

## prometheus监控部署

基于 <http://wiki.tongdun.me/pages/viewpage.action?pageId=36279058> 于天策6.0.1版本实践经验

支持的组件 <https://prometheus.io/docs/instrumenting/exporters/>

部署两种方案(基于hostname客户变更ip仅需修改/etc/hosts映射):

### 1. 基于ip配置:

```
#获取IP
`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}' |cut -d/ -f1`
```

### 2. 基于hostname配置:

```
#1. 修改/etc/hosts映射关系, 以td01主机名为例: 10.57.17.223 td01
vi /etc/hosts
#2. 添加如下内容(多台集群则每台映射均需规划配置)
10.57.17.223 td01
#3. 修改主机名
hostnamectl set-hostname td01
#或(两种方式等价, 上面命令会直接修改 /etc/hostname 文件)
vim /etc/hostname
td01
#4. 测试, ping通且显示当前机器ip
ping td01

#获取hostname
`hostname`
```

## 主机监控信息

1、Node\_export node\_export 用来监控主机信息, 每台机器上都要部署一个该组件 (1) 解压【node\_exporter-1.0.1.linux-amd64.zip】压缩文件到指定目录

```
unzip node_exporter-1.0.1.linux-amd64.zip
```

(2) 可选在sh脚本中更改发布的端口号(对应 --web.listen-address=":9100")

```
#直接替换
sed -i "s#:9001#:9001#g" node_exporter.sh

#或
vim node_exporter.sh
#根据需要默认还是修改端口9100内容
--web.listen-address=":9100"
```

(3) 进入目录, 执行命令: sh node\_exporter.sh [start | stop | restart], 表示启动、停止和重启。

```
sh node_exporter.sh start
```

(4) 启动成功后, 可通过访问 `http://{ip}:9100/metrics` 查看监控信息, 有监控指标输出即为部署成功

```
curl "http://`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}'
|cut -d/ -f1`:9100/metrics"
#或
curl "http://`hostname`:9100/metrics"
```

(5) 参考资料: [https://github.com/prometheus/node\\_exporter](https://github.com/prometheus/node_exporter)

2、Mysql\_export mysql\_export 用来监控 Mysql 服务器的信息, 每台部署 Mysql 的机器上都要部署一个该组件 (1) 解压【mysql\_exporter-0.12.1.linux-amd64.zip】压缩文件到指定目录

```
unzip mysql_exporter-0.12.1.linux-amd64.zip
```

(2) 进入目录, 修改 my.cnf 文件, 填写 Mysql 数据库的用户名、密码

```
sed -i "s#user=xxx#user=kratos#g" my.cnf
sed -i "s#password=xxx#password=123456#g" my.cnf
cat my.cnf
```

(3) 可选在sh脚本中更改发布的端口号(对应 `--web.listen-address=":9104"`)

```
#直接替换
sed -i "s#:9104#9104#g" mysql_export.sh

#或
vim mysql_export.sh
#根据需要默认还是修改端口9104内容
--web.listen-address=":9104" &
```

(4) 执行命令: `sh mysql_export.sh [start | stop | restart]`, 表示启动、停止和重启。

```
sh mysql_export.sh start
```

(5) 启动成功后, 可通过访问 `http://{ip}:9104/metrics` 查看监控信息, 有监控指标输出即为部署成功

```
curl "http://`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}'
|cut -d/ -f1`:9104/metrics"
或
curl "http://`hostname`:9104/metrics"
```

(6) 参考资料: [https://github.com/prometheus/mysql\\_exporter](https://github.com/prometheus/mysql_exporter)

3、Kafka\_export kafka\_export 用来监控 kafka 服务器信息, 每台部署 Kafka 的机器上都要部署一个该组件 (1) 解压【kafka\_exporter-1.2.0.linux-amd64.zip】压缩文件到指定目录

```
unzip kafka_exporter-1.2.0.linux-amd64.zip
```

(2) 通过修改 kafka\_export.sh 脚本来配置要监控的kafka机器, 或者是修改端口号

```
#基于ip一键替换
sed -i 's#kafka.server=10.58.10.73:9092#kafka.server=`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}' |cut -d/ -f1`:9092#g'
kafka_exporter.sh
grep "kafka.server" kafka_exporter.sh

#或基于hostname
sed -i 's#kafka.server=10.58.10.73:9092#kafka.server=`hostname`:9092#g'
kafka_exporter.sh
grep "kafka.server" kafka_exporter.sh

#或
vim kafka_exporter.sh
修改对应启动脚本nohup ./kafka_exporter --web.listen-address=:9308 --
kafka.version=1.1.0 --kafka.server=10.58.10.73:9092
```

