第一章 安装准备

1.1 规划

服务端：

OS ：Ubuntu 14.04.4 x64
IP ：10.1.5.204
JDK ：1.8.0_171
Tomcat ：8.5.33.0
TCP：8080
路径：/pinpoint/tomcat-8080-pp
pinpoint-web ：1.8.0-RC1
路径：/pinpoint/tomcat-8080-pp/webapps/ppweb
web：http://10.1.5.204:8080/ppweb
账单密码：无
角色：提供 web 展示
pinpoint-collector ：1.8.0-RC1
TCP ：9994
UDP ：9995,9996
路径：/pinpoint/tomcat-8080-pp/webapps/ppcol
角色：服务端程序，接收客户端的数据上报
Hbase:1.2.6
路径：/pinpoint/hbase-1.2.6/
数据目录：/pinpoint/hbase-data/
角色：服务端存储数据

客户端：

OS ：Ubuntu 14.04.4 x64
JDK ：1.8.0_171
Tomcat ：8.5.33.0
pinpoint-agent ：1.8.0-RC1
IP：172.16.7.51
程序：openapi 等多各 spring boot 程序的，也有 tomcat 启动的程序
角色：收集数据，上报到服务端

1.2 版本选择

pinpoint 和 hbase、jdk 的版本匹配关系，以及 agent 和 collector 的版本对应关系见 https://github.com/naver/pinpoint
下面是部分截图:

Pinpoint Version	Agent	Collector	Web
1.0.x	6-8	6+	6+
1.1.x	6-8	7+	7+
1.5.x	6-8	7+	7+
1.6.x	6-8	7+	7+
1.7.x	6-8	8+	8+
1.8.x	6-10	8+	8+

HBase compatibility table:

Pinpoint Version	HBase 0.94.x	HBase 0.98.x	HBase 1.0.x	HBase 1.2.x
1.0.x	yes	no	no	no
1.1.x	no	not tested	yes	not tested
1.5.x	no	not tested	yes	not tested
1.6.x	no	not tested	not tested	yes
1.7.x	no	not tested	not tested	yes
1.8.x	no	not tested	not tested	yes

Agent - Collector compatibility table:

Agent Version	Collector 1.0.x	Collector 1.1.x	Collector 1.5.x	Collector 1.6.x	Collector 1.7.x	Collector 1.8.x
1.0.x	yes	yes	yes	yes	yes	yes
1.1.x	not tested	yes	yes	yes	yes	yes
1.5.x	no	no	yes	yes	yes	yes
1.6.x	no	no	not tested	yes	yes	yes
1.7.x	no	no	no	no	yes	yes
1.8.x	no	no	no	no	no	yes

1.3 程序用户

我这里直接使用 root 启动程序，生产环境中建议使用普通用户启动程序。

1.4 jdk 环境配置

保证能看到如下效果

```
root@demo02:~# java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
root@demo02:~#
```

步骤这里略过。

1.5 tomcat 安装配置

服务端用来运行 pinpoint-collector 和 pinpoint-web，
客户端用来运行 java 项目
安装过程这里略过。

第二章 服务端 Hbase 安装配置

为了简化安装步骤，这里使用 hbase 单机模式，zookeeper 使用 hbase 自带的，不再安装 zk 集群，实际使用中可根据需要灵活配置。

2.1 Hbase 下载

到这里
http://archive.apache.org/dist/hbase/
选择合适版本，我选了 1.2.6 版本

```
wget http://archive.apache.org/dist/hbase/1.2.6/hbase-1.2.6-bin.tar.gz
```

并解压到规划的路径

```
tar xf hbase-1.2.6-bin.tar.gz -C /pinpoint/
```

2.2 创建目录

```
#创建 hbase 的 zk 数据目录
mkdir /pinpoint/hbase-zk-data
#创建 hbase 的数据目录
mkdir /pinpoint/hbase-data
```

2.3 修改配置文件

cat /pinpoint/hbase-1.2.6/conf/hbase-site.xml
保证有以下内容

```
<configuration>
<property>
       <name>hbase.rootdir</name>
       <value>file:///pinpoint/hbase-data</value>
   </property>
   <property>
       <name>hbase.zookeeper.property.dataDir</name>
       <value>/pinpoint/hbase-zk-data</value>
   </property>
</configuration>
```

