

COLLECT众测平台使用指南

Group: 191250166_outlast

1.项目搭建

- 1、下载jenkins.war
- 2、在8082端口启动jenkins

```
nohup java -jar /home/molgypeter/jenkins.war --ajp13Port=-1 --httpPort=8082 >/dev/null 2>&1 &
```

jenkins目录在/home/molgypeter/.jenkins

- 3、创建前端和后端的流水线。

2、webhook配置

- 1、jenkins中添加gitlab插件，配置gitlab。

Gitlab

☒ Enable authentication for '/project' end-point

GitLab connections

Connection name

frontend

A name for the connection

Gitlab host URL

https://git.nju.edu.cn

The complete URL to the Gitlab server (e.g. http://gitlab.mydomain.com)

Credentials

GitLab API token (frontend) ▼

添加 ▼

API Token for accessing Gitlab

高级...

Test Connection

删除

Connection name

backend

A name for the connection

- 2、在前端和后端流水线中配置gitlab的构建触发器，生成Secret token，在git.nju.edu.cn上配置webhook触发构建。

构建触发器

- ☐ 其他工程构建后触发
- ☐ 定时构建
- ☒ Build when a change is pushed to GitLab. GitLab webhook URL: <http://121.5.6.104:8082/project/backend-outlast>

Enabled GitLab triggers

- ☒ Push Events
- ☐ Push Events in case of branch delete
- ☒ Opened Merge Request Events
- ☐ Build only if new commits were pushed to Merge Request
- ☐ Accepted Merge Request Events
- ☐ Closed Merge Request Events

Rebuild open Merge Requests

Never

URL

<http://121.5.6.104:8082/project/backend-outlast>

URL must be percent-encoded if it contains one or more special characters.

Secret token

041894467919be64d834cfd8fe9ff195

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.

Trigger

- ☒ Push events

Branch name or wildcard pattern to trigger on (leave blank for all)

Push to the repository.

3. 脚本实现构建及部署

前端分为：'Git'获取远程git仓库的代码，'Build'进行npm构建，'Deploy'将构建好的文件复制到特定目录下（使用了nginx进行页面的部署），最后将构建结果返回git仓库。

```
pipeline {
  agent any

  stages {
    stage('Git') {
      steps {
        git credentialsId: 'ed08ca8b-f78c-40d4-8170-0eaf29cd6b7b', url:
'https://git.nju.edu.cn/191250194/frontend-outlast.git'
      }
    }
    stage('Build') {
      steps {
        //sh "chmod -R a+x ./node_modules/.bin/*"
        //sh "rm -rf ./node_modules"
        sh "npm install"
        sh "npm run build"
      }
    }
    stage('Deploy') {
      steps {
        sh "rm -rf /home/molgypeter/frontend/*"
        sh "cp -r dist/* /home/molgypeter/frontend"
```

```

    }
  }
}

post {
  success{
    updateGitlabCommitStatus name: 'build', state: 'success'
  }
  failure {
    updateGitlabCommitStatus name: 'build', state: 'failed'
  }
}
}
}

```

后端分为以下的步骤：'Git'获取远程git仓库的代码，'SQL'执行SQL语句，'CleanFiles'清楚上传的静态文件，'Build'进行构建并生成覆盖率报告（使用nginx部署页面），'Deploy'执行shell脚本进行部署，最后将构建结果返回git仓库。

```

pipeline {
  agent any

  stages {
    stage('Git') {
      steps {
        git credentialsId: 'ed08ca8b-f78c-40d4-8170-0eaf29cd6b7b', url:
'https://git.nju.edu.cn/191250194/backend-outlast.git'
      }
    }
    stage('SQL') {
      steps {
        sh "mysql -uroot -p123456 -h localhost <
/home/molgypeter/.jenkins/workspace/backend-outlast/sql/collect.sql "
      }
    }
    stage('CleanFiles') {
      steps {
        sh "rm -rf /home/molgypeter/static/"
      }
    }
    stage('Build') {
      steps {
        sh "mvn clean package"
        sh "rm -rf /home/molgypeter/jacoco/* && cp -r
target/site/jacoco/* /home/molgypeter/jacoco"
      }
    }
    stage('Deploy') {
      steps {
        withEnv(['JENKINS_NODE_COOKIE=dontkillme']) {
          sh "sh /home/molgypeter/backend/run_backend.sh"
        }
      }
    }
  }

  post {
    success{

