# SeaTunnel部署

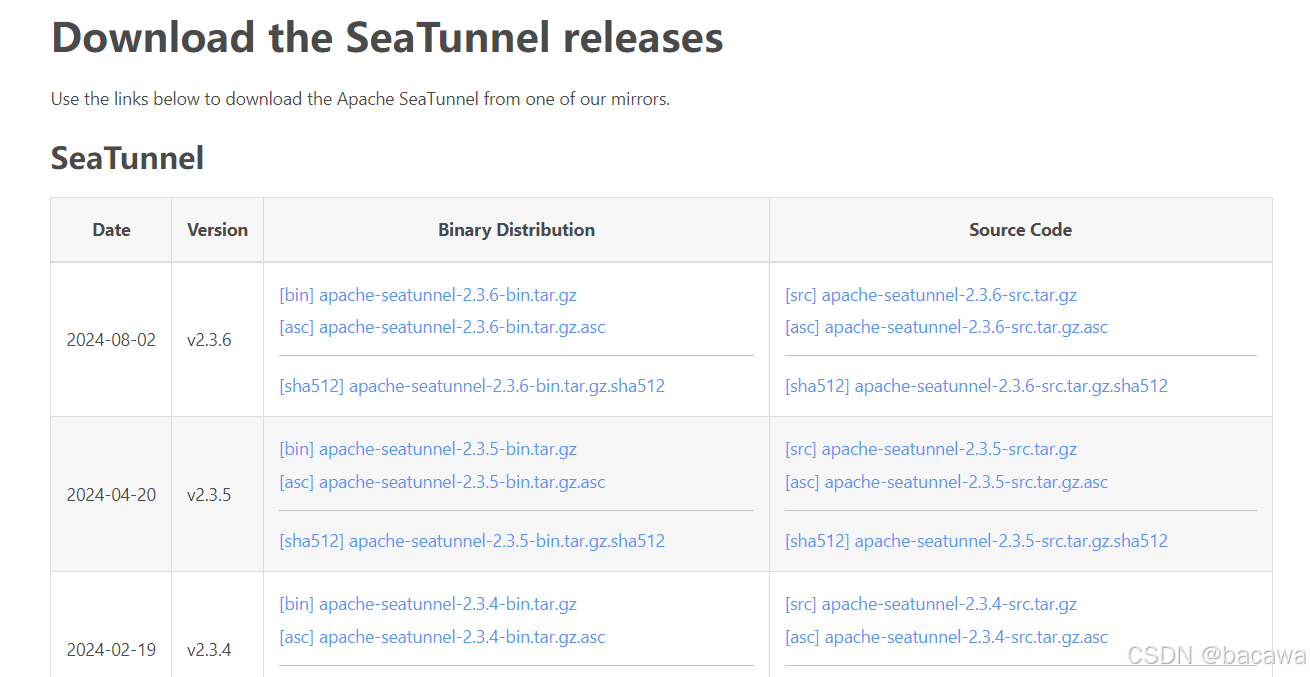
**一、准备工作**

**1.1 环境**

|  |  |
| --- | --- |
| 类目 | 要求（基础环境安装请自行百度） |
| 操作系统 | CentOS 7.6.18\_x86\_64 内核 3.10.0-1160.el7.x86\_64 |
| Java | Java 8 或 11， 其他高于Java 8的版本理论上也可以工作， 以及设置 JAVA\_HOME |
| Maven | 版本>= 3.6.3 |
| 数据库 | Mysql 8.x |

**1.2 下载**

进入 SeaTunnel [下载页面](https://seatunnel.apache.org/download/) 下载最新版本的发布版[安装包](https://so.csdn.net/so/search?q=%E5%AE%89%E8%A3%85%E5%8C%85&spm=1001.2101.3001.7020" \t "_blank)https://seatunnel.apache.org/download/



**二、SeaTunnel安装**

**2.1 解压安装包**

tar -xvf apache-seatunnel-2.3.6-bin.tar.gz

**2.2 配置环境变量**

export SEATUNNEL\_HOME=/seatunnel/apache-seatunnel-2.3.6

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

source /etc/profile

**2.3 下载SeaTunnel相关jar包**

要想连接Mysql，Oracle，SqlServer，Hive，Kafka，Clickhouse，Doris等时，还需下载对应的jar包。

SeaTunnel下载connector的jar时，使用mvnw来下载jar包，默认是从https://repo.maven.apache.org 下载，速度有点慢，确实是有点慢，好处是不用再一个个拷贝Jar包，所以我还是建议用这种方式进行下载。

