

部署文档

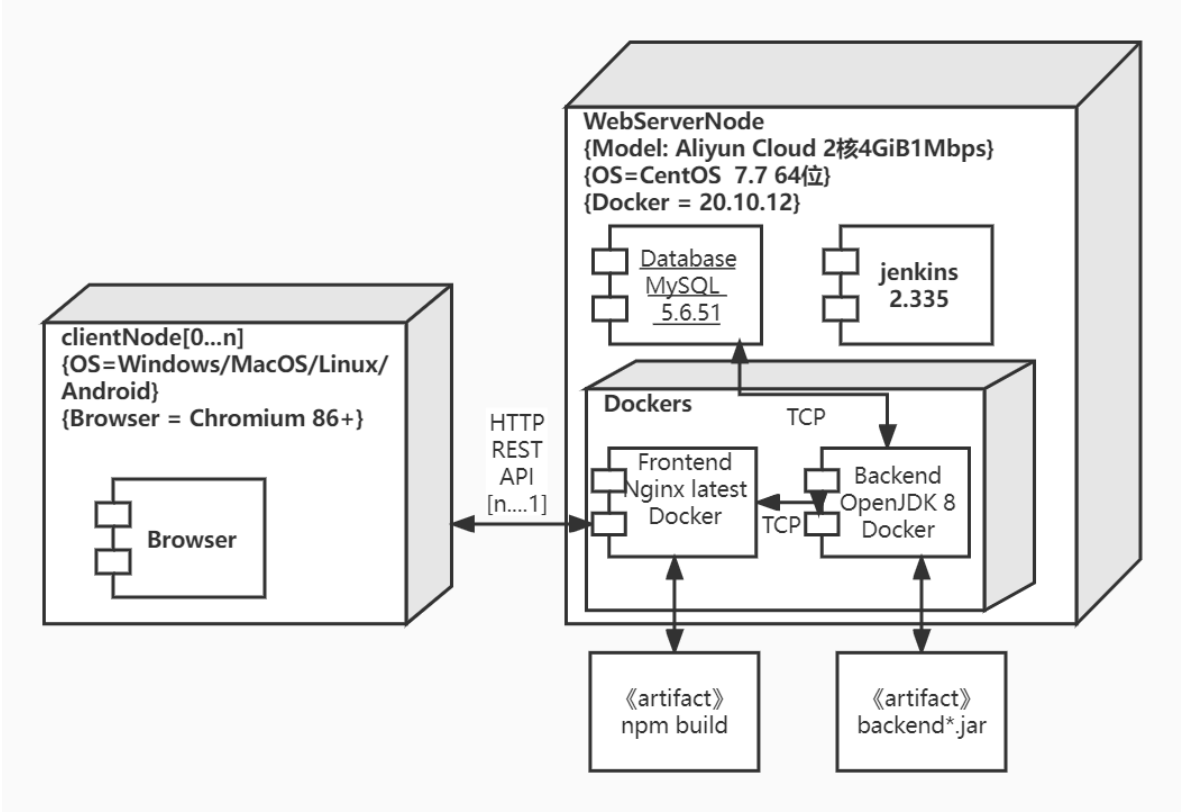
创建者：王文渊

创建时间：2021年3月31日

文档修改记录

时间	操作者	修改内容	版本号
2022年3月31日	王文渊	创建文档，初步写好	v1.0
2022年4月2日	王文渊	修改增加文档	v2.0

部署图



部署信息

Jenkins

地址：106.14.182.244:8082/jenkins/

1.通过tomcat运行，下载jenkins的war包放webapps，服务器配置安全组

```
1 | vim server.xml | 修改到8082端口
2 | ./startup.sh | 启动
```

2.jenkins设置并安装配置java，Maven，git，NodeJS；安装blue ocean，jacoco等插件

3.新建任务配置仓库url，凭证，hook(进入gitlab集成配置页面，复制 jenkins 触发器配置页面的 url 及 Token)，配置全局凭据等。

本项目jenkins任务：

mirror后缀的前后端任务

监测按照原来仓库而加的 并发回构建成果与否信息到git.nju.edu.cn （检测develop/master两个分支）只进行build测试依赖性，不进行部署

The screenshot shows a GitLab commit page for the repository 'Frontend-TestPlatform-Mirror'. The commit is titled 'fix:修复score分数计算' and was authored 18 hours ago by 'L-nianlhy'. The commit hash is '19050f08'. The page shows the commit's parent as 'f3a79a60' on the 'develop' branch. Below the commit details, there is a section for 'Pipeline #34592' which passed with stage 'latest' in 0 seconds. At the bottom, there is a table showing the pipeline's status and stages.

