Seata 官方中文文档：<http://seata.io/zh-cn/docs/overview/what-is-seata.html>

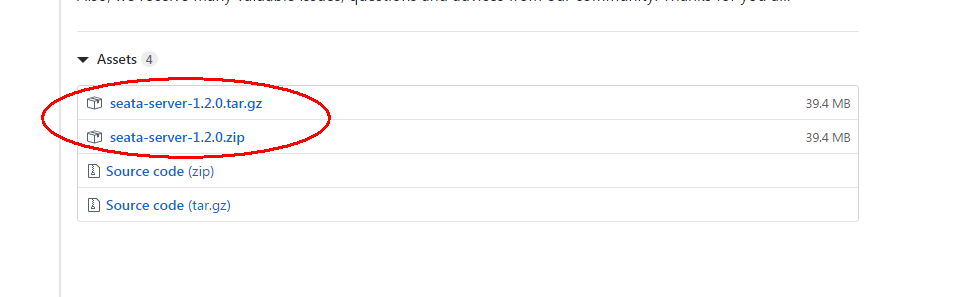
Seata分TC、TM和RM三个角色，TC（Server端）为单独服务端部署，TM和RM（Client端）由业务系统集成。

接下来先介绍server端的安转部署,

目安装最新seata最新版本1.2.0（http://seata.io/zh-cn/docs/ops/deploy-guide-beginner.html）官方有比较全的文档，可参考