(3) 进入目录，执行命令：sh kafka\_exporter.sh [start | stop | restart]，表示启动、停止和重启。

```
sh kafka_exporter.sh start
```

(4) 启动成功后，可通过访问 http://{ip}:9308/metrics 查看监控信息，有监控指标输出即为部署成功

```
curl "http://`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}' |cut -d/ -f1`:9308/metrics"
或
curl "http://`hostname`:9308/metrics"
```

(5) 参考资料：[https://github.com/danielqsj/kafka\\_exporter](https://github.com/danielqsj/kafka_exporter)

4、Elasticsearch\_export elasticsearch\_export 用来监控 es 服务器信息，每台部署 es 的机器上都要部署一个该组件 (1) 解压【elasticsearch\_exporter-1.0.4rc1.linux-amd64.zip】压缩文件到指定目录

```
unzip elasticsearch_exporter-1.0.4rc1.linux-amd64.zip
```

(2) 通过修改 elasticsearch\_export.sh 脚本来配置要监控的es机器，或者是修改端口号

```
#基于ip一键替换
sed -i 's#10.57.34.45#`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}' |cut -d/ -f1`:#g' elasticsearch_export.sh
grep "es.uri" elasticsearch_export.sh

#或基于hostname
sed -i 's#10.57.34.45#`hostname`:#g' elasticsearch_export.sh
grep "es.uri" elasticsearch_export.sh

#或
vim elasticsearch_export.sh
#修改对应启动脚本
--web.listen-address=:9109 --es.uri http://10.57.34.45:9200 &

###注意：
#如果es有配置用户名和密码，需对应增加用户名及密码信息
--web.listen-address=:9109 --es.uri http://用户:密码@10.57.34.45:9200 &
```

(3) 进入目录，执行命令：sh elasticsearch\_export.sh [start | stop | restart]，表示启动、停止和重启。

```
sh elasticsearch_export.sh start
```

(4) 启动成功后，可通过访问 http://{ip}:9109/metrics 查看监控信息，有监控指标输出即为部署成功

```
curl "http://`ip` | grep inet | grep -v 127.0.0.1 | grep -v inet6 | awk '{print $2}'  
| cut -d/ -f1`:9109/metrics"  
#或  
curl "http://`hostname`:9109/metrics"
```

(5) 参考资料：[https://github.com/justwatchcom/elasticsearch\\_exporter](https://github.com/justwatchcom/elasticsearch_exporter)

5、Aerospike\_export aerospike\_export 用来监控 aerospike 服务器信息，每台部署 aerospike 的机器上都要部署一个该组件 (1) 解压【aerospike-prometheus-exporter-1.0.0.zip】压缩文件到指定目录

```
unzip aerospike-prometheus-exporter-1.0.0.zip
```

(2) 修改 /etc/aerospike-prometheus-exporter/ape.toml 配置文件中db\_host，或直接使用下面shell命令

```
#基于ip一键替换  
sed -i 's#db_host=".*"#db_host="\`ip` | grep inet | grep -v 127.0.0.1 | grep -v  
inet6 | awk '{print $2}' | cut -d/ -f1`:9109/metrics"#g' ./etc/aerospike-prometheus-  
exporter/ape.toml  
grep "db_host" ./etc/aerospike-prometheus-exporter/ape.toml  
  
#或基于hostname  
sed -i 's#db_host=".*"#db_host="\`hostname`:9109/metrics"#g' ./etc/aerospike-prometheus-  
exporter/ape.toml  
grep "db_host" ./etc/aerospike-prometheus-exporter/ape.toml  
  
#或  
vim ./etc/aerospike-prometheus-exporter/ape.toml  
#修改对应启动脚本  
db_host=
```

(3) 进入目录，执行命令：sh aerospike\_export.sh [start | stop | restart]，表示启动、停止和重启。

```
sh aerospike_export.sh start
```

(4) 启动成功后，可通过访问 http://{ip}:9145/metrics 查看监控信息，有监控指标输出即为部署成功，或执行shell命令

```
curl "http://`ip` | grep inet | grep -v 127.0.0.1 | grep -v inet6 | awk '{print $2}'  
| cut -d/ -f1`:9145/metrics"  
#或  
curl "http://`hostname`:9145/metrics"
```

