一、代码管理

P1、Maven部署

1、安装maven

```
[root@jumpserver ~]# ssh -p2222 operation@192.168.1.252
operation@192.168.1.252's password:
Opt> p          #输入p
  ID  | 主机名                                     | IP              | 平台       | 组织       | 备注
------+-------------------------------------------+----------------+----------+----------+--------
-
  1   | harbor                                     | 192.168.1.30 | Linux     | Default  |
  2   | nacos                                      | 192.168.1.13  | Linux     | Default  |
  3   | rocketmq                                   | 192.168.1.14  | Linux     | Default  |
  4   | test                                       | 192.168.1.101 | Linux     | Default  |
页码：1，每页行数：21，总页数：1，总数量：4
提示：输入资产ID直接登录，二级搜索使用 // + 字段，如：//192 上一页：b 下一页：n
搜索：
[Host]> 4    #进入test主机
[op@test ~]$ sudo -s
```

```
拷贝软件到test主机
[root@jumpserver ~]# scp /root/package/node-v16.13.0-linux-x64.tar.gz /root/package/apache-maven-3.6.3-bin.tar.gz  /root/package/jenkins* /root/package/repository.tar.gz 192.168.1.101:/root/

#安装Maven
[root@test op]# cd /root/
[root@test ~]# tar -xf apache-maven-3.6.3-bin.tar.gz
[root@test ~]# mv apache-maven-3.6.3 /usr/local/maven
[root@test ~]# yum -y install java-11-openjdk-devel    #安装JDK11环境

#maven环境变量
[root@test ~]# vim /etc/bashrc      #文件末尾添加
...
export MAVEN_HOME="/usr/local/maven"
export PATH=${MAVEN_HOME}/bin/:$PATH
[root@test ~]# source /etc/bashrc
[root@test ~]# mvn -v                #测试Maven指令
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /usr/local/maven
Java version: 11.0.20, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-11-openjdk-11.0.19.0.7-4.el8.x86_64
Default locale: en_US, platform encoding: ANSI_X3.4-1968
OS name: "linux", version: "4.18.0-372.26.1.el8_6.x86_64", arch: "amd64", family: "unix"


#配置Maven本地仓库
[root@test ~]# vim /usr/local/maven/conf/settings.xml
 55   <localRepository>/usr/local/maven/repository/</localRepository>

直接解压/root/repository.tar.gz，把里面的内容放到/usr/local/maven/即可
[root@test ~]# tar -xf /root/repository.tar.gz -C /usr/local/maven/
```

2、代码准备

给test主机绑定公网IP，拉取代码（由于之前已经有购买公网ip经验，绑定多次，此次不再重复截图）

```
拉取代码
[root@test op]# cd /root/
[root@test ~]# yum -y install git
[root@test ~]# git clone https://gitee.com/cc-0001/tea
[root@test ~]# ls tea/code/page/
admin-page              #网站商品管理页面
front-page              #网站商品展示页面

[root@test ~]# ls tea/code/backend/
tarena-passport         #验证码中心组件
tarena-tp-attach        #附件中心组件
tarena-tp-tea           #商品服务组件
```

1、商品服务组件

```
#编写配置文件
[root@test ~]# cd /root/tea/code/backend/tarena-tp-tea/
[root@test tarena-tp-tea]# vim tea-server/tea-server-main/src/main/resources/application-vm.yaml

spring:
  # 数据源配置
  datasource:
    url: jdbc:mysql://192.168.1.12:3306/tarena_tp_tea?useSSL=false&useUnicode=true&characterEncoding=utf-8&serverTimezone=Asia/Shanghai&allowMultiQueries=true&allowPublicKeyRetrieval=true
    username: teauser
    password: Taren123
    driver-class-name: com.mysql.cj.jdbc.Driver