2.4 添加环境变量（可选）

保证/etc/profile 中有如下内容:

```
export HBASE_HOME=/pinpoint/hbase-1.2.6
export PATH=$HBASE_HOME/bin:$JAVA_HOME/bin:$PATH
```

2.5 启动

```
/pinpoint/hbase-1.2.6/bin/start-hbase.sh
```

另外停止命令是:

```
/pinpoint/hbase-1.2.6/bin/stop-hbase.sh
```

2.6 hbase 初始化

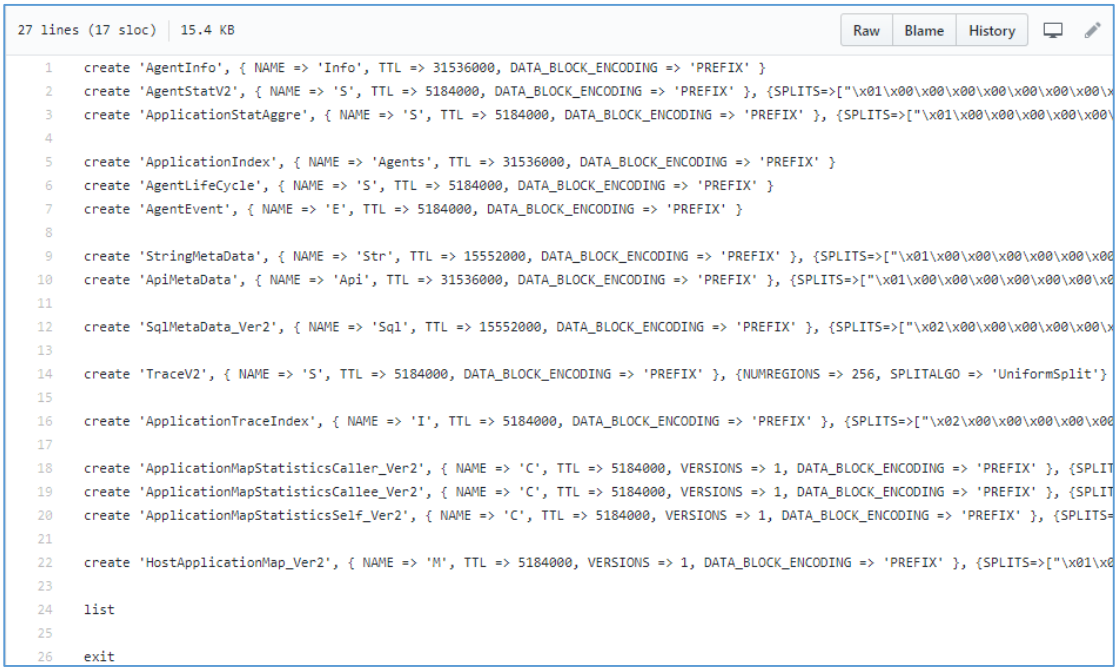
使用如下命令初始化 hbase

```
hbase shell /root/hbase-create.hbase
```

hbase 初始化，就是在空的 hbase 中创建一些表等动作，该文件是通用的，可以到 github 的源码中查，地址:

