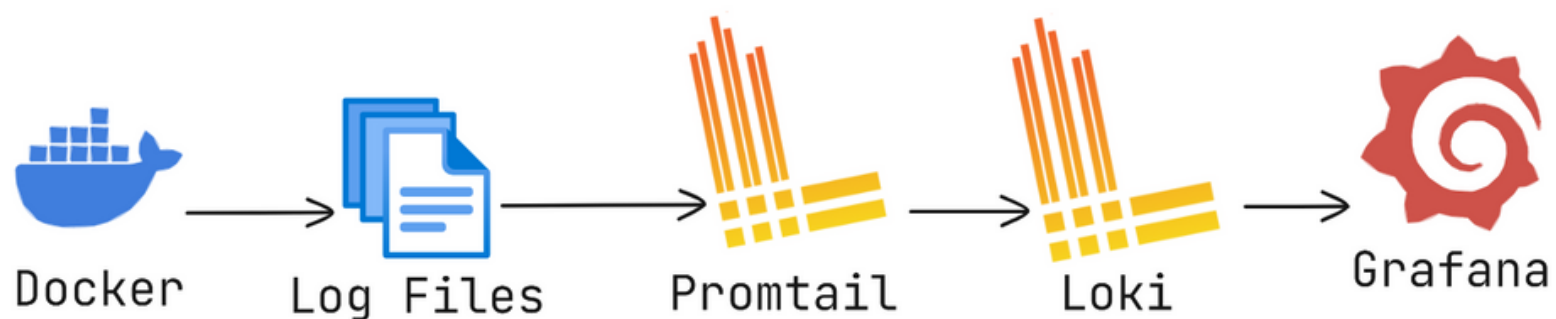


# Dataflow

1. First Docker container generate logs .
2. Then Promtail continuously tails the log files and sends the log data to Loki .
3. Loki stores the log entries .
4. At last Grafana queries Loki for log data and displays it through dashboard .



## 1. Install Grafana

```
wget https://dl.grafana.com/enterprise/release/grafana-enterprise-10.2.3.linux-amd64.tar.gz
```

```
tar -zxvf grafana-enterprise-10.2.3.linux-amd64.tar.gz
```

## 2. Install Loki and start Loki

```
wget https://raw.githubusercontent.com/grafana/loki/v2.9.4/cmd/loki/loki-local-config.yaml -O loki-config.yaml
```

```
docker run --name loki -d -v $(pwd):/mnt/config -p 3100:3100 grafana/loki:2.9.4 -config.file=/mnt/config/loki-config.yaml
```

### 3. Get Promtail config file

```
wget  
https://raw.githubusercontent.com/grafana/loki/v2.9.4/clients/  
cmd/promtail/promtail-docker-config.yaml -O promtail-  
config.yaml
```

### 4. Edit Promtail config file

Add this under the **static configs** . This scrape the docker container logs .

```
- targets:  
  - localhost  
  labels:  
    job: docker-logs  
    __path__: /var/lib/docker/containers/*/log
```

## **5. Run Promtail using the Promtail config file .**

```
docker run --name promtail -d -v $(pwd):/mnt/config -v /var/lib/docker/containers:/var/lib/docker/containers --link loki grafana/promtail:2.9.4 -config.file=/mnt/config/promtail-config.yaml
```

