Seata 分布式事务解决方案环境搭建

seata 官网：<http://seata.io/zh-cn/>

seata 版本：1.2.0

seata事务方案：AT模式

spring-cloud-alibaba 版本：2.2.1.RELEASE

本案例中需要其它组件：mysql，nacos

# 下载seata服务端

<http://seata.io/zh-cn/blog/download.html>

# 创建seata服务端数据库与相关表

1)：创建数据库名称为seata (可以是其它名称，后面相关db配置统一即可)

2)：执行以下sql脚本

SQL脚本获取：<https://github.com/seata/seata/tree/develop/script/server/db>

# 创建客户端业务模块undo\_log表

在参与全局事务业务服务中的数据库中创建undo\_log表，用于事务回滚用的。

SQL脚本获取：<https://github.com/seata/seata/tree/develop/script/client/at/db>

# 配置seata服务端注册与配置中心

1)：修改的文件：seata\conf\registry.conf

2)：修改内容：

registry {

# file 、nacos 、eureka、redis、zk、consul、etcd3、sofa

type = "nacos"

nacos {

application = "seata-server"

serverAddr = "localhost"

namespace = "seata"

cluster = "default"

username = "nacos"

password = "nacos"

}

}

config {

# file、nacos 、apollo、zk、consul、etcd3

type = "nacos"

nacos {

serverAddr = "localhost"

namespace = "seata"

group = "SEATA\_GROUP"

username = "nacos"

password = "nacos"

}

}

# 启动nacos，并创建namespace

上一步中配置了nacos namespace选项，因为seata的配置项很多，在nacos中创建一个namespace用于区分

注意：上一步中的namespace id是seata（可自定义为其它值）

# seata服务端配置提交至nacos配置中心

1)：配置文件参考：<https://github.com/seata/seata/tree/develop/script/config-center>/[config.txt](https://github.com/seata/seata/blob/develop/script/config-center/config.txt)

2)：执行脚本，把config.txt配置提交至配置中心

脚本位置：<https://github.com/seata/seata/tree/develop/script/config-center/nacos>

Nacos-config.py，nacos-config.sh 示例如下：

注意：config.txt在本地的位置可能需要修改.py或.sh脚本中的config.txt的位置

Shell：

sh nacos-config.sh -h localhost -p 8848 -g SEATA\_GROUP -t seata

参数说明：

-h: host, the default value is localhost.

-p: port, the default value is 8848.

-g: Configure grouping, the default value is 'SEATA\_GROUP'.

-t: Tenant information, corresponding to the namespace ID field of Nacos, the default value is ''.

-u: username, nacos 1.2.0+ on permission control, the default value is ''.

-w: password, nacos 1.2.0+ on permission control, the default value is ''.

python:

python ${SEATAPATH}/script/config-center/nacos/nacos-config.py localhost:8848

详细参考：<https://github.com/seata/seata/tree/develop/script/config-center>

备注：本案是使用mysql，可以修改config.txt中的store.db开头的配置，再提交到nacos，或者提交后再到nacos控制台修改即可。

比如（mysql 8）：

store.mode=db

store.db.datasource=druid

store.db.dbType=mysql

store.db.driverClassName=com.mysql.cj.jdbc.Driver

store.db.url=jdbc:mysql://localhost:3306/seata?useUnicode=true&useSSL=false&characterEncoding=utf8&serverTimezone=Asia/Shanghai&allowPublicKeyRetrieval=true

store.db.user=root

store.db.password=root

# 启动seata服务端

脚本：seata\bin\seata-server.bat、seata-server.sh

运行seata-server.sh [-h ip -p 端口] 观察启动有无错误即可

# 客户端项目配置

1)：maven坐标

<dependency>

<groupId>com.alibaba.cloud</groupId>

<artifactId>spring-cloud-starter-alibaba-seata</artifactId>

<exclusions>

<exclusion>

<groupId>io.seata</groupId>

<artifactId>seata-spring-boot-starter</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>io.seata</groupId>

<artifactId>seata-spring-boot-starter</artifactId>

<version>${seata.version}</version>

</dependency>

2)：spring配置

配置文件参考：<https://github.com/seata/seata/tree/develop/script/client/spring>

本示例配置如下application.yml ：

seata:

enabled: true

tx-service-group: my\_test\_tx\_group

enable-auto-data-source-proxy: true

config:

type: nacos

nacos:

server-addr: 127.0.0.1:8848

group: SEATA\_GROUP

namespace: seata

registry:

type: nacos

nacos:

application: seata-server

server-addr: 127.0.0.1:8848

namespace: seata

cluster: default

3)：业务代码示例

@GlobalTransactional(rollbackFor = Exception.class)

public User createRandomUser() {

//本地db操作

User user = new User();

user.setPassword("123456");

user.setUsername(StringUtils.random(6, StringUtils.RandomType.ALL));

user.setRealName("临时用户");

this.baseMapper.insert(user);

//调用远程服务

LogDTO logDto = new LogDTO();

logDto.setCostTime(100L);

logDto.setIp("localhost");

logDto.setMethod("test");

logDto.setLogContent("测试分布式事务");

Result<LogDTO> result = logFeign.add(logDto);

this.logger.info("invoke log feign , result {}", result.isSuccess());

//模拟出错，测试是否回滚事务

int a=10/0; //测试结论：本地与远程服务都会回滚事务的

return user;

}

# 高可用seata服务端搭建

按照以上的第1,6两步即可把你的seata新节点接入到同一个nacos集群,配置&注册中心中,由于是同一个配置中心,所以db也是采用的共同配置.至此高可用搭建已经顺利完结,如果你想测试,仅需关掉其中一个server节点,验证服务是否可用即可.

# 配置出现过的问题与注意事项

1）：出现过的问题

项目中集成了spring-sleuth，导至启动的时候会报feign服务找不到的错误

需配置以下：

spring:

sleuth:

feign:

enabled: false

2）：事务分组配置要一致

客户端配置：

seata:

tx-service-group: **my\_test\_tx\_group**

服务端nacos配置中心的配置：

service.vgroupMapping.**my\_test\_tx\_group**=**default**

服务端seata\conf\registry.conf配置

nacos {

application = "seata-server"

serverAddr = "localhost"

namespace = "seata"

cluster = "**default**"

username = "nacos"

password = "nacos"

}

以上几处蓝色字体配置要一致，否则可能会找不到TC