```

```

        updateGitlabCommitStatus name: 'build', state: 'success'
    }
    failure {
        updateGitlabCommitStatus name: 'build', state: 'failed'
    }
}
}
}

```

其中的run_backend.sh需要结束占用8081端口的进程，执行新的jar部署到8081：

```

#!/bin/bash

SOURCE_DIR=/home/molgypeter/.jenkins/workspace/backend-outlast/target
TARGET_DIR=/home/molgypeter/backend

JAR_NAME=`ls $SOURCE_DIR | grep '.*\.jar$'`
if [ -n "$JAR_NAME" ];then
    echo '删除旧版本'
    rm $TARGET_DIR/*.jar && cp $SOURCE_DIR/$JAR_NAME $TARGET_DIR
fi
echo '检查8081端口是否被占用...'
pid_blog=`netstat -tunpl | grep 8081|awk '{printf $7}'|cut -d/ -f1`
if [ "$pid_blog" != "" ];
then
    echo '8081端口被占用'
    kill -9 "$pid_blog"
    echo $pid_blog '进程已被杀死'
else
    echo "端口未被占用"
fi
echo '后台运行jar包...'
fileName=`date +%Y-%m-%dT%H%M%S.txt`
nohup java -jar $TARGET_DIR/$JAR_NAME --server.port=8081
>$TARGET_DIR/logs/$fileName 2>&1 &
echo 'jar包部署结束'

```

目前流水线脚本和shell脚本中的文件目录需要视具体情况而自定义，准备在迭代三中使用环境变量来方便路径的配置。

4. 项目使用前提

在application.yml中配置mysql数据库密码和服务器的文件存储目录。

```

datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/collect?setUnicode=true&characterEncoding=
    username: root
    password: 123456

web:
    file-upload-path: /home/molgypeter/static/

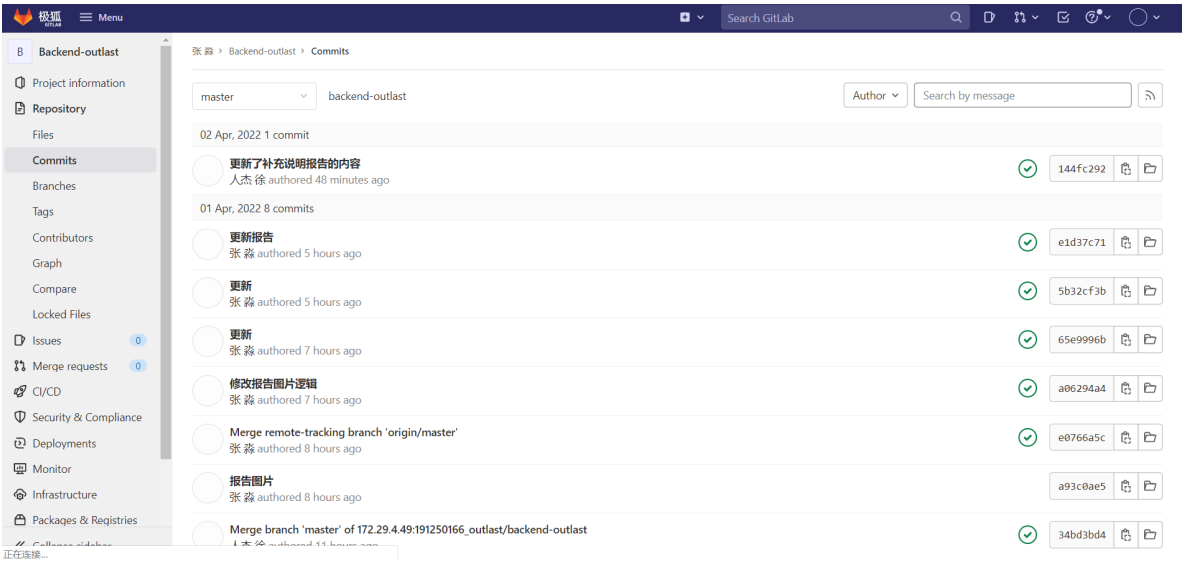
```