## **6. Access the metrics at port 3100**

<http://localhost:3100/metrics>

## 7. Start grafana server

```
cd grafana-v10.2.3/bin/
```

```
./grafana-server
```

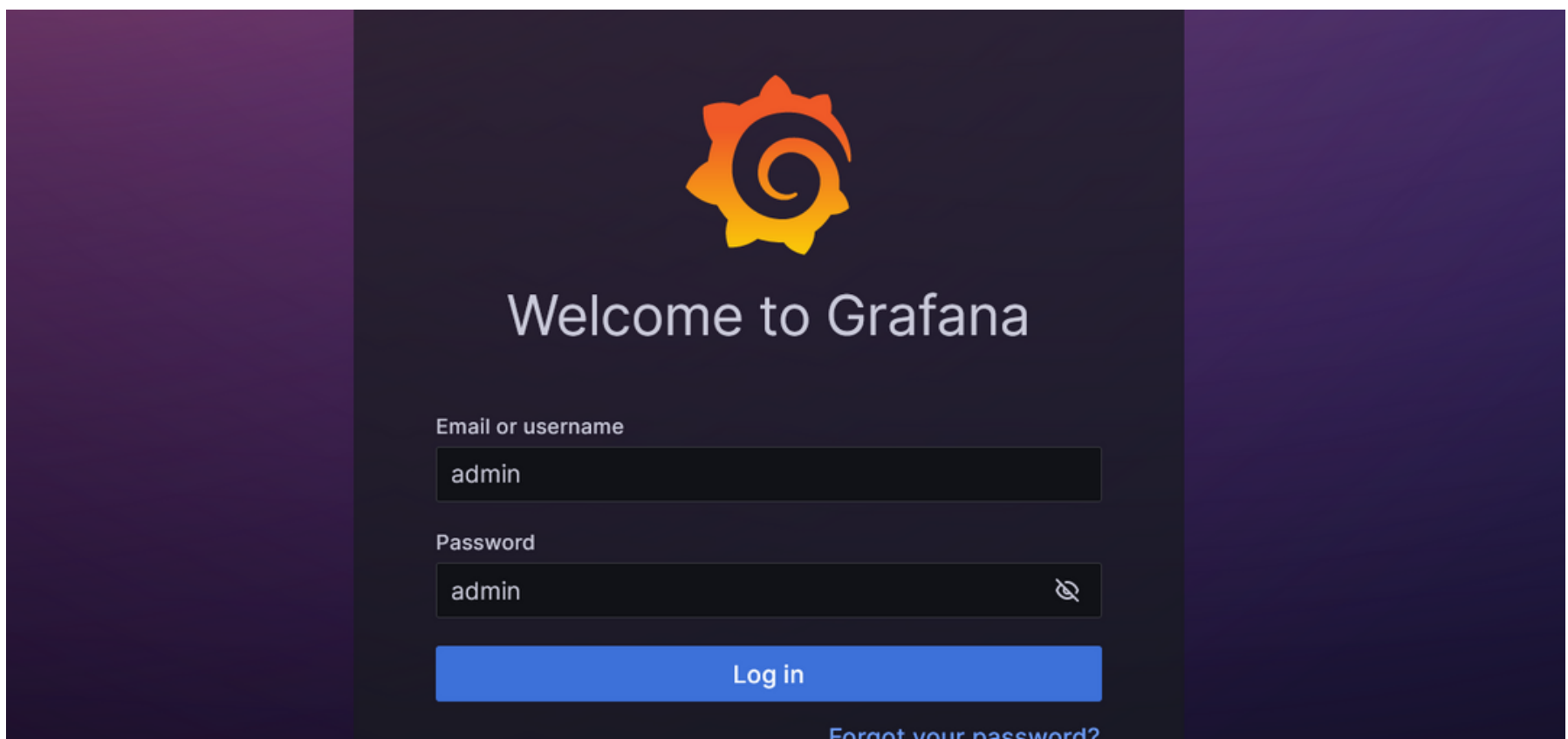
## 8. Connect grafana server

```
http://localhost:3000/login
```