系统默认自动下载时会下载 plugin\_config 文件中的所有连接器，路径如下：

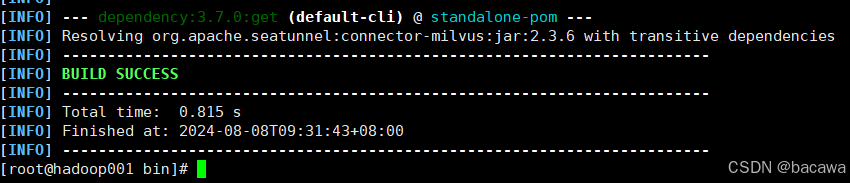
/seatunnel/apache-seatunnel-2.3.6/config/plugin\_config

执行下载命令

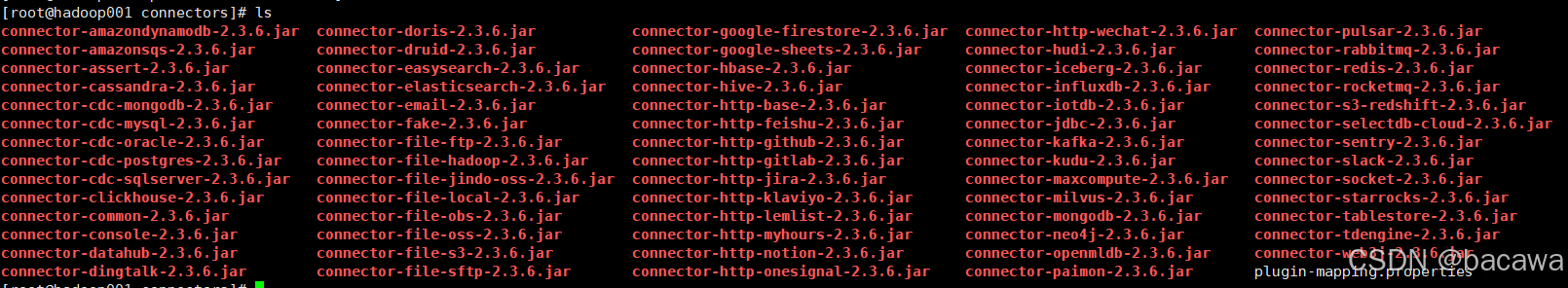
/seatunnel/apache-seatunnel-2.3.6/bin/install-plugin.sh

