

Docker部署Seata

参考链接:

<http://seata.io/zh-cn/docs/ops/deploy-by-docker.html>

本示例以1.5.1版本为例，如果你要安装其他版本请自行修改版本号，可以通过下面链接查看镜像支持的版本号

<https://hub.docker.com/r/seataio/seata-server/tags>

1 安装前准备

1.1 准备目录

创建一个专门的目录来存储 seata 相关配置文件和数据

```
1 mkdir /home/seata
2 cd /home/seata
```

1.2 拉取镜像

```
1 docker pull seataio/seata-server:1.5.1
```

1.3 运行镜像

```
1 docker run --name seata -p 8091:8091 -d seataio/seata-server:1.5.1
```

1.4 检查结果

通过 `docker ps` 命令查看，如果看到下图所示的效果表示运行成功。

```
[root@localhost nacos-docker]# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
f8bc45edcff2   seataio/seata-server:1.5.1         "java -Djava.securit..." 30 seconds ago Up 29 seconds
seata
```

如果运行不成功，根据提示解决问题。多半是端口占用，或者容器名称重复等等。

1.5 获取Seata配置

从刚才创建好的容器中获取配置文件

```
1 docker cp seata:/seata-server/resources /home/seata/seata-config
```

```
[root@localhost seata]# docker cp seata:/seata-server/resources /home/seata/seata-config
[root@localhost seata]# ls seata-config/
application.example.yml  application.yml  banner.txt  io  logback  logback-spring.xml  lua
```

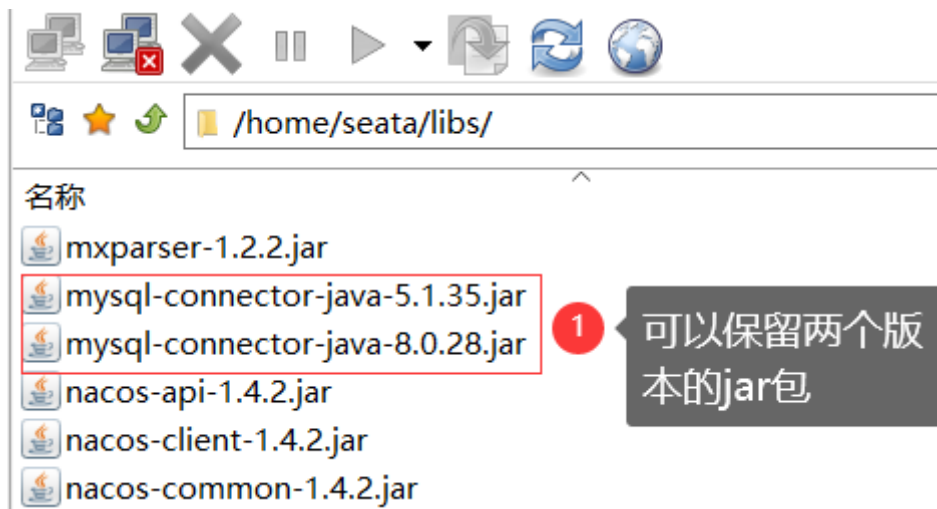
1.6 获取 Seata 库

为了让 seata 支持 mysql8 驱动，此时需要对 libs 进行扩充。所以需要在 seata 的 libs 基础之上进行扩充。

首先复制容器里面的 libs 目录。

```
1 docker cp seata:/seata-server/libs /home/seata/libs
```

然后检查 libs 里面的 jar 包，如果没有 mysql8 驱动 jar 包，需要手动上传 mysql8 版本驱动 jar 包到 /home/seata/libs 目录下面，上传后的效果如下。