    hikari:
      minimum-idle: 1
      auto-commit: true
      idle-timeout: 600000
      minimum: 5
      max-lifetime: 1800000
      #数据库连接超时时间,默认30秒，即30000
      connection-timeout: 120000
      connection-test-query: SELECT 1 FROM DUAL

  redis:
    host: 192.168.1.147    #此地址写自己redis的负载均衡地址
    port: 6379
    password: "Taren123"
    database: 8
    jedis:
      pool:
        max-active: 8
        min-idle: 5
        max-idle: 8
        max-wait: PT15M
        time-between-eviction-runs: PT15M
rocketmq:
  name-server: 192.168.1.14:9876
  producer:
    group: ${spring.application.name}


mq:
  auto-close-delay-level: 15
  group:
    auto-close: order-auto-close-group
    success-paid: success-paid-group
  topic:
    pay-success-paid-topic: pay-success-paid
    order-cancel-topic: order-cancel-topic
    auto-close-topic: order-auto-close-topic
    order-success-paid-topic: order-success-paid-topic

pay:
  publicFilePath: '/usr/local/project/tea/publicKey.txt'
  redirectPath: 'https://dev-cs-pay-center.tctm.life/payCenter/queryOrderInfo'


#url相关
image_path: /home/images/vm/tea_attach/
url: http://122.9.48.80:30080/tea_attach/   #写自己华为云负载均衡ELB的IP


jwt:
  rsa:
    enabled: true
    private-key: |
      -----BEGIN PRIVATE KEY-----
      MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBANAn5giVT7oy/drc
      o0evOFJUYSnHLbYtXG4MUuEUM2QdGX9d1q3y5k9orwX3wrt4WP4gIkQdB0EzQApp
      WZMUbbVsibVWkVSTQ5aInPgjLjfaDlcP5U4Z1hh3QiPZ01i3AWYY8HS+2SBIlRLH
      S5X0MjZkdbOVEPODU73JNSjFKz4XAgMBAAECgYBwXblt1LxPNYuYBBcYcVwkBWzW
      ErF4cJA+z+RFoJQFbTgAa3WiPUdagpZI0HCMpvDTbiFRR6JM7g+aDzjNju+RUCW9
      0iaDrB5vKyh3INMzcdCfm4btGectxb/ZCYFkXPBvqtqzK/H9ANk/Xlq1KFw0O2x2
      fgV7swYuiZ3kSYLbEQJBAPx2gXPkpraqEPmt//ce7AP5fHxC8Td2JFROsRfyweJP
      0dUdiLDgMxwpFZIhidU1sJBWX62Vqv6Jh/MvoMBhttsCQQDTEnqy9P+VAS1huYkl
      XwWjbZKxHEIaxW+Bg/ombXLhGudpy+chHFy4QEgA1Puda7kxgcTMhRj1i1EliytB
      ecR1AkEA9STJPNS4wURQKXG4y6f9+zoFk8+Y1IbmberfcWaqt2tUUEFWWpHJbz3B
      kVz5rstsgCsuyqo7GOBI323PBR+c4wJBAJxA3dyFSn+AM5xYwZKM2Zu2jhXXGYjA
      CGAU16aC97x1MkM704rLeEheLe+PvAU5rgtcSdgt3+BGlnf4orkB79ECQE2uNOTP
      enyfRVLgN0j3ZLSyOSqw+kqGsoU2jpcxFCGy0RGk3z1fHVccHfspqCA7AMj9jK5O
      3EULSLOmkXQHPLQ=
      -----END PRIVATE KEY-----
    expiration: 1000000
    public-key: |
      -----BEGIN PUBLIC KEY-----
      MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDQJ+YIlU+6Mv3a3KNHrzhSVGEp
      xy22LVxuDFLhFDNkHRl/Xdat8uZPaK8F98K7eFj+ICJEHQdBM0AKaVmTFG21bIm1
      VpFUk0OWiJz4Iy432g5XD+VOGdYYd0Ij2dNYtwFmGPB0vtkgSJUSx0uV9DI2ZHWz
      lRDzg1O9yTUoxSs+FwIDAQAB
      -----END PUBLIC KEY-----

gateway:
  whitelists:
      - /doc.html
      - /register/**
      - /**/*.html"
      - /*.html"
      - /favicon.ico
      - /**/*.html
      - /**/*.css
      - /**/*.js
      - /swagger-resources/**
      - /v2/api-docs/**
      - /sso/valid/code
      - /*.js
      - /**.html
      - /user/login
      - /menu/**/**
      - /content/**/**
      - /images/**

