

hello I need to create setup in monitoring loges in my Apache Tomcat server using in Prometheus and Loki and grafan in my organigation more logs apache tomacat server then creating this dashborad in grafan and monitering loges then you creating me all proper set up i am using in ubuntu 18 you creating me proper set up

ChatGPT said:

ChatGPT

To set up monitoring of Apache Tomcat logs using Prometheus, Loki, and Grafana on Ubuntu 18, follow the step-by-step instructions below. This guide assumes you have administrative access to your server.

1. Install Prerequisites

Ensure you have the necessary tools installed:

- **Java:** Required for Apache Tomcat.
- **Docker and Docker Compose:** Simplifies the deployment of Prometheus, Loki, and Grafana.

Update and Install Basic Tools

bash

Copy code

```
sudo apt update && sudo apt upgrade -y  
sudo apt install curl wget git unzip openjdk-11-jdk -y
```

Install Docker

bash

Copy code

```
sudo apt install -y docker.io  
sudo systemctl enable docker  
sudo systemctl start docker  
sudo usermod -aG docker $USER
```

Install Docker Compose

bash

Copy code

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/$(curl -s  
https://api.github.com/repos/docker/compose/releases/latest | grep -oP
```

```
'"tag_name": "\K(.*)"?=)"')/docker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

2. Install and Configure Apache Tomcat

Download and Install Apache Tomcat

bash

Copy code

wget

```
https://downloads.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz
```

```
tar -xzf apache-tomcat-9.0.80.tar.gz
```

```
mv apache-tomcat-9.0.80 /opt/tomcat
```

1.

Start Apache Tomcat

bash

Copy code

```
/opt/tomcat/bin/startup.sh
```

2.

3. **Access the Tomcat UI** Visit <http://<your-server-ip>:8080> to verify that Tomcat is running.

3. Set Up Loki for Log Collection

Create a Loki Configuration File Save the following as `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
    max_chunk_age: 1h
    chunk_target_size: 1048576
    chunk_retain_period: 30s
  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/index
      cache_location: /loki/boltdb-cache
      shared_store: filesystem
    filesystem:
      directory: /loki/chunks
  limits_config:
    enforce_metric_name: false
    reject_old_samples: true
    reject_old_samples_max_age: 168h
  chunk_store_config:
    max_look_back_period: 0s
  table_manager:
    retention_deletes_enabled: true
    retention_period: 672h
```

1.

Run Loki Using Docker

bash

Copy code

```
docker run -d --name=loki -p 3100:3100 -v
```

```
$(pwd)/loki-config.yml:/etc/loki/local-config.yml grafana/loki:latest  
-config.file=/etc/loki/local-config.yml
```

2.

4. Set Up Promtail (Log Forwarder)

Create a Promtail Configuration File Save this as `promtail-config.yml`:

yaml

Copy code

```
server:  
  http_listen_port: 9080  
  grpc_listen_port: 0  
  
positions:  
  filename: /tmp/positions.yml  
  
clients:  
  - url: http://localhost:3100/loki/api/v1/push  
  
scrape_configs:  
  - job_name: system  
    static_configs:  
      - targets:  
          - localhost  
        labels:  
          job: apache-tomcat  
          __path__: /opt/tomcat/logs/*.log
```

1.

Run Promtail Using Docker

bash

Copy code

```
docker run -d --name=promtail -v  
$(pwd)/promtail-config.yml:/etc/promtail/config.yml -v  
/opt/tomcat/logs:/opt/tomcat/logs grafana/promtail:latest  
-config.file=/etc/promtail/config.yml
```

2.

5. Set Up Prometheus

Create a Prometheus Configuration File Save this as `prometheus.yml`:

yaml

Copy code

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'loki'
    static_configs:
      - targets: ['localhost:3100']
```

1.

Run Prometheus Using Docker

bash

Copy code

```
docker run -d --name=prometheus -p 9090:9090 -v
$(pwd)/prometheus.yml:/etc/prometheus/prometheus.yml
prom/prometheus:latest
```

2.

6. Set Up Grafana

Run Grafana Using Docker

bash

Copy code

```
docker run -d --name=grafana -p 3000:3000 grafana/grafana
```

1.

2. **Access Grafana** Visit `http://<your-server-ip>:3000`. The default credentials are:

- **Username:** admin
- **Password:** admin

3. **Add Loki as a Data Source**

- Navigate to **Configuration > Data Sources > Add Data Source**.
- Select **Loki** and set the URL to `http://<your-server-ip>:3100`.

4. **Create a Dashboard**

- Import a pre-made Loki dashboard or create a custom one for Apache Tomcat logs.

