pinpoint是韩国人开源的 APM (Application Performance Management/应用性能管理)工具 - Pinpoint。它基于google Dapper开发，目标就是为n(n>=1)层架构开发新的跟踪平台，为n层架构的系统提供解决方案。pinpoint能够对基于java的大规模分布式系统和应用做调用链的跟踪。pinpoint提供了一个web页面展示分布式系统的拓扑图以及系统这各个组件之间关系。pinpoint 有3个主要组件组成：日志收集器[Collector]、控制台[Web]、代理[Agent]，采用HBase进行存储。请结合下文的Pinpoin架构图来理解各组件之间如何分工合作的。

Pinpoint的特点如下:

分布式事务跟踪，跟踪跨分布式应用的消息

自动检测应用拓扑，帮助你搞清楚应用的架构

水平扩展以便支持大规模服务器集群

提供代码级别的可见性以便轻松定位失败点和瓶颈

使用字节码增强技术，添加新功能而无需修改代码

安装探针不需要修改哪怕一行代码及trace server端部署简单，支持hdfs存储

具有简单的阀值触发报警功能

移植性比较强的，会比较讨人喜欢（相比cat）

自定义插件功能（参考https://github.com/naver/pinpoint/wiki/Pinpoint-Plugin-Developer-Guide）

本文的部署方式也适用于最新的1.8.3版。参考博文：<https://blog.csdn.net/qq_31598113/article/details/90672693>

**集群角色规划**

|  |  |  |
| --- | --- | --- |
| **IP address** | **Hostname** | **Service** |
| **10.203.11.2** | **cdh-test-1-4** | **zk,nn1,zkfc,hmaster,pp\_collector,pp\_web** |
| **10.203.11.3** | **cdh-test-1-5** | **zk,nn2,zkfc,hmaster,pp\_agent** |
| **10.203.11.4** | **cdh-test-1-1** | **zk,datanode,nm,hrs,jn** |
| **10.203.11.5** | **cdh-test-1-3** | **datanode,nm,hrs,jn** |
| **10.203.11.6** | **cdh-test-1-2** | **datanode,nm,hrs,jn** |

**Pinpoint collctor采集器和web服务部署在4号机：**

****

**pinpoint agent和测试用的项目(tomcat服务)部署在5号机：**

****

**Pinpoint部署步骤**

1、在cdh-test-1-4主机配JDK、装hbase、装Pinpoint的collector和web组件

**1.1 系统需要安装Java环境**

从oracle官网下载、安装jdk-1.8.0后，在/etc/profile文件配置环境变量

export PATH=$PATH:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7\_6.x86\_64/bin

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7\_6.x86\_64

export HADOOP\_HOME=/tony\_soft/hadoop-2.8.4

export HBASE\_HOME=/tony\_soft/hbase-1.4.9

export ZOOKEEPER\_HOME=/tony\_soft/zookeeper-3.4.14

使配置生效：source /etc/profile

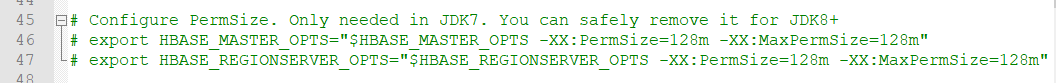
**1.2 安装Hbase**

解压Hbase放入指定目录

修改 /hbase/conf/hbase-env.sh 中的JAVA\_HOME环境变量位置

export JAVA\_HOME=/usr/java/jdk8/

在使用JDK8的情况下， hbase-env.sh中这两行可以注释，避免启动警告。



修改Hbase的配置信息  
[hadoop@cdh-test-1-4 ~]$ grep --color -C 3 "hbase.rootdir" $HBASE\_HOME/conf/hbase-site.xml

-->

<configuration>

<property>

<name>hbase.rootdir</name>

<value>hdfs://tony/hbase</value>

<description>指定hdfs上的HRegionServers共享目录</description>

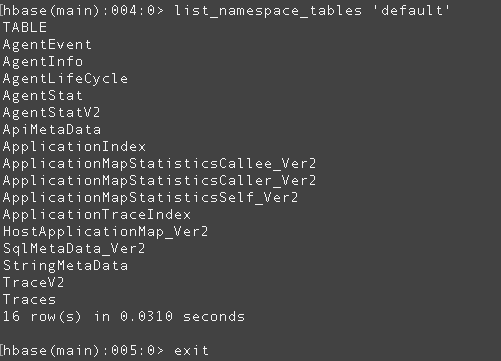
</property>

先启动hbase（我这里是hbase集群模式） cd /tony\_soft/hbase/bin && ./start-hbase.sh

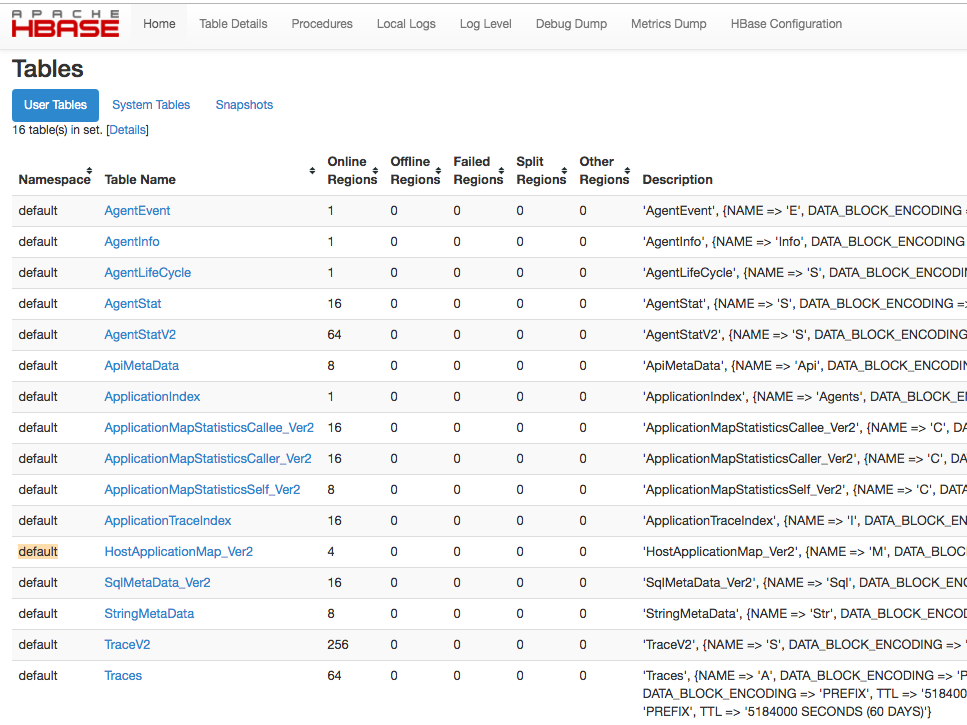
再初始化Hbase的pinpoint库: 执行pinpoint提供的Hbase初始化语句用于创建pinpoint所需的表

./hbase shell /home/pp\_res/hbase-create.hbase

创建成功后进到hbase shell可看到16张表已建好：



在web UI也能看到HBase的16张表



**1.3 安装部署pinpoint的collector组件**

解压tomcat到指定位置，用于部署pinpoint-collector

cd /home/pp\_res/

tar -zxvf apache-tomcat-8.5.40.tar.gz

mv apache-tomcat-8.5.40/ /tony\_soft/pp\_col

# 修改pp\_col的Tomcat配置，修改端口，避免与下面部署的pp\_web服务有关端口冲突。

cd /tony\_soft/pp\_col/conf/

sed -i 's/port="8005"/port="18115"/g' server.xml

sed -i 's/port="8080"/port="18181"/g' server.xml

sed -i 's/port="8443"/port="18443"/g' server.xml

sed -i 's/port="8009"/port="18119"/g' server.xml

sed -i 's/redirectPort="8443"/redirectPort="18443"/g' server.xml

解压pinpoint的collector的war包

cd /home/pp\_res/

rm -rf /tony\_soft/pp\_col/webapps/\* # 清空默认配置

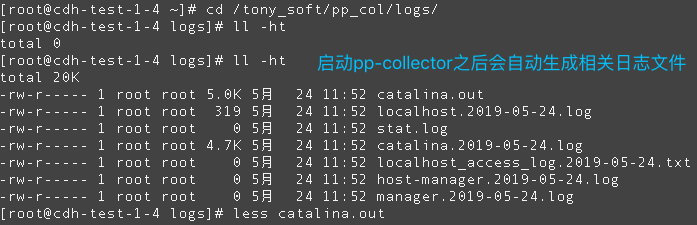
用unzip解压pinpoint的collector组件：

unzip /tony\_soft/pinpoint/resource/pinpoint-collector-1.7.3.war -d /tony\_soft/pp\_col/webapps/ROOT