**下载过程耗时比较长，可以去划个水先，耐心等待即可**

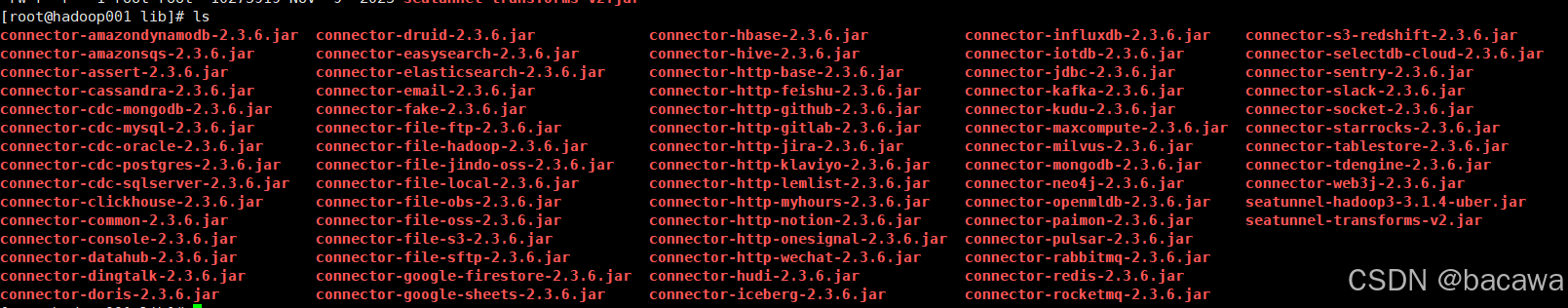
执行完毕之后，会有类似于下图的结果显示页。



自动下载完成之后， 将 /seatunnel/apache-seatunnel-2.3.6/connectors



下所有的jar包都拷贝到 /seatunnel/apache-seatunnel-2.3.6/lib 目录下

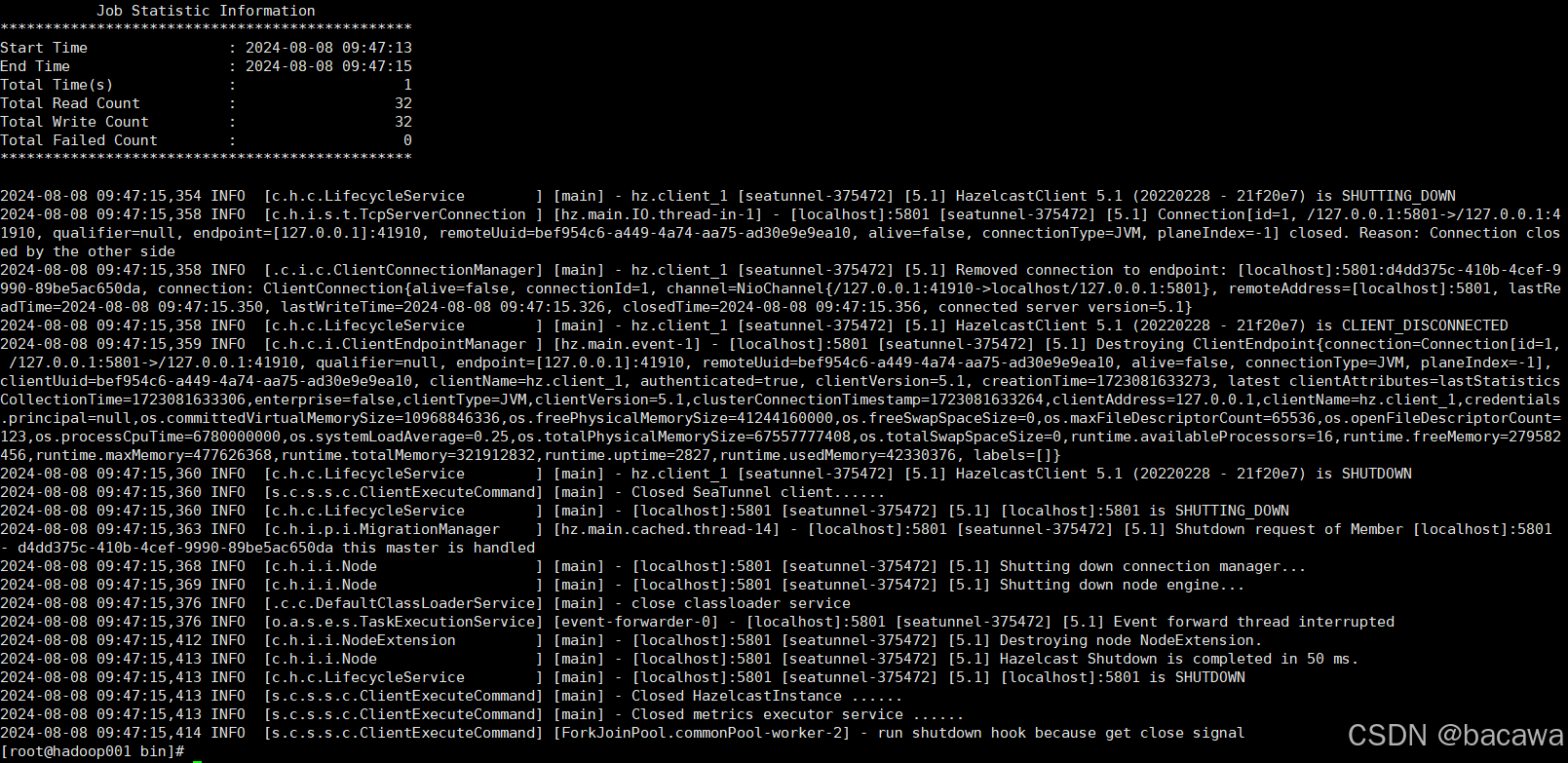


**2.4 测试验证**

*#进入SeaTunnel bin目录 启动服务*

./seatunnel.sh --config ../config/v2.batch.config.template -e local