(5) 参考资料: • aerospike+prometheus部署: <https://www.aerospike.com/docs/tools//monitors/tack/index.html> • aerospike输出指标解释: <https://www.aerospike.com/docs/reference/metrics/>

6、Process\_export process\_export 用来监控进程信息, 该进程会定时执行 ps 命令来监控进程信息, 通常我们需要在部署了 zk、kafka、es、as 的每台机器上部署该组件 (1) 解压【process-exporter-0.5.0.linux-amd64.zip】压缩文件到指定目录

```
unzip process-exporter-0.5.0.linux-amd64.zip
```

(2) 进入目录, 修改 process\_export.yml 配置文件, 配置上在当前机器上你想要监控的进程。 • name: 在grafana面板上显示的进程名称 • cmdline: 系统的进程名称, 相当于 ps -ef | grep xxx, xxx 就是该属性的值

```
#根据需要替换版本号
sed -i "s#kafka1.1.0#kafka_2.11-1.0.0#g" process_export.yml
sed -i "s#zookeeper-3.4.8#zookeeper-3.4.8#g" process_export.yml
sed -i "s#elasticsearch-5.0.2#elasticsearch-5.6.8#g" process_export.yml
cat process_export.yml

#根据需要新增
- name: "aerospike"
  cmdline:
  - 'aerospike'
```

(3) 可通过修改 process\_export.sh 脚本来修改端口号(--web.listen-address=":9256")

```
#直接替换
sed -i "s#9256#9256#g" process_export.sh

#或
vim process_export.sh
#修改对应启动脚本
--web.listen-address=":9256" -config.path process_export.yml &
```

(4) 执行命令: sh process\_export.sh [start | stop | restart], 表示启动、停止和重启。

```
sh process_export.sh start
```

(5) 启动成功后, 可通过访问 http://{ip}:9256/metrics 查看监控信息, 有监控指标输出即为部署成功

```
curl "http://`ip a|grep inet |grep -v 127.0.0.1 |grep -v inet6|awk '{print $2}'
|cut -d/ -f1`:9256/metrics"
#或
curl "http://`hostname`:9256/metrics"
```

(6) 参考资料: <https://github.com/ncabatoff/process-exporter>

Prometheus 部署 普罗米修斯 (Prometheus) 相当于一个数据库, 它会定时对我们配置的job去拉取监控数据并保存。所以整个银河监控只部署一个普罗米修斯实例即可。 (1) 解压【prometheus-2.19.2.linux-amd64.zip】压缩文件到指定目录

```
unzip prometheus-2.19.2.linux-amd64.zip
```

(2) 进入目录，修改 prometheus.yml 配置文件，在 scrape\_configs 配置上我们刚才部署的组件（包括上面的监控程序和我们银河服务的实例），按照原来的配置文件你只需要修改各个job的IP即可。• 以前置服务为例，一个job就代表一个服务，job\_name不要改，只需要修改targets对应的ip、port即可。  
- targets: ['10.58.10.73:8090'] • 比如 node\_export 部署在了两台机器上，两个ip公用一个job\_name，这里没有配置 metrics\_path，默认路径是 /metrics - targets: ['10.58.10.73:9100','10.57.34.45:9100']

```
#根据需要替换为ip或hostname
sed -i "s#10.57.34.45:9093#10.57.34.114:9093#g" prometheus.yml
sed -i "s#10.58.10.73:8090#10.57.34.114:8070#g" prometheus.yml
sed -i "s#10.58.10.73:9088#10.57.34.114:8077#g" prometheus.yml
sed -i "s#10.57.17.148:9077#10.57.34.114:8075#g" prometheus.yml
sed -i
"s#10.58.10.73:9100','10.57.34.45:9100#10.57.34.114:9100','10.57.17.223:9100#g"
prometheus.yml
sed -i "s#10.58.10.73:9308#10.57.17.223:9308#g" prometheus.yml
sed -i "s#10.57.34.45:9104#10.57.17.223:9104#g" prometheus.yml
sed -i "s#10.57.34.45:9256#10.57.17.223:9256#g" prometheus.yml
sed -i "s#10.57.34.45:9109#10.57.17.223:9109#g" prometheus.yml
sed -i "s#10.57.34.45:9145#10.57.17.223:9145#g" prometheus.yml
sed -i "s#10.57.241.21:8071#10.57.34.114:8071#g" prometheus.yml
sed -i "s#10.58.10.73:9004#10.57.34.114:8074#g" prometheus.yml
sed -i "s#10.58.10.73:9002#10.57.34.114:9002#g" prometheus.yml
sed -i "s#10.59.75.138:8088#10.57.34.114:8072#g" prometheus.yml
cat prometheus.yml
```

