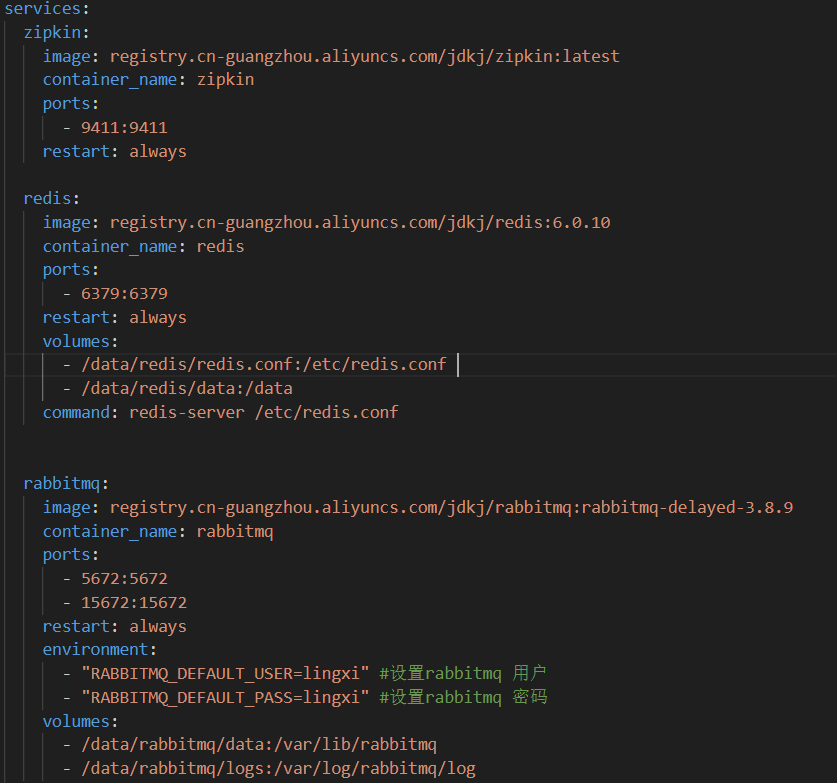
**测试环境部署文档**

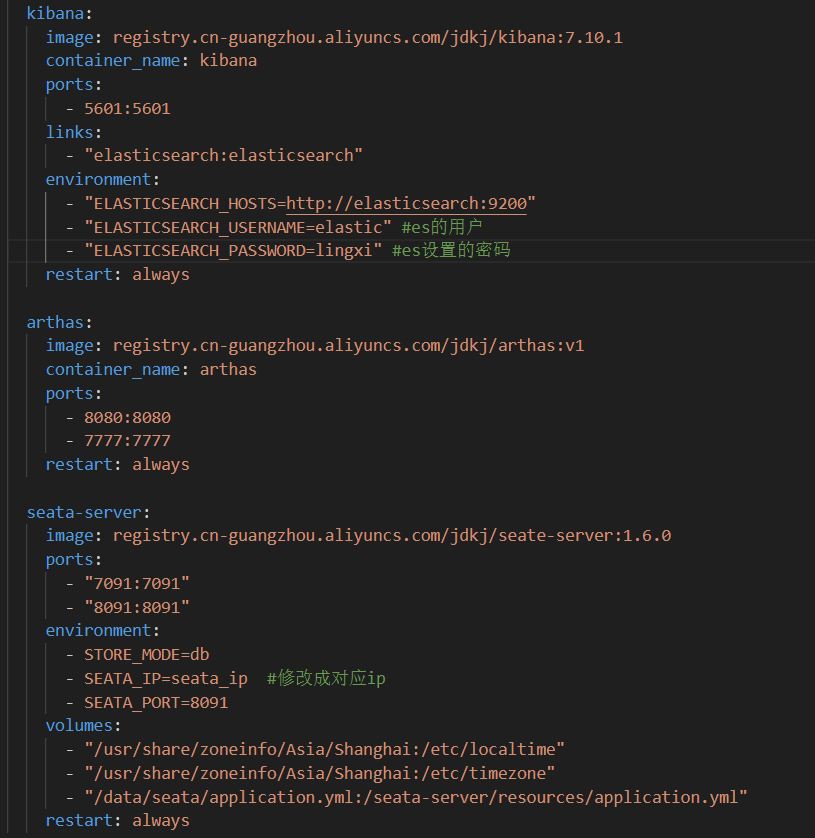
# 1.服务器环境初始化

## 1.1部署应用中间件

(1)连接上服务器, 