（如果没有unzip,执行yum -y install unzip安装unzip）

启动第一个Tomcat服务：cd /tony\_soft/pp\_col/bin/ && ./startup.sh

成功后可在/tony\_soft/pp\_col/logs目录下的 catalina.out看到启动日志，查看启动过程是否出现错误



**1.4 部署pinpoint的web组件**

再次解压Tomcat后move到指定位置，这个tomcat用于部署pp\_web

cd /home/pp\_res/

tar -zxvf apache-tomcat-8.5.40.tar.gz

mv apache-tomcat-8.5.40/ /tony\_soft/pp\_web

# 同样的，修改一下pp\_web有关的端口号

cd /tony\_soft/pp\_web/conf/

sed -i 's/port="8005"/port="28005"/g' server.xml

sed -i 's/port="8080"/port="28080"/g' server.xml

sed -i 's/port="8443"/port="28443"/g' server.xml

sed -i 's/port="8009"/port="28009"/g' server.xml

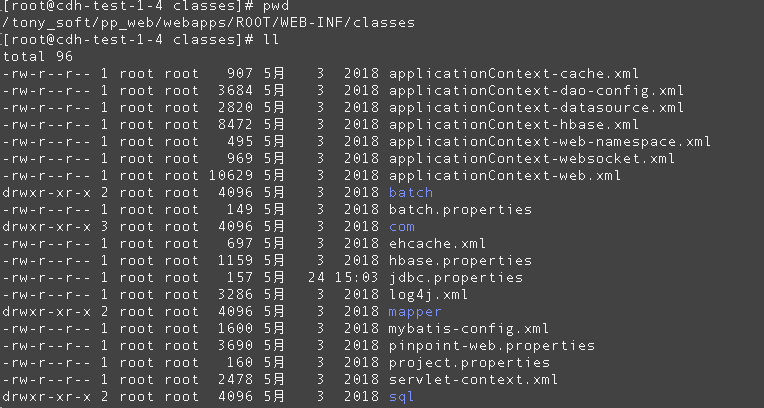
sed -i 's/redirectPort="8443"/redirectPort="28443"/g' server.xml

解压pinpoint web组件的war包

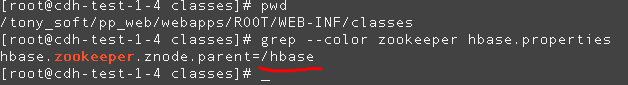
rm -rf /tony\_soft/pp\_web/webapps/\* # 清空默认配置

unzip pinpoint的web组件[root@cdh-test-1-4 classes]# unzip /tony\_soft/pinpoint/resource/pinpoint-web-1.7.3.war -d /tony\_soft/pp\_web/webapps/ROOT

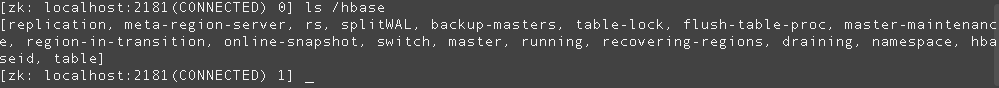
/tony\_soft/pp\_web/webapps/ROOT/WEB-INF/classes下的配置文件说明：



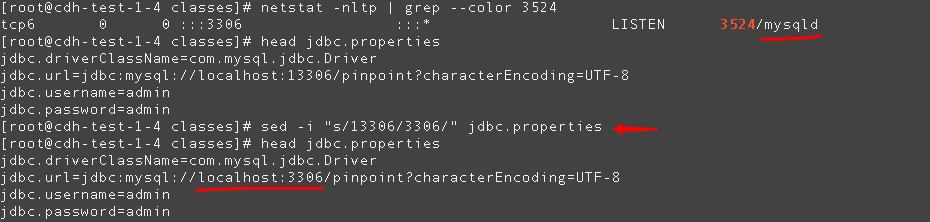
* **hbase.properties**：配置pp\_web从哪个数据源获取采集数据，这里我只指定Hbase的zk地址(在4号机)







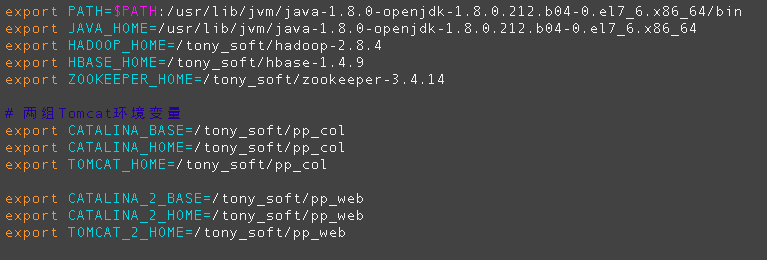
* **jdbc**.**properties** ：pp\_web连接自身Mysql数据库的连接认证配置文件



* sql目录 pp\_web本身有些数据需要存放在MySQL数据库中，需初始化表结构(执行两个.sql脚本即可)
* **pinpoint**的web组件.**properties** ：这里pp\_web集群的配置文件(如果你需要pp\_web集群)
* **applicationContext**-\* .xml ：这些文件在未来调优中会用到
* **log4j**.**xml** ：日志相关配置

因为我要在同一个主机下同时运行两个Tomcat，需要先干三件事——

第一件事，先停掉刚才启动的collector的tomcat服务，再去配置两套环境变量：

****

使这两组环境变量生效：source /etc/profile

第二件事，server.xml中配置端口不冲突(请参考上文步骤1.3)

第三件事，修改collector组件的Tomcat脚本startup.sh、shutdown.sh，在各自文件末尾都添加这几行:

export JAVA\_HOME=$JAVA\_HOME

export PATH=$PATH:$JAVA\_HOME/bin

export CLASSPATH=$JAVA\_HOME/lib

export CATALINA\_HOME=$CATALINA\_HOME

export CATALINA\_BASE=$CATALINA\_BASE

修改web组件的Tomcat的脚本startup.sh、shutdown.sh，在各自文件末尾都添加这几行：

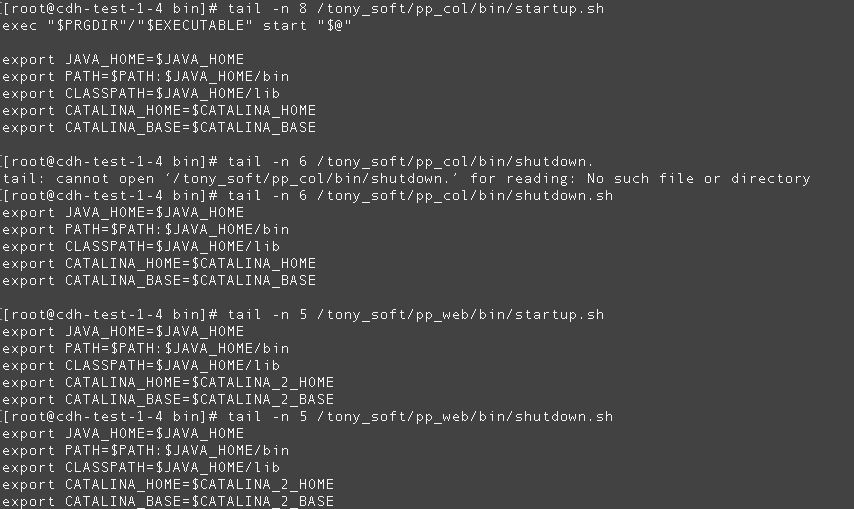
export JAVA\_HOME=$JAVA\_HOME

export PATH=$PATH:$JAVA\_HOME/bin

export CLASSPATH=$JAVA\_HOME/lib

export CATALINA\_HOME=$CATALINA\_2\_HOME

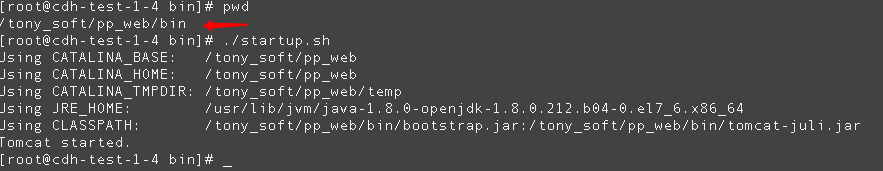
export CATALINA\_BASE=$CATALINA\_2\_BASE

