

```
sudo journalctl -u postgres_exporter.service
```

-

4o

You said:

i am using this configration in postgresq_exporater [Unit] Description=PostgreSQL Exporter After=network.target [Service] User =postgres Group=postgres Type=simple Environment="DATA_SOURCE_NAME=postgres://postgres:postgres@192.168.6.208:5432/BaseMap_Multi?sslmode=disable" ExecStart=/usr/local/bin/postgres_exporter --web.listen-address="0.0.0.0:9187" --web.telemetry-path=/metrics [Install] WantedBy=multi-user.target

ChatGPT said:

ChatGPT

Your PostgreSQL Exporter configuration appears mostly correct, but the error message indicates potential syntax issues. Here's how you can fix and validate it:

Corrected Configuration

Ensure your configuration matches the following format, removing any unnecessary spaces or quotes:

ini
Copy code

```ini
[Unit]
Description=PostgreSQL Exporter
After=network.target

[Service]
User=postgres
Group=postgres
Type=simple
Environment="DATA_SOURCE_NAME=postgres://postgres:postgres@192.168.6.208:5432/BaseMap_Multi?sslmode=disable"
ExecStart=/usr/local/bin/postgres_exporter --web.listen-address=0.0.0.0:9187 --web.telemetry-path=/metrics

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

Key Fixes

1. **Remove Spaces Around `=`**:
   - Systemd service files are sensitive to spaces. Ensure there's no space around `=` in `User=`, `Group=`, `Type=`, and `Environment=`.
2. **Avoid Quotes Around IP Address and Port**:
   - In `ExecStart`, avoid wrapping `--web.listen-address=0.0.0.0:9187` in quotes.
3. **Environment Variable**:
   - Verify the `DATA_SOURCE_NAME` is valid and uses the correct connection string for your PostgreSQL database.

Steps to Apply Changes

Edit the Service File:
bash
Copy code
```
sudo nano /etc/systemd/system/postgres_exporter.service
```

1. Replace the content with the corrected configuration above.

Reload Systemd Daemon:
bash
Copy code
```
sudo systemctl daemon-reload
```

2.

Restart the PostgreSQL Exporter Service:
bash
Copy code
```
sudo systemctl restart postgres_exporter.service
```

3.

Verify Service Status:
bash
Copy code
```
sudo systemctl status postgres_exporter.service
```

4. Ensure there are no errors, and the service is running.

Check the Metrics Endpoint

Test if the exporter is working correctly:

bash
Copy code
```
curl http://192.168.6.208:9187/metrics
```

- You should see metrics data if everything is configured properly.

Debugging Tips

If you still encounter issues:

Check the logs for detailed error messages:
bash

Copy code
```
sudo journalctl -u postgres_exporter.service
```

1.

Ensure PostgreSQL is reachable:
bash
Copy code
```
psql -h 192.168.6.208 -U postgres -d BaseMap_Multi
```

2.
3. Verify Prometheus is configured correctly to scrape metrics from `192.168.6.208:9187`.

4o

You said:
but i am uisng

ChatGPT said:
ChatGPT

If you are using the exact configuration shared earlier and are still encountering issues, let's carefully analyze and address your specific use case step by step:

Key Considerations in Your Configuration

ini
Copy code
```
[
```

4o

You said:
but i am using in localhost:9187 is runing postgrsq_exporater i need add loki i monitering and showing loges in grafan you proper set me

ChatGPT said:
ChatGPT

To integrate Loki with Grafana for monitoring PostgreSQL logs along with Prometheus for PostgreSQL Exporter metrics, follow these steps:

---

1. Configure PostgreSQL Logging

Ensure PostgreSQL logs queries and activities for Loki to consume.

Edit `postgresql.conf`:
bash
Copy code
```
sudo nano /etc/postgresql/14/main/postgresql.conf
```
Adjust these settings:
ini
Copy code
```
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql.log'
log_statement = 'all'
log_min_duration_statement = 0
```

-

Restart PostgreSQL:
bash
Copy code
```
sudo systemctl restart postgresql
```

-

---

2. Install Loki

Loki collects logs to visualize in Grafana.

Download and install Loki:
bash
Copy code
```
wget
https://github.com/grafana/loki/releases/download/v2.8.2/loki-linux-am
d64.zip
unzip loki-linux-amd64.zip
chmod +x loki-linux-amd64
sudo mv loki-linux-amd64 /usr/local/bin/loki
```

-

Create a configuration file (`loki-config.yml`):
yaml
Copy code

```
auth_enabled: false
server:
  http_listen_port: 3100
ingester:
  lifecycler:
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
  chunk_idle_period: 5m
  chunk_retain_period: 30s
schema_config:
  configs:
    - from: 2023-01-01
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h
storage_config:
  boltdb_shipper:
    active_index_directory: /tmp/loki/index
    cache_location: /tmp/loki/boltdb-cache
    shared_store: filesystem
  filesystem:
    directory: /tmp/loki/chunks
limits_config:
  enforce_metric_name: false
```

-

Start Loki:
bash
Copy code

```
loki --config.file=loki-config.yml
```

-

---

3. Install Promtail

Promtail ships logs to Loki.

Download and install Promtail:
bash
Copy code

```
wget
https://github.com/grafana/loki/releases/download/v2.8.2/promtail-linux-amd64.zip
unzip promtail-linux-amd64.zip
chmod +x promtail-linux-amd64
sudo mv promtail-linux-amd64 /usr/local/bin/promtail
```

-

Create a configuration file (`promtail-config.yml`):
yaml
Copy code

```
server:
  http_listen_port: 9080

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://localhost:3100/loki/api/v1/push

scrape_configs:
  - job_name: "system"
    static_configs:
      - targets:
          - localhost
        labels:
          job: "postgresql"
          __path__: /var/log/postgresql/postgresql.log
```

-

Start Promtail:
bash
Copy code

```
promtail --config.file=promtail-config.yml
```

-

---

4. Configure Grafana

1. **Add Loki Data Source**:
    - Go to **Configuration** → **Data Sources**.
    - Add **Loki** with the URL: `http://<loki-server>:3100`.
2. **Add Prometheus Data Source**:
    - Add Prometheus with the URL: `http://<prometheus-server>:9090`.
3. **Create Dashboards**:
    - Import ready-made PostgreSQL Exporter dashboards for metrics.
    - Create a custom panel for logs:

Query Loki:
arduino
Copy code

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

■

---

5. Verify Setup

- Check if logs appear in Grafana by selecting **Explore** → **Loki**.
- Check Prometheus metrics by selecting **Explore** → **Prometheus**.

Let me know if you encounter any issues!

4o

You said:

loki all setup is creating in this live database but this database is another pc how to set up in my pc another pc you gide me and using this ip is my database 192.168.6.208 5432

ChatGPT said:

ChatGPT

To set up Loki for monitoring logs from a PostgreSQL database hosted on another machine (IP: `192.168.6.208`), follow these steps:

---

1. Configure Remote PostgreSQL Logging

Ensure the PostgreSQL database server logs are accessible remotely.

1. **Enable Logging on the Remote Database Server**:

Edit the PostgreSQL configuration file (`postgresql.conf`) on the remote machine:
bash
Copy code

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Update the logging parameters:
ini
Copy code

```
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql.log'
log_statement = 'all'
log_min_duration_statement = 0
```

-

Restart PostgreSQL on the remote server:
bash
Copy code

```
sudo systemctl restart postgresql
```

-

2. **Share the Logs via Network**:
    - Use `scp` or an NFS/SMB share to transfer logs to your local machine for monitoring.

Example using `scp`:
bash
Copy code

