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
```
[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,

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
```