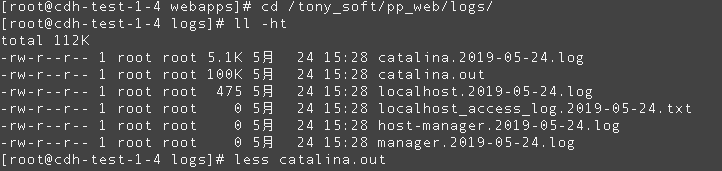


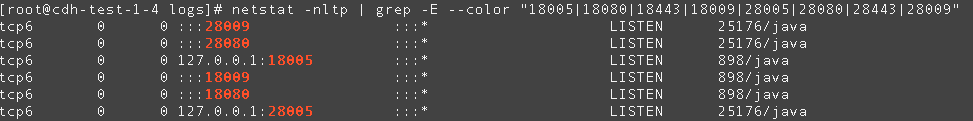
先启动第一个Tomcat：cd /tony\_soft/pp\_col/bin/ && ./startup.sh

再启动第二个Tomcat：cd /tony\_soft/pp\_web/bin/ && ./startup.sh

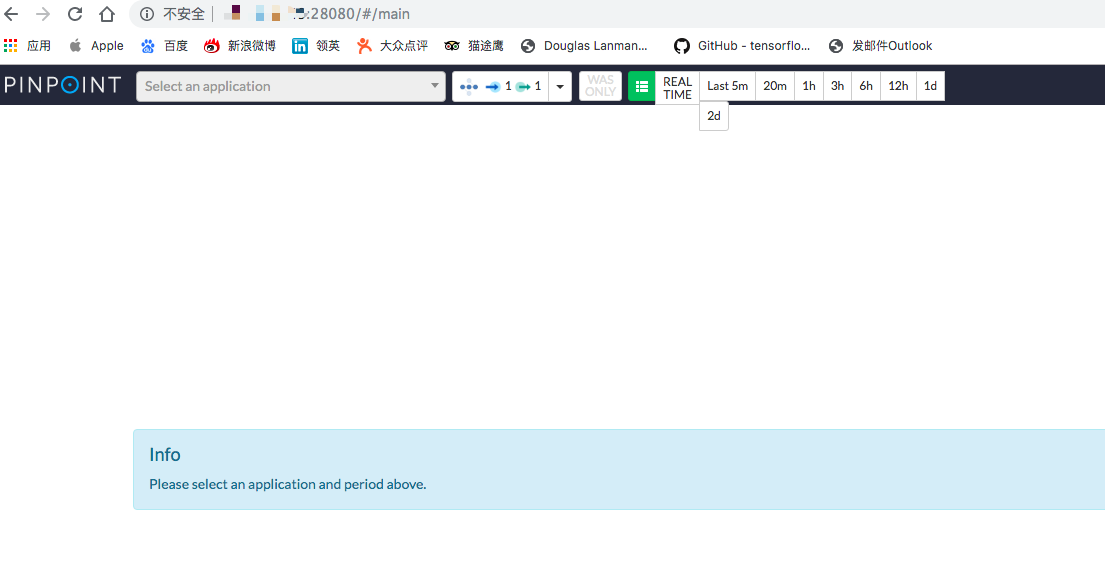
同样的，可以打开catalina.out文件来查看启动是否完成、启动过程是否出现错误等



  
pp\_col和pp\_web启动完成，可看到相关的端口处于LISTEN状态，1开头的端口号是pinpoint collector有关的服务端口，2开头的是pinpoin web有关的：



去浏览器中进行访问。输入地址http://${yourIPaddress}:28080 进入PinPoint的Web页面:



2、在cdh-test-1-5主机上部署测试项目及PinPoint-Agent组件

首先，也要把所需文件放到cdh-test-1-5主机上的的 /tony\_soft/pp\_test 目录下，包括pinpoint-agent-1.7.3.tar.gz、jdk、apache-tomcat-8.5.40.tar.gz、test.war

**2.1 参考上文1.1的安装、配置jdk**

**2.2 部署测试项目**

解压tomcat到指定目录，用于部署测试项目

[root@cdh-test-1-5]# cd /tony\_soft/pp\_test

[root@cdh-test-1-5 pp\_test]# tar -zxvf apache-tomcat-8.5.40.tar.gz

[root@cdh-test-1-5]# mv apache-tomcat-8.5.40 /tony\_soft/pp\_test/

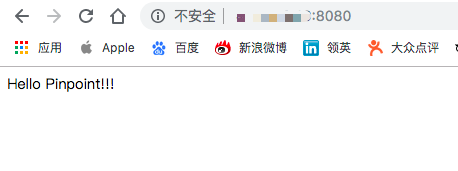
从war包部署测试项目

cd /tony\_soft/pp\_test/

rm -rf /tony\_soft/pp\_test/webapps/\*

[root@cdh-test-1-5 bin]# unzip /tony\_soft/test.war -d /tony\_soft/pp\_test/webapps/ROOT

运行后直接访问8080端口或者8080/test即可。其中/test是会发生异常的，方便后面的警报功能测试。



在继续安装pp\_agent开始监控应用之前，做一个小结：

(1)如果Hbase不是和pp-web, pp-col装在一台机器，需要安装zookeeper，只需安装并确认在监听2181端口

(2)如果zookeeper安装在独立机器上，需修改pp\_col 和 pp\_web的配置文件pinpoint-collector.properties和pinpoint-web.properties，否则会导致pp\_col和pp\_web模块无法启动

**2.3 安装配置pp\_agent**

cd /tony\_soft/pp\_test

mkdir -p /tony\_soft/pp\_agent

tar -zxvf pinpoint-agent-1.7.3.tar.gz -C /tony\_soft/pp\_agent  
修改/tony\_soft/pp\_agent/pinpoint.config，把里面的 profiler.collector.ip 的值改为装有pinpoint collector组件的ip地址。

再去修改启动脚本/tony\_soft/pp\_test/bin/catalina.sh，通过添加下列三行来添加探针：



这些探针是为了agent把被监控应用的相关数据发送到collector。三行代码分别代表：

* pp\_agent目录下 pinpoint-bootstrap-1.7.3.jar包的绝对路径
* pp\_agent的唯一ID，不能与被监控的其他项目相同
* 被采集项目的名称，pp\_web页面中下拉框选择项目时显示的名字