7. Verify Setup

- **Prometheus:** Visit <http://<your-server-ip>:9090> to verify metrics are collected.
- **Grafana:** Check your dashboard to view the Apache Tomcat logs in real-time.

This setup captures and monitors logs from Apache Tomcat using Prometheus, Loki, and Grafana. If you encounter issues, let me know!

4o

You said:

but apache tomcat server is another ip means another pc and i am creating setup in my pc to monitoring then give me proper set up then another pc ip 192.168.6.51:8586 this ip and port in another pc this is running in apache tomcat then proper guide me set up using this ip all set up in my pc and you properly guide me with setup in which pc my pc ip is 192.168.6.108 ok

ChatGPT said:

ChatGPT

Memory updated

To set up monitoring for an Apache Tomcat server running on a different PC, follow the instructions below. We will distinguish between actions required on **your PC (192.168.6.108)** and the **Tomcat server PC (192.168.6.51:8586)**.

Setup on the Apache Tomcat Server (192.168.6.51:8586)

1. Install Promtail (Log Forwarder)

1. Download and Configure Promtail

On the Tomcat server, install Docker:

bash

Copy code

```
sudo apt update && sudo apt install -y docker.io
sudo systemctl enable docker
sudo systemctl start docker
```

```
sudo usermod -aG docker $USER
```

○

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

yaml

Copy code

```
server:
```

```
  http_listen_port: 9080
```

```
  grpc_listen_port: 0
```

```
positions:
```

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

```
clients:
```

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

```
scrape_configs:
```

```
  - job_name: apache-tomcat
```

```
    static_configs:
```

```
      - targets:
```

```
        - localhost
```

```
      labels:
```

```
        job: tomcat-logs
```

```
        __path__: /path/to/tomcat/logs/*.log
```

- Replace `/path/to/tomcat/logs` with the actual path to Tomcat's logs (e.g., `/opt/tomcat/logs/`).

Run Promtail

bash

Copy code

```
docker run -d --name=promtail -v
```

```
$(pwd)/promtail-config.yml:/etc/promtail/config.yml -v
```

```
/opt/tomcat/logs:/opt/tomcat/logs grafana/promtail:latest
```

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

2.

Setup on the Monitoring PC (192.168.6.108)

1. Install and Configure Loki

Create a Loki Configuration File Save this as `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
    max_chunk_age: 1h
    chunk_target_size: 1048576
    chunk_retain_period: 30s
  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/index
      cache_location: /loki/boltdb-cache
      shared_store: filesystem
    filesystem:
      directory: /loki/chunks
  limits_config:
    enforce_metric_name: false
```



```
    reject_old_samples: true
    reject_old_samples_max_age: 168h
chunk_store_config:
    max_look_back_period: 0s
table_manager:
    retention_deletes_enabled: true
    retention_period: 672h
```

1.

Run Loki

bash

Copy code

```
docker run -d --name=loki -p 3100:3100 -v
$(pwd)/loki-config.yml:/etc/loki/local-config.yml grafana/loki:latest
-config.file=/etc/loki/local-config.yml
```

2.

2. Install and Configure Prometheus

Create a Prometheus Configuration File Save this as `prometheus.yml`:

yaml

Copy code

```
global:
    scrape_interval: 15s

scrape_configs:
  - job_name: 'loki'
    static_configs:
      - targets: ['localhost:3100']
```

1.

Run Prometheus

bash

Copy code

```
docker run -d --name=prometheus -p 9090:9090 -v
$(pwd)/prometheus.yml:/etc/prometheus/prometheus.yml
prom/prometheus:latest
```

2.

3. Install and Configure Grafana

Run Grafana

bash

Copy code

```
docker run -d --name=grafana -p 3000:3000 grafana/grafana
```

- 1.
 2. **Access Grafana**
 - Open <http://192.168.6.108:3000> in your browser.
 - Log in with the default credentials ([admin/admin](#)) and set a new password.
 3. **Add Loki as a Data Source**
 - Navigate to **Configuration > Data Sources > Add Data Source**.
 - Select **Loki** and set the URL to <http://192.168.6.108:3100>.
 4. **Create a Dashboard**
 - Use a pre-made Loki dashboard or create a custom one for Apache Tomcat logs.
-

Verify Setup

On the Tomcat Server: Check if Promtail is running properly.

bash

Copy code

```
docker logs promtail
```

- 1.
2. **On the Monitoring PC:** Verify that Loki and Prometheus are receiving data:
 - Loki: Visit <http://192.168.6.108:3100/metrics>
 - Prometheus: Visit <http://192.168.6.108:9090>
3. **On Grafana:** Ensure logs are visible on your dashboard.

Let me know if you encounter any issues!