



链滴

# SkyWalking 8.3 快速上手教程

作者: [CodingOX](#)

原文链接: <https://ld246.com/article/1606895244388>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 前言

Skywalking 5.0 和 6.0 都写过安装记录，但是存储用的都是 [Elasticsearch](#)

这次随着 8.3 版本的更新，修改了之前版本的一些问题，存储也替换为更为简单的 [MySQL](#) 了，这里小的做一个记录。

2020年12月02日 修改为 [skwalking 8.3](#) 版本

## 正文

Skywalking 整体分为 3 个部分：[agent](#), [backend](#), [UI](#)

- [agent](#) 代理，应用启动的时候后，通过该 [agent](#) 收集数据并上报 [backend](#)。
- [backend](#) 数据收集器，[agent](#) 收集的数据会发送给 [backend](#)，然后数据处理后入库。
- [ui](#) 也就是提供可视化服务的界面。

在安装前，请在官方网址进行下载：[下载地址](#)，下载的时候，请一定注意了不是第一行的 [Source Code](#) 是第二行的 [Binary Distribution](#)。

## backend

如上所述：[backend](#) 的目的是收集 [agent](#) 的上报的数据，然后持久化到存储中，起到承上启下的功能所以首先安装 [backend](#)。

## 前置知识

与之先关的目录和文件主要是如下几个：

- `bin`目录下的`oapService.sh`，它就是 `backend` 的启动脚本
- `config`目录下的`application.yml`，它是`backend`的配置信息
- `oap-libs`是`backend`启动需要的所有`jar`包
- `logs`目录下的`skywalking-oap-server.log`是启动后的日志文件。

## 启动步骤

以下内容是基于MySQL作为存储来讲解的。

第一步：修改`config`下的配置文件`application.yml`

- `storage.selector`修改为`mysql`，默认是用于演示的`h2`，你也可以选择自己的存储介质。
- 然后找到 `store`下的 `mysql`节点
- 修改 `mysql.properties`中基本信息：`jdbcUrl/dataSource.user/dataSource.password`

2020年08月10日 补充

低估了 `skywalking` 使用数据空间的能力，差点导致了生产事故，幸好收到了短信提示，及时处理。里补充下注意事项

在该配置文件下还有：`recordDataTTL`和`metricsDataTTL`，表示的数据数据保存的时间，单位是天分别为 3天和 7天。

我这个服务有 3 个服务节点，12 个小时就占用了18G 的内存空间，这样算下来，基本上一个节点一时最起码就占用了 0.5G 的空间。

所以请大家一定要注意评估你数据库的存储空间！

第二步：将`mysql`的驱动包，拷贝到`oap-libs`目录中。官方可没有自带`mysql`的驱动包，所以请自行载后拷贝到这个目录。

- `mysql8`一定要找对应的`mysql8`的 `jar` 包，不要找`mysql5.7`的驱动包，[MySql8驱动](#)
- `mysql5.7`的数据库，最好也是5.7的（理论上 8 的也是兼容），[MySQL5.7驱动](#)

如果上述流程完成后，尝试运行`bin`目录下的`oapService.sh`

```
sh oapService.sh
```

然后查看`logs/skywalking-oap-server.log`的日志，如果失败会有日志，接下来你就见招拆招吧。

启动后，你可以通过如下几种方式观察是否成功。

第一就是通过查看端口运行情况判定：

```
lsof -i:12800 #数据上报端口，很重要的端口
lsof -i:11800
```

如果端口被占用，日志又没有错误保存，那么基本启动成功了。

第二就是查看`jdbcUrl`对应的数据库是否已经默认创建好了数据表。

## Bug1

我在本文的时候，发现日志报错了一个错误，提示：[Data too long for column 'statement' at row 1。](#)

简单查了下应该是Segment表的字段statement，原本是200的长度，我将其改为2000后，该问题前为止没有再输出。

还有就是top\_n\_database\_statement有这个字段，我暂时也将其修改到了2000。

同时提了issue官方也很快做出了响应，建议修改为2000。[直达地址](#)

## Bug2

提示：[Specified key was too long; max key length is 767 bytes](#)

最简单的方法就是将参数：[innodb\\_large\\_prefix](#)修改为ON，其他方法请自行百度了解

## ui

该组件将数据可视化出来，供大家更直观的查看数据。

## 前置知识

与之关联的目录和文件大体如下：

- bin下的webappService.sh为UI的启动脚本
- webapp目录，其中包括了2个重要的东西
  - webapp.yml，包括UI启动的端口和上报地址
  - skywalking-webapp.jar，启动jar包
- logs下的文件 webapp.log

## 启动步骤

第一步：修改webapp.yml，其默认的端口是经典的8080，所以还是修改为一个特殊点的端口，避免动出错。

第二部：修改collecotr.ribbon.listofServers的ip实际运行的ip

第三步：通过bin下的webappService.sh启动

```
sh webappService.sh
```

不要看到控制台输出SkyWalking Web Application started successfully!就认为OK了。

还是要自己看看日志webapp.log有没有报错内容。

## agent

agent的作用是同应用一起启动，将应用的指标数据上传。agent我记得是可以共享的，就是所有的

用都用同一个 agent，但是为了安全，还是建议每个应用一个 agent。

与之相关的目录为agent，其中最关键的就是agent/config/agent.config这个配置文件

以下是使用步骤：

1. 拷贝一份 agent到任意一个你喜欢的地方，我喜欢放在 jar包的同级目录。
2. 修改 agent.config中的参数
  1. agent.service\_name修改实际的，同一个应用多个实例，其 service\_name相同
  2. agent.instance\_name每个实例都应该是唯一的，建议自行修改，这样UI上可视化后，更有意义。
  3. collector.backend\_service修改为实际的backend所在主机的ip和端口。
3. 在 jar包启动脚本中添加如下命令 `-javaagent:/xxx/agent/skywalking-agent.jar`

## 结语

按照上面的步骤后，启动应用，你可以通过UI的界面去看看是否 OK

以上是skywalking 8.3在使用MySQL时的最基础的配置，更多的信息请参考：[官方文档](#)

当然你有问题，也可以留言讨论。