配置好后启动测试项目

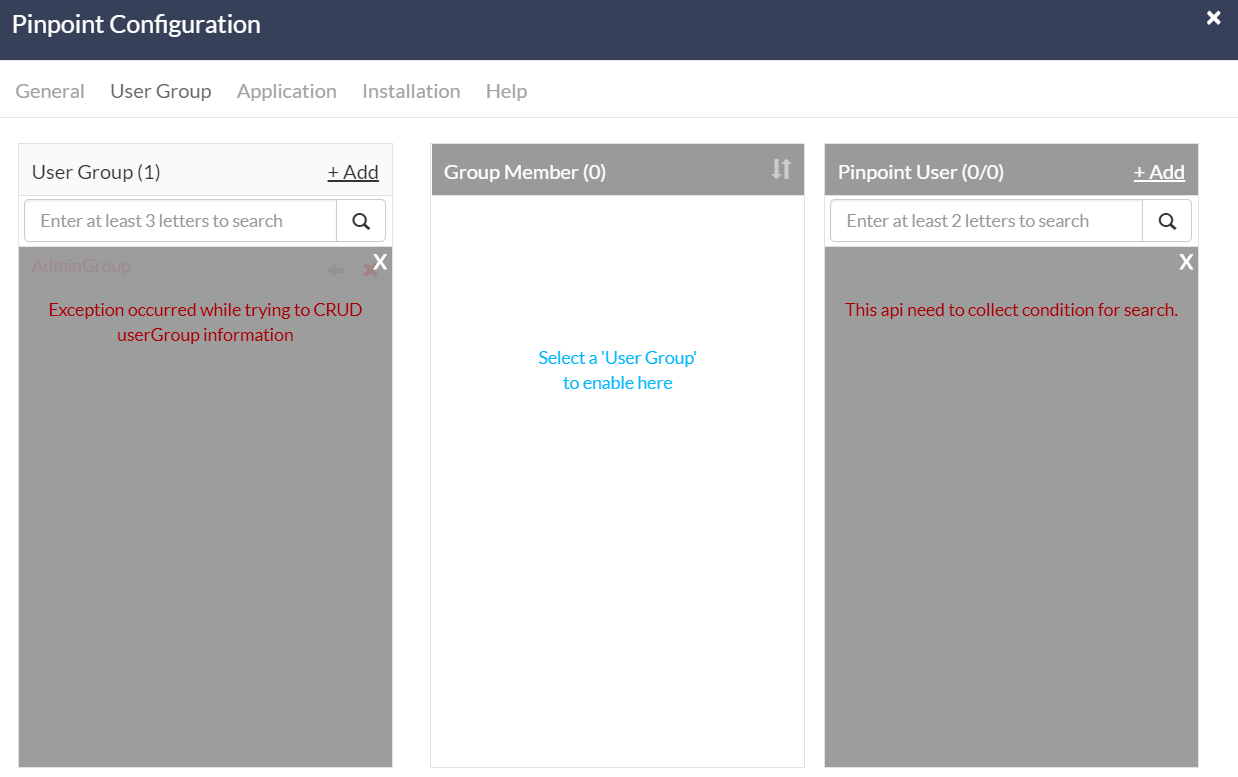
cd /tony\_soft/pp\_test/bin/ && ./startup.sh

3、Pinpoint的Alarm功能需要MySQL服务

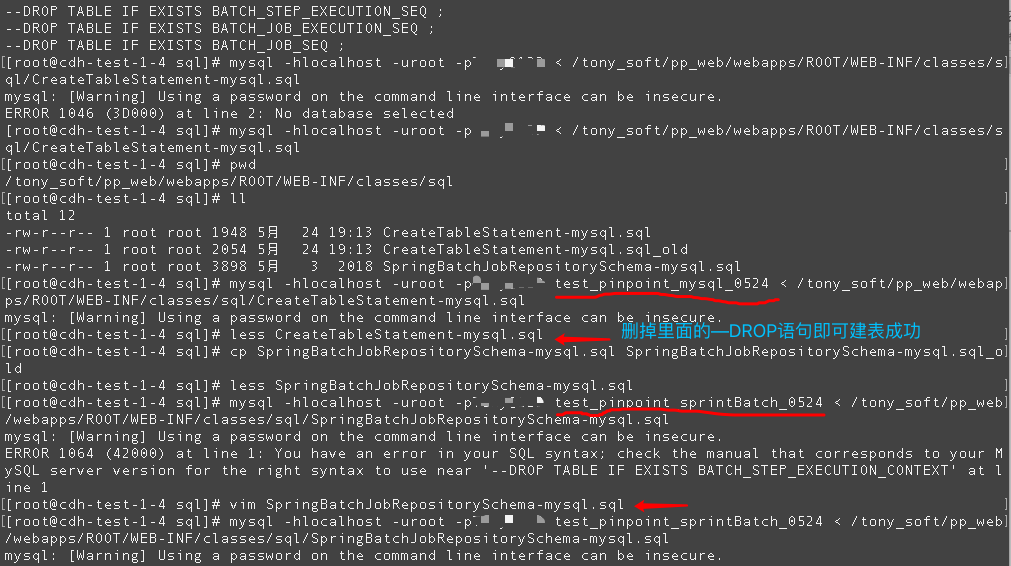
先安装MYSQL服务器机器客户端：

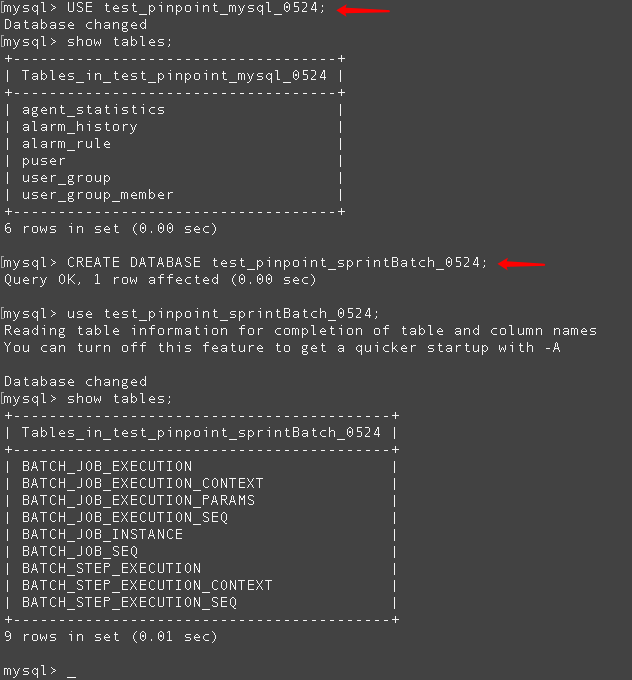
CentOS7 yum方式安装MySQL5.7 ：<https://www.cnblogs.com/luohanguo/p/9045391.html>

如果要使用Pinpoint的Alarm功能需要MySQL服务支持，否则点击pp web页面右上角的齿轮后，其中一些功能（如编辑用户、用户组、报警等功能）会出现如图所示的异常：



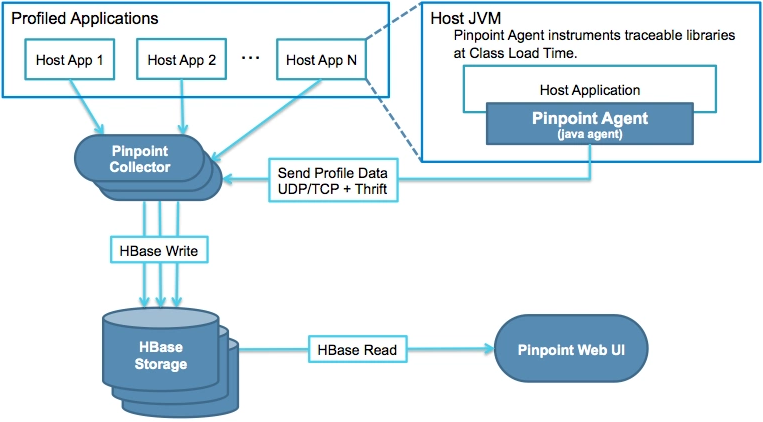
所以需要先使用pp\_web组件自带的sql文件创建有关的表





在MYSQL里面建了两张表以后，pp\_web页面仍然无法正常显示User Group，解决办法请看下文“2019-50-28第三次踩坑记录”。

最后来看一下pinpoint的架构：



架构中各组件的角色：

**pinpoint web UI**显示的内容来自于后端的**pinpoint web**服务，

pinpoint web服务会去**hbase**中取数，

**collector**会把来自于**agent**的数据采集上来，由collector存入hbase(本案例采用hbase集群模式，不是单机HBASE)。

Demo展示时，可以把pp collector服务和pp web服务部署在同一个主机上，方便collector接收agent发来的数据后发给web服务，web服务再对hbase读/写；agent服务部署在应用端，方便将数据发送给collector。