## 9. Login to grafana server

Default username : admin

Default password : admin



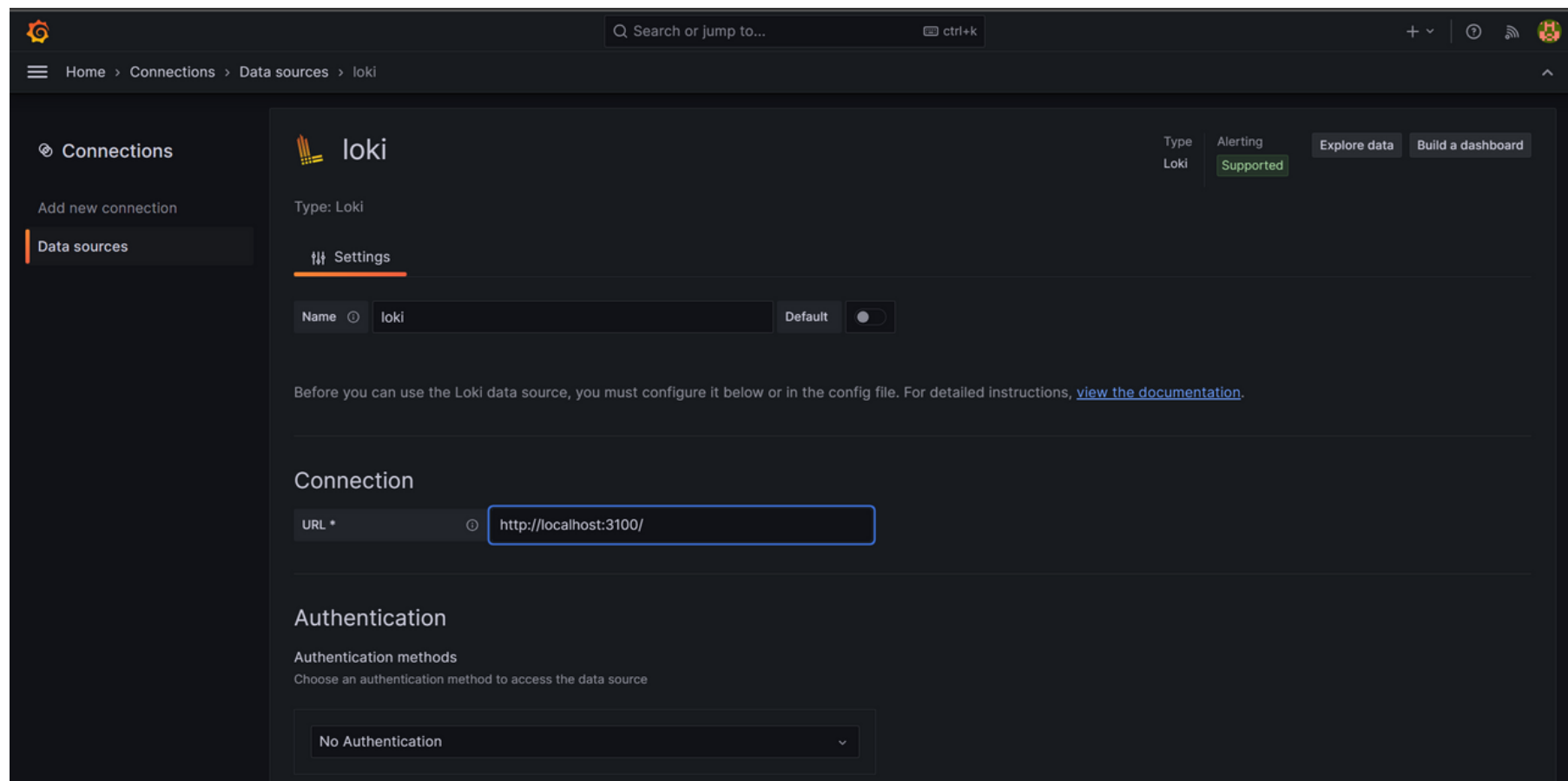
## 10. Integrate Loki and grafana

Go to grafana dashboard

Add a loki data source

Enter the URL in which loki server is running .

After entering the URL then click save and test .



## 11. See the docker logs

Create a new dashboard using loki as data source .  
Use this `{job="docker-logs"} |> `error`` to see logs where you are getting error keyword .

The screenshot displays the Grafana dashboard interface. At the top, the browser address bar shows `localhost:3000/dashboard/new?orgId=1&editPanel=1`. The dashboard header includes a search bar, navigation links, and buttons for 'Discard', 'Save', and 'Apply'. The main panel area shows a 'Panel Title' with a list of log entries, each containing a timestamp, caller, message, and error details. Below the panel, the 'Query' section is active, showing the data source 'loki' and the query `{job="docker-logs"} |> `error``. The 'Query options' section shows 'MD = auto = 1120' and 'Interval = 20s'. The 'Label filters' section shows 'job = docker-logs'. The 'Line contains' section shows 'error'. The 'Options' section shows 'Type: Range' and 'Line limit: 1000'. On the right side, the 'Logs' panel is visible, showing a search bar and a 'Repeat by variable' dropdown menu.