

日志系统部署文档

一、相关部署文件

1、docker-compose.yml (其中相关变量见.env文件)

```
version: "3.7"

services:
  filebeat:
    env_file:
      - .env
    user: root
    container_name: filebeat
    image: docker.elastic.co/beats/filebeat:7.7.1
    restart: always
    volumes:
      - ${filebeat_yml_volume}:/usr/share/filebeat/filebeat.yml
      - ${log_volume}:/filebeat/log/node.INFO
      - ${filebeat_registry_volume}:/usr/share/filebeat/data/registry
    networks:
      - localnet
    depends_on:
      - redis

  redis:
    user: root
    container_name: redis
    image: redis:latest
    restart: always
    ports:
      - ${redis_port}:6379
    command: redis-server /etc/conf/redis.conf
    volumes:
      - ${redis_data_volume}:/data
      - ${redis_conf_volume}:/etc/conf
    privileged: true
    networks:
      - localnet
    environment:
      - TZ=Asia/Shanghai
      - LANG=en_US.UTF-8

#创建docker共享网络
networks:
  localnet:
    external:
      name: localnet
```

2、.env

#配置宿主机filebeat启动配置文件挂载路径。(可以结合具体部署环境自定义路径)

filebeat_yml_volume=/home/logstah/filebeat.docker.yml

#配置宿主机log文件挂载路径，这个需要子豪、杜总那边提供此log文件然后指向便可。(可以结合具体部署环境自定义路径)

log_volume=/home/logstah/log/node.INFO

#配置宿主机filebeat上报index持久化挂载路径。(可以结合具体部署环境自定义路径)

filebeat_registry_volume=/home/logstah/registry

#配置宿主机redis数据持久挂载路径。(可以结合具体部署环境自定义路径)

redis_data_volume=/home/logstah/redis/data

#配置宿主机redis_conf挂载路径，当前目录下创建redis.conf文件，见下面此文件内容。(可以结合具体部署环境自定义路径)

redis_conf_volume=/home/logstah/redis/conf

#配置宿主机redis映射端口。(可以结合具体部署环境自定义)

redis_port=6379

3、filebeat.docker.yml 这个文件创建到：\${filebeat_yml_volume}目录下，文本内容直接复制下面就可以。

filebeat.inputs:

- type: log

enabled: true

##配置你要收集的日志目录，这里指定的是filebeat容器内部的日志目录可自定义，但是一定要用这个路径去bind到上面docker-compose.yml文件中的这行： -

\${log_volume}:/filebeat/log/node.INFO

paths:

- /filebeat/log/node.INFO

include_lines: ['[[USER]]']

processors:

- drop_fields:

when:

has_fields: ['log_source']

fields: ["input", "@timestamp", "@metadata", "ecs", "host", "agent", "log"]

fields:

log_source: msg-test-1

fields_under_root: true

#控制台打印，本地调试可以打开，注意7.x后只支持配置一个output节点

#output.file:

#path: "/tmp/filebeat"

#filename: filebeat

#上报的redis配置信息

output.redis:

hosts: ["redis:6379"]

```
key: sys_log
db: 0
```

4、**redis.conf** 这个文件创建到: \${redis_conf_volume}目录下, 文本内容直接复制下面就可以。

```
#bind 127.0.0.1
protected-mode no
port 6379
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize no
supervised no
pidfile /var/run/redis_6379.pid
loglevel notice
logfile ""
databases 16
always-show-logo yes

save 900 1
save 300 10
save 60 10000

stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
# The filename where to dump the DB
dbfilename dump.rdb
rdb-del-sync-files no
dir ./
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-diskless-load disabled
repl-disable-tcp-nodelay no
replica-priority 100
acllog-max-len 128
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
replica-lazy-flush no
lazyfree-lazy-user-del no
appendonly no
# The name of the append only file (default: "appendonly.aof")
appendfilename "appendonly.aof"
# appendfsync always
appendfsync everysec
# appendfsync no
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
aof-load-truncated yes
aof-use-rdb-preamble yes
lua-time-limit 5000
slowlog-log-slower-than 10000
```

```
slowlog-max-len 128
latency-monitor-threshold 0
notify-keyspace-events ""
hash-max-ziplist-entries 512
hash-max-ziplist-value 64
list-max-ziplist-size -2
list-compress-depth 0
set-max-intset-entries 512
zset-max-ziplist-entries 128
zset-max-ziplist-value 64
hll-sparse-max-bytes 3000
stream-node-max-bytes 4096
stream-node-max-entries 100
activerehashing yes
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit replica 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60
hz 10
dynamic-hz yes
aof-rewrite-incremental-fsync yes
rdb-save-incremental-fsync yes
jemalloc-bg-thread yes
```

二、部署依赖关系

注意：redis、filebeat、skadi后端指定同一个network

FileBeat依赖Redis。

Skadi后端依赖Redis。

Redis启动成功后FileBeat、Skadi可以按随机顺序起来。