创建/data目录且创建basic编写docker-compose.yml

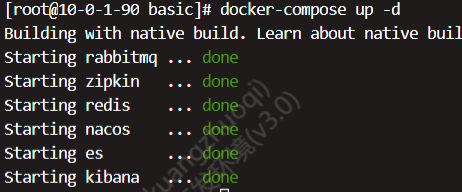






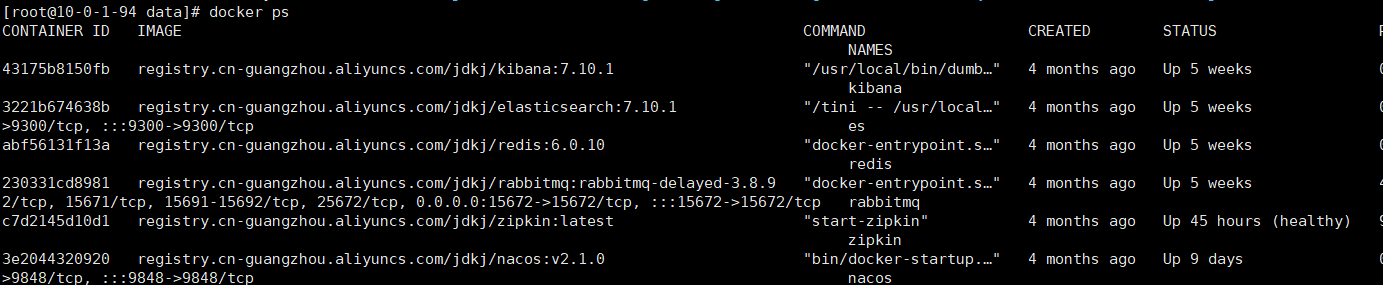
(2) 使用docker-compose启动容器(中间件、服务注册中心)