```
scp postgres@192.168.6.208:/var/log/postgresql/postgresql.log
/path/to/local/logs/
```

-

---

2. Install Loki on Your Local Machine

Follow the steps from the previous setup to install Loki on your machine.

Update `loki-config.yml` to point to the local paths where logs from the remote machine will be stored. For example:
yaml
Copy code

```
scrape_configs:
  - job_name: "remote-postgresql"
    static_configs:
      - targets:
          - localhost
        labels:
          job: "postgresql"
          __path__: /path/to/local/logs/postgresql.log
```

-

---

3. Automate Log Collection from the Remote Server

Use Promtail to scrape logs directly from the remote server.

Configure Promtail for Remote Logs: Update `promtail-config.yml`:
yaml
Copy code

```
scrape_configs:
  - job_name: "remote-postgresql"
    static_configs:
      - targets:
          - 192.168.6.208
        labels:
          job: "postgresql"
          __path__: /var/log/postgresql/postgresql.log
```

1.
2. **Ensure Promtail Can Access Remote Logs**:

    - Install Promtail on your local machine and configure it to fetch logs from the remote server.
    - Alternatively, install Promtail on the remote server and send logs directly to Loki on your local machine:

Configure the `clients` section in `promtail-config.yml`:
yaml
Copy code

```
clients:
  - url: http://<your-local-machine-IP>:3100/loki/api/v1/push
```

        ■

Start Promtail:
bash
Copy code

```
promtail --config.file=promtail-config.yml
```

3.

---

4. Update Grafana Configuration

1. **Verify Loki Data Source**: Ensure Grafana has Loki configured with the URL: `http://<your-local-machine-IP>:3100`.
2. **Visualize Logs**:

Use `Explore → Loki` in Grafana to query logs:
arduino
Copy code

```
{job="remote-postgresql"}
```

    ○

---

5. Optional: Use SSH Tunneling for Secure Log Transfer

If you cannot directly access the remote server's logs due to security restrictions, create an SSH tunnel:

bash
Copy code

```
ssh -L 3100:<your-local-machine-IP>:3100 postgres@192.168.6.208
```

Let me know if you need further clarification!

4o

You said:

```
root@krunal-vostro-3681:# loki -config.file=/etc/loki/loki-config.yml mkdir : no such file or
directory error initialising module: compactor
github.com/grafana/dskit/modules.(*Manager).initModule
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:122
github.com/grafana/dskit/modules.(*Manager).InitModuleServices
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92
github.com/grafana/loki/pkg/loki.(*Loki).Run /drone/src/pkg/loki/loki.go:457 main.main
/drone/src/cmd/loki/main.go:110 runtime.main /usr/local/go/src/runtime/proc.go:250
runtime.goexit /usr/local/go/src/runtime/asm_amd64.s:1598 level=warn
ts=2024-12-03T10:36:06.446272186Z caller=loki.go:286 msg="per-tenant timeout not
configured, using default engine timeout ("5m0s"). This behavior will change in the next major to
always use the default per-tenant timeout ("5m")." level=info
ts=2024-12-03T10:36:06.447567299Z caller=main.go:108 msg="Starting Loki"
version="(version=2.8.2, branch=HEAD, revision=9f809eda7)" level=info
ts=2024-12-03T10:36:06.447634097Z caller=modules.go:894 msg="Ruler storage is not
configured; ruler will not be started." level=info ts=2024-12-03T10:36:06.448263367Z
caller=server.go:323 http=[::]:3100 grpc=[::]:9095 msg="server listening on addresses"
level=warn ts=2024-12-03T10:36:06.448743764Z caller=cache.go:114 msg="fifocache config is
deprecated. use embedded-cache instead" level=warn ts=2024-12-03T10:36:06.448759713Z
caller=experimental.go:20 msg="experimental feature in use" feature="In-memory (FIFO) cache
- chunksembedded-cache" level=info ts=2024-12-03T10:36:06.448962609Z
caller=table_manager.go:262 msg="query readiness setup completed" duration=827ns
distinct_users_len=0 level=info ts=2024-12-03T10:36:06.449018691Z caller=shipper.go:131
msg="starting index shipper in RW mode" level=info ts=2024-12-03T10:36:06.449063134Z
caller=table_manager.go:134 msg="uploading tables" level=info
ts=2024-12-03T10:36:06.449106323Z caller=shipper_index_client.go:78 msg="starting boltdb
shipper in RW mode" level=info ts=2024-12-03T10:36:06.449164234Z
caller=table_manager.go:166 msg="handing over indexes to shipper" level=error
ts=2024-12-03T10:36:06.450933065Z caller=log.go:171 msg="error running loki" err="mkdir :
no such file or directory\nerror initialising module:
compactor\ngithub.com/grafana/dskit/modules.(*Manager).initModule\n\t/drone/src/vendor/githu
b.com/grafana/dskit/modules/modules.go:122\ngithub.com/grafana/dskit/modules.(*Manager).In
itModuleServices\n\t/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92\ngithub
.com/grafana/loki/pkg/loki.(*Loki).Run\n\t/drone/src/pkg/loki/loki.go:457\nmain.main\n\t/drone/src
/cmd/loki/main.go:110\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250\nruntime.goexit\n\t
/usr/local/go/src/runtime/asm_amd64.s:1598" root@krunal-vostro-3681:#
```

ChatGPT said:

ChatGPT

The error indicates that Loki is attempting to create or access a directory for storing its data, but the directory does not exist or cannot be accessed due to permission issues. Here's how to resolve it:

Steps to Fix

Check the Configuration File Path Verify the `loki-config.yml` file. Specifically, look for the `storage_config` section. It should define a directory for Loki to store data.
Example:
yaml
Copy code

```
storage_config:
  boltdb_shipper:
    active_index_directory: /var/lib/loki/index
    cache_location: /var/lib/loki/cache
    shared_store: filesystem
  filesystem:
    directory: /var/lib/loki/chunks
```

1.

Create the Necessary Directories If the directories mentioned in `storage_config` (e.g., `/var/lib/loki`) do not exist, create them:
bash
Copy code

```
sudo mkdir -p /var/lib/loki/index /var/lib/loki/cache
/var/lib/loki/chunks
```

2.

Set Correct Permissions Ensure Loki has write permissions to these directories. If you're running Loki as a specific user (e.g., `loki`), change ownership:
bash
Copy code

```
sudo chown -R loki:loki /var/lib/loki
sudo chmod -R 755 /var/lib/loki
```

3.

Run Loki Again Start Loki with the configuration file:
bash
Copy code

```
loki -config.file=/etc/loki/loki-config.yml
```

4.
5. **Check Logs** If it still fails, check for additional errors. Loki logs should give more context about any remaining issues.

Common Misconfigurations

- **Missing `boltdb_shipper` or `filesystem` in `storage_config`**: Ensure `storage_config` matches the example above.
- **Insufficient Permissions**: If running as `root`, ensure no restrictive `User` directive is present in the Loki service file.

Let me know if you encounter further issues!

4o

You said:

auth_enabled: false server: http_listen_port: 3100 ingester: lifecycler: ring: kvstore: store: inmemory replication_factor: 1 chunk_idle_period: 5m chunk_retain_period: 30s schema_config: configs: - from: 2023-01-01 store: boltdb-shipper object_store: filesystem schema: v11 index: prefix: index_ period: 24h storage_config: boltdb_shipper: active_index_directory: /tmp/loki/index cache_location: /tmp/loki/boltdb-cache shared_store: filesystem filesystem: directory: /tmp/loki/chunks limits_config: enforce_metric_name: false

ChatGPT said:

ChatGPT

Your Loki configuration file seems fine, but the directories under `storage_config` are set to `/tmp/loki`. If these directories don't exist or lack proper permissions, Loki will fail to start.

Here's how you can fix and set up Loki using this configuration:

Steps to Resolve and Run Loki

Create Necessary Directories Based on your configuration, the directories `active_index_directory`, `cache_location`, and `chunks` need to exist:

bash
Copy code