https://github.com/naver/pinpoint/blob/master/hbase/scripts/hbase-create.hbase

拷贝如下图中的内容，保存到 hbase-create.hbase 文件中即可。



27 lines (17 sloc) | 15.4 KB — Raw | Blame | History

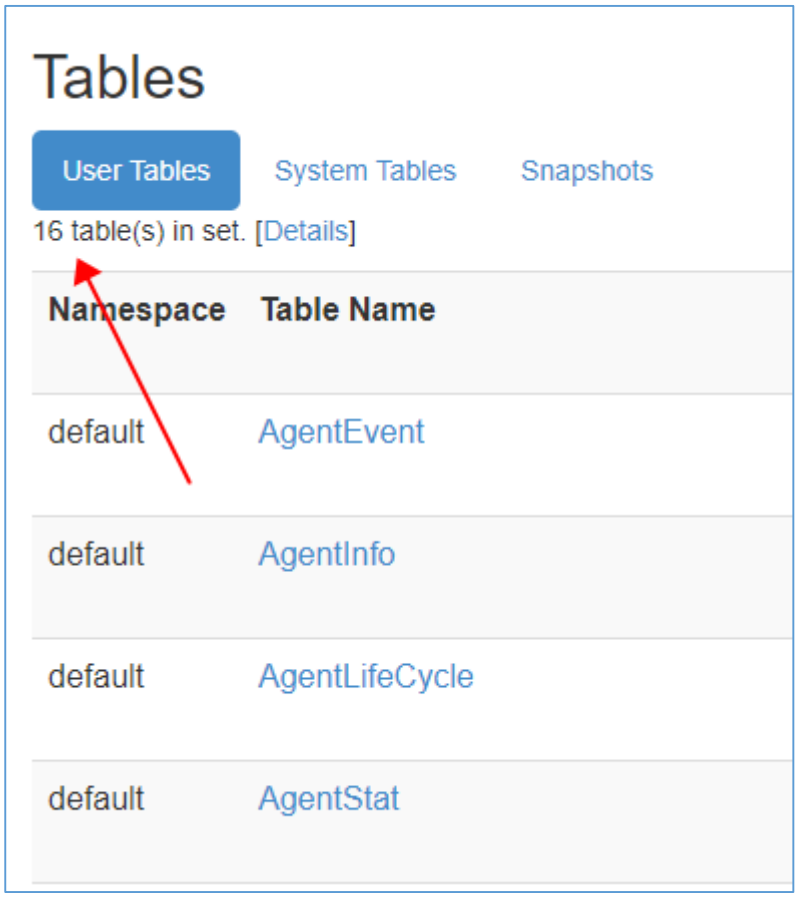
```
create 'AgentInfo', { NAME => 'Info', TTL => 31536000, DATA_BLOCK_ENCODING => 'PREFIX' }
create 'AgentStatV2', { NAME => 'S', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x01\x00\x00\x00\x00\x00\x00\x00\x
create 'ApplicationStatAggre', { NAME => 'S', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x01\x00\x00\x00\x00\x00\

create 'ApplicationIndex', { NAME => 'Agents', TTL => 31536000, DATA_BLOCK_ENCODING => 'PREFIX' }
create 'AgentLifeCycle', { NAME => 'S', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }
create 'AgentEvent', { NAME => 'E', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }

create 'StringMetaData', { NAME => 'Str', TTL => 15552000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x01\x00\x00\x00\x00\x00\x00
create 'ApiMetaData', { NAME => 'Api', TTL => 31536000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x01\x00\x00\x00\x00\x00\x00\x0

create 'SqlMetaData_Ver2', { NAME => 'Sql', TTL => 15552000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x02\x00\x00\x00\x00\x00\x

create 'TraceV2', { NAME => 'S', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }, {NUMREGIONS => 256, SPLITALGO => 'UniformSplit'}

create 'ApplicationTraceIndex', { NAME => 'I', TTL => 5184000, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x02\x00\x00\x00\x00\x00

create 'ApplicationMapStatisticsCaller_Ver2', { NAME => 'C', TTL => 5184000, VERSIONS => 1, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLIT
create 'ApplicationMapStatisticsCallee_Ver2', { NAME => 'C', TTL => 5184000, VERSIONS => 1, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLIT
create 'ApplicationMapStatisticsSelf_Ver2', { NAME => 'C', TTL => 5184000, VERSIONS => 1, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=

create 'HostApplicationMap_Ver2', { NAME => 'M', TTL => 5184000, VERSIONS => 1, DATA_BLOCK_ENCODING => 'PREFIX' }, {SPLITS=>["\x01\x0

list

exit
```

创建的过程如下:

```
lx2@iZuf6lnha4v6rymngnvx79Z:~$
lx2@iZuf6lnha4v6rymngnvx79Z:~$ /pinpoint/hbase-1.2.6.1/bin/hbase shell hbase-create.hbase
2018-09-12 16:42:05,389 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
0 row(s) in 1.8320 seconds

0 row(s) in 4.2480 seconds

0 row(s) in 2.2670 seconds

0 row(s) in 1.2200 seconds

0 row(s) in 1.2290 seconds

0 row(s) in 1.2250 seconds

0 row(s) in 1.2220 seconds

0 row(s) in 1.2230 seconds

0 row(s) in 1.2250 seconds

0 row(s) in 8.2490 seconds

0 row(s) in 1.2230 seconds
```