5. 自动化部署演示视频

6. CI/CD阶段记录

后端

commit记录：



jenkins构建记录（因为新增了'CleanFiles'步骤而没有显示更早的阶段视图）：

Dashboard

backend-outlast

完整阶段视图

重命名

流水线语法

Build History

构建历史

Filter builds...

#114

2022-4-2 上午12:04

Started by GitLab push by 张 森

#113

2022-4-1 下午7:47

Started by GitLab push by 张 森

#112

2022-4-1 下午7:22

Started by GitLab push by 张 森

#111

2022-4-1 下午5:49

Started by GitLab push by 张 森

#110

2022-4-1 下午5:06

Started by GitLab push by 张 森

#109

2022-4-1 下午4:15

Started by GitLab push by 张 森

#108

2022-4-1 下午1:45

Started by GitLab push by 张 森

#107

2022-3-31 下午10:05

Created by GitLab push by 张 森

Average stage times:

(Average full run time: ~27s)

#114

Apr 02 00:04

1 commit

#113

Apr 01 19:47

1 commit

Git	SQL	CleanFiles	Build	Deploy	Declarative: Post Actions
683ms	895ms	357ms	23s	521ms	653ms
675ms	890ms	342ms	22s	457ms	545ms
692ms	900ms	372ms	23s	585ms	762ms

相关链接

- 最近一次构建(#114),50 分之前
- 最近稳定构建(#114),50 分之前
- 最近成功的构建(#114),50 分之前
- 最近失败的构建(#103),3 天 2 小时之前
- 最近未成功的构建(#103),3 天 2 小时之前
- 最近完成的构建(#114),50 分之前

Dashboard

backend-outlast

Build History

构建历史

Filter builds...