(3) 可通过修改 prometheus.sh 脚本来修改端口号(--web.listen-address=":9091")

```
#直接替换
sed -i "s#.:9091#.:9091#g" prometheus.sh

#或
vim prometheus.sh
#修改对应启动脚本
--web.listen-address=":9091" &
```

(4) 修改完 prometheus.yml 以后，执行命令：sh prometheus.sh [start | stop | restart]，表示启动、停止和重启。

```
sh prometheus.sh start
```

(5) 启动成功后，可通过访问 http://{ip}:9091/graph 查看普罗米修斯主页面

```
curl "http://`ip a | grep inet | grep -v 127.0.0.1 | grep -v inet6 | awk '{print $2}'`
| cut -d/ -f1`:9091/metrics"
```

(6) 查看我们配置的 job 是否正常拉取，通过选择 Status -> Targets 查看，这里有我们在配置文件里配置的所有job，up 即为正常，down 即为失败

Grafana 部署 Prometheus 相当于一个数据库，用来保存收集到的指标数据；而 Grafana 则是专门用来展示数据的，它提供了很多丰富的面板配置，帮助我们进行数据的可视化（1）解压【grafana-7.0.6.zip】压缩文件到指定目录

```
unzip grafana-7.0.6.zip
```

(2) Grafana默认端口为 3000，要修改端口，进入 conf 目录修改 defaults.ini 文件

```
sed -i "s#http_port = 3000#http_port = 3000#g" conf/defaults.ini  
grep "http_port = " conf/defaults.ini
```

(3) 执行命令：sh grafana.sh [start | stop | restart]，表示启动、停止和重启。

```
sh grafana.sh start
```

(4) 启动成功后，访问 <http://{ip}:3000>，进入Grafana主页面，账号密码为：admin / Td@123456

```
http://10.57.34.114:3000
```

(5) 登录后

需要配置 Prometheus 数据源，选择左侧目录 Data Sources

点击Configuration标签 -> Data Sources标签(新增或基于历史数据配置)

1. 基于历史数据配置: 点击选择 Prometheus
2. HTTP -> URL 配置为: <http://10.57.34.114:9091> (安装的Prometheus 地址及端口)
3. 点击 save & test
4. 点击Dashboards标签 -> Manage标签 -> 整体监控
5. 配置完数据源后，选择左侧目录 Manager，选择面板查看即可

Alertmanager 部署 普罗米修斯将数据采集和告警通知分成了两个模块。报警规则配置在普罗米修斯上（警报规则文件），然后发送报警信息到 AlertManger，AlertManager来管理这些报警信息，同时提供了聚合分组、告警抑制等高级功能，还支持通过 Email、WebHook 等多种方式发送告警消息提示。

(1) 解压【alertmanager-0.21.0.linux-amd64.zip】压缩文件到指定目录

```
unzip alertmanager-0.21.0.linux-amd64.zip
```

(2) 进入目录，修改 alertmanager.yml 配置文件，该文件用于配置告警通知，这里提供的是163邮件的通知的样例。你需要在这里配置上邮箱的 SMTP 服务器配置

```
smtp_smarthost: 'smtp.163.com:25'  
smtp_from: 'yunuotianming@163.com'  
smtp_auth_username: 'yunuotianming@163.com'  
smtp_auth_password: 'xxxxxooooo'
```

• 在这里配置上告警通知的接收人。mail-error 表示严重等级的告警通知（比如服务宕机），mail-warning 表示紧急等级的告警通知（比如内存使用快满了）

```
receivers:

- name: 'mail-error'
  email_configs:
    - to: xxx@163.com

- name: 'mail-warning'
  email_configs:
    - to: xxx@163.com
```

(3) 默认端口为 9093，可通过修改sh脚本修改端口

```
sed -i "#9093#9093#" alertmanager.sh
```

(4) 如果修改了端口，需要在普罗米修斯的配置文件（prometheus.yml）中对应修改端口，然后重启普罗米修斯，使得普罗米修斯和 Alertmanager 可以正常通信。

(5) 执行命令：sh alertmanager.sh [start | stop | restart]，表示启动、停止和重启。

```
sh alertmanager.sh start
```