2.7 确认初始化成功

进入如下 hbase 的 web 管理界面:
http://10.1.5.204:16010/master-status
看到有如下表就表示初始化正常



第三章 服务端 web/col 程序部署

3.1 安装 pinpoint-web

下载地址：https://github.com/naver/pinpoint/releases
下载 war 包，并解压到指定目录

```
unzip pinpoint-web-1.8.0-RC1.war -d /pinpoint/tomcat-8080-pp/webapps/ppweb
```

3.2 安装 pinpoint-collector

下载地址：https://github.com/naver/pinpoint/releases
下载 war 包，并解压到指定目录

```
unzip pinpoint- collector-1.8.0-RC1.war -d /pinpoint/tomcat-8080-pp/webapps/ppcol
```

3.3 修改配置（不是必须）

服务端程序通过 zk 连接 hbase，本次测试时都放在同一台机器，所以以下内容可以不修改，如果 zk 在其他地方需要修改以下两个文件内容：

/pinpoint/tomcat-8080-pp/webapps/ppweb/WEB-INF/classes/hbase.properties
/pinpoint/tomcat-8080-pp/webapps/ppcol/WEB-INF/classes/hbase.properties
文件中以下两行按照实际进行修改。

```
hbase.client.host=localhost
hbase.client.port=2181
```

3.4 启动服务端

其实就是启动 tomcat

```
/pinpoint/tomcat-8080-pp/bin/startup.sh
```

第四章 客户端 agent 安装

这一步在需要监控的客户机上操作，其工作过程大致就是运行时包含进去一个特殊的 jar 包，该包会将程序内部的各种请求发送到服务端。

6.1 安装 pinpoint-agent

下载地址：https://github.com/naver/pinpoint/releases
下载 war 包，并解压到指定目录

```
mkdir -pv /data/pp-agent/
tar xf pinpoint-agent-1.8.0-RC1.tar.gz -c /data/pp-agent
```

修改配置文件
/data/pp-agent/pinpoint.config
中的内容，保证如下配置项指向的是服务端：

```
profiler.collector.ip=10.1.5.204
```

6.2 spring boot 启动

启动命令中添加三个参数，例如
-javaagent:/data/pp-agent/pinpoint-bootstrap-1.8.0-RC1.jar
#jar 包的路径
-Dpinpoint.agentId=openapiuat-1
#agentid，对于多机部署的情况，使用这个加以区分，且全局唯一，即不同 name 下也要不同
-Dpinpoint.applicationName=open-api-uat
#应用名字，必须唯一，不同应用使用不同的名字

例如：
nohup java -jar -javaagent:/data/pp-agent/pinpoint-bootstrap-1.8.0-RC1.jar -Dpinpoint.agentId=open-api-1 -Dpinpoint.applicationName=open-api-uat /data/louxe/louxe-open-api.jar --server.port=8081 > /data/louxe/open-api.log &

特别提醒，经验证，

1) applicationName 不支持/ +等特殊符号，但可以使用中横线-，下划线_等，否则无法上报数据。
2) applicationName 不可太长，我设置为 louxe-manage_merchapi_yycontrol 这样就太长了（无法上报），具体限制未验证

6.3 tomcat 启动

在 tomcat/bin/catalina.sh 中添加以下内容即可：

```
CATALINA_OPTS="$CATALINA_OPTS -javaagent:/data/pp-agent/pinpoint-bootstrap-1.8.0-RC1.jar"
CATALINA_OPTS="$CATALINA_OPTS -Dpinpoint.agentId=louxe-manage-web-uat-1"
CATALINA_OPTS="$CATALINA_OPTS -Dpinpoint.applicationName=louxe-manage-web-uat"
```