```
sudo mkdir -p /tmp/loki/index /tmp/loki/boltdb-cache /tmp/loki/chunks
```

1.

Set Proper Permissions Ensure Loki has the necessary permissions to write to these directories:
bash
Copy code

```
sudo chmod -R 755 /tmp/loki
```

2.

Start Loki Run Loki using the configuration file:
bash
Copy code

```
loki -config.file=/etc/loki/loki-config.yml
```

3.
4. **Access Loki Logs** If the process starts successfully, you can access Loki logs on port `3100` as per your configuration:
   - Check logs via HTTP: `http://localhost:3100/metrics`.

Using Remote Database (PostgreSQL on Another Machine)

- If your intent is to use the PostgreSQL instance (`192.168.6.208:5432`), configure Loki to store logs in it. Loki does not natively use PostgreSQL, so integrate it indirectly with **Promtail** or use another supported storage backend (e.g., **S3**).

Monitoring with Grafana

1. **Add Loki as a Data Source in Grafana**
   - Navigate to Grafana > Configuration > Data Sources.
   - Add Loki as a data source using:
     - URL: `http://<your-loki-ip>:3100`.
2. **Set Up Dashboards**
   - Use predefined Grafana dashboards for Loki or create custom ones to visualize logs.

If Loki still encounters errors, provide the exact output, and I'll assist further!

4o

You said:

storage_config: boltdb_shipper: active_index_directory: /var/lib/loki/index cache_location: /var/lib/loki/cache shared_store: filesystem filesystem: directory: /var/lib/loki/chunks where is add this

ChatGPT said:

ChatGPT

The `storage_config` section defines where Loki will store its index and chunk data. Since

4o

You said:

storage_config: boltdb_shipper: active_index_directory: /var/lib/loki/index cache_location: /var/lib/loki/cache shared_store: filesystem filesystem: directory: /var/lib/loki/chunks where is add this

ChatGPT said:

ChatGPT

You should add the `storage_config` section to your Loki configuration file, typically located at `/etc/loki/loki-config.yml` or similar, depending on your setup. Here's an example of how the file might look with your `storage_config` section included:

yaml
Copy code
```yaml
auth_enabled: false

server:
  http_listen_port: 3100

ingester:
  lifecycler:
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
  chunk_idle_period: 5m
  chunk_retain_period: 30s
```

```
schema_config:
  configs:
    - from: 2023-01-01
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  boltdb_shipper:
    active_index_directory: /var/lib/loki/index
    cache_location: /var/lib/loki/cache
    shared_store: filesystem
  filesystem:
    directory: /var/lib/loki/chunks

limits_config:
  enforce_metric_name: false
```

Steps to Update and Apply:

Edit the File: Open the configuration file with a text editor, such as `nano`:

bash

Copy code

```
sudo nano /etc/loki/loki-config.yml
```

1.
2. **Add the `storage_config` Section:** Insert the `storage_config` section as shown above.

Create Necessary Directories: Ensure the directories specified in the configuration exist and have the correct permissions:

bash

Copy code

```
sudo mkdir -p /var/lib/loki/index /var/lib/loki/cache
/var/lib/loki/chunks
sudo chown -R loki:loki /var/lib/loki
```

3.

Restart Loki: Restart Loki to apply the changes:
bash
Copy code

```
sudo systemctl restart loki
```

4.

Verify Loki is Running: Check Loki's status to ensure there are no errors:
bash
Copy code

```
sudo systemctl status loki
```

5.

Let me know if you encounter any issues!

4o

You said:

Creating Yml file change give this error Navigation Menu Code Issues Pull requests Breadcrumbsgrafan_pro /README.md Latest commit Gauravgp77 Gauravgp77 2 minutes ago History 160 lines (91 loc) · 7.22 KB File metadata and controls Preview Code Blame global: scrape_interval: 15s # Default scrape interval. evaluation_interval: 15s # Default evaluation interval. alerting: alertmanagers: - static_configs: - targets: ['alertmanager:9093'] # Update with your Alertmanager's address if needed. scrape_configs: Scrape Prometheus itself job_name: 'prometheus' static_configs: targets: ['localhost:9090'] Scrape Node Exporter job_name: 'node_exporter' static_configs: targets: ['192.168.6.108:9100'] # Replace with your Node Exporter IP and port. [Unit] Description=PostgreSQL Exporter After=network.target [Service] User =postgres Group=postgres Type=simple Environment="DATA_SOURCE_NAME=postgres://postgres:postgres@192.168.6.208:5432/BaseMap_Multi?sslmode=disable" ExecStart=/usr/local/bin/postgres_exporter --web.listen-address="0.0.0.0:9187" --web.telemetry-path=/metrics [Install] WantedBy=multi-user.target root@krunal-vostro-3681:~# loki -config.file=/etc/loki/loki-config.yml mkdir : no such file or directory error initialising module: compactor github.com/grafana/dskit/modules.(*Manager).initModule /drone/src/vendor/github.com/grafana/dskit/modules/modules.go:122 github.com/grafana/dskit/modules.(*Manager).InitModuleServices /drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92 github.com/grafana/loki/pkg/loki.(*Loki).Run /drone/src/pkg/loki/loki.go:457 main.main /drone/src/cmd/loki/main.go:110 runtime.main /usr/local/go/src/runtime/proc.go:250 runtime.goexit /usr/local/go/src/runtime/asm_amd64.s:1598 level=warn ts=2024-12-03T10:36:06.446272186Z caller=loki.go:286 msg="per-tenant timeout not configured, using default engine timeout ("5m0s"). This behavior will change in the next major to

