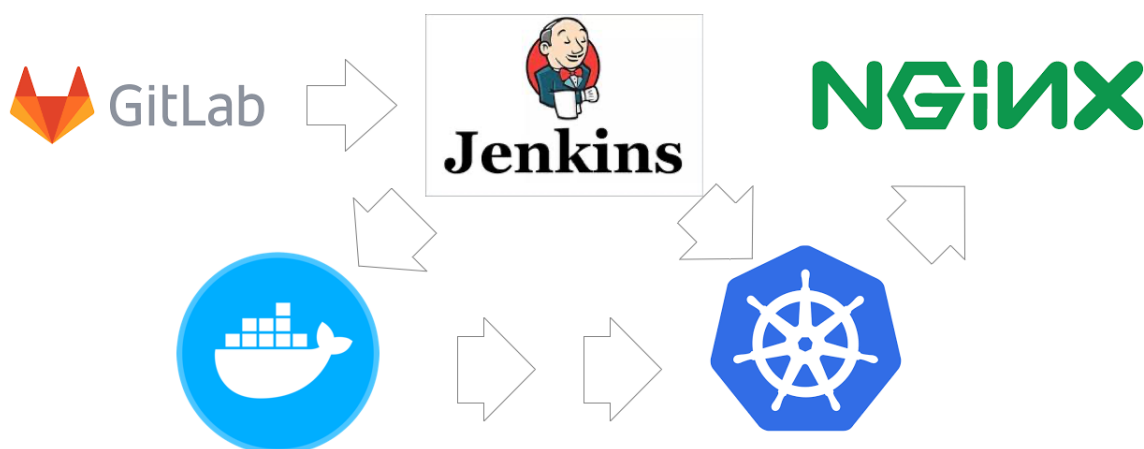


容器虚拟化技术和自动化部署



初识pinpoint

pinpoint是开源在github上的一款APM监控工具，它是用java编写的，Pinpoint是一款全链路分析工具，提供了无侵入式的调用链监控、方法执行详情查看、应用状态信息监控等功能。基于GoogleDapper论文进行的实现，与另一款开源的全链路分析工具Zipkin类似，但相比Zipkin提供了无侵入式、代码维度的监控等更多的特性。Pinpoint支持的功能比较丰富，可以支持如下几种功能：

- 服务拓扑图：对整个系统中应用的调用关系进行了可视化的展示，单击某个服务节点，可以显示该节点的详细信息，比如当前节点状态、请求数量等
- 实时活跃线程图：监控应用内活跃线程的执行情况，对应用的线程执行性能可以有比较直观的了解
- 请求响应散点图：以时间维度进行请求计数和响应时间的展示，拖过拖动图表可以选择对应的请求查看执行的详细情况
- 请求调用栈查看：对分布式环境中每个请求提供了代码维度的可见性，可以在页面中查看请求针对到代码维度的执行详情，帮助查找请求的瓶颈和故障原因。
- 应用状态、机器状态检查：通过这个功能可以查看相关应用程序的其他的一些详细信息，比如CPU使用情况，内存状态、垃圾收集状态，TPS和JVM信息等参数。

架构组成

Pinpoint 主要由 3 个组件外加 Hbase 数据库组成，三个组件分别为：Agent、Collector 和 Web UI。

- Agent组件：用于收集应用端监控数据，无侵入式，只需要在启动命令中加入部分参数即可
- Collector组件：数据收集模块，接收Agent发送过来的监控数据，并存储到HBase
- WebUI：监控展示模块，展示系统调用关系、调用详情、应用状态等，并支持报警等功能

docker官网

```
1 | https://hub.docker.com/r/pinpointdocker/pinpoint-web
2 | https://hub.docker.com/r/pinpointdocker/pinpoint-hbase
3 | https://hub.docker.com/r/pinpointdocker/pinpoint-collector
4 | https://hub.docker.com/r/pinpointdocker/pinpoint-agent
5 | https://hub.docker.com/r/pinpointdocker/pinpoint-quickstart
6 | https://hub.docker.com/r/pinpointdocker/pinpoint-mysql
7 |
8 | docker-github官网
9 | https://github.com/naver/pinpoint-docker
```