#打包测试：-D'maven.test.skip'=true：构建项目时跳过运行测试。通过设置为true，可以忽略测试环节，直接进行打包
[root@tea tarena-tp-tea]# mvn clean package -Dmaven.test.skip=true
...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for tp-tea 1.0.0-SNAPSHOT:
[INFO]
[INFO] tp-tea ............................................. SUCCESS [ 27.827 s]
[INFO] tea-common ......................................... SUCCESS [ 13.587 s]
[INFO] tea-po ............................................. SUCCESS [  0.943 s]
[INFO] tea-server ......................................... SUCCESS [  0.045 s]
[INFO] tea-server-protocol ................................ SUCCESS [ 23.567 s]
[INFO] tea-server-dao-api ................................. SUCCESS [ 24.172 s]
[INFO] tea-server-domain .................................. SUCCESS [ 17.005 s]
[INFO] tea-server-dao-impl ................................ SUCCESS [  0.038 s]
[INFO] tea-server-infrastructure .......................... SUCCESS [  0.448 s]
[INFO] tea-web-adapter .................................... SUCCESS [  5.663 s]
```

```
[INFO] tea-server-admin ................................... SUCCESS [ 43.449 s]
[INFO] tea-admin .......................................... SUCCESS [  0.015 s]
[INFO] tea-admin-protocol ................................. SUCCESS [  0.565 s]
[INFO] tea-admin-dao-api .................................. SUCCESS [  0.236 s]
[INFO] tea-admin-domain ................................... SUCCESS [  0.944 s]
[INFO] tea-admin-dao-impl ................................. SUCCESS [  0.042 s]
[INFO] tea-admin-infrastructure ........................... SUCCESS [  0.513 s]
[INFO] tea-admin-adapter .................................. SUCCESS [  0.626 s]
[INFO] tea-admin-main ..................................... SUCCESS [  3.421 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:44 min
[INFO] Finished at: 2023-04-16T02:04:22+08:00
[INFO] ------------------------------------------------------------------------

#查看结果并启动jar测试
#-Dfile.encoding=utf-8 是设置Java虚拟机的文件编码为UTF-8，
#-Xmx128M -Xms128M -Xmn64m -XX:MaxMetaspaceSize=128M -XX:MetaspaceSize=128M：配置Java虚拟机的内存分配和垃圾回收相关设置的
[root@test tarena-tp-tea]# ls tea-server/tea-server-main/target/tea-server-admin-1.0.0-SNAPSHOT.jar
[root@test tarena-tp-tea]# java -Dfile.encoding=utf-8 -jar tea-server/tea-server-main/target/tea-server-admin-1.0.0-SNAPSHOT.jar -Xmx128M -Xms128M -Xmn64m -XX:MaxMetaspaceSize=128M -XX:MetaspaceSize=128M --server.port=30091 --spring.profiles.active=vm


#创建存放图片的目录，解压图片tar包到此目录，同时也是为后期实验做准备
[root@test tarena-tp-tea]# mkdir -p /home/images/vm/tea_attach/
[root@test tarena-tp-tea]# tar -xf /root/tea/teaimg.tar.gz -C /home/images/vm/tea_attach/
```

2、商品管理组件

```
#编写配置文件
[root@test tarena-tp-tea]# vim tea-admin/tea-admin-main/src/main/resources/application-vm.yaml
spring:
  # 数据源配置
  datasource:
    url: jdbc:mysql://192.168.1.12:3306/tarena_tp_tea?useSSL=false&useUnicode=true&characterEncoding=utf-8&serverTimezone=Asia/Shanghai&allowMultiQueries=true&allowPublicKeyRetrieval=true