Status	Pipeline	Triggerer	Stages
✓ 18 hours ago	fix:修复score分数计算 #34592 develop parent f3a79a60 latest	👤	✓

frontend-testplatform backend（后端更改更新过的流水线版本）

进行自动化的部署（只在master时触发）



Build #29 (2022-4-2 14:21:18)

Started by GitLab push by 王文渊



Changes

1. 增加测试 ([details](#))



Started by GitLab push by 王文渊



Revision: 650e18dfbca2ac123831d7d830b8322d55650491

Repository: http://172.29.4.49/191250090_lwxz/backend-testplatform.git

• refs/remotes/origin/master



Test Result (没有失败)



Jacoco - Overall Coverage Summary

INSTRUCTION	51%	<div><div></div></div>
BRANCH	26%	<div><div></div></div>
COMPLEXITY	43%	<div><div></div></div>
LINE	70%	<div><div></div></div>
METHOD	60%	<div><div></div></div>
CLASS	84%	<div><div></div></div>

frontend

106.14.182.244:8080

安装docker新建nginx镜像，准备nginx.conf配置文件并修改

```
1 listen 8080;          *# 监听8080端口*
2
3 server_name 106.14.182.244; *# 服务器地址
```

挂载配置文件，就是把装有docker宿主机上面的nginx.conf配置文件映射到启动的nginx容器里面，这需要先准备好

```
1 前端项目更改配置
2 #main.js
3 axios.defaults.baseURL = "http://106.14.182.244:9000"
```

在master分支push新的代码后会自动部署

jenkins配置构建后执行shell，重新生成docker容器把./dist以及之下的页面，配置，logs放到nginx容器内

--privileged=true使用该参数，container内的root拥有真正的root权限。

```

1  cnpm install
2  npm run build
3  cp -r ./dist /root/nginx/www #相同会覆盖 给予权限避免403
4  docker stop front
5  docker rm front
6  docker run -d -p 8080:80 --name front --privileged=true -v
   /root/nginx/www:/usr/share/nginx/html -v
7  /root/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v
   /root/nginx/logs:/var/log/nginx -d nginx

```

backend

106.14.182.244:9000

```

1  后端代码更改配置
2  #application.properties
3  spring.datasource.url=jdbc:mysql://106.14.182.244:3306/collect_db?
   setUnicode=true&characterEncoding=utf8&useSSL=false&serverTimezone=GMT%2B8
4  spring.datasource.username=root
5  spring.datasource.password=password
6  server.port=9000
7
8  #generatorConfig.xml
9  connectionURL="jdbc:mysql://106.14.182.244:3306/collect_db?
   setUnicode=true&characterEncoding=utf8&useSSL=false&serverTimezone=GMT%2B8"

```

在master分支push新的代码后会自动部署

编写流水线项目pipeline，配置脚本路径jenkins/Jenkinsfile

```

1  #Jenkinsfile
2  pipeline {
3      agent any
4      stages {
5          stage('Build') {
6              steps {
7                  sh 'mvn -B -DskipTests clean package'
8              }
9          }
10         stage('Test') {
11             steps {
12                 sh 'mvn package'
13                 sh "chmod +x mvnw"
14                 junit 'target/surefire-reports/*.xml'
15                 step([
16                     $class: 'JacocoPublisher',
17                     execPattern: 'target/*.exec',
18                     classPattern: 'target/classes',
19                     sourcePattern: 'src/main/java',
20                     exclusionPattern: 'src/test*'
21                 ])
22             }
23         }
24         stage('deploy') {
25             steps {

```

```

27         sh 'chmod +x ./jenkins/scripts/start_back.sh'
28         sh './jenkins/scripts/start_back.sh'
29     }
30 }
31 }
32 }

```

在部署阶段，调用以下脚本，重新生成docker，build时根据dockerfile，把java8的镜像作为基础，把项目的jar包放到docker里启动，暴露9000。

```

1 #start_back.sh
2 #!/usr/bin/env bash
3 ls -al
4 docker stop back
5 docker rm back
6 docker build -t backend . --no-cache
7 docker run -d -p 9000:9000 --name back backend

```

```

1 #Dockerfile
2 FROM java:8
3 VOLUME /tmp
4 ADD ./target/*.jar app.jar
5 RUN bash -c 'touch /app.jar'
6 EXPOSE 9000
7 ENTRYPOINT ["java","-jar","/app.jar"]

```

database

106.14.182.244:3306 密码:password

- 1.安装mysql覆盖掉MariaDB，配置/etc/my.cnf字符编码。
- 2.grant all privileges on . to root@'% 'identified by 'password'; 允许远程连接访问，配置安全组
- 3.source sql脚本

流程