基础镜像

```
1 | docker pull zookeeper:3.4
2 | docker pull flink:1.3.1
3 | docker pull pinpointdocker/pinpoint-web
4 | docker pull pinpointdocker/pinpoint-collector:2.1.0
5 | docker pull pinpointdocker/pinpoint-hbase
6 | docker pull pinpointdocker/pinpoint-agent:2.1.0
7 | docker pull pinpointdocker/pinpoint-mysql:2.1.0
8 | docker pull pinpointdocker/pinpoint-quickstart:latest
```

```
1 | docker save \
2 | zookeeper:3.4 \
3 | flink:1.3.1 \
4 | pinpointdocker/pinpoint-web \
5 | pinpointdocker/pinpoint-collector:2.1.0 \
6 | pinpointdocker/pinpoint-hbase \
7 | pinpointdocker/pinpoint-agent:2.1.0 \
8 | pinpointdocker/pinpoint-mysql:2.1.0 \
9 | pinpointdocker/pinpoint-quickstart:latest \
10 | -o pinpoint2.1.0.tar
```

```
1 | docker load -i pinpoint2.1.0.tar
```

注意8000、9997、8080、9994-9996、16010、6123、2181、2888、3888、8081、8079端口不能被占用

安装pinpoint

启动pinpoint

```
1 | 安装pinpoint
2 | unzip pinpoint-docker-2.1.0.zip
3 |
4 | cd pinpoint-docker-2.1.0
5 |
6 | 拉取镜像
7 | docker-compose pull
8 |
9 | 运行镜像
10 | docker-compose up -d
```

WebUI

```
1 | 启动时间大概需要3-10分钟:
2 | http://192.168.198.142:8079/
```

flink

```
1 | http://192.168.198.142:8081/
```

QuickStart

```
1 | http://192.168.198.142:8000/
```

Hbase

```
1 | http://192.168.198.142:16010/
```

agent安装

官网下载地址

```
1 | https://github.com/naver/pinpoint/releases/download/v2.1.0/pinpoint-agent-2.1.0.tar.gz
```

安装

```
1 | tar xzf pinpoint-agent-2.1.0.tar.gz
2 | rm -rf pinpoint-agent-2.1.0.tar.gz
3 | mv pinpoint-agent-2.1.0 pinpoint
```

修改pinpoint.config 文件

```
1 vi /data/pinpoint/profiles/release
2
3 改配置项 profiler.collector.ip=127.0.0.1 为你自己的Collector组件的IP（Pinpoint服务
  器IP）
4
5 profiler.transport.grpc.collector.ip=192.168.198.142
6 profiler.collector.ip=192.168.198.142
```

war包

配置tomcat

```
1 将apache-tomcat-8.5.57.tar.gz上传154节点/opt目录中
2
3 cd /opt
4 tar zxf apache-tomcat-8.5.57.tar.gz
5
6 mv apache-tomcat-8.5.57 tomcat8
7 rm -rf apache-tomcat-8.5.57.tar.gz
8
9 将war包复制到webapps目录
10 cd /data
11 mv /data/skywalking-springmvc.war /opt/tomcat8/webapps/
```

集成pinpoint

```
1 CATALINA_OPTS="$CATALINA_OPTS -javaagent:/data/pinpoint/pinpoint-bootstrap-
  2.1.0.jar"
2 CATALINA_OPTS="$CATALINA_OPTS -Dpinpoint.agentId=jenkinsagent154-1"
3 CATALINA_OPTS="$CATALINA_OPTS -Dpinpoint.applicationName=springmvcdemo"
```

启动tomcat

```
1 启动tomcat8
2 cd /opt/tomcat8/bin
3
4 ./startup.sh
5
6
7 http://192.168.198.154:8080/
8
9 http://192.168.198.154:8080/skywalking-springmvc/
10 http://192.168.198.154:8080/skywalking-springmvc/hello/sayHello.controller
11
12 停止tomcat8
13 ./shutdown.sh
```

jar包

```
1 java -javaagent:/data/pinpoint/pinpoint-bootstrap-2.1.0.jar -
  Dpinpoint.agentId=jenkinsagent154-2 -
  Dpinpoint.applicationName=springbootdemo1 -jar skywalkingspringbootdemo1.jar
```

docker

Dockerfile

```
1 FROM openjdk:8-alpine3.9
2 # 作者信息
3 MAINTAINER laosiji Docker pinpoint springboot "laosiji@lagou.com"
4 # 修改源
5 RUN echo "http://mirrors.aliyun.com/alpine/latest-stable/main/" >
  /etc/apk/repositories && \
6     echo "http://mirrors.aliyun.com/alpine/latest-stable/community/" >>
  /etc/apk/repositories
7
8 # 安装需要的软件，解决时区问题
9 RUN apk --update add curl bash tzdata && \
10     rm -rf /var/cache/apk/*
11
12 #修改镜像为东八区时间
13 ENV TZ Asia/Shanghai
14
15 COPY pinpoint/ /opt/pinpoint/
16
17 VOLUME /tmp
18 ARG JAR_FILE
19 COPY ${JAR_FILE} /app.jar
20
21 EXPOSE 8081
22
23 ENTRYPOINT java -javaagent:/opt/pinpoint/pinpoint-bootstrap-2.1.0.jar -
  Dpinpoint.agentId=jenkinsagent154-2 -
  Dpinpoint.applicationName=springbootdemo1 -jar /app.jar
```