(6) 启动成功后，访问 <http://{ip}:9093>，进入Alertmanager主页面，当监控数据触发告警后，我们可以在主页面看到告警信息

```
http://10.57.34.114:9093
```

(7) 部署普罗米修斯的时候已经内置了告警规则，所以告警规则不用我们再去配置了，我们可以进入普罗米修斯的主页面，查看当前已配置的告警规则 <http://{普罗米修斯部署的IP}:9091/alerts>

```
http://10.57.34.114:9091/alerts
```

## 注意事项

针对集群下"硬件资源监控 /Aerospike 监控大盘"集群节点数显示异常问题

1. 点击"硬件资源监控 /Aerospike 监控大盘"右上角的"Dashboard Settings"设置按钮
2. 点击左侧菜单栏的"Variables"
3. 点击"Variable"中的"cluster"
4. "Selection Options"中的"Multi-value", "Include All option"关闭
5. 点击"Update"即可

针对preserver日志打印频繁问题

修改 AuthFilter 类



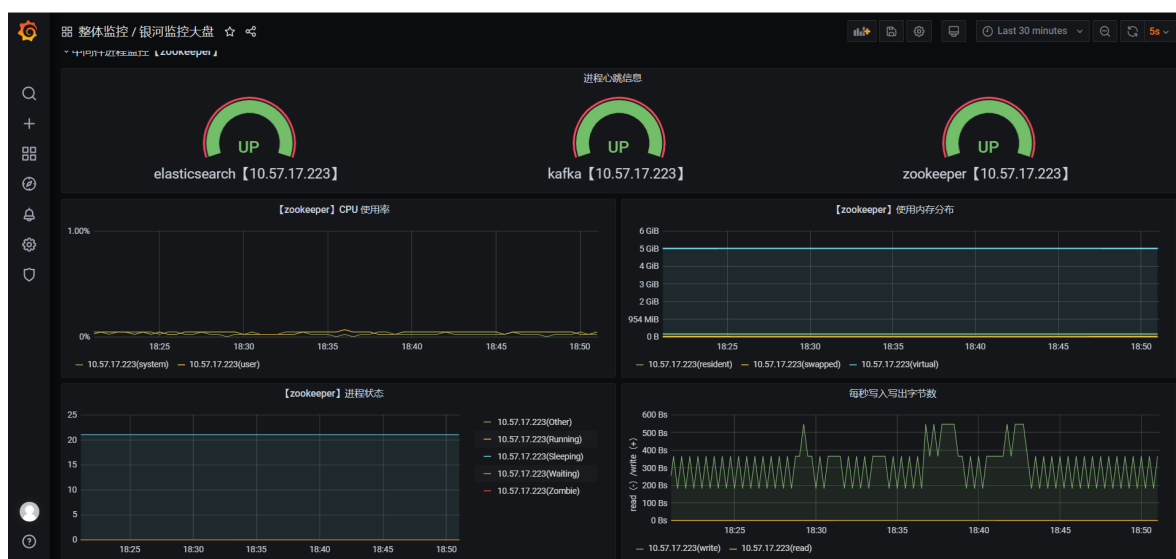
```
private static final List<String> DONT_PRINT_RESP_LIST = new ArrayList<>();
static {
    DONT_PRINT_RESP_LIST.add("/metrics");
    DONT_PRINT_RESP_LIST.add("/api/scriptField/all");
    DONT_PRINT_RESP_LIST.add("/api/systemField/all");
    DONT_PRINT_RESP_LIST.add("/api/application/all");
    //新增过滤 prometheus 打印行为
    DONT_PRINT_RESP_LIST.add("/api/actuator/prometheus");
}
```

## 监控系统

大盘: 应用目前需要银河4.0以上, 天策需要6.0以上(目前天策6.0没有TPS RT埋点)



大盘: 可以查看zookeeper的内存状态, 读写状态



as: 查看内存使用状态(注意: 内存不够用了salaxy的指标是不会累计的, 具体需要结合as配置(多少百分比不写入))