======2019-05-27第一次踩坑=======

在重启pp\_col和pp\_web时出现故障：pp\_col能重启，但是无法启动pp\_web

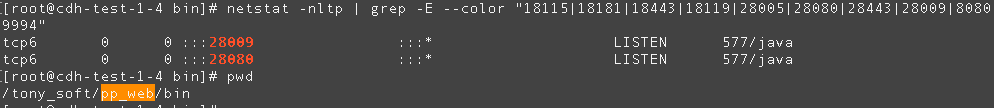
（1）重启pp\_col、pp\_web，pp\_col成功重启、pp\_web失败重启

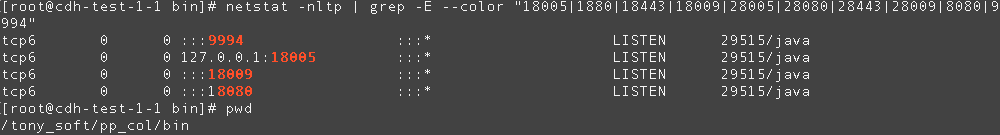
（2）重启pp\_web、pp\_col，pp\_col成功重启、pp\_web失败重启

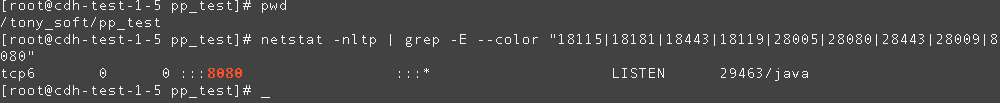
（3）查看相关日志后发现有is very like to cause memory leak字样

**（4）原因：在4号机上同时启动pp\_col和pp\_web服务会导致内存不足，导致了无法同时运行col服务和web服务。那么改成pp\_col在4号机启动，pp\_web在1号机启动(http端口号是28080)，之后发现在浏览器访问28080端口页面时打不开。原因很有可能是之前只是放行了从外网访问4号机IP地址的28080端口，而没有放行从外网访问1号机IP地址的28080端口。**

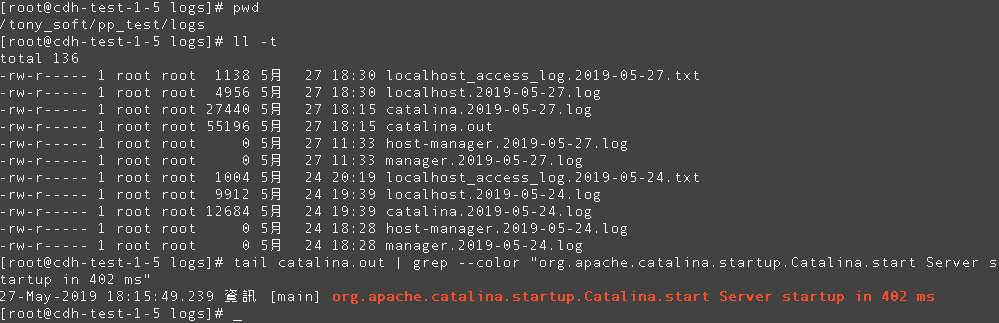
**所以现在改成在4号机启动pp\_web服务，在1号机启动pp\_col服务。需要修改相关的/etc/profile(CATALINA\_HOME、CATALINA\_BASE环境变量)、/tony\_soft/pp\_col/conf/server.xml(有关端口)后成功启动各自的pp\_col、pp\_web ：**

****

****



测试的项目日志也表明启动成功：



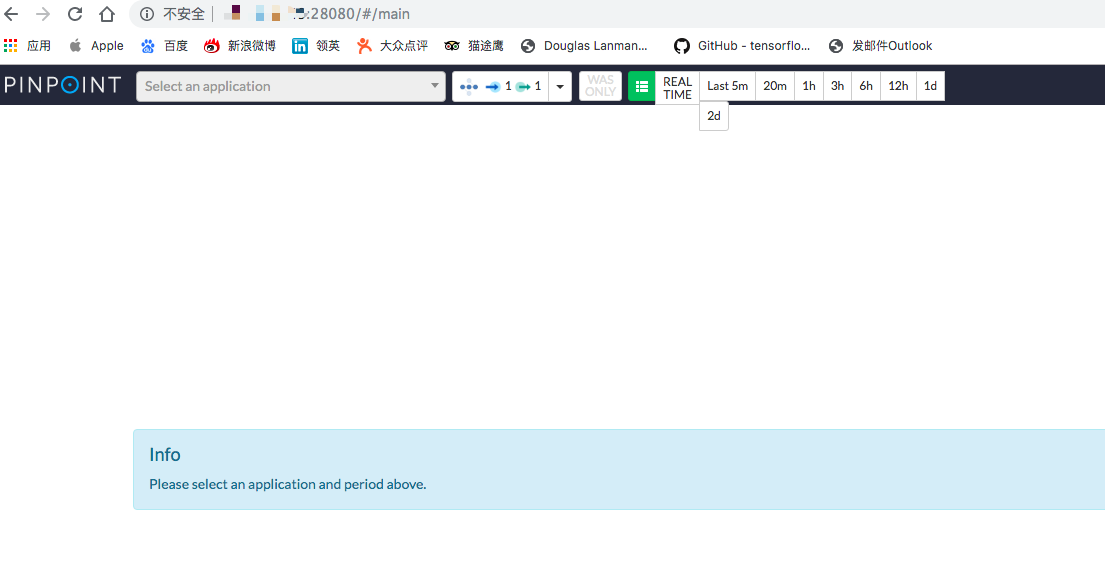
======2019-05-28第二次踩坑记录======

【解决web页不显示应用问题】

昨天和今天都被一个问题困扰，就是web页面不显示已添加到pp\_agent的java应用，下面是解决过程。

第一次尝试——

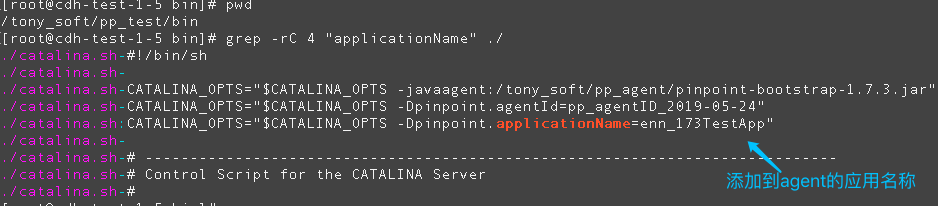
先是分别重启pp\_collector、pp\_web、pp\_agent、pp\_test(java应用程序)，仍不显示被agent打了探针的应用, 如截图：