# docker-compose up -d



1. 查看容器是否启动成功

# docker ps



## 1.2 部署数据库应用

（1）安装 postgresql-12

#yum install -y readline readline-devel openssl openssl-devel zlib zlib-devel

#yum -y install <https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm>

#yum -y install postgresql12

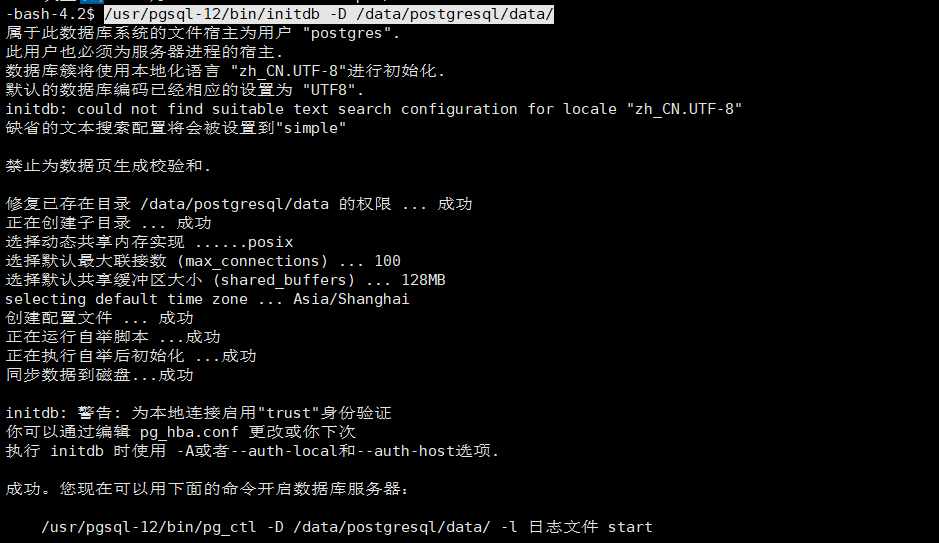
#yum -y install postgresql12-server

#mkdir -p /data/postgresql/data

#chown -R postgres.postgres /data/postgresql/data

#su - postgres

-bash-4.2$ /usr/pgsql-12/bin/initdb -D /data/postgresql/data/



切换到root用户：

#vim /usr/lib/systemd/system/postgresql-12.service

修改Environment=PGDATA=/data/postgresql/data/

#systemctl daemon-reload

#systemctl start postgresql-12

#systemctl enable postgresql-12

#systemctl status postgresql-12

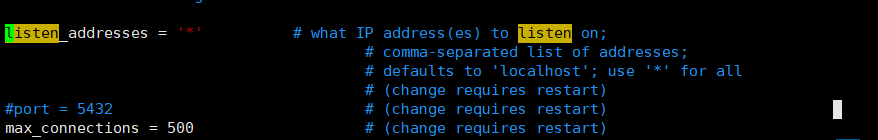
切换用户修改配置文件

#su - postgres

-bash-4.2$ vim /data/postgresql/data/postgresql.conf

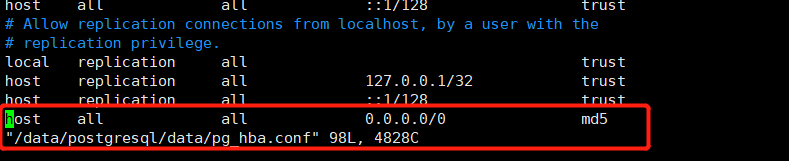
listen\_addresses = '\*'

max\_connections = 500



-bash-4.2$ vim /data/postgresql/data/pg\_hba.conf

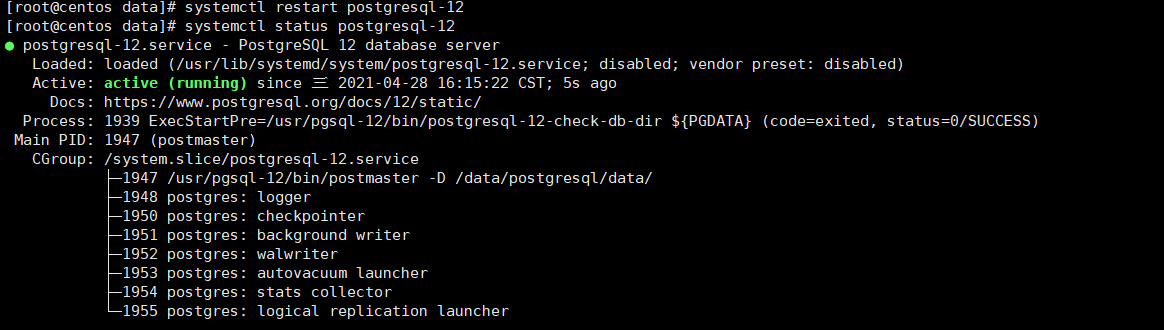
追加host all all 0.0.0.0/0 md5



切换root重启

#systemctl restart postgresql-12

#systemctl status postgresql-12



#su - postgres

-bash-4.2$ psql

修改postgres密码

postgres=# alter user postgres ENCRYPTED PASSWORD 'YOUR\_PASSWORD';

创建数据库：

postgres=# create user test with password 'YOUR\_PASSWORD';

postgres=# create database test\_db owner test;

postgres=# grant all privileges on database test\_db to test;

**1.3 部署NGinx**

## 1.3 环境变量

1. 创建 相关的后端服务
2. 将lingxi复制到/data目录下(将原来/data目录下的lingxi替换掉)，重命名为lingxi-test，

由于jenkins连接发布主机的ssh验证用户一般都是www，因此我们需要给lingxi-test目录修改所属主， # chown -R [www.www](http://www.www) /data/lingxi-test