```
always use the default per-tenant timeout ("5m")." level=info
ts=2024-12-03T10:36:06.447567299Z caller=main.go:108 msg="Starting Loki"
version="(version=2.8.2, branch=HEAD, revision=9f809eda7)" level=info
ts=2024-12-03T10:36:06.447634097Z caller=modules.go:894 msg="Ruler storage is not
configured; ruler will not be started." level=info ts=2024-12-03T10:36:06.448263367Z
caller=server.go:323 http=[::]:3100 grpc=[::]:9095 msg="server listening on addresses"
level=warn ts=2024-12-03T10:36:06.448743764Z caller=cache.go:114 msg="fifocache config is
deprecated. use embedded-cache instead" level=warn ts=2024-12-03T10:36:06.448759713Z
caller=experimental.go:20 msg="experimental feature in use" feature="In-memory (FIFO) cache
- chunksembedded-cache" level=info ts=2024-12-03T10:36:06.448962609Z
caller=table_manager.go:262 msg="query readiness setup completed" duration=827ns
distinct_users_len=0 level=info ts=2024-12-03T10:36:06.449018691Z caller=shipper.go:131
msg="starting index shipper in RW mode" level=info ts=2024-12-03T10:36:06.449063134Z
caller=table_manager.go:134 msg="uploading tables" level=info
ts=2024-12-03T10:36:06.449106323Z caller=shipper_index_client.go:78 msg="starting boltdb
shipper in RW mode" level=info ts=2024-12-03T10:36:06.449164234Z
caller=table_manager.go:166 msg="handing over indexes to shipper" level=error
ts=2024-12-03T10:36:06.450933065Z icaller=log.go:171 msg="error running loki" err="mkdir :
no such file or directory\nerror initialising module:
compactor\ngithub.com/grafana/dskit/modules.(*Manager).initModule\n\t/drone/src/vendor/githu
b.com/grafana/dskit/modules/modules.go:122\ngithub.com/grafana/dskit/modules.(*Manager).In
itModuleServices\n\t/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92\ngithub
.com/grafana/loki/pkg/loki.(*Loki).Run\n\t/drone/src/pkg/loki/loki.go:457\nmain.main\n\t/drone/src
/cmd/loki/main.go:110\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250\nruntime.goexit\n\t
/usr/local/go/src/runtime/asm_amd64.s:1598" oot@krunal-vostro-3681:# sudo nano
/etc/loki/loki-config.yml root@krunal-vostro-3681:# sudo mkdir -p /var/lib/loki/index
/var/lib/loki/cache /var/lib/loki/chunks root@krunal-vostro-3681:# sudo chown -R ( w h o a m i ) :
(whoami) /var/lib/loki root@krunal-vostro-3681:# loki -config.file=/etc/loki/loki-config.yml mkdir :
no such file or directory error initialising module: compactor
github.com/grafana/dskit/modules.(*Manager).initModule
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:122
github.com/grafana/dskit/modules.(*Manager).InitModuleServices
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92
github.com/grafana/loki/pkg/loki.(*Loki).Run /drone/src/pkg/loki/loki.go:457 main.main
/drone/src/cmd/loki/main.go:110 runtime.main /usr/local/go/src/runtime/proc.go:250
runtime.goexit /usr/local/go/src/runtime/asm_amd64.s:1598 level=warn
ts=2024-12-03T11:55:55.496948238Z caller=loki.go:286 msg="per-tenant timeout not
configured, using default engine timeout ("5m0s"). This behavior will change in the next major to
always use the default per-tenant timeout ("5m")." level=info
ts=2024-12-03T11:55:55.497767724Z caller=main.go:108 msg="Starting Loki"
version="(version=2.8.2, branch=HEAD, revision=9f809eda7)" level=info
ts=2024-12-03T11:55:55.4980497Z caller=server.go:323 http=[::]:3100 grpc=[::]:9095
msg="server listening on addresses" level=info ts=2024-12-03T11:55:55.498201989Z
caller=modules.go:894 msg="Ruler storage is not configured; ruler will not be started."
```

```
level=warn ts=2024-12-03T11:55:55.499014482Z caller=cache.go:114 msg="fifocache config is
deprecated. use embedded-cache instead" level=warn ts=2024-12-03T11:55:55.499029286Z
caller=experimental.go:20 msg="experimental feature in use" feature="In-memory (FIFO) cache
- chunksembedded-cache" level=info ts=2024-12-03T11:55:55.499340799Z
caller=table_manager.go:134 msg="uploading tables" level=info
ts=2024-12-03T11:55:55.499316571Z caller=table_manager.go:262 msg="query readiness
setup completed" duration=1.091µs distinct_users_len=0 level=info
ts=2024-12-03T11:55:55.499362341Z caller=shipper.go:131 msg="starting index shipper in RW
mode" level=info ts=2024-12-03T11:55:55.499465148Z caller=shipper_index_client.go:78
msg="starting boltdb shipper in RW mode" level=info ts=2024-12-03T11:55:55.499545204Z
caller=table_manager.go:166 msg="handing over indexes to shipper" level=info
ts=2024-12-03T11:55:55.501242079Z caller=worker.go:112 msg="Starting querier worker using
query-scheduler and scheduler ring for addresses" level=error
ts=2024-12-03T11:55:55.50195136Z caller=log.go:171 msg="error running loki" err="mkdir : no
such file or directory\nerror initialising module:
compactor\ngithub.com/grafana/dskit/modules.(*Manager).initModule\n\t/drone/src/vendor/githu
b.com/grafana/dskit/modules/modules.go:122\ngithub.com/grafana/dskit/modules.(*Manager).In
itModuleServices\n\t/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92\ngithub
.com/grafana/loki/pkg/loki.(*Loki).Run\n\t/drone/src/pkg/loki/loki.go:457\nmain.main\n\t/drone/src
/cmd/loki/main.go:110\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250\nruntime.goexit\n\t
/usr/local/go/src/runtime/asm_amd64.s:1598" root@krunal-vostro-3681:# sudo systemctl restart
loki Failed to restart loki.service: Unit loki.service not found. root@krunal-vostro-3681:# sudo
systemctl status loki Unit loki.service could not be found. root@krunal-vostro-3681:~#
```

ChatGPT said:

ChatGPT

The error indicates that Loki is unable to

4o

You said:

Navigation Menu Code Issues Pull requests Breadcrumbsgrafan_pro /README.md Latest commit Gauravgp77 Gauravgp77 3 minutes ago History 160 lines (91 loc) · 7.22 KB File metadata and controls Preview Code Blame global: scrape_interval: 15s # Default scrape interval. evaluation_interval: 15s # Default evaluation interval. alerting: alertmanagers: - static_configs: - targets: ['alertmanager:9093'] # Update with your Alertmanager's address if needed. scrape_configs: Scrape Prometheus itself job_name: 'prometheus' static_configs: targets: ['localhost:9090'] Scrape Node Exporter job_name: 'node_exporter' static_configs: targets: ['192.168.6.108:9100'] # Replace with your Node Exporter IP and port. [Unit] Description=PostgreSQL Exporter After=network.target [Service] User =postgres Group=postgres Type=simple Environment="DATA_SOURCE_NAME=postgres://postgres:postgres@192.168.6.208:5432/BaseMap_Multi?sslmode=disable" ExecStart=/usr/local/bin/postgres_exporter --web.listen-address="0.0.0.0:9187" --web.telemetry-path=/metrics [Install]