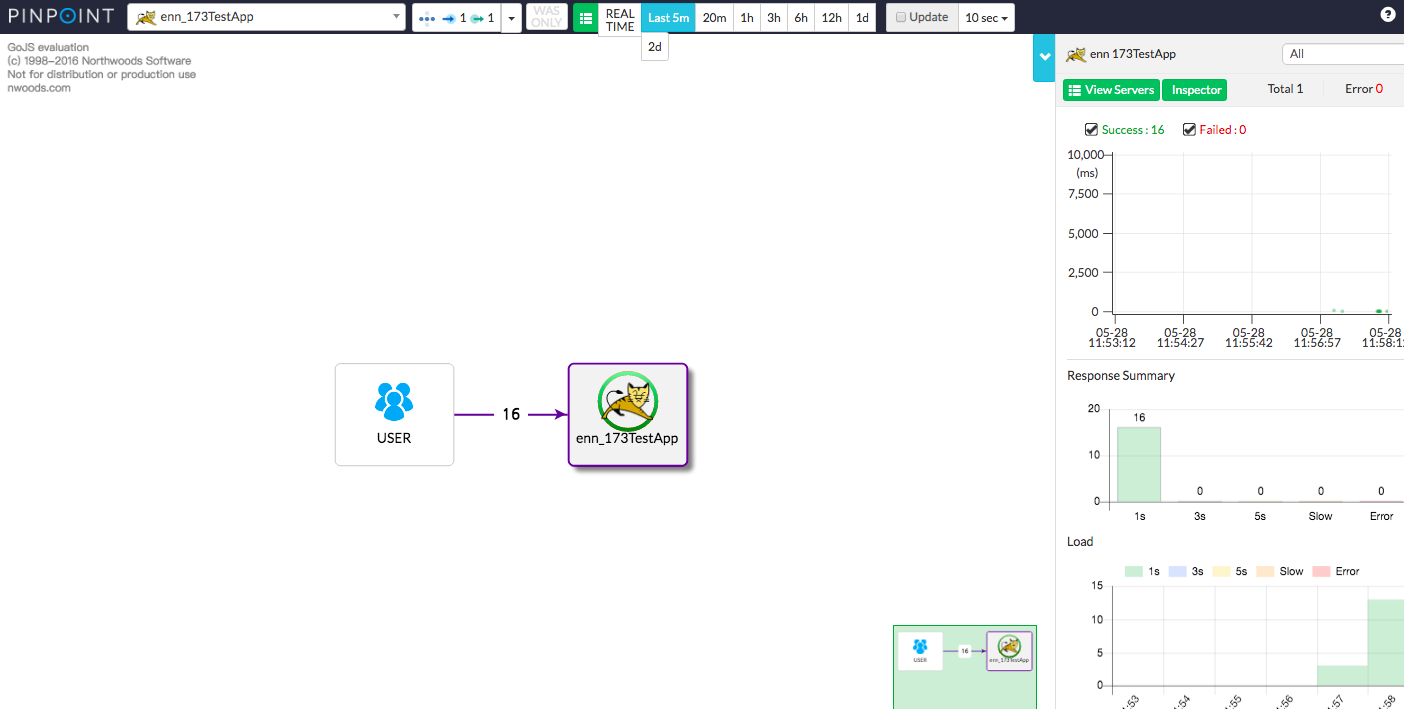


第二次尝试——

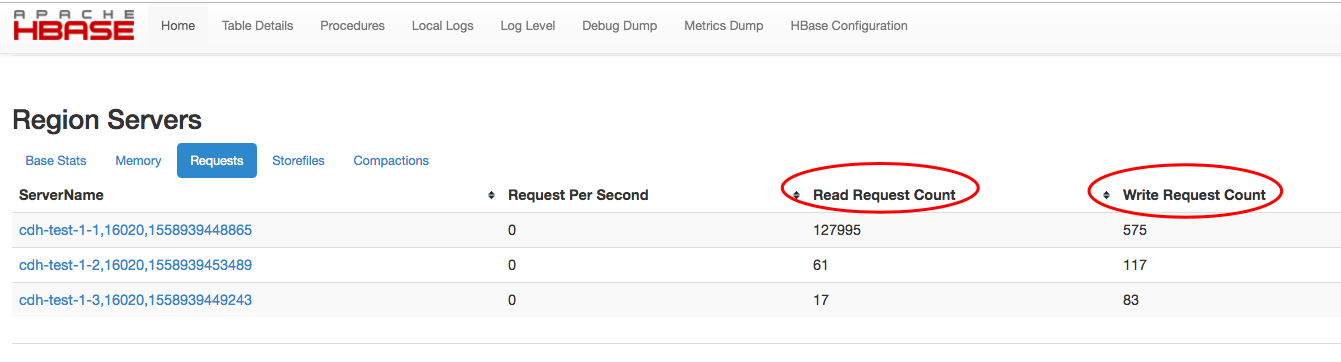
检查过各主机的/etc/hosts文件，检查过主机间的网络通信，都没问题。

百思不得其解，再次查看pinpoint架构图后突然灵光一闪！联想到pp\_web显示数据的来源是hbase，hbase的数据来源是collector，而collector的数据由agent发来。既然agent已经随着java的test应用启动了，那么agent是否将应用的数据成功发送到collector了呢？如果成功发送了应用的数据，那么web页应该显示已添加的application呀！然后去检查/tony\_soft/pp\_agent/pinpoint.config配置文件，发现配置项profiler.collector.ip写的不是collector所在的IP地址。推测这个错误很可能是导致web页面的application list下拉框不显示已添加的application原因。改成collector所在主机的IP地址后，依次重启pp\_collector、pp\_web服务、pp\_agent、pp\_test应用，再去查看web页，证明推测是对的：





到pp\_test也就是测试用的java应用的web页面http://${yourIP}:8080/多访问几下，发现hbase的requests数目也大于0了，证实了推测正确：

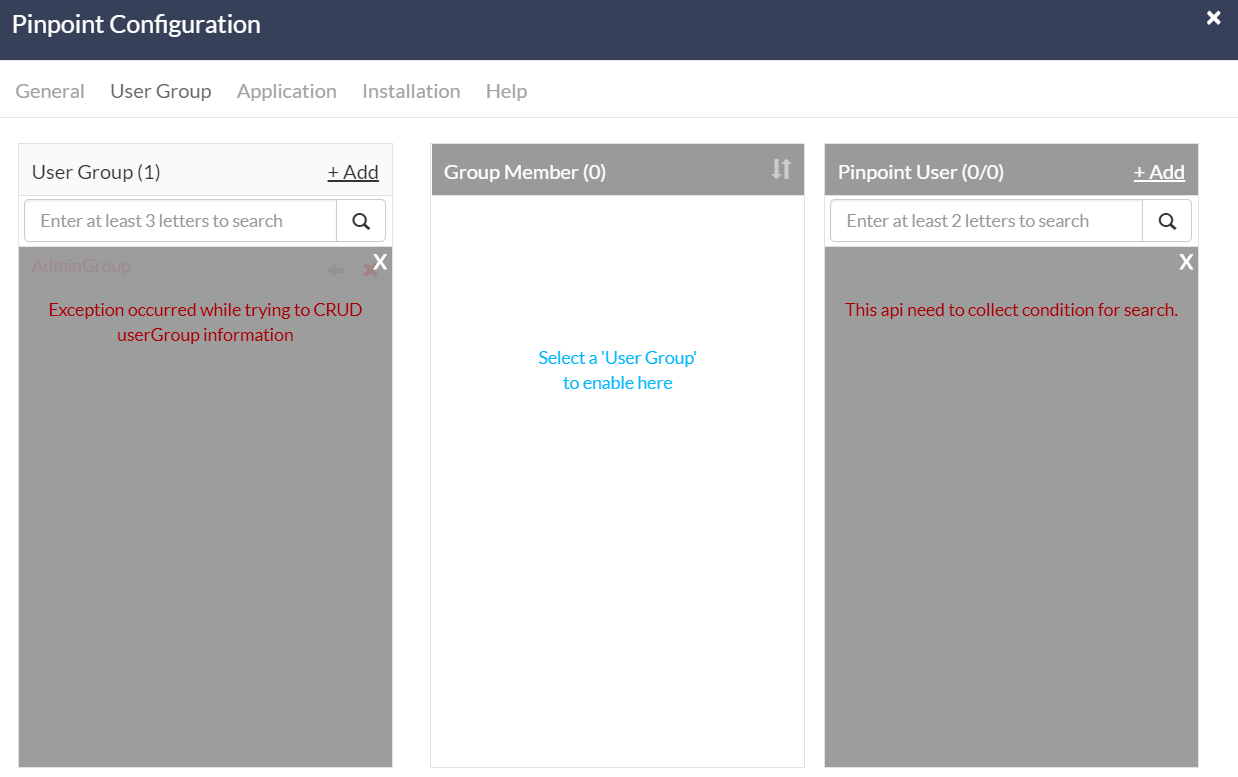


===========2019-50-28第三次踩坑记录==========

【web页面用户组管理页问题】

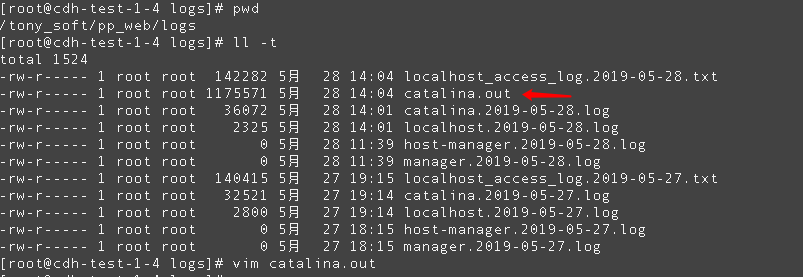
现象——

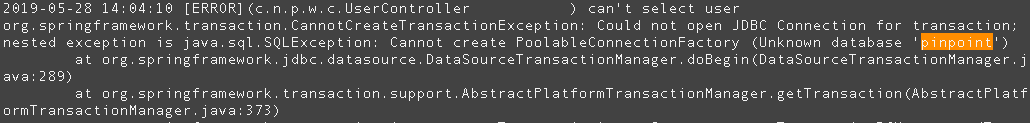
为了将来使用Alarm功能对指定用户提供告警，需要配置用户和用户组。在web页点击右上角的齿轮后，发现查看User Group时发现错误提示，如截图：