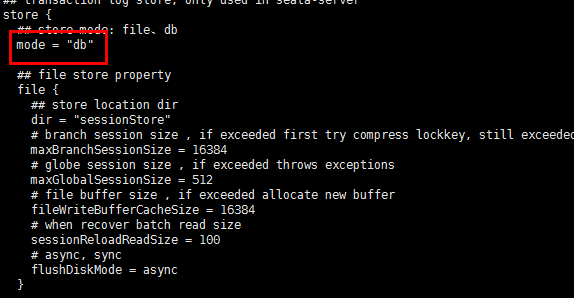
部署流程：

1. 下载安装包：<https://github.com/seata/seata/releases>

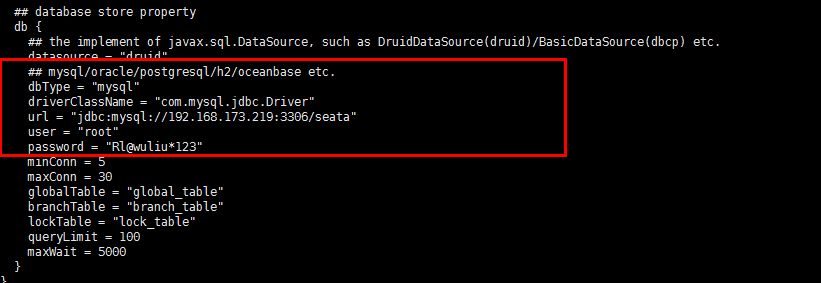


格式不一样而已，这两个都可以

1. 解压 tar -zxvf seata-server-1.2.0.tar.gz
2. 修改conf下配置文件 file.conf、registry.conf

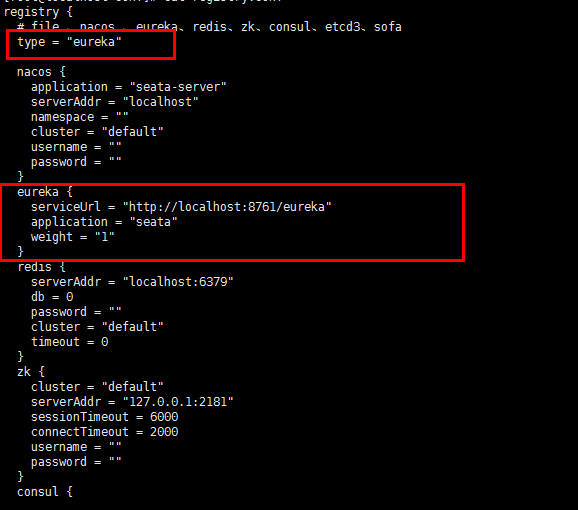


默认配置为file、因为我们要用到注册中心且分布式部署，修改存储模式为db。



修改对应的数据库配置

接下来修改registry.conf



这里的type默认为file，这里修改为eureka,并对eureka配置相应修改

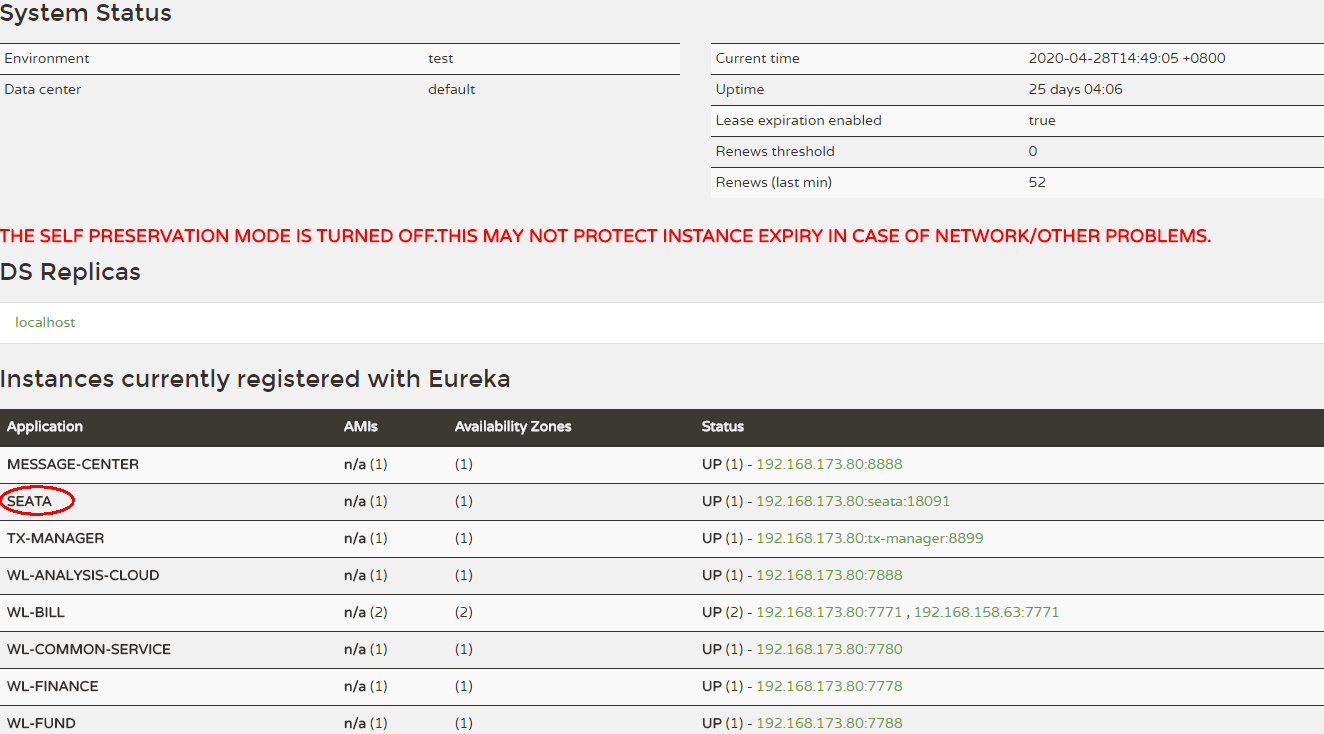
最后还需为server提供单独的库，建表。全局事务会话信息由3块内容构成，全局事务-->分支事务-->全局锁，对应表global\_table、branch\_table、lock\_table

启动命令 bin目录下seata-server.sh -p 18091 -n 1

参数说明：

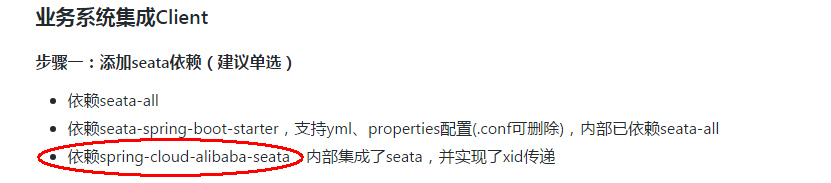
-h: 注册到注册中心的ip -p: Server rpc 监听端口 -m: 全局事务会话信息存储模式，file、db，优先读取启动参数 -n: Server node，多个Server时，需区分各自节点，用于生成不同区间的transactionId，以免冲突 -e: 多环境配置参考 <http://seata.io/en-us/docs/ops/multi-configuration-isolation.html>

看下eureka这边注册上来了



1. 项目整合

查看官方文档发现：



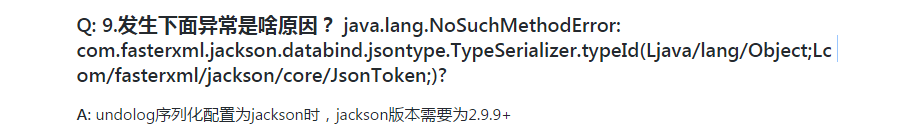
所以这边选用 spring-cloud-alibaba-seata

选用 spring-cloud-alibaba-seata集成方式的前提，还需引入 com.alibaba.cloud 这边有个版本兼容的问题：目前项目中使用的是springCloud E 版本+spring boot 1.5.X