(3)设置环境变量，编辑文件，中间件、consul等账号密码根据docker-compose编排的来写

# vi /data/ling-test/.config

export LINGXI\_DB\_PORT=5432 #数据库端口

export LINGXI\_DB\_ADDR=10.0.1.182 #数据库地址

export LINGXI\_DB\_USER=jfzx\_pgsql\_test #数据库账号，测试环境专用帐号

export LINGXI\_DB\_PWD= jfzx\_pgsql\_test #密码

export LINGXI\_DB\_NAME=lingxi\_test\_db #业务库名

export LINGXI\_DB\_PAAS\_NAME=lingxi\_paas\_test\_db #Paas库名

export LINGXI\_HIKARI\_MIN=2

export LINGXI\_HIKARI\_MAX=20

export LINGXI\_CONSUL\_ADDR=127.0.0.1 #consul地址（本机部署）

export LINGXI\_CONSUL\_PORT=8500 #consul端口

export LINGXI\_ZIPKIN\_ADDR=127.0.0.1 #zipkin地址（本机部署）

export LINGXI\_ZIPKIN\_PORT=9411 #zipkin端口

export LINGXI\_REDIS\_ADDR=127.0.0.1 #redis地址（本机部署）

export LINGXI\_REDIS\_PORT=6379 #redis 端口

export LINGXI\_REDIS\_PWD=jfzx\_redis #redis 密码

export LINGXI\_REDIS\_DB=0 #redis 默认db

export LINGXI\_MQ\_ADDR=127.0.0.1 #rabbitmq地址（本机部署）

export LINGXI\_MQ\_PORT=5672 #rabbitmq端口

export LINGXI\_MQ\_USER=jfzx\_rabbitmq #rabbitmq账号

export LINGXI\_MQ\_PWD=jfzx\_rabbitmq #rabbitmq密码

export LINGXI\_ES\_URIS=http://127.0.0.1:9200 #es地址（本机部署）

export LINGXI\_ES\_USER=

export LINGXI\_ES\_PWD=

export LINGXI\_LIBREOFFICE\_HOME=/opt/libreoffice6.4

export PAY\_NOTIFY\_URL=http://lingxi-gateway-test.shushangyun.com:12880

#网关服务地址，同上，后面会在nginx中配置此域名

export LINGXI\_AUTH\_URL= http://lingxi-gateway-test.shushangyun.com:12880

export LINGXI\_ESIGN\_APPID=7438817714

export LINGXI\_ESIGN\_SECRET=f0c7a973274fe884ed09b5515324a432

export LINGXI\_ESIGN\_SANDBOX=true

export LINGXI\_ESIGN\_URL= http://lingxi-gateway-test.shushangyun.com:12880

export LINGXI\_ESIGN\_DEBUG=true

export FILE\_SERVER\_TYPE=1

#阿里云oss文件服务地址

export LINGXI\_ALIYUN\_ENDPOINT=http://oss-cn-shenzhen.aliyuncs.com

export LINGXI\_ALIYUN\_ACCESSKEY=LTAI4FogcK9Ftw81JNDTUgXF #oss访问密钥

export LINGXI\_ALIYUN\_SECRET=43TCFNGzSlAvXv4keRZ8CY9ztL1RkI

export LINGXI\_ALIYUN\_BUCKET=shushangyun01 #oss存储桶

export LINGXI\_ALIYUN\_FOLDER=11

export LINGXI\_ALIYUN\_WEBURL=oss-cn-shenzhen.aliyuncs.com

#网关服务地址

export LINGXI\_AUTH\_URL= http://lingxi-gateway-test.shushangyun.com:12880

export LINGXI\_AUTH\_FILE\_PATH=/data/tmpfiles

export LINGXI\_SMTP\_HOST=smtp.exmail.qq.com

export LINGXI\_SMTP\_PORT=465

export LINGXI\_SMTP\_ACCOUNT=lingxi@shushangyun.com

export LINGXI\_SMTP\_USER=lingxi

export LINGXI\_SMTP\_PWD=SsyLX888

export LINGXI\_IM\_APPID=1400493302

export LINGXI\_IM\_KEY=2e2316f7379ed4af198ed5b592a979aade2a1562fe12d1d22aaf0711612ca977

export JPUSH\_KEY=c7c06a8602e436b96b423854

export JPUSH\_SECRET=45877b81a9e8bf31e79c47cd

export JPUSH\_APNS\_PRO=true

# 3.微服务启动脚本

(1) 修改好环境变量之后，需要去修改每个微服务的启动脚本，/data/lingxi-test/下的xxx-start.sh

图形用户界面, 文本

描述已自动生成

一般需要修改两个地方，第一个是服务所在目录，第二个启动参数: 默认语言，-Duser.language，在修改这个之前需要和开发负责人确认，服务的程序语言需要是英文还是中文，中文参数是“zh”, 英文是“en”

文本

描述已自动生成

1. 我们需要修改所有微服务的启动脚本，可使用for循环配合sed进行批量修改，首先修改部署jar包的目录，在/data/lingxi-test下执行：

# for service in `ls -l | awk '{print $9}' | grep service`;do sed -i 's/lingxi/lingxi-test/g' $service/$service-start.sh ;done

修改启动参数“语言”：例如中文改成英文

# for service in `ls -l | awk '{print $9}' | grep service`;do sed -i 's/language=zh/ language=en/g' $service/$service-start.sh ;done

执行完之后再检查下脚本有没有修改到位

# 4. nginx配置

(1)需要配置五个基本域名(瓴犀项目最少都有五个)，分别是中台(paas)、平台后台(admin)、能力中心(mall)、商城(b2b);