看到如下页面，证明配置没有问题，就可以启动 Seatunnel 服务了。



**2.5 启动服务**

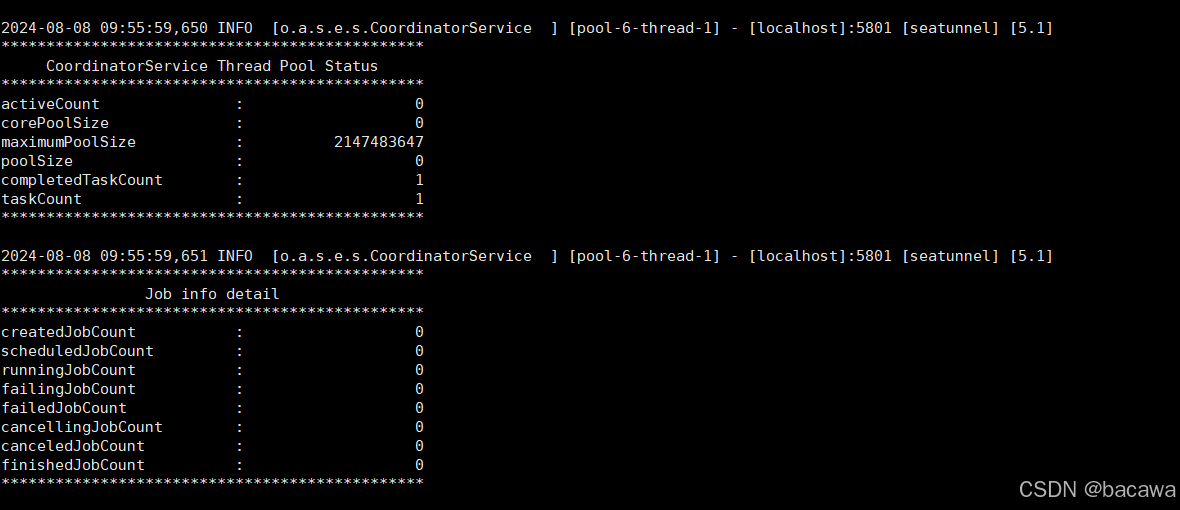
*#进入SeaTunnel bin目录 启动服务*

nohup sh seatunnel-cluster.sh 2>&1 &

在seatunnel的安装目录下查看日志

tail -100f logs/seatunnel-engine-server.log

看到如下所示页面，即标识启动成功。

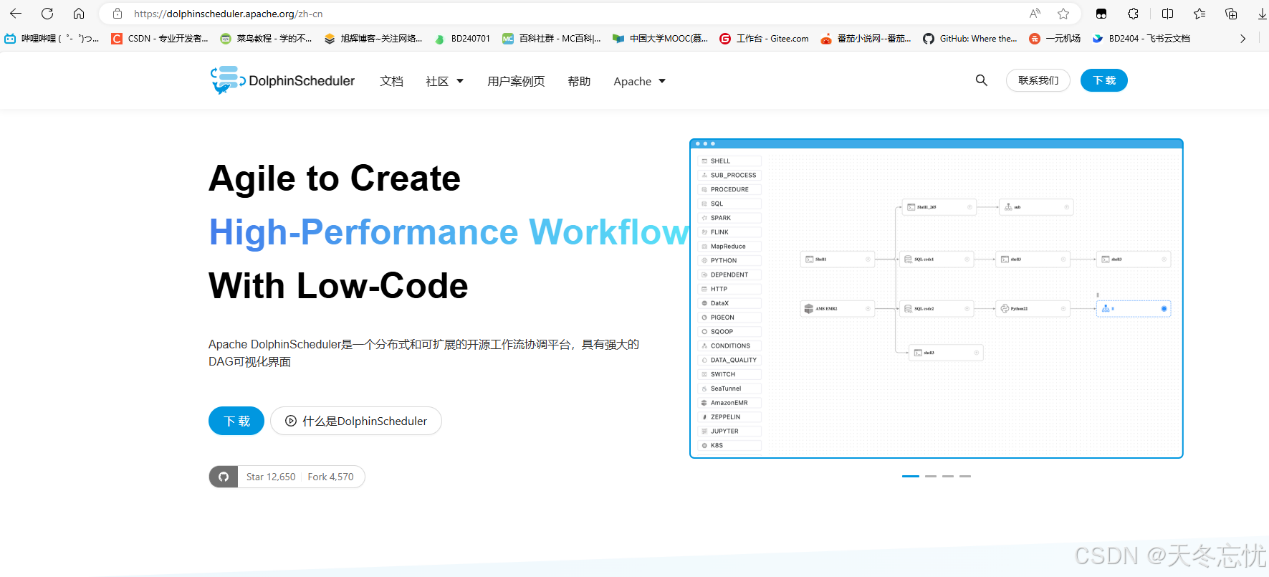


# 海豚调度部署

**1. 下载DolphinScheduler**

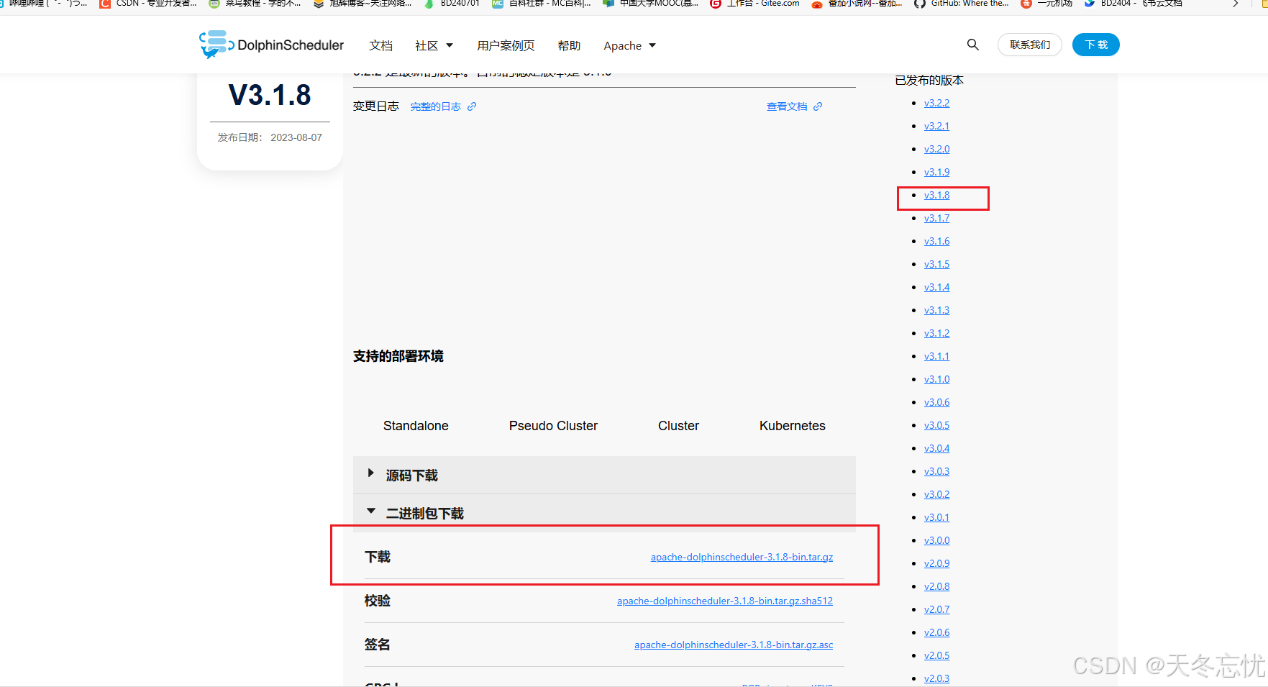
您可以从Apache DolphinScheduler的官方网站或GitHub仓库下载最新版本的DolphinScheduler。

<https://dolphinscheduler.apache.org/>



**2: 选择版本**

在官网的下载页面，你将看到不同版本的 DolphinScheduler。从版本列表中选择 **v3.1.8**。



**3.拷贝MySQL驱动**

将MySQL的JDBC驱动包拷贝到所有服务的jars目录下。我是在/opt/installs/目录下

cd /opt/installs/apache-dolphinscheduler-3.1.8-bin

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./alert-server/libs/standalone-server/