#114

2022-4-2 上午12:04

Started by GitLab push by 张 森

#113

2022-4-1 下午7:47

Started by GitLab push by 张 森

#112

2022-4-1 下午7:22

Started by GitLab push by 张 森

#111

2022-4-1 下午5:49

Started by GitLab push by 张 森

#110

2022-4-1 下午5:06

Started by GitLab push by 张 森

#109

2022-4-1 下午4:15

Started by GitLab push by 张 森

#108

2022-4-1 下午1:45

Started by GitLab push by 张 森

#107

2022-3-31 下午10:05

Started by GitLab push by 张 森

#106

2022-3-31 下午7:41

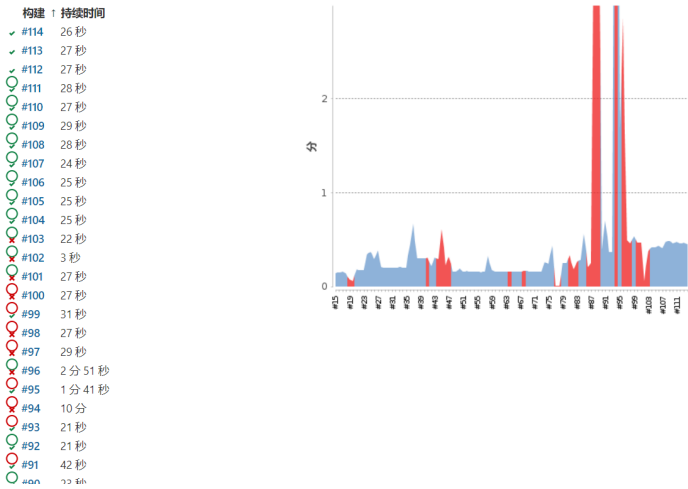
Started by GitLab push by 张 森

#105

2022-3-31 上午11:51

Started by GitLab push by 张 森

构建时间趋势



前端

commit记录：

极狐

Menu

Search GitLab

Frontend-outlast

Project information

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Locked Files

Issues

Merge requests

CI/CD

Security & Compliance

Deployments

Monitor

Infrastructure

Packages & Registries

张森 > Frontend-outlast > Commits

master frontend-outlast

Author Search by message

31 Mar, 2022 2 commits

更新文件上传

张森 authored 1 day ago

47137e62

更新文件上传

张森 authored 1 day ago

b678ffc2

29 Mar, 2022 5 commits

push

191250156 authored 3 days ago

b884b2fa

接口修改

191250156 authored 3 days ago

ef7f0647

注册接口修改

191250156 authored 3 days ago

4cdb8206

修改注册界面

191250156 authored 3 days ago

4b4edc97

修改个人信息界面

191250156 authored 3 days ago

4c5990ff

22 Mar, 2022 2 commits

jenkins构建记录：

Dashboard > frontend-outlast >

完整阶段视图

重命名

流水线语法

Build History 构建历史

Filter builds...

#48 2022-3-31 下午10:04

Started by GitLab push by 张森

#47 2022-3-29 下午4:45

Started by GitLab push by 张森

#46 2022-3-29 下午4:18

Started by GitLab push by 张森

#45 2022-3-29 下午4:12

Started by GitLab push by 张森

#44 2022-3-29 下午3:47

Started by GitLab push by 张森

#43 2022-3-29 下午3:31

Started by GitLab push by 张森

#42 2022-3-22 下午3:02

Started by GitLab push by 张森

#41 2022-3-22 下午1:49

Started by GitLab push by 张森

Average stage times:

(Average full run time: ~42s)

	Git	Build	Deploy	Declarative: Post Actions
#48	809ms	39s	644ms	788ms
#47	939ms	37s	630ms	512ms
#46	656ms	32s	651ms	1s
#45	719ms	31s	606ms	641ms
#44	623ms	32s	620ms	626ms
#43	772ms	32s	657ms	1s
#42	1s	46s	632ms	924ms

Dashboard > frontend-outlast >

Build History 构建历史

Filter builds...

#48 2022-3-31 下午10:04

Started by GitLab push by 张森

#47 2022-3-29 下午4:45

Started by GitLab push by 张森

#46 2022-3-29 下午4:18

Started by GitLab push by 张森

#45 2022-3-29 下午4:12

Started by GitLab push by 张森

#44 2022-3-29 下午3:47

Started by GitLab push by 张森

#43 2022-3-29 下午3:31

Started by GitLab push by 张森

#42 2022-3-22 下午3:02

Started by GitLab push by 张森

#41 2022-3-22 下午1:49

Started by GitLab push by 张森

#40 2022-3-19 下午2:40

Started by GitLab push by 张森

#39 2022-3-17 下午3:46

Started by GitLab push by 张森

构建时间趋势

构建 持续时间

#48 40 秒

#47 36 秒

#46 34 秒

#45 35 秒

#44 36 秒

#43 49 秒

#42 36 秒

#41 40 秒

#40 39 秒

#39 1 分 18 秒

#38 2 分 23 秒

#37 3.1 秒

#36 1 分 29 秒

#35 57 秒

#34 1 分 24 秒

#33 1 分 16 秒

#32 1 分 47 秒

#31 1 分 2 秒

#30 42 秒

#29 17 秒

#28 46 秒

#27 30 秒

#26 52 秒

#25 55 秒

构建时间趋势图

构建 持续时间

构建时间趋势图

构建时间趋势图