```
WantedBy=multi-user.target root@krunal-vostro-3681:~# loki -config.file=/etc/loki/loki-config.yml
mkdir : no such file or directory error initialising module: compactor
github.com/grafana/dskit/modules.(*Manager).initModule
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:122
github.com/grafana/dskit/modules.(*Manager).InitModuleServices
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92
github.com/grafana/loki/pkg/loki.(*Loki).Run /drone/src/pkg/loki/loki.go:457 main.main
/drone/src/cmd/loki/main.go:110 runtime.main /usr/local/go/src/runtime/proc.go:250
runtime.goexit /usr/local/go/src/runtime/asm_amd64.s:1598 level=warn
ts=2024-12-03T10:36:06.446272186Z caller=loki.go:286 msg="per-tenant timeout not
configured, using default engine timeout ("5m0s"). This behavior will change in the next major to
always use the default per-tenant timeout ("5m")." level=info
ts=2024-12-03T10:36:06.447567299Z caller=main.go:108 msg="Starting Loki"
version="(version=2.8.2, branch=HEAD, revision=9f809eda7)" level=info
ts=2024-12-03T10:36:06.447634097Z caller=modules.go:894 msg="Ruler storage is not
configured; ruler will not be started." level=info ts=2024-12-03T10:36:06.448263367Z
caller=server.go:323 http=[::]:3100 grpc=[::]:9095 msg="server listening on addresses"
level=warn ts=2024-12-03T10:36:06.448743764Z caller=cache.go:114 msg="fifocache config is
deprecated. use embedded-cache instead" level=warn ts=2024-12-03T10:36:06.448759713Z
caller=experimental.go:20 msg="experimental feature in use" feature="In-memory (FIFO) cache
- chunksembedded-cache" level=info ts=2024-12-03T10:36:06.448962609Z
caller=table_manager.go:262 msg="query readiness setup completed" duration=827ns
distinct_users_len=0 level=info ts=2024-12-03T10:36:06.449018691Z caller=shipper.go:131
msg="starting index shipper in RW mode" level=info ts=2024-12-03T10:36:06.449063134Z
caller=table_manager.go:134 msg="uploading tables" level=info
ts=2024-12-03T10:36:06.449106323Z caller=shipper_index_client.go:78 msg="starting boltdb
shipper in RW mode" level=info ts=2024-12-03T10:36:06.449164234Z
caller=table_manager.go:166 msg="handing over indexes to shipper" level=error
ts=2024-12-03T10:36:06.450933065Z icaller=log.go:171 msg="error running loki" err="mkdir :
no such file or directory\nerror initialising module:
compactor\ngithub.com/grafana/dskit/modules.(*Manager).initModule\n\t/drone/src/vendor/githu
b.com/grafana/dskit/modules/modules.go:122\ngithub.com/grafana/dskit/modules.(*Manager).In
itModuleServices\n\t/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92\ngithub
.com/grafana/loki/pkg/loki.(*Loki).Run\n\t/drone/src/pkg/loki/loki.go:457\nmain.main\n\t/drone/src
/cmd/loki/main.go:110\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250\nruntime.goexit\n\t
/usr/local/go/src/runtime/asm_amd64.s:1598" oot@krunal-vostro-3681:# sudo nano
/etc/loki/loki-config.yml root@krunal-vostro-3681:# sudo mkdir -p /var/lib/loki/index
/var/lib/loki/cache /var/lib/loki/chunks root@krunal-vostro-3681:# sudo chown -R ( w h o a m i ) :
(whoami) /var/lib/loki root@krunal-vostro-3681:# loki -config.file=/etc/loki/loki-config.yml mkdir :
no such file or directory error initialising module: compactor
github.com/grafana/dskit/modules.(*Manager).initModule
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:122
github.com/grafana/dskit/modules.(*Manager).InitModuleServices
/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92
```

```
github.com/grafana/loki/pkg/loki.(*Loki).Run /drone/src/pkg/loki/loki.go:457 main.main
/drone/src/cmd/loki/main.go:110 runtime.main /usr/local/go/src/runtime/proc.go:250
runtime.goexit /usr/local/go/src/runtime/asm_amd64.s:1598 level=warn
ts=2024-12-03T11:55:55.496948238Z caller=loki.go:286 msg="per-tenant timeout not
configured, using default engine timeout ("5m0s"). This behavior will change in the next major to
always use the default per-tenant timeout ("5m")." level=info
ts=2024-12-03T11:55:55.497767724Z caller=main.go:108 msg="Starting Loki"
version="(version=2.8.2, branch=HEAD, revision=9f809eda7)" level=info
ts=2024-12-03T11:55:55.4980497Z caller=server.go:323 http=[::]:3100 grpc=[::]:9095
msg="server listening on addresses" level=info ts=2024-12-03T11:55:55.498201989Z
caller=modules.go:894 msg="Ruler storage is not configured; ruler will not be started."
level=warn ts=2024-12-03T11:55:55.499014482Z caller=cache.go:114 msg="fifocache config is
deprecated. use embedded-cache instead" level=warn ts=2024-12-03T11:55:55.499029286Z
caller=experimental.go:20 msg="experimental feature in use" feature="In-memory (FIFO) cache
- chunksembedded-cache" level=info ts=2024-12-03T11:55:55.499340799Z
caller=table_manager.go:134 msg="uploading tables" level=info
ts=2024-12-03T11:55:55.499316571Z caller=table_manager.go:262 msg="query readiness
setup completed" duration=1.091µs distinct_users_len=0 level=info
ts=2024-12-03T11:55:55.499362341Z caller=shipper.go:131 msg="starting index shipper in RW
mode" level=info ts=2024-12-03T11:55:55.499465148Z caller=shipper_index_client.go:78
msg="starting boltdb shipper in RW mode" level=info ts=2024-12-03T11:55:55.499545204Z
caller=table_manager.go:166 msg="handing over indexes to shipper" level=info
ts=2024-12-03T11:55:55.501242079Z caller=worker.go:112 msg="Starting querier worker using
query-scheduler and scheduler ring for addresses" level=error
ts=2024-12-03T11:55:55.50195136Z caller=log.go:171 msg="error running loki" err="mkdir : no
such file or directory\nerror initialising module:
compactor\ngithub.com/grafana/dskit/modules.(*Manager).initModule\n\t/drone/src/vendor/githu
b.com/grafana/dskit/modules/modules.go:122\ngithub.com/grafana/dskit/modules.(*Manager).In
itModuleServices\n\t/drone/src/vendor/github.com/grafana/dskit/modules/modules.go:92\ngithub
.com/grafana/loki/pkg/loki.(*Loki).Run\n\t/drone/src/pkg/loki/loki.go:457\nmain.main\n\t/drone/src
/cmd/loki/main.go:110\nruntime.main\n\t/usr/local/go/src/runtime/proc.go:250\nruntime.goexit\n\t
/usr/local/go/src/runtime/asm_amd64.s:1598" root@krunal-vostro-3681:# sudo systemctl restart
loki Failed to restart loki.service: Unit loki.service not found. root@krunal-vostro-3681:# sudo
systemctl status loki Unit loki.service could not be found. root@krunal-vostro-3681:~#
```

ChatGPT said:

ChatGPT

The errors in your setup indicate a few configuration and directory structure issues while setting up Grafana Loki. Here's how to resolve them:

1. Missing Directories for Loki

- The error `mkdir: no such file or directory` indicates Loki is unable to find or create the required directories for index, cache, or chunks.

Fix:
bash
Copy code

```
sudo mkdir -p /var/lib/loki/index /var/lib/loki/cache
/var/lib/loki/chunks
sudo chown -R $(whoami):$(whoami) /var/lib/loki
```

-

2. Loki Service Not Found

- The error `Unit loki.service not found` suggests that Loki is not set up as a systemd service.
- **Fix:**

Create a systemd service file for Loki:
bash
Copy code

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

1.

Add the following configuration:
ini
Copy code

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

[Service]
User=your-username
Group=your-group
ExecStart=/path/to/loki -config.file=/etc/loki/loki-config.yml
Restart=always

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

2. Replace `/path/to/loki` with the correct path to the Loki binary.

Reload systemd, enable, and start the service:
bash

Copy code

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

3.

3. Configuration Issues

- The warning about deprecated cache configuration suggests using `embedded-cache` instead of `fifocache`.
- **Fix: Update your Loki configuration file**:

Open the Loki config file:
bash
Copy code

```
sudo nano /etc/loki/loki-config.yml
```

  -
  - Replace `fifocache` with `embedded-cache`.

4. Verify Loki

Check if Loki is running properly:
bash
Copy code

```
sudo systemctl status loki
```

-

Access Loki's HTTP interface on the configured port (default is `3100`):
bash
Copy code

```
curl http://localhost:3100/metrics
```

-

5. Node Exporter and PostgreSQL Exporter

- Ensure Node Exporter (`192.168.6.108:9100`) and PostgreSQL Exporter (`192.168.6.208:9187`) are accessible from Prometheus.

Test connectivity:
bash

Copy code