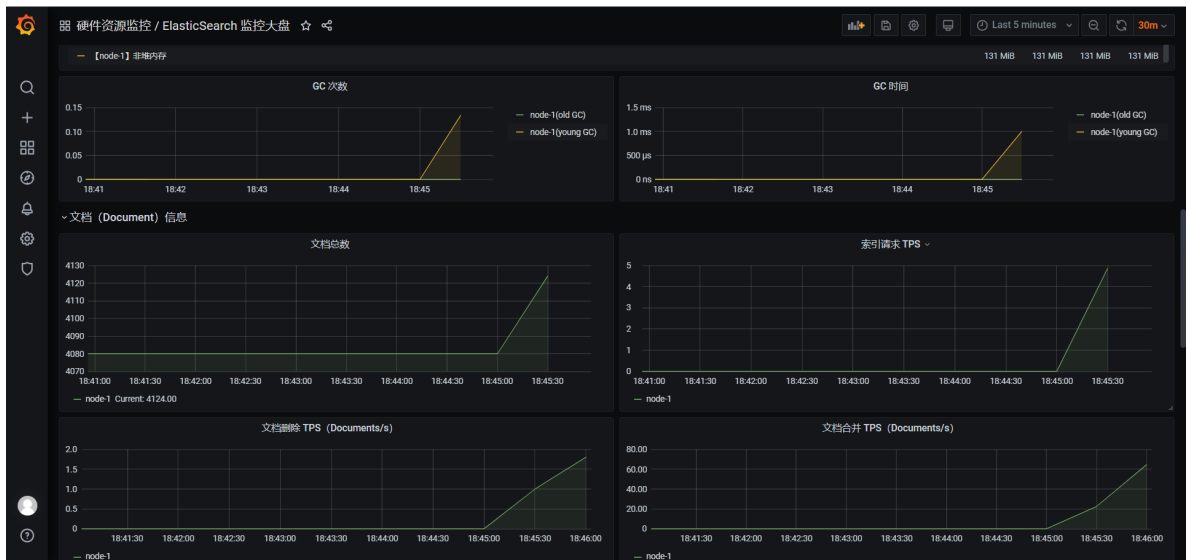
as: 可以查看读写速度



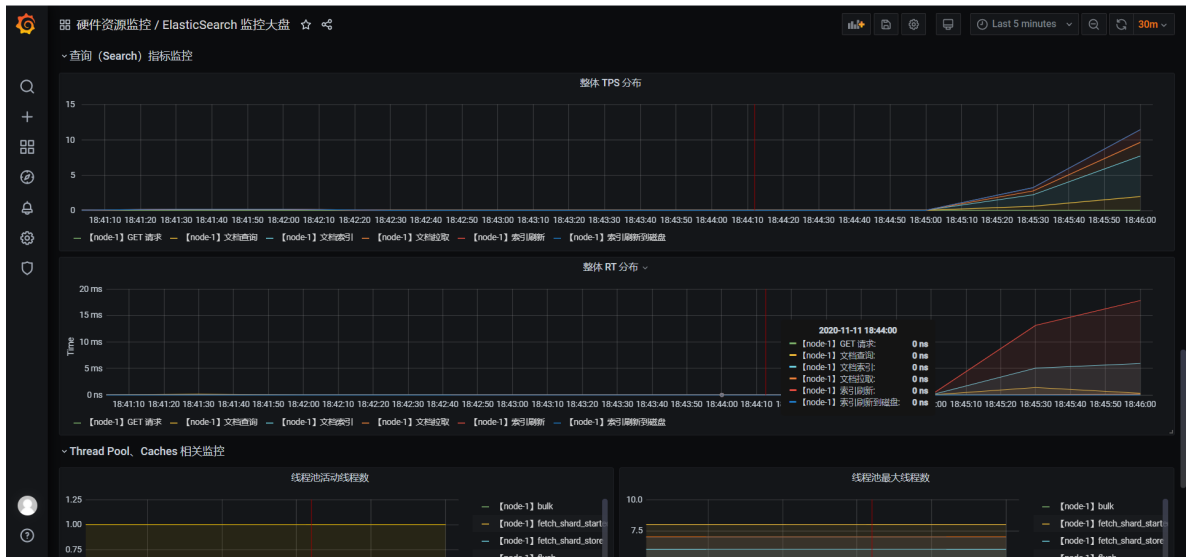
es: 查看es集群状态(green为正常, yellow/red就不健康了, 可以发现目前有9个分片没有被分配到节点上)