第一次排查——

进到pp\_web服务所在的日志目录，查看最新的日志catalina.out，发现有Unknow database ‘pinpoint’字样：



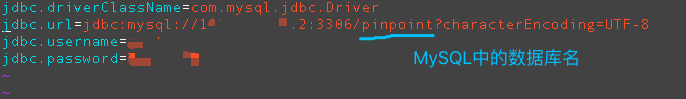


2019-05-28 14:04:10 [ERROR](c.n.p.w.c.UserGroupController ) Exception occurred while trying to CRUD userGroup information

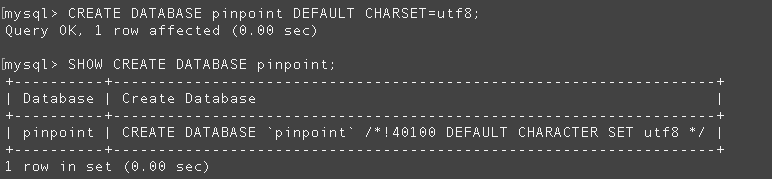
org.springframework.transaction.CannotCreateTransactionException: Could not open JDBC Connection for transaction; nested exception is java.sql.SQLException: Cannot create PoolableConnectionFactory (Unknown database 'pinpoint')

at org.springframework.jdbc.datasource.DataSourceTransactionManager.doBegin(DataSourceTransactionManager.java:289)

然后检查$CATALINA\_HOME//webapps/ROOT/WEB-INF/classes/jdbc.properties，发现日志里提到的库名没在mysql中：

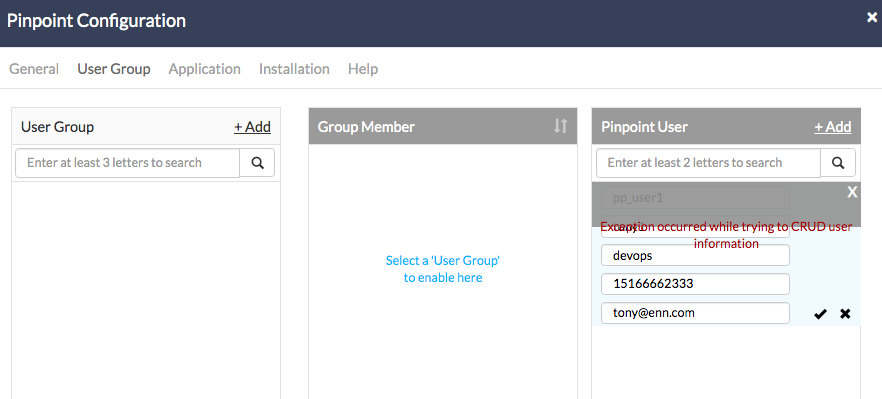


原因是之前通过两个.sql脚本初始化的数据库表都没在pinpoint库名下。于是先到数据库中创建pinpoint库:



然后分别重启pp\_web、pp\_col、test应用，再到web页面查看User Group，仍然报错。

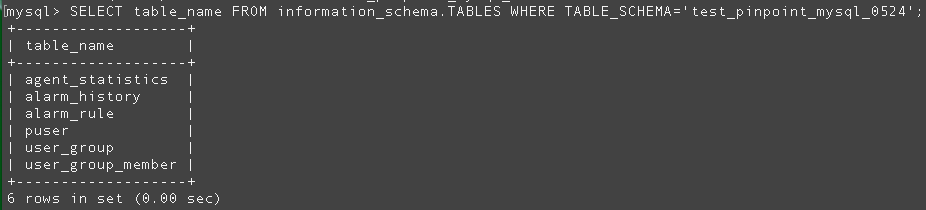
第二次排查——

通过web页面的User Group尝试创建用户，报错：  


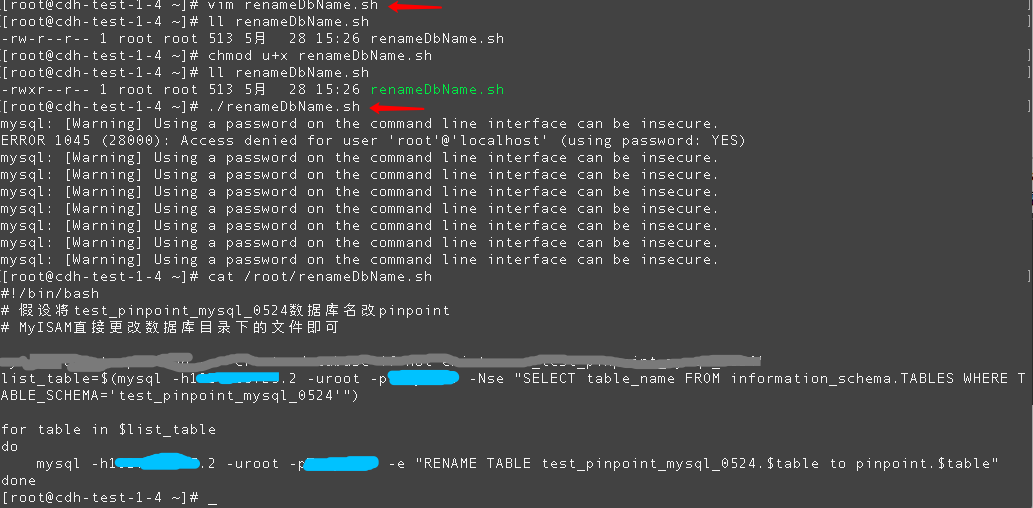
再去查看最新的catalina.out日志，发现有提示说pinpoint下面的puser表不存在：

Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Table 'pinpoint.puser' doesn't exist

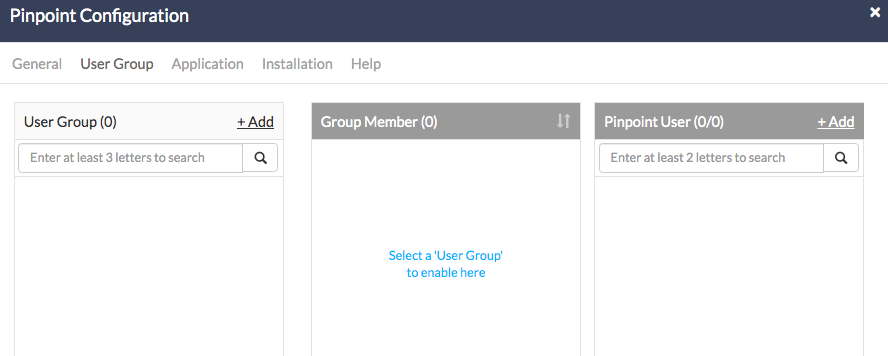
进到mysql shell检查之前通过pp\_web组件自带的sql文件创建的表，发现puser表不属于pinpoint库