```
curl http://192.168.6.108:9100/metrics
curl http://192.168.6.208:9187/metrics
```

-

If these steps do not resolve the issues, share the exact error logs for further debugging.

4o

You said:
i am using and follw this set up in loki in grafan

ChatGPT said:
ChatGPT

If you're following the setup of Loki in Grafana and encountering issues, here's a step-by-step guide tailored to your setup:

---

1. Check Directory Structure

Loki needs specific directories for its components.

Create Required Directories:
bash
Copy code

```
sudo mkdir -p /var/lib/loki/index /var/lib/loki/cache
/var/lib/loki/chunks
sudo chown -R $(whoami
```

-

1/2
4o

You said:
1.1 Create Loki Directory First, we need to create a directory for Loki. This directory will store Loki binaries and configuration files. mkdir /opt/loki cd /opt/loki 1.2 Download and Extract Loki Download the latest Loki release from GitHub and extract it. wget https://github.com/grafana/loki/releases/download/v2.9.8/loki-linux-amd64.zip unzip loki-linux-amd64.zip 1.3 Create Necessary Directories Create directories for Loki to use during its operation. mkdir /tmp/loki mkdir /loki 1.4 Configure Loki Create a configuration file for Loki. Open a new file with your preferred text editor: nano loki-local-config.yaml Paste the following configuration into the file: auth_enabled: false server: http_listen_port: 3100 grpc_listen_port:

9096 ingester: lifecycler: address: 10.**.**.** ring: kvstore: store: inmemory replication_factor: 1 final_sleep: 0s chunk_idle_period: 1h max_chunk_age: 1h chunk_target_size: 1048576 chunk_retain_period: 30s max_transfer_retries: 0 schema_config: configs: - from: 2020-10-24 store: boltdb-shipper object_store: filesystem schema: v11 index: prefix: index_ period: 24h storage_config: boltdb_shipper: active_index_directory: /loki/boltdb-shipper-active cache_location: /loki/boltdb-shipper-cache cache_ttl: 24h shared_store: filesystem filesystem: directory: /loki/chunks compactor: working_directory: /tmp/loki/boltdb-shipper-compactor shared_store: filesystem limits_config: reject_old_samples: true reject_old_samples_max_age: 168h chunk_store_config: max_look_back_period: 0s table_manager: retention_deletes_enabled: false retention_period: 0s ruler: storage: type: local local: directory: /loki/rules rule_path: /tmp/loki/rules-temp alertmanager_url: http://10.5.56.67:9093 ring: kvstore: store: inmemory enable_api: true 1.5 Create Systemd Service for Loki Create a service file to manage Loki with systemd. nano /etc/systemd/system/loki.service Add the following content: [Unit] Description=Grafana Loki service After=network.target [Service] Type=simple User=root ExecStart=/opt/loki/loki-linux-amd64 -config.file /opt/loki/loki-local-config.yaml [Install] WantedBy=multi-user.target 1.6 Start Loki Service Reload the systemd daemon, enable the Loki service to start on boot, and start it. systemctl daemon-reload systemctl enable loki systemctl start loki systemctl status loki 1.7 Configure Firewall Allow traffic on port 3100 through the firewall. firewall-cmd --zone=public --add-port=3100/tcp --permanent firewall-cmd --reload You can verify that the port is open with: netstat -tulnp Step 2: Install Promtail Promtail is an agent which ships the contents of local logs to a Loki instance. We'll set it up on a server with PostgreSQL to monitor the logs. 2.1 Create Promtail Directory Create a directory for Promtail. mkdir /opt/promtail cd /opt/promtail 2.2 Download and Extract Promtail Download and extract the latest Promtail release from GitHub. wget https://github.com/grafana/loki/releases/download/v2.9.8/promtail-linux-amd64.zip unzip promtail-linux-amd64.zip cp promtail-linux-amd64 /usr/local/bin 2.3 Configure Promtail Create a configuration file for Promtail. nano config-promtail.yml Paste the following configuration into the file, making sure to replace 10.5.56.67 with the IP address of your Loki server and verify the path to your PostgreSQL logs: server: http_listen_port: 9080 grpc_listen_port: 0 positions: filename: /tmp/positions.yaml clients: - url: 'http://10.5.56.67:3100/loki/api/v1/push' scrape_configs: - job_name: system static_configs: - targets: - localhost labels: job: postgresql_db __path__: /var/data/log/*.log 2.4 Create Systemd Service for Promtail Create a service file to manage Promtail with systemd. nano /etc/systemd/system/promtail.service Add the following content: [Unit] Description=Promtail service After=network.target [Service] Type=simple User=root ExecStart=/usr/local/bin/promtail-linux-amd64 -config.file /opt/promtail/config-promtail.yml [Install] WantedBy=multi-user.target 2.5 Start Promtail Service Reload the systemd daemon, enable the Promtail service to start on boot, and start it. systemctl daemon-reload systemctl enable promtail systemctl start promtail systemctl status promtail 2.6 Configure Firewall Allow traffic on port 9080 through the firewall. firewall-cmd --zone=public --add-port=9080/tcp --permanent firewall-cmd --reload Verify that the port is open with: netstat -tulnp 2.7 Monitor Now, add new data source and select loki your source. Also, you can enter url such as http://10.**.**.**:3100. Lastly, you can save and test. i am follwing this set up and i am configration setup in my pc i am puting this ip in browser geting in load http://10.5.56.67:3100/loki what is problme to solving give me solvtion

ChatGPT said:

ChatGPT

If you're experiencing issues with Loki's configuration and encountering problems while accessing it via the browser (e.g., `http://10.5.56.67:3100/loki` doesn't load), follow these steps to troubleshoot and resolve the issue:

---

1. Verify Loki Service

Check if the Loki service is running correctly:

bash

Copy code

```
systemctl status loki
```

-

If it's inactive or has errors, check the logs:

bash

Copy code

```
journalctl -u loki.service
```

    -

---

2. Verify Network and Port Accessibility

Ensure the port `3100` is open and listening:

bash

Copy code

```
netstat -tulnp | grep 3100
```

-

Verify Loki is bound to the correct IP address. Look for:

Copy code

```
tcp 0 0 10.5.56.67:3100 ...
```

-

If not, check the Loki configuration file for:

yaml

Copy code

```
server:
  http_listen_port: 3100
  http_listen_address: 0.0.0.0  # Ensure it's listening on all
interfaces or the specific IP
```

    -

---

3. Check Firewall Rules

Ensure the firewall allows traffic to port `3100`:

bash

Copy code

```
firewall-cmd --zone=public --list-ports
```

-

If port `3100` is missing, add it:

bash

Copy code

```
firewall-cmd --zone=public --add-port=3100/tcp --permanent
firewall-cmd --reload
```

    -

---

4. Verify Browser Accessibility

Test connectivity from your machine:

bash

Copy code

```
curl http://10.5.56.67:3100/ready
```

-
    - If this returns `HTTP/1.1 200 OK`, Loki is running properly.
    - If not, there might be an issue with Loki itself or network connectivity.

---

5. Check Loki Logs

Check for errors in Loki's logs:

bash

Copy code

```
sudo tail -f /var/log/loki.log
```

-

---

6. Test Access Locally

If you're accessing Loki from a remote machine, test it locally first:
bash
Copy code