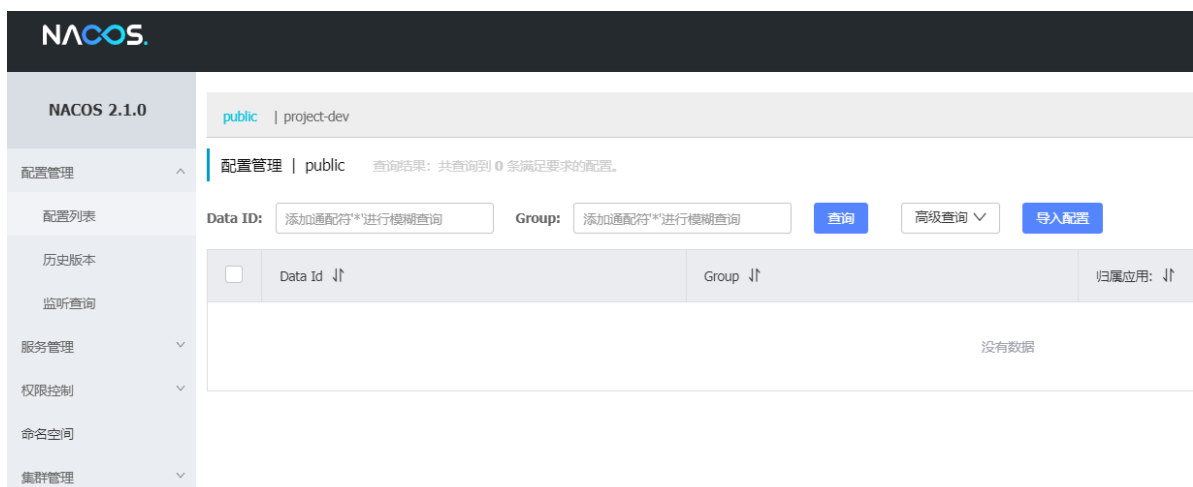


1.7 停止与删除容器

```
1 docker stop seata
2 docker rm seata
```

1.8 检查 Nacos 服务

如果没有启动 nacos 就参照之前的文档，启动后在浏览器，输入 <http://ip:8848/nacos> 用默认的用户名密码登录



1.9 准备 Seata 事务库

执行 `资源/db/server-db.sql` 脚本即可。

1.10 配置防火墙

注意：如果是云服务器端口放行需要到控制台页面去开放，一般情况操作的系统的防火墙是关闭的，不需要通过下面的命令行开放防火墙。

```
1 firewall-cmd --add-port 8091/tcp --permanent
2 firewall-cmd --reload
```

2 安装 Seata

2.1 修改配置

编辑 `application.yml` 配置

```
1 vi seata-config/application.yml
```

配置内容参考如下，对应参考文件路径为 `资源/seata-config/application.yml`，你可以直接上传到服务器覆盖对应文件后再进行修改。

```
1 server:
2   port: 7091
3 spring:
4   application:
5     name: seata-server
6 logging:
7   config: classpath:logback-spring.xml
8   file:
9     path: ${user.home}/logs/seata
10  extend:
11    logstash-appender:
12      destination: 127.0.0.1:4560
13    kafka-appender:
14      bootstrap-servers: 127.0.0.1:9092
15      topic: logback_to_logstash
16 console:
17   user:
18     username: seata
19     password: seata
20 seata:
21   config:
22     type: nacos
23     nacos:
24       application: seata-server
25       server-addr: 192.168.220.128:8848
26       group: SEATA_GROUP
27       namespace: ""
28       cluster: default
29       username: nacos
30       password: nacos
```

```

31     data-id: seataServer.properties
32 registry:
33     type: nacos
34     nacos:
35         application: seata-server
36         server-addr: 192.168.220.128:8848
37         group: SEATA_GROUP
38         namespace: ""
39         cluster: default
40         username: nacos
41         password: nacos
42     store:
43         mode: db
44         db:
45             datasource: druid
46             db-type: mysql
47             driver-class-name: com.mysql.cj.jdbc.Driver
48             url: jdbc:mysql://192.168.220.128:3306/seata?
rewriteBatchedStatements=true&useUnicode=true&characterEncoding=utf8&autoRec
onnect=true&useSSL=false&serverTimezone=Asia/Shanghai&allowPublicKeyRetrieval=true
49         user: root
50         password: 123456
51         min-conn: 5
52         max-conn: 100
53         global-table: global_table
54         branch-table: branch_table
55         lock-table: lock_table
56         distributed-lock-table: distributed_lock
57         query-limit: 100
58         max-wait: 5000
59     security:
60         secretKey: SeataSecretKey0c382ef121d778043159209298fd40bf3850a017
61         tokenValidityInMilliseconds: 1800000
62         ignore:
63             urls:
/,/**/*.css,//**/*.js,//**/*.html,//**/*.map,//**/*.svg,//**/*.png,//**/*.ico,/con
sole-fe/public/**,/api/v1/auth/login