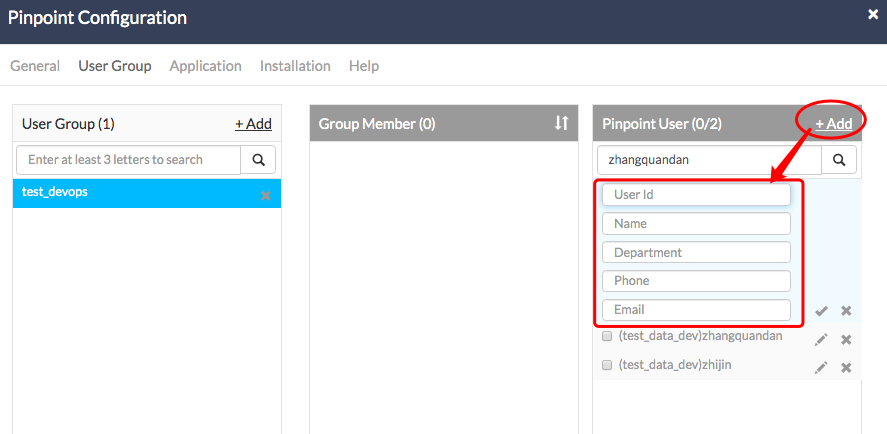
解决办法是把那些表都拷贝到pinpoint库下面：

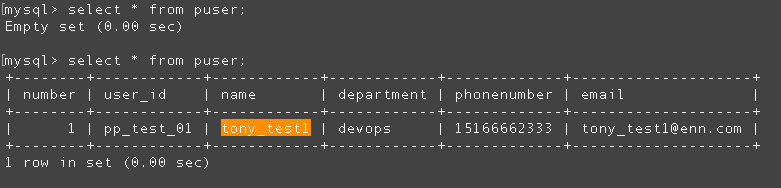


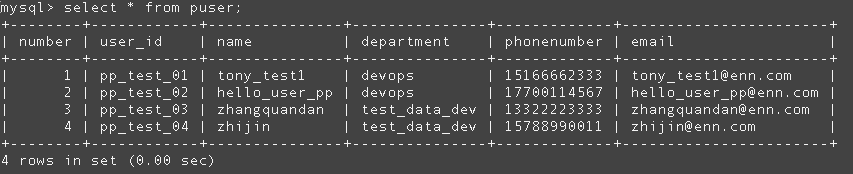
拷贝完以后，不用重启pp\_web服务，到web页面重新查看和操作用户和用户组，发现没有报错了：



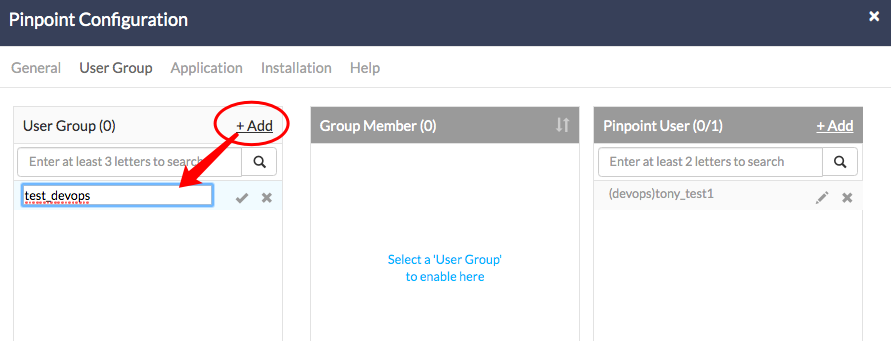
再添加多个用户以后，能在mysql shell中查看到该用户了：



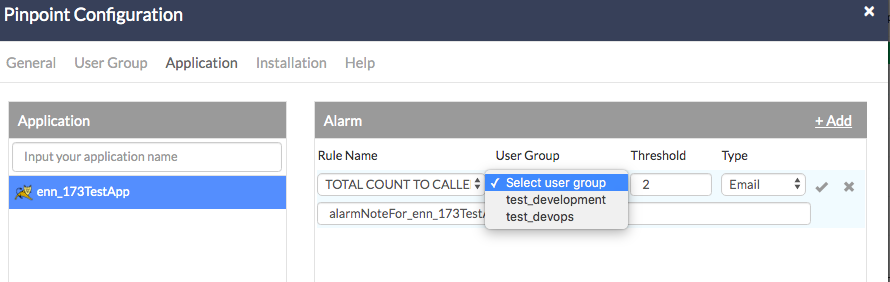




添加用户组：



创建用户组成功后，将用户tony\_test1添加到组里：



=========下一步=========

需要加入更多的JAVA应用到pinpoint-agent当中，以便监控java作业运行情况、方便定位应用故障、方便发现应用的性能问题和大数据集群的性能问题。

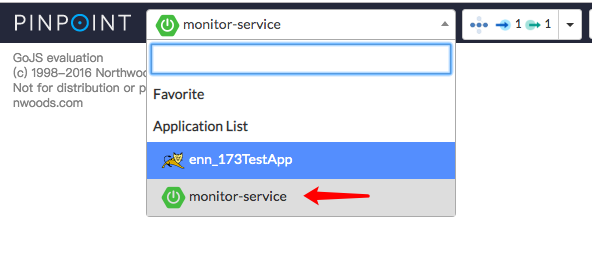
例如，在部署了pinpoint-web、pinpoint-collector、hbase集群、pinpoint-agent的集群上，如果想要监控一个通过jar包启动的应用，那么可以通过下列方式为应用加入pinpoint-agent探针：

**java -javaagent:/tony\_soft/pp\_agent/pinpoint-bootstrap-1.7.3.jar \**

**-Dpinpoint.agentId=pp\_agentID\_2019-05-28 -Dpinpoint.applicationName=monitor-service \**

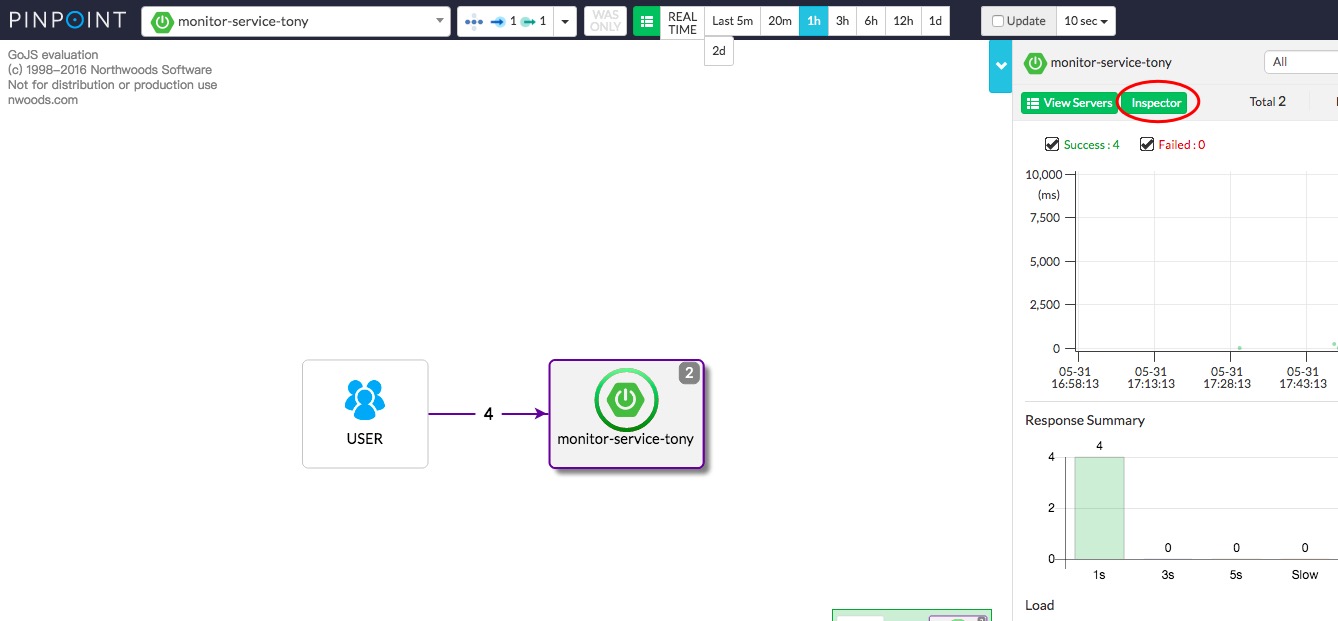
-jar monitor-service-0.0.1-SNAPSHOT.jar

应用启动后，可以在web页的应用列表里看到：



监控其他java应用的做法也是类似的，就是通过在启动java应用的时候像刚才那样添加启动参数即可，不需要改java应用的代码。

jar包monitor-service-0.0.1-SNAPSHOT.jar启动之后，里面的应用程序会去查mysql数据库，那么pinpoint-agent能跟踪到，但是我的web页中的拓扑图没显示MySQL组件，如截图：



但是，点击页面的Inspetor按钮之后，就能在页面底部Mysql有关信息，如截图：