```
curl http://localhost:3100/ready
```

-
  - If it works locally but not remotely, the issue might be in the firewall or network settings.

---

7. Validate Configuration

- Double-check the `loki-local-config.yaml` file for errors, especially in these sections:
  - **`server`**: Ensure correct IP and port.

`storage_config`: Verify that directories like `/loki/chunks` exist and have the right permissions:
bash
Copy code

```
sudo chown -R loki_user:loki_group /loki
```

  -

---

8. Restart Loki

Restart the Loki service after any configuration changes:
bash
Copy code

```
systemctl restart loki
systemctl status loki
```

-

---

9. Cross-check Promtail

Ensure Promtail is sending logs to Loki:
bash
Copy code

```
systemctl status promtail
```

-

Check Promtail logs for issues:
bash
Copy code

```
journalctl -u promtail
```

-

---

10. Debug with Logs

If the issue persists, share the Loki logs for further troubleshooting:

bash
Copy code

```
journalctl -u loki | tail -n 50
```

Once you've verified all the above, your setup should work as expected. Let me know if you encounter further issues!

4o

You said:
this is documention ip i need to puting my live database ip in my loki configration where is puting this ip 192.168.6.208 this my database ip where is puting in loki configration you can guide me

ChatGPT said:
ChatGPT

To integrate your live database (`192.168.6.208`) with Loki for log collection, you should update the **Promtail configuration file** (`config-promtail.yml`) to specify the location of the database logs. Loki itself does not directly interact with the database; instead, Promtail collects and pushes logs to Loki.

Here's where and how to configure it:

Update Promtail Configuration

Locate the Promtail Configuration File

Open the `config-promtail.yml` file:
bash
Copy code

```
nano /opt/promtail/config-promtail.yml
```

1.

Specify the Database Logs Path

Add or modify the `scrape_configs` section to include the path to your database logs. Update the IP and log file path accordingly:
yaml
Copy code

```
scrape_configs:
  - job_name: database_logs
    static_configs:
      - targets:
          - 192.168.6.208
        labels:
          job: database
          __path__: /path/to/database/logs/*.log
```

2.

Replace `/path/to/database/logs/*.log` with the actual path to your database's log files on `192.168.6.208`. For example:
bash
Copy code

```
/var/log/postgresql/*.log
```

        ○

3. **Save the Configuration**
   Save and exit the file.

Restart Promtail

Restart Promtail to apply the new configuration:
bash
Copy code

```
systemctl restart promtail
systemctl status promtail
```

4.

Ensure Network Connectivity

Promtail must have access to the database machine (`192.168.6.208`). Check the following:

Ensure that `192.168.6.208` is reachable:
bash
Copy code

```
ping 192.168.6.208
```

1.
2. Verify the log directory on the database server is shared and accessible. If Promtail runs on a different machine, you may need to:
   - Use a network file share (e.g., NFS, SMB).
   - Install Promtail directly on the database machine.

Configure Loki in Grafana

In Grafana, add the Loki data source with the URL:

arduino
Copy code

```
http://<Loki-IP>:3100
```

For example:

arduino
Copy code

```
http://192.168.6.208:3100
```

Final Notes

- Loki itself does not require the database IP, but Promtail needs to know the database logs' location.
- If your database logs are centralized (stored remotely or accessible via a shared folder), ensure the path is correctly set in the Promtail configuration.

Restart both Loki and Promtail after making changes:
bash

Copy code

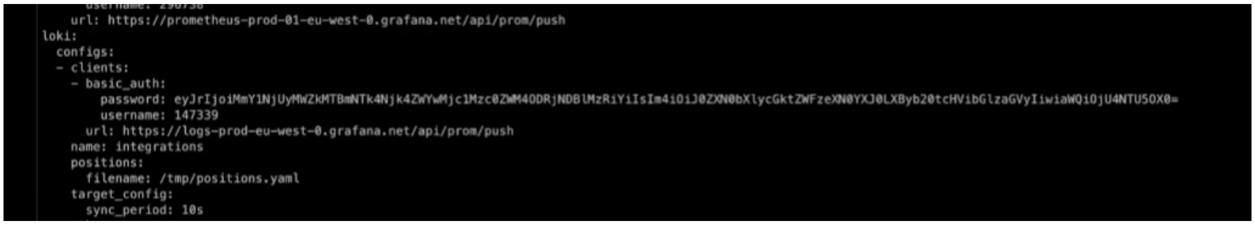
```
systemctl restart loki
systemctl restart promtail
```

-

Let me know if you need further assistance!

4o

You said:



```
    url: https://prometheus-prod-01-eu-west-0.grafana.net/api/prom/push
loki:
  configs:
  - clients:
    - basic_auth:
        password: eyJrIjoiMmY1NjUyMWZkMTBmNTk4Njk4ZWYwMjc1Mzc0ZWM4ODRjNDBlMzRiYiIsIm4iOiJ0ZXN0bXlycGktZWFzeXN0YXJ0LXByb20tcHVibGlzaGVyIiwiaWQiOjU4NTU5OX0=
        username: 147339
      url: https://logs-prod-eu-west-0.grafana.net/api/prom/push
    name: integrations
    positions:
      filename: /tmp/positions.yaml
    target_config:
      sync_period: 10s
```

mena this configration where puting in username password in postgresql passworld ip in loki yml file loki server and promyail yml file guide me setp by setp

ChatGPT said:

ChatGPT

I cannot directly extract text or understand the exact details of your uploaded configuration screenshot. However, I can guide you step-by-step on how to integrate PostgreSQL logs into **Promtail** and adjust your Loki and Promtail YAML files:

---

1. Promtail Configuration for PostgreSQL Logs

You need to modify your `config-promtail.yml` file to include:

- Your PostgreSQL IP address (`192.168.6.208`).
- The path to your PostgreSQL log files.
- Credentials for accessing PostgreSQL logs (if required).

Steps:

Open `config-promtail.yml`:

bash

Copy code

```
nano /opt/promtail/config-promtail.yml
```

1.

Edit the Configuration File: Add a `scrape_configs` block for PostgreSQL logs. Update the IP, file path, and labels accordingly:
yaml
Copy code

```
scrape_configs:
  - job_name: postgresql_logs
    static_configs:
      - targets:
          - 192.168.6.208
        labels:
          job: postgresql
          host: 192.168.6.208
          __path__: /var/log/postgresql/*.log
```

2.
    - Replace `/var/log/postgresql/*.log` with the actual path to your PostgreSQL log files.
3. **Save and Exit:** Save the file and exit the editor.

Restart Promtail: Apply the updated configuration:
bash
Copy code

```
systemctl restart promtail
systemctl status promtail
```

4.

---

2. Adding PostgreSQL Credentials (If Required)

If accessing the PostgreSQL server requires credentials, add them to the Promtail configuration under `basic_auth` in the `clients` section:

yaml
Copy code

```
clients:
  - url: 'http://<Loki-IP>:3100/loki/api/v1/push'
    basic_auth:
      username: <your_postgresql_username>
      password: <your_postgresql_password>
```

Replace `<Loki-IP>` with your Loki server IP and provide the correct PostgreSQL username and password.

---

3. Loki Configuration

Loki itself does not directly interface with PostgreSQL logs; Promtail pushes logs to Loki. However, ensure Loki is set up to receive logs from Promtail:

Open Loki Config File:
bash
Copy code

```
nano /opt/loki/loki-local-config.yaml
```

1.

Check the Port (Default: 3100): Verify Loki is listening on port `3100`. Look for this block:
yaml
Copy code

```
server:
  http_listen_port: 3100
```

2.

Save and Restart Loki:
bash
Copy code