```

注意：需要修改上面 `seata.config`、`seata.registry`、`seata.store` 节点对应的 IP、端口以及用户名和密码。

详细配置可以参考 `application.example.yml`，该文件存放了所有可使用的详细配置。

2.2 服务编排

在 `/home/seata` 目录中建立 `docker-compose.yml`，并写入下列内容。

```

1 version: "3"
2 services:
3     seata-server:
4         image: seataio/seata-server:1.5.1
5         hostname: seata-server
6         container_name: seata-server
7         ports:

```

```
8      - "8091:8091"
9      environment:
10         - SEATA_PORT=8091
11         - SEATA_IP=192.168.220.128
12      volumes:
13         - /home/seata/seata-config:/seata-server/resources
14         - /home/seata/libs:/seata-server/libs
15         - /home/seata/sessionStore:/seata-server/sessionStore
16         - /home/seata/logs:/root/logs
```

注意：修改成你的服务器 IP

2.3 启动服务

首先进入启动服务试试，执行下面的命令

```
1 docker-compose up
```

启动成功后可以看到下图所示的效果

```
[root@localhost seata]# docker-compose up
[+] Running 2/2
 # Network seata_default Created
 # Container seata-server Created
Attaching to seata-server
seata-server | SEATA
seata-server |
seata-server |
seata-server |
seata-server |
seata-server |
seata-server | 12:30:41.410 INFO --- [ main] io.seata.
arting ServerApplication using Java 1.8.0_212 on seata-server with PID 1 (/s
in /seata-server)
seata-server | 12:30:41.414 INFO --- [ main] io.seata.
active profile set, falling back to default profiles: default
seata-server | 12:30:43.196 INFO --- [ main] o.s.b.w.e
mcats initialized with port(s): 7091 (http)
seata-server | 12:30:43.215 INFO --- [ main] o.a.coyot
initializing ProtocolHandler ["http-nio-7091"]
seata-server | 12:30:43.217 INFO --- [ main] o.apache.
arting service [Tomcat]
seata-server | 12:30:43.217 INFO --- [ main] org.apach
arting Servlet engine: [Apache Tomcat/9.0.55]
seata-server | 12:30:43.316 INFO --- [ main] o.a.c.c.C
initializing Spring embedded WebApplicationContext
seata-server | 12:30:43.317 INFO --- [ main] w.s.c.Ser
ot WebApplicationContext: initialization completed in 1827 ms
seata-server | 12:30:43.992 INFO --- [ main] o.s.b.a.w
ding welcome page: class path resource [static/index.html]
seata-server | 12:30:44.323 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/'] with []
seata-server | 12:30:44.324 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.css'] with []
seata-server | 12:30:44.324 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.js'] with []
seata-server | 12:30:44.325 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.html'] with []
seata-server | 12:30:44.332 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.map'] with []
seata-server | 12:30:44.332 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.svg'] with []
seata-server | 12:30:44.332 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.png'] with []
seata-server | 12:30:44.333 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/**/*.ico'] with []
seata-server | 12:30:44.333 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/console-fe/public/**'] with []
seata-server | 12:30:44.333 INFO --- [ main] o.s.s.web
ll secure Ant [pattern='/api/v1/auth/login'] with []
seata-server | 12:30:44.389 INFO --- [ main] o.s.s.web
ll secure any request with [org.springframework.security.web.context.request
ter@3dbd7107, org.springframework.security.web.context.SecurityContextPersis
ework.security.web.header.HeaderWriterFilter@149274cb, org.springframework.s
goutFilter@670342a2, io.seata.console.filter.JwtAuthenticationTokenFilter@4a
.web.savedrequest.RequestCacheAwareFilter@2f95653f, org.springframework.secu
olderAwareRequestFilter@646427f7, org.springframework.security.web.authentic
5f8a02cf, org.springframework.security.web.session.SessionManagementFilter@e
.web.access.ExceptionTranslationFilter@74c9e11, org.springframework.security
Interceptor@790d8fdd]
seata-server | 12:30:44.434 INFO --- [ main] o.a.coyot
arting ProtocolHandler ["http-nio-7091"]
seata-server | 12:30:44.465 INFO --- [ main] o.s.b.w.e
mcats started on port(s): 7091 (http) with context path ''
seata-server | 12:30:44.479 INFO --- [ main] io.seata.
arted ServerApplication in 4.065 seconds (JVM running for 4.92)
seata-server | 12:30:44.780 INFO --- [ main] i.s.core.
rver started, service listen port: 8091
seata-server | 12:30:44.799 INFO --- [ main] io.seata.
ata server started in 317 milliseconds
```