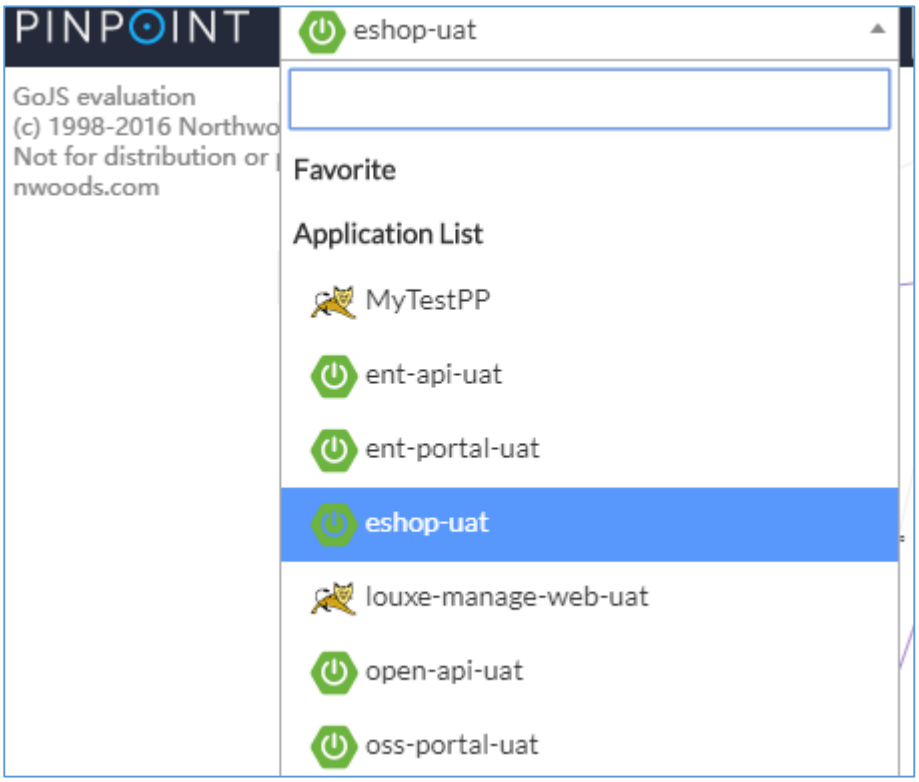
各项的含义于 4.2 相同

第五章 观察效果

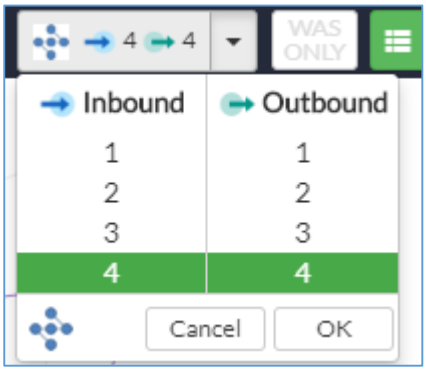
使用浏览器进入以下地址：

http://10.1.5.204:8080/ppweb

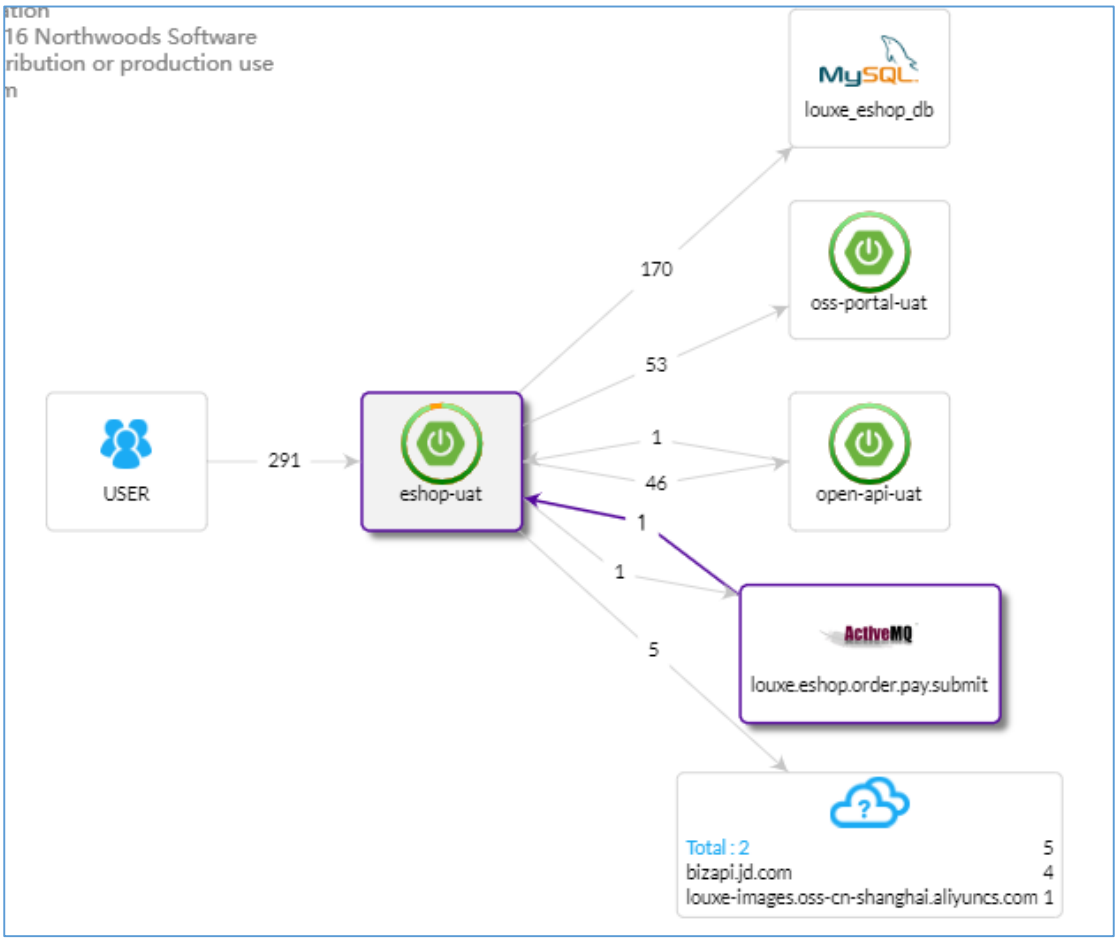
如下图，这里选择 app name



如下图图，可以选择某个应用进来和出去的交互的应用显示的范围或者深度：



如下是其中一个拓扑图

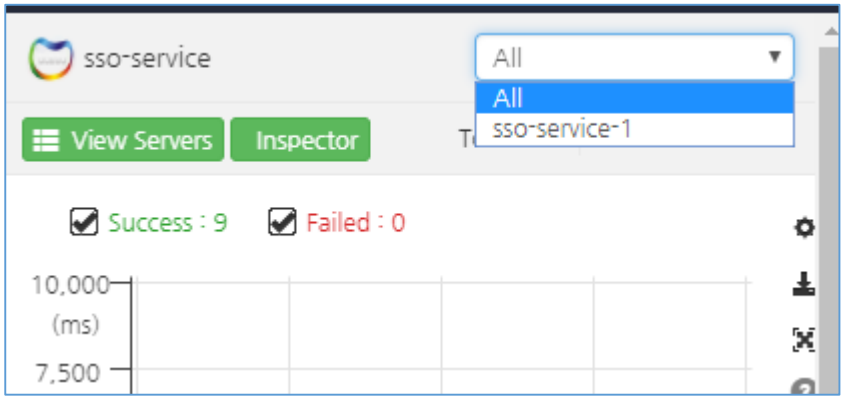


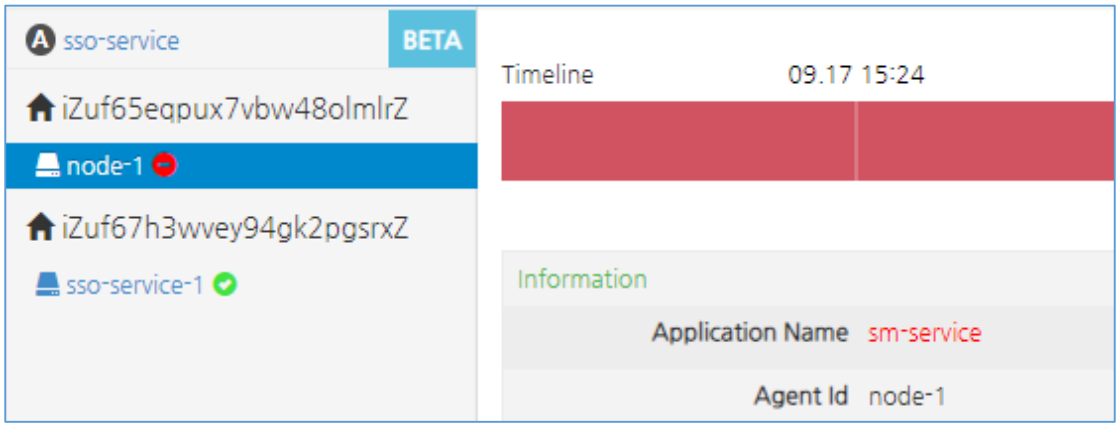
功能很强大，还有很多其他功能，这不是本文档的重点，不再介绍。

第六章 其他

6.1 客户端上报信息修改

如下图所示，其中给一个节点 node-1 失效了，造成这个的原因是，在 4.2/4.3 中配置时，Dpinpoint.agentId 参数变动了。我开始配置了很多 node-1/node-2 等，后来修改成唯一，但这个记录还是在的，需要修改 hbase 库进行订正。