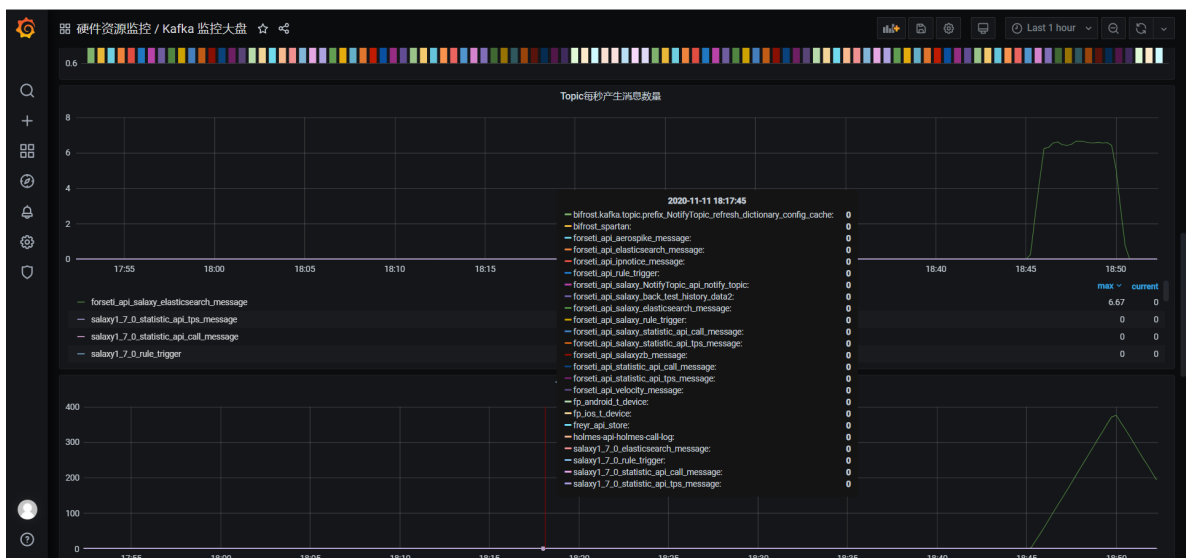
es: gc状况, 以及TPS



es: 每秒请求, 响应耗时



kafka: 产生消息的速度(秒/分钟)