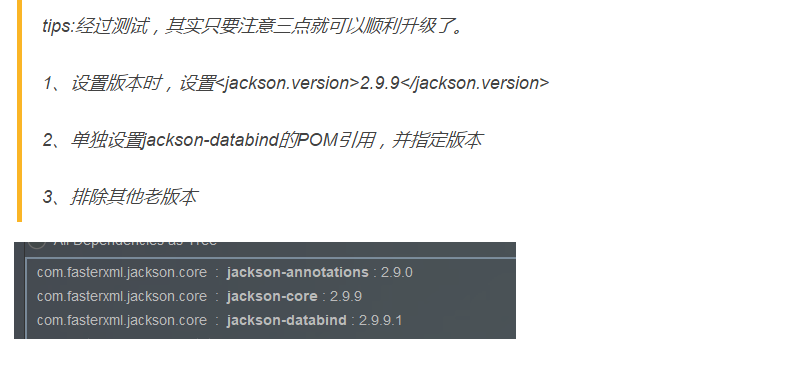
故选用集成 spring cloudAlibaba 1.5.1



另外，发现undolog默认序列化为jackson在,这边要求配置2.9.9+



Ok.换jackson



Pom 文件

、、、、、、、parent

<properties>

<spring-cloud-alibaba.version>1.5.1.RELEASE</spring-cloud-alibaba.version> <spring-cloud-starter-alibaba-seata.version>2.2.0.RELEASE

</spring-cloud-starter-alibaba-seata.version>

<seata.verson>1.2.0</seata.verson>

<jackson-databind.version>2.9.9.3</jackson-databind.version>

<jackson-core.version>2.9.9</jackson-core.version>

<jackson-annotations.version>2.9.0</jackson-annotations.version>

</properties>

<dependencyManagement>

<dependencyManagement>

<dependency>

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

<artifactId>spring-cloud-alibaba-dependencies</artifactId>

<version>${spring-cloud-alibaba.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

<dependency>

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

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

<version>${spring-cloud-starter-alibaba-seata.version}</version>

</dependency>

<dependency>

<artifactId>seata-all</artifactId>

<groupId>io.seata</groupId>

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

</dependency>

<dependency>

<groupId>io.seata</groupId>

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

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

</dependency>

<!-- 解决seata要求jackson版本2.9.9+构建失败问题 -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>${jackson-databind.version}</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-core</artifactId>

<version>${jackson-core.version}</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-annotations</artifactId>

<version>${jackson-annotations.version}</version>

</dependency>

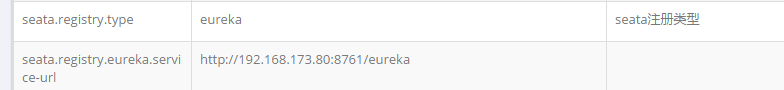
、、、、、、、、

依赖加完了,接下来需要配置下文件了

接下来开始配置

具体参数请查看：<https://seata.io/zh-cn/docs/user/configurations.html>

这边介绍几个必须的配置，模块比较多，公用的配置，这边放在apollo common



以上为公共配置，接下来为每个微服务单独配置：（微服务整合看下面介绍即可）

1.pom 添加依赖

<dependency>

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

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

</dependency>

<dependency>

<groupId>io.seata</groupId>

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

</dependency>

注意：这里有个jar包冲突，如果pom文件依赖了spring-cloud-starter-eureka

将上面的依赖声明在spring-cloud-starter-eureka之前

2.@SpringBootApplication(exclude = GlobalTransactionAutoConfiguration.class)

3.apollo配置（thirdp举例）

seata.tx-service-group thirdpt\_tx\_group

seata.service.vgroup-mapping.thirdpt\_tx\_group seata

4.建表

-- for AT mode you must to init this sql for you business database. the seata server not need it.

CREATE TABLE IF NOT EXISTS `undo\_log`

(

`id` BIGINT(20) NOT NULL AUTO\_INCREMENT COMMENT 'increment id',

`branch\_id` BIGINT(20) NOT NULL COMMENT 'branch transaction id',

`xid` VARCHAR(100) NOT NULL COMMENT 'global transaction id',

`context` VARCHAR(128) NOT NULL COMMENT 'undo\_log context,such as serialization',

`rollback\_info` LONGBLOB NOT NULL COMMENT 'rollback info',

`log\_status` INT(11) NOT NULL COMMENT '0:normal status,1:defense status',

`log\_created` DATETIME NOT NULL COMMENT 'create datetime',

`log\_modified` DATETIME NOT NULL COMMENT 'modify datetime',

PRIMARY KEY (`id`),

UNIQUE KEY `ux\_undo\_log` (`xid`, `branch\_id`)

) ENGINE = InnoDB

AUTO\_INCREMENT = 1

DEFAULT CHARSET = utf8 COMMENT ='AT transaction mode undo table';

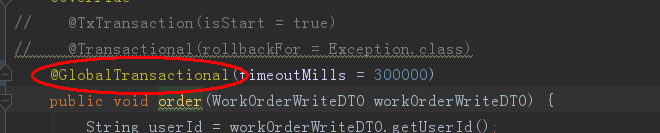
5.使用

@GlobalTransaction 全局事务注解

@GlobalLock 防止脏读和脏写，又不想纳入全局事务管理时使用。（不需要rpc和xid传递等成本）

1.消费者（调用方）

超时时间默认1分钟，可设置



2.（生产者）被调用方加上本地事务即可