进入 hbase 查看，输入以下命令，可看到表里面的内容

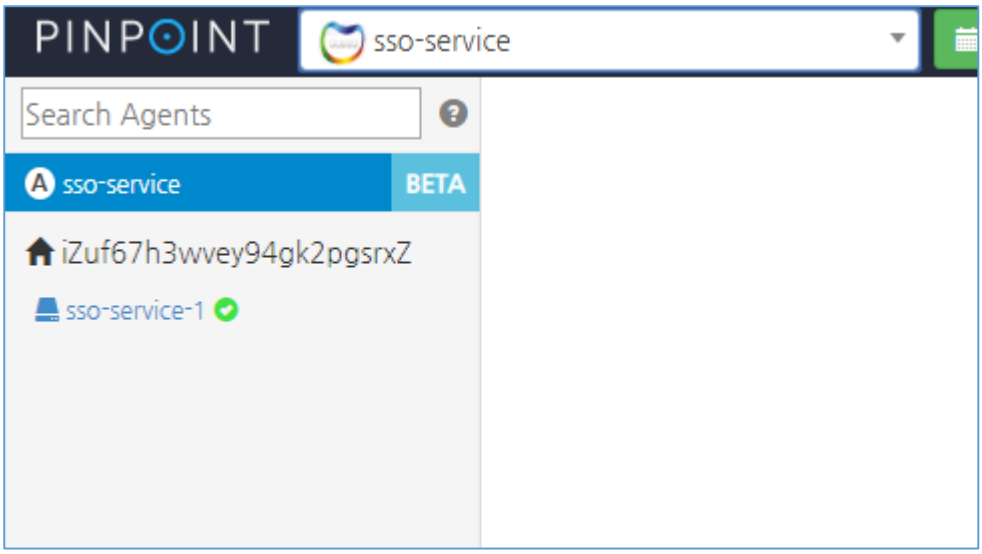
scan 'ApplicationIndex'

```
sm-service                              column=Agents:node-1, timestamp=1536891080806, value=\x04V
sm-service                              column=Agents:sm-service-1, timestamp=1539140059889, value=\x04V
sm-service                              column=Agents:sm-service-2, timestamp=1539162574125, value=\x04V
sso-service                             column=Agents:node-1, timestamp=1536826090060, value=\x04V
sso-service                             column=Agents:sso-service-1, timestamp=1539160965002, value=\x04V
19 row(s) in 0.0910 seconds

hbase(main):037:0> get 'ApplicationIndex','sso-service',{COLUMN=>Agents:node-1, timestamp=1536826090060, value=\x04V}
```

使用如下命令删除这条记录即可。

delete 'ApplicationIndex','sso-service','Agents:node-1'

删除后，不再有 node-1 节点，如下图：



6.2 hbase 数据重新生成

如果需要将 hbase 的数据迁移，或则删除所有旧的数据并重新初始化，操作步骤如下:

1） 停止 hbase
2） 停止 tomcat-pp，即第三章启动的程序
3） 删除 hbase-date 目录里面的所有数据
rm –rf /pinpoint/hbase-data/*
4） 删除 zookeeper 里面的所有数据
rm –rf /pinpoint/hbase-zk-data/*
5） 重新启动 hbase
6） 重新启动 tomcat-pp