(2) 连接内网的nginx服务器：10.0.1.182

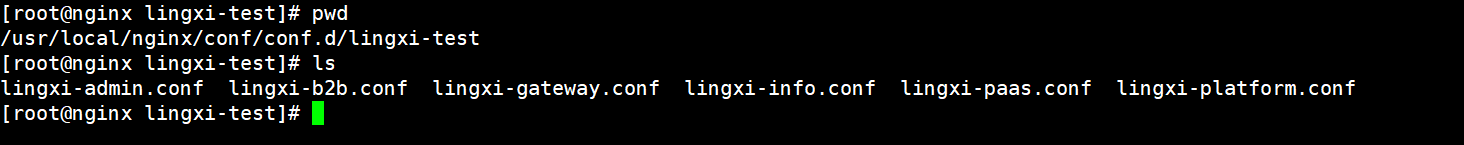
(3) 进入目录/usr/local/nginx/conf/conf.d/ 可以看到一些项目的nginx配置文件，

文本

描述已自动生成

我们需要单独为新项目再创建一个配置文件目录，比如 lingxi-test，可以复制其他项目再重命名一下，然后再修改配置

# cp -r lxy-test-v4 lingxi-test



(4)下面说明前端各域名需要修改的内容

中台(Paas)

# vi lingxi-paas.conf

文本

描述已自动生成

平台后台(admin)

# vi lingxi-admin.conf

文本

描述已自动生成

能力中心(platform)修改的内容也和paas、平台后台一样，配置好内网端口、相应的域名、发布目录、和代理的后端地址即可

网关域名配置

网关域名不需要配置内网端口，内网访问可以用后端服务器ip:端口的方式访问：还有要注意的就是代理的地址有三个地方要改

文本

描述已自动生成

文本

描述已自动生成

商城类域名(b2b、info等)，不需要在nginx配置根目录，商城类项目都是由nodejs启动的，测试或者uat环境项目的nodejs一般部署在后端服务器，因此发布目录在后端服务器配置

文本

描述已自动生成

文本

描述已自动生成

1. 配置完域名之后，在/usr/local/nginx/html下创建相应的发布目录，并修改目录的所属主www用户。

文本

描述已自动生成

1. 执行nginx -t检查nginx没问题之后，再执行nginx -s reload重载nginx配置

文本

描述已自动生成

# 5. jenkins配置与发布

## 5.1 内网jenkins说明

目前部署在内网的项目主要使用这两台jenskins，一主一从

Jenkins主机: 主要用于构建和发布后端的微服务

内网地址：10.0.1.180:8080

外网地址：183.3.136.70:33972

Jenkins从机：主要用于构建和发布前端项目

内网地址：10.0.1.196

## 5.2 配置发布的目标机器

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

在Publish over SSH模块下，点击新增

图形用户界面, 文本

描述已自动生成

图形用户界面, 应用程序, 电子邮件

描述已自动生成

## 5.3 创建后端服务的job

每个job对应瓴犀其中一个微服务,目前瓴犀的微服务在25个左右

表格

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

以after-sale-service为例：

图形用户界面, 文本, 应用程序, 电子邮件, Teams

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

后端服务全部构建完成后，可以有多种方式验证:

1. 访问consul的web页面，查看服务是否都有注册上去，访问方式是浏览器访问后端服务器ip:8500

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. 在服务器使用ps命令查看java的进程，看服务的进程是否都在运行，不过这种方式可能没办法完全验证微服务是否全部都正常运行，有可能日志也有报错，日志目录一般在/data/logs下，该目录需要提前建好并把目录所属主改成www用户，当服务进行构建并执行启动之后会有日志产生

## 5.4 配置前端项目的job

(1)jenkins由()中的内容进行权限分配，Test环境分配给开发人员使用，而UAT环境无特殊情况是不分配权限给开发人员的

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

(2) 前端项目需要指定运行节点slave-1

图形用户界面, 应用程序

描述已自动生成

中台(paas)的配置:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

平台后台的配置：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

商城类项目的配置：

商城类项目都是由nodejs渲染并由pm2进程管理的，测试或者uat环境项目的nodejs一般部署在后端服务器，因此商城类项目部署在后端服务器

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件, Teams

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

商城项目构建：首次构建部署需要全部勾选，后面构建根据开发人员的要求勾选

图形用户界面, 应用程序

描述已自动生成

能力中心：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

# 6.访问测试

项目构建完成且构建过程无报错的情况下，使用 域名:12880的方式访问，公司的内网服务器用的是专线ip, 企业专线80、443这些端口是不开放的，因此需要加上12880端口，比如访问paas, http://lingxi-paas-test.shushangyun.com:12880

中台

图形用户界面, 网站

描述已自动生成

平台后台

图片包含 游戏机, 自然

描述已自动生成

能力中心

图形用户界面, 应用程序, Teams

描述已自动生成

B2B商城

社交网络的手机截图

描述已自动生成

商城类项目可以有好几个，除了b2b, 还有info资讯、self自营商城等等，这里就不一一列举了