    username: teauser
    password: Taren123
    driver-class-name: com.mysql.cj.jdbc.Driver
    hikari:
      minimum-idle: 1
      auto-commit: true
      idle-timeout: 600000
      minimum: 5
      max-lifetime: 1800000
      #数据库连接超时时间,默认30秒，即30000
      connection-timeout: 120000
      connection-test-query: SELECT 1 FROM DUAL

  redis:
    host: 192.168.1.147    #写自己redis负载均衡的地址
    port: 6379
    password: "Taren123"
    database: 8
    jedis:
```

```yaml
      pool:
        max-active: 8
        min-idle: 5
        max-idle: 8
        max-wait: PT15M
        time-between-eviction-runs: PT15M

#图片路径配置
image_path: /home/images/vm/tea_attach/
url: http://122.9.48.80:30080/tea_attach/   #写自己华为云负载均衡ELB的IP


rocketmq:
  name-server: 192.168.1.14:9876
  producer:
    group: ${spring.application.name}

mq:
  topic:
    order-cancel-topic: order-cancel-topic

business-mq:
  topic:
    stock-rollback: 'order-cancel-topic'
  group:
    rollback-listener-group: 'stock-rollback-listener-group'


jwt:
  rsa:
    enabled: true
    private-key: |
      -----BEGIN PRIVATE KEY-----
      MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBANAn5giVT7oy/drc
      o0evOFJUYSnHLbYtXG4MUuEUM2QdGX9d1q3y5k9orwX3wrt4WP4gIkQdB0EzQApp
      WZMUbbVsibVWkVSTQ5aInPgjLjfaDlcP5U4Z1hh3QiPZ01i3AWYY8HS+2SBIlRLH
      S5X0MjZkdbOVEPODU73JNSjFKz4XAgMBAAECgYBwXblt1LxPNYuYBBcYcVwkBWzW
      ErF4cJA+z+RFoJQFbTgAa3WiPUdagpZI0HCMpvDTbiFRR6JM7g+aDzjNju+RUCW9
      0iaDrB5vKyh3INMzcdCfm4btGectxb/ZCYFkXPBvqtqzK/H9ANk/Xlq1KFw0O2x2
      fgV7swYuiZ3kSYLbEQJBAPx2gXPkpraqEPmt//ce7AP5fHxC8Td2JFROsRfyweJP
      0dUdiLDgMxwpFZIhidU1sJBWX62Vqv6Jh/MvoMBhttsCQQDTEnqy9P+VAS1huYkl
      XwWjbZKxHEIaxW+Bg/ombXLhGudpy+chHFy4QEgA1Puda7kxgcTMhRj1i1EliytB
      ecR1AkEA9STJPNS4wURQKXG4y6f9+zoFk8+Y1IbmberfcWaqt2tUUEFWWpHJbz3B
      kVz5rstsgCsuyqo7GOBI323PBR+c4wJBAJxA3dyFSn+AM5xYwZKM2Zu2jhXXGYjA
      CGAU16aC97x1MkM704rLeEheLe+PvAU5rgtcSdgt3+BGlnf4orkB79ECQE2uNOTP
      enyfRVLgN0j3ZLSyOSqw+kqGsoU2jpcxFCGy0RGk3z1fHVccHfspqCA7AMj9jK5O
      3EULSLOmkXQHPLQ=
      -----END PRIVATE KEY-----
    expiration: 1000000
    public-key: |
      -----BEGIN PUBLIC KEY-----
      MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDQJ+YIlU+6Mv3a3KNHrzhSVGEp
      xy22LVxuDFLhFDNkHRl/Xdat8uZPaK8F98K7eFj+ICJEHQdBM0AKaVmTFG21bIm1
      VpFUk0OWiJz4Iy432g5XD+VOGdYYd0Ij2dNYtwFmGPB0vtkgSJUSx0uV9DI2ZHWz
      lRDzg1O9yTUoxSs+FwIDAQAB
      -----END PUBLIC KEY-----

gateway:
  whitelists:
    - /doc.html
    - /register/**
    - /**/*.html"
    - /*.html"
    - /favicon.ico
    - /**/*.html
    - /**/*.css
    - /**/*.js
    - /swagger-resources/**
    - /v2/api-docs/**
    - /sso/valid/code
    - /*.js
    - /**.html
    - /user/login
    - /menu/**/**
    - /content/**/**
    - /*.jpg
    - /*.jpeg
    - /*.png
    - /images/**

#打包测试
[root@test tarena-tp-tea]# mvn clean        #清理测试文件
[root@test tarena-tp-tea]# mvn clean package -D'maven.test.skip'=true
