

Linux下安装prometheus & grafana

1 安装prometheus

1.1 下载prometheus

1.2 安装

1.3 启动prometheus

1.4 访问prometheus

2 安装grafana

2.1 下载grafana

2.2 安装

2.3 启动grafana

2.4 访问grafana

3 安装kafka_exporter

3.1 安装

3.2 解压自选目录下，并重命

3.3 启动

3.4 浏览器访问

4 Prometheus配置

3.1 进入prometheus安装目录，编辑prometheus.yml

3.2 重启Prometheus

1 安装prometheus

1.1 下载prometheus

下载地址

<https://prometheus.io/download/#prometheus>

下载

```
wget https://github.com/prometheus/prometheus/releases/download/v2.42.0/prometheus-2.42.0.linux-amd64.tar.gz
```

1.2 安装

```
# 新建目录，并进入目标目录  
mkdir -p /middleware/prometheus && cd /middleware/prometheus
```

```
1 # 解压  
2 tar -vxzf /root/prometheus-2.42.0.linux-amd64.tar.gz -C /middleware/prometheus  
3  
4 cd /middleware/prometheus  
5 mv prometheus-2.42.0.linux-amd64 prometheus
```

1.3 启动prometheus

```
1 前台启动prometheus  
2 cd /middleware/prometheus  
3 ./prometheus --config.file=prometheus.yml
```

后台启动prometheus，并且重定向输入日志到当前目录的prometheus.out

```
nohup ./prometheus --config.file=prometheus.yml >> /middleware/prometheus/prometheus/prometheus.out 2>&1 &
```

1.4 访问prometheus

prometheus启动完后，监听端口为9090

<http://hadoop001:9090/>

2 安装grafana

2.1 下载grafana

下载地址

```
1 https://grafana.com/
2 https://grafana.com/grafana/download?pg=get&plcmt=自我管理-盒子1-cta1&edition=oss
```

下载

```
 wget https://dl.grafana.com/oss/release/grafana-9.4.2.linux-amd64.tar.gz
```

2.2 安装

```
1 # 新建目录，并进入目标目录
2 mkdir -p /middleware/grafana && cd /middleware/grafana
3
4 # 解压
5 tar -vxzf /middleware/grafana-9.4.2.linux-amd64.tar.gz -C /middleware/grafana
6
7 cd /middleware/grafana
8 mv grafana-9.4.2 grafana
```

2.3 启动grafana

grafana默认的配置文件在\$GRAFANA_HOME/conf/defaults.ini 中，该文件中的内容不要修改。复制一份，命名为grafana.ini；自定义的配置文件通过 --config 来指定加载路径

```
1 cp /middleware/grafana/grafana/conf/defaults.ini /middleware/grafana/grafana/conf/grafana.ini
```

后台启动grafana

Plain Text | 复制代码

```
1 nohup /middleware/grafana/grafana/bin/grafana-server \
2 -config "/middleware/grafana/grafana/conf/grafana.ini" \
3 -homepath "/middleware/grafana/grafana" \
4 -pidfile "/middleware/grafana/grafana.pid" web \
5 >> /middleware/grafana/grafana.out 2>&1 &
```

```
nohup /opt/industry_cloud/kafka_exporter/grafana/bin/grafana-server \
-config "/opt/industry_cloud/kafka_exporter/grafana/conf/grafana.ini" \
-homepath "/opt/industry_cloud/kafka_exporter/grafana" \
-pidfile "/opt/industry_cloud/kafka_exporter/grafana/grafana.pid" web \
>> /opt/industry_cloud/kafka_exporter/grafana/grafana.out 2>&1 &
```

- config: 自定义的配置文件的路径
- homepath: homepath的路径, 否则程序启动不了
- pidfile: pid文件的路径

2.4 访问grafana

grafana启动完后, 监听端口为3000; 默认的用户名和密码是: admin/admin

<http://192.168.10.223:3000/>

参考资料

Plain Text | 复制代码

```
1 https://blog.csdn.net/huangjinjin520/article/details/129311778
2 https://blog.csdn.net/qq_43386944/article/details/122727712
3 https://www.cnblogs.com/ezops/p/16607845.html
```

3 安装kafka_exporter

3.1 安装

kafka_exporter 用来收集 Topic, Broker, ConsumerGroup 的相关信息，可以无缝对接 Prometheus 和 Grafana，使用起来比较方便。

安装配置kafka_exporter **注：1个kafka集群只需要1个exporter，在集群上的任意1台服务器部署。**

安装包下载，也可以wget在线下载，这里我下载到了home目录下

```
Plain Text | 复制代码  
1 wget https://github.com/danielqsj/kafka_exporter/releases/download/v1.2.0/kafka_exporter-1.2.0.linux-amd64.tar.gz
```

3.2 解压自选目录下，并重命

```
Plain Text | 复制代码  
1 # tar -zxf kafka_exporter-1.2.0.linux-amd64.tar.gz  
2 # mv kafka_exporter-1.2.0.linux-amd64/ kafka_exporter
```

3.3 启动

前台启动，默认监听端口9308

```
Plain Text | 复制代码  
1 ./kafka_exporter --kafka.server=hadoop001:9092
```

后台启动

```
Plain Text | 复制代码  
1 nohup ./kafka_exporter --kafka.server=hadoop001:9092
```

3.4 浏览器访问

<http://hadoop001:9308>

Kafka Exporter

[Metrics](#)

也可以通过<http://hadoop001:9308/metrics>, 看抓取的指标信息。

4 Prometheus配置

3.1 进入prometheus安装目录，编辑prometheus.yml

```
1 # vim prometheus.yml      //加入prometheus监控中, targets可配置多个监控目标
2   - job_name: 'kafka'
3     static_configs:
4       - targets: ['hadoop001:9308']
```

3.2 重启Prometheus

```
1 # systemctl restart prometheus
```

浏览器查看监控信息：<http://172.20.32.218:9090/targets>

离线下载json

<https://grafana.com/grafana/dashboards/7589-kafka-exporter-overview/>