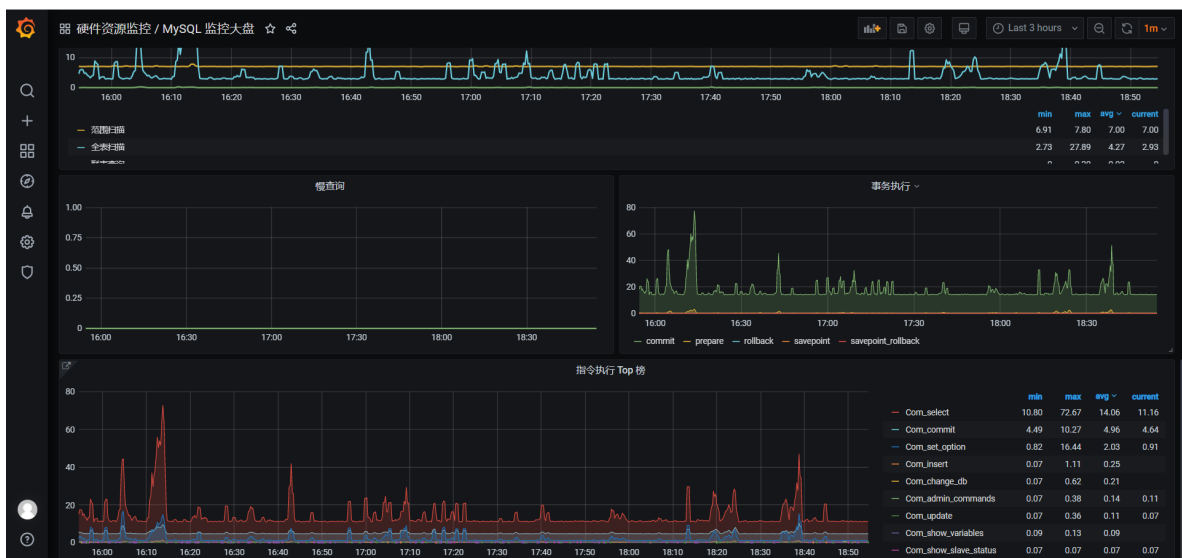
kafka: 关注消息是否产生堆积, 消费速度是否正常



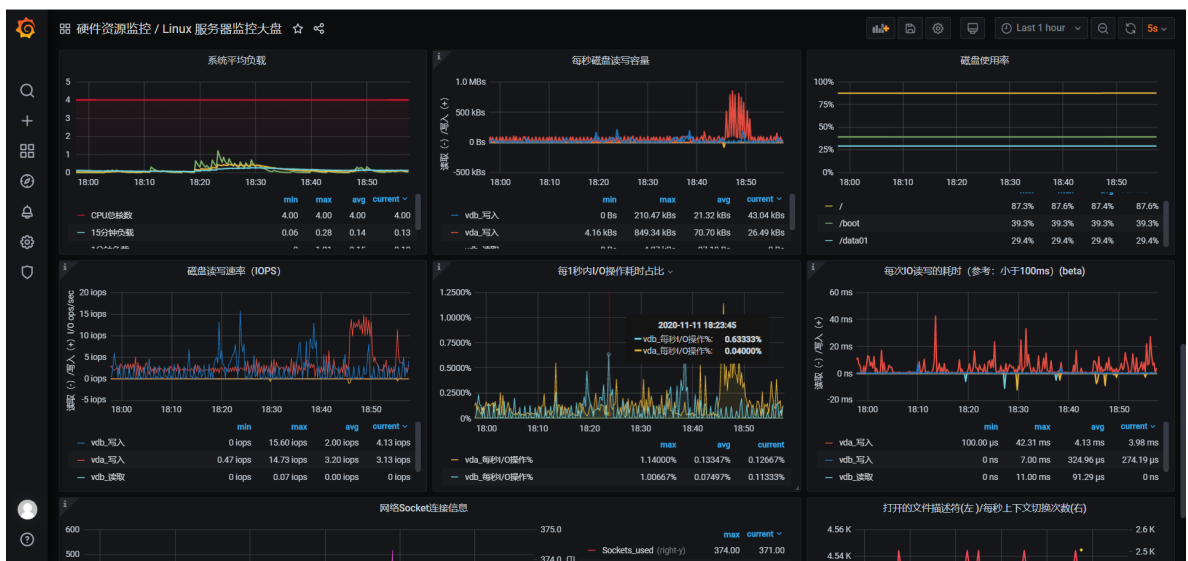
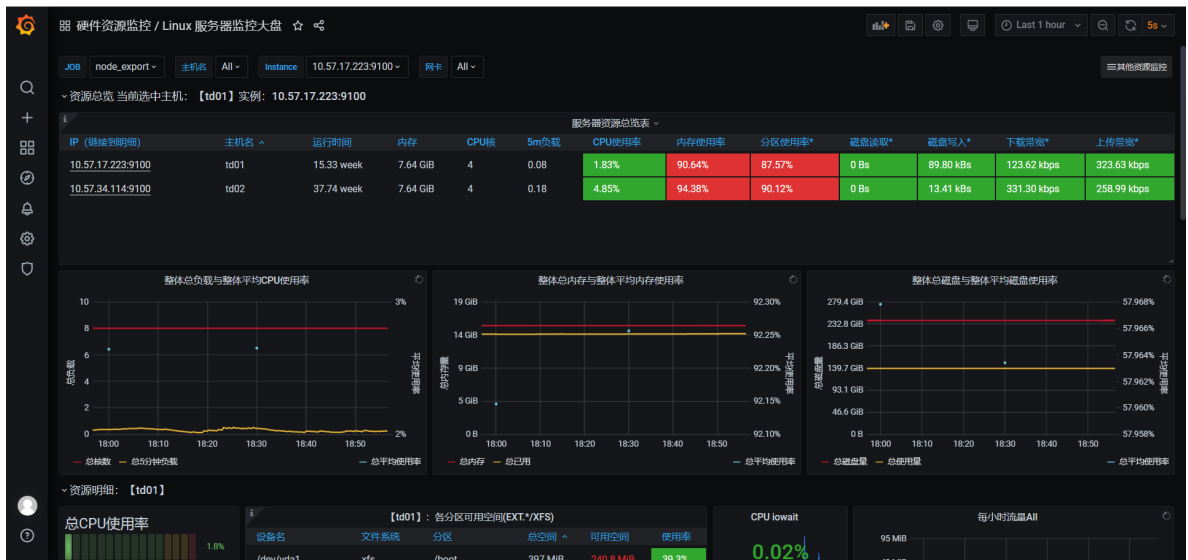
## mysql: 应关注IO及锁的争用情况



## mysql: 关注慢查询



## linux: 服务器性能-> CPU 磁盘空间 IO 带宽



consumer: 对应应用的 cpu 线程状态



consumer: 年轻代老年代内存占用趋势



## consumer: gc情况