进入 Nacos 注册中心控制面板查看是否有 seata 服务，如下图所示。



如果一切正常，那么服务器安装成功了，接下来可以切换到后台启动模式。

ctrl + c 结束前台启动，使用下面命令进行后台启动。

```
1 docker-compose up -d
```

3 扩展内容

3.1 配置中心填充

在你 seata 所在的 namespace 中新建配置，配置 data-id 为 seataServer.properties，分组为 SEATA_GROUP，这些属性和我们前面安装 seata 修改的配置是匹配的，如下图所示。

新建配置

* Data ID: seataServer.properties 1

* Group: SEATA_GROUP 2

[更多高级选项](#)

描述:

配置格式: ☐ TEXT ☐ JSON ☐ XML ☐ YAML ☐ HTML ☒ Properties 3

* 配置内容: ? :

```
1 service.vgroupMapping.project_tx_group=default
2 store.mode=db
3 store.db.datasource=druid
4 store.db.dbType=mysql
5 store.db.driverClassName=com.mysql.cj.jdbc.Driver
6 store.db.url=jdbc:mysql://192.168.220.128:3306/seata?rewriteBatch
7 store.db.user=root
8 store.db.password=123456
```

配置内容示例

```
1 service.vgroupMapping.project_tx_group=default
2 store.mode=db
3 store.db.datasource=druid
4 store.db.dbType=mysql
5 store.db.driverClassName=com.mysql.cj.jdbc.Driver
6 store.db.url=jdbc:mysql://192.168.220.128:3306/seata?
  rewriteBatchedStatements=true&useUnicode=true&characterEncoding=utf8&autoReco
  nnect=true&useSSL=false&serverTimezone=Asia/Shanghai&allowPublicKeyRetrieval=
  true
7 store.db.user=root
8 store.db.password=123456
```

发布成功后，可以看到下图所示效果。

public | project-dev

配置管理 | public 查询结果: 共查询到 1 条满足要求的配置。

Data ID: Group: 查询 高级查询 ▾ 导入配置

<input type="checkbox"/>	Data Id ⚡	Group ⚡	归属应用: ⚡
<input type="checkbox"/>	seataServer.properties	SEATA_GROUP	

更多配置内容参考下面链接，选择性的复制配置到你的项目中。

<https://github.com/seata/seata/blob/develop/script/config-center/config.txt>

3.2 数据库表补充

需要在业务数据库中新增 `undo_log` 表，表的内容参考 `资源/db/client-db.sql` 脚本。