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./api-server/libs/standalone-server/

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./master-server/libs/standalone-server/

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./standalone-server/libs/standalone-server/

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./tools/libs/standalone-server/

cp /opt/installs/hive/lib/mysql-connector-java-8.0.26.jar ./worker-server/libs/standalone-server/

**4. 初始化数据库**

DolphinScheduler使用H2数据库作为默认的存储解决方案。您需要在Linux终端中运行数据库初始化脚本来初始化数据库。

2）在Linux黑窗口运行（连接刚创建的dolphinscheduler数据库）

export DATABASE=mysql

export SPRING\_PROFILES\_ACTIVE=dolphinscheduler

export SPRING\_DATASOURCE\_URL="jdbc:mysql://bigdata01:3306/dolphinscheduler?useUnicode=true&characterEncoding=UTF-8&useSSL=false&allowPublicKeyRetrieval=true"

export SPRING\_DATASOURCE\_USERNAME=dolphinscheduler

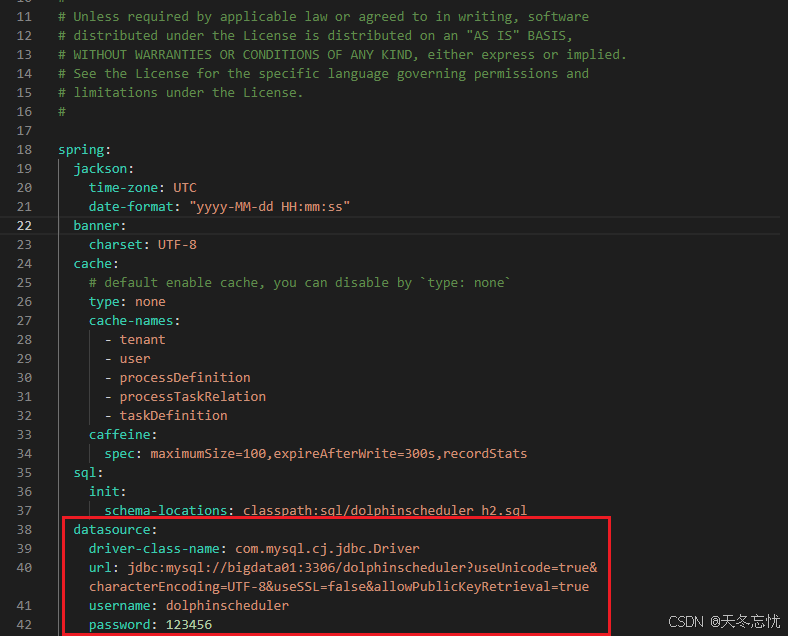
export SPRING\_DATASOURCE\_PASSWORD=123456

**5.初始化数据库（在Linux黑窗口运行）**

bash tools/bin/upgrade-schema.sh

4. 修改 DolphinScheduler 的配置文件

编辑位于apache-dolphinscheduler-3.1.8-bin/standalone-server/conf目录下的application.yaml 文件，将其中关于数据库连接的部分修改为指向你的 MySQL 实例，包括用户名、密码、数据库名等信息。以下是修改后的示例：（注释后第一个spring）



**6. 启动DolphinScheduler**

在Linux命令行中运行以下命令启动DolphinScheduler

注意: 单机模式（Standalone） 仅建议 20 个以下工作流使用，因为其采用内存式的 H2 Database, Zookeeper Testing Server，任务过多可能导致不稳定，并且如果重启或者停止 standalone-server 会导致内存中数据库里的数据清空。

在/opt/installs/apache-dolphinscheduler-3.1.8-bin/目录下运行

如何启动和关闭单机版：

# 启动 Standalone Server 服务

bash ./bin/dolphinscheduler-daemon.sh start standalone-server

# 停止 Standalone Server 服务

bash ./bin/dolphinscheduler-daemon.sh stop standalone-server

# 查看 Standalone Server 状态

bash ./bin/dolphinscheduler-daemon.sh status standalone-server

**7. 访问UI界面**

启动成功后，您可以通过浏览器访问DolphinScheduler的UI界面，默认地址为http://localhost:12345/dolphinscheduler。

localhost为你自己虚拟机的IP地址

默认账号和密码： admin dolphinscheduler123