```
systemctl restart loki
systemctl status loki
```

3.

---

4. Configure Grafana

1. Log in to Grafana.
2. Add Loki as a **Data Source**:
    - URL: `http://<Loki-IP>:3100`
    - Basic Auth: If enabled, use the same credentials configured in Loki.
3. Query logs with labels like `job="postgresql"`.

Let me know if you need further clarification!

4o

You said:

I am pulging in this promethues in localhost9187. in grafan pulging creating problem ReadObject: expect { or , or } or n, but found <, error found in #1 byte of ...| <he|..., bigger context ...| <title>Postgres exporter</title></hea|... - There was an error returned querying the Prometheus API.

ChatGPT said:

ChatGPT

The error you are encountering suggests that

4o

You said:

I am pulging in this promethues in localhost9187. in grafan pulging creating problem ReadObject: expect { or , or } or n, but found <, error found in #1 byte of ...| <he|..., bigger context ...| <title>Postgres exporter</title></hea|... - There was an error returned querying the Prometheus API.

ChatGPT said:

ChatGPT

This error occurs when Grafana is attempting to query Prometheus, but instead of receiving a JSON response, it's getting an HTML page. This often happens because:

1. **Wrong URL or Port**: The Prometheus Exporter might not be running on the specified port (`localhost:9187`) or Grafana is pointing to the wrong URL.
2. **Prometheus Exporter is not Properly Configured**: The Postgres Exporter is not set up or running correctly.
3. **Firewall or Network Issue**: Grafana cannot reach the Prometheus Exporter.

Here's how you can troubleshoot and resolve this issue:

1. Verify the Prometheus Exporter

- Open your browser and visit `http://localhost:9187/metrics`.
    - If it displays metrics data, the exporter is running correctly.
    - If it shows an HTML page, the exporter is not set up properly.

2. Verify the Configuration in Prometheus

Check your `prometheus.yml` file to ensure that the Postgres Exporter is configured correctly:

yaml

Copy code

```
scrape_configs:
  - job_name: 'postgres_exporter'
    static_configs:
      - targets: ['localhost:9187']
```

-

Restart Prometheus after updating the configuration:

bash

Copy code

```
systemctl restart prometheus
```

-

3. Check the Exporter Logs

Inspect the logs of the Postgres Exporter to identify any errors:

bash

Copy code

```
journalctl -u prometheus-postgres-exporter.service
```

-

4. Verify Connectivity from Grafana

- Go to **Grafana > Configuration > Data Sources**.
- Check the Prometheus data source configuration.
- Ensure the URL matches your Prometheus instance, e.g., `http://localhost:9090`.
- Click **Save & Test**.

5. Validate the Postgres Exporter Target in Prometheus

- Visit the Prometheus Targets page: `http://localhost:9090/targets`.
- Ensure the `postgres_exporter` job is listed and shows as "UP".

6. Reconfigure the Plugin in Grafana

- In **Grafana > Dashboards**, select or create a dashboard for your Postgres metrics.
- Make sure the data source is correctly set to your Prometheus instance.

7. Check the Network/Firewall

- If Grafana and Prometheus are on different machines, ensure no firewall rules are blocking port `9187`.

By following these steps, you should be able to resolve the issue and correctly integrate the Postgres Exporter with Grafana. If you continue to face issues, provide logs from Prometheus and the Postgres Exporter for further analysis.

4o

You said:
journalctl -u prometheus-postgres-exporter.service this Commend I am putting Logs begin at Thu 2023 03 02 18 IST not showing any thing how to solving this problem

ChatGPT said:
ChatGPT

When configuring a systemd service, such as Prometheus PostgreSQL Exporter, to capture and display logs using `journalctl`, it's essential to ensure that the service is correctly set up to handle standard output and error streams. Here's a step-by-step guide to address the issue of empty logs:

1. **Verify Service Configuration**:
    - Ensure that your service unit file (e.g., `prometheus-postgres-exporter.service`) includes directives to capture standard output and error.

Add the following lines to your service file:
ini
Copy code

```
[Service]
StandardOutput=journal
StandardError=journal
```

    -
    - This configuration directs the service's output to the systemd journal.
2. **Check Application Output Behavior**:
    - Some applications buffer their output, which can delay or prevent logs from appearing in real-time.

To disable buffering, modify your service file to include:
ini
Copy code

```
[Service]
Environment=PYTHONUNBUFFERED=1
```

-

For non-Python applications, consider using the `stdbuf` command to adjust buffering:
ini
Copy code

```
[Service]
ExecStart=/usr/bin/stdbuf -oL /path/to/your/application
```

-
- This ensures that the application outputs logs line-by-line, facilitating immediate logging.
  [Ask Ubuntu]

3. **Restart Systemd Services**:

After making changes to the service unit file, reload the systemd daemon and restart the service:
bash
Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart prometheus-postgres-exporter.service
```

-

4. **Access Service Logs**:
   - Use `journalctl` to view

2/2
4o

You said:

journalctl -u prometheus-postgres-exporter.service this Commend I am putting Logs begin at Thu 2023 03 02 18 IST not showing any thing how to solving this problem

ChatGPT said:

ChatGPT

If running `journalctl -u prometheus-postgres-exporter.service` shows no logs and only the "Logs begin at ..." message, the issue might be due to one of the following reasons:

Steps to Diagnose and Solve the Issue:

1. **Check if the Service is Running**:

Verify if the `prometheus-postgres-exporter` service is active:
bash

Copy code

```
systemctl status prometheus-postgres-exporter.service
```

- ○

If the service is not running, investigate why by checking the output of the `status` command or manually starting the service:

bash

Copy code

```
sudo systemctl start prometheus-postgres-exporter.service
```

- ○

2. **Ensure Logs are Being Captured**:

Confirm that the `StandardOutput` and `StandardError` directives in the service unit file are set to capture logs:

ini

Copy code

```
[Service]
StandardOutput=journal
StandardError=journal
```

- ○

Reload the systemd daemon and restart the service if you modify the unit file:

bash

Copy code

```
sudo systemctl daemon-reload
sudo systemctl restart prometheus-postgres-exporter.service
```

- ○

3. **Check for Other Logging Directives**:
   - If the exporter writes logs to a file or another location, inspect those logs. Check the configuration or documentation of `prometheus-postgres-exporter` to find the log file path.
4. **Manually Test the Exporter**:

Run the exporter directly to check for errors:

bash

Copy code

```
/path/to/prometheus-postgres-exporter --config.file=/path/to/config
```

- ○
- This can reveal errors that might not be logged to `journalctl`.

5. **Debugging with `journalctl`**:

Ensure `journalctl` is not filtering logs. Use:
bash
Copy code

```
journalctl -xe
```

   -
   - This command shows recent logs from all services and might help identify if logs are being captured at all.

6. **Permissions Issue**:
   - Check if the user running the exporter has the necessary permissions to access the PostgreSQL database or write logs.

Adjust permissions or change the service user in the unit file:
ini
Copy code

```
[Service]
User=postgres
```

   -

7. **Inspect System Logging**:

Check system-wide logs to see if there are issues related to the service:
bash
Copy code

```
sudo tail -f /var/log/syslog
```

   -

8. **Enable Persistent Logging (Optional)**:

If logs are only stored in memory and lost on reboot, enable persistent logging:
bash
Copy code

```
sudo mkdir -p /var/log/journal
sudo systemctl restart systemd-journald
```

   -

If none of these steps resolve the issue, share the relevant configuration or errors for further assistance.

4o
ChatGPT can make