制作镜像

```
1 cd /data
2 docker build --rm -t lagou/pinpointdemo1:1.0 --build-arg
  JAR_FILE=skywalkingspringbootdemo1.jar .
```

运行镜像

```
1 docker run -itd --name pinpointdemo1 -p 8081:8081 lagou/pinpointdemo1:1.0
```

测试项目

```

1 docker logs -f pinpointdemo1
2
3 http://192.168.198.154:8081/sayBoot
4
5 docker stop pinpointdemo1
6 docker rm pinpointdemo1

```

数据库

application.yml文件

包含mysql和mariadb数据库配置信息

```

1 server:
2   port: 8082
3 spring:
4   datasource:
5     #driver-class-name: com.mysql.jdbc.Driver
6     driver-class-name: org.mariadb.jdbc.Driver
7     username: root
8     password: admin
9     #url: jdbc:mysql://192.168.198.154:3306/lagou?characterEncoding=utf-
10    url: jdbc:mariadb://192.168.198.154:3307/lagou?characterEncoding=utf-
11    8&useSSL=false&useTimezone=true&serverTimezone=GMT%2B8
12    8&useSSL=false&useTimezone=true&serverTimezone=GMT%2B8
13 mybatis-plus:
14   configuration:
15     log-impl: org.apache.ibatis.logging.stdout.StdOutImpl
16   type-aliases-package: com.lagou.skywalkingspringbootdemo2.entity
17   mapper-locations: mapper/*.xml

```

数据库

```

1 创建tbuser表
2 CREATE TABLE `tbuser` (
3   `userid` int(11) NOT NULL AUTO_INCREMENT,
4   `username` varchar(20) COLLATE utf8_bin DEFAULT NULL,
5   `password` varchar(20) COLLATE utf8_bin DEFAULT NULL,
6   `userroles` varchar(2) COLLATE utf8_bin DEFAULT NULL,
7   `nickname` varchar(50) COLLATE utf8_bin DEFAULT NULL,
8   PRIMARY KEY (`userid`)
9 ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_bin

```

```

1 INSERT INTO tbuser (username,PASSWORD,userroles,nickname) VALUES
  ('admin','1234','04','管理员'),('lagou','1234','03','拉勾教育')

```

基础镜像

```
1 docker pull mariadb:10.5.2
2 docker load -i mariadb.10.5.2.tar
3
4 docker pull bitnami/mysql:5.7.30
5 docker load -i bitnami.mysql_5.7.30.tar
```

运行镜像

```
1 docker run -itd --name mariadb --restart always --privileged=true -p
  3307:3306 -v /data/mariadb:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=admin
  mariadb:10.5.2 --character-set-server=utf8mb4 --collation-
  server=utf8mb4_unicode_ci
```

启动agent测试

```
1 java -javaagent:/data/pinpoint/pinpoint-bootstrap-2.1.0.jar -
  Dpinpoint.agentId=jenkinsagent154-3 -
  Dpinpoint.applicationName=springbootdemo2 -jar skywalkingspringbootdemo2.jar
```

测试项目

```
1 http://192.168.198.154:8082/users
```

运行镜像

```
1 此镜像为非根镜像。没有root权限。挂载目录需要授权，命令如下：
2 mkdir -p /data/mysql/
3 chown -R 1001 /data/mysql/
4
5 docker run -itd --name mysql5.7 --restart always -p 3306:3306 -e
  MYSQL_ROOT_PASSWORD=admin -e TZ=Asia/Shanghai -v
  /data/mysql:/bitnami/mysql/ bitnami/mysql:5.7.30
```

启动agent测试

```
1 java -javaagent:/data/pinpoint/pinpoint-bootstrap-2.1.0.jar -
  Dpinpoint.agentId=jenkinsagent154-4 -
  Dpinpoint.applicationName=springbootdemo3 -jar skywalkingspringbootdemo2.jar
```

测试项目

1 | <http://192.168.198.154:8082/users>