...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for tp-tea 1.0.0-SNAPSHOT:
[INFO]
[INFO] tp-tea ............................................. SUCCESS [  0.081 s]
[INFO] tea-common ......................................... SUCCESS [  0.004 s]
[INFO] tea-po ............................................. SUCCESS [  0.003 s]
[INFO] tea-server ......................................... SUCCESS [  0.003 s]
[INFO] tea-server-protocol ................................ SUCCESS [  0.003 s]
[INFO] tea-server-dao-api ................................. SUCCESS [  0.004 s]
[INFO] tea-server-domain .................................. SUCCESS [  0.005 s]
[INFO] tea-server-dao-impl ................................ SUCCESS [  0.004 s]
[INFO] tea-server-infrastructure .......................... SUCCESS [  0.003 s]
[INFO] tea-web-adapter .................................... SUCCESS [  0.007 s]
[INFO] tea-server-admin ................................... SUCCESS [  0.015 s]
[INFO] tea-admin .......................................... SUCCESS [  0.004 s]
[INFO] tea-admin-protocol ................................. SUCCESS [  0.002 s]
[INFO] tea-admin-dao-api .................................. SUCCESS [  0.006 s]
[INFO] tea-admin-domain ................................... SUCCESS [  0.006 s]
[INFO] tea-admin-dao-impl ................................. SUCCESS [  0.004 s]
[INFO] tea-admin-infrastructure ........................... SUCCESS [  0.003 s]
[INFO] tea-admin-adapter .................................. SUCCESS [  0.009 s]
[INFO] tea-admin-main ..................................... SUCCESS [  0.007 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.065 s
[INFO] Finished at: 2023-04-16T02:11:35+08:00
[INFO] ------------------------------------------------------------------------

查看结果并启动jar测试
[root@test tarena-tp-tea]# ls tea-admin/tea-admin-main/target/tea-admin-main-1.0.0-SNAPSHOT.jar
[root@test tarena-tp-tea]# java -Dfile.encoding=utf-8 -jar tea-admin/tea-admin-main/target/tea-admin-main-1.0.0-SNAPSHOT.jar -Xmx128M -Xms128M -Xmn64m -XX:MaxMetaspaceSize=128M -XX:MetaspaceSize=128M --server.port=30092 --spring.profiles.active=vm

#清理测试文件
[root@test tarena-tp-tea]# mvn clean
```

3、附件中心服务组件

```
#编写配置文件
[root@test tarena-tp-tea]# cd /root/tea/code/backend/tarena-tp-attach/
[root@test tarena-tp-attach]# vim attach-server/attach-server-
main/src/main/resources/application-vm.yaml
spring:
  #文件大小配置
  http:
    multipart:
      enabled: true
      max-file-size: 30MB
      max-request-size: 300MB
  # 数据源配置
  datasource:
    url: jdbc:mysql://192.168.1.12:3306/tarena_tp_tea?
useSSL=false&useUnicode=true&characterEncoding=utf-
8&serverTimezone=Asia/Shanghai&allowMultiQueries=true&allowPublicKeyRetrieval=true
    username: teauser
    password: Taren123
    driver-class-name: com.mysql.cj.jdbc.Driver
    hikari:
      minimum-idle: 1
      auto-commit: true
      idle-timeout: 600000
      minimum: 5
      max-lifetime: 1800000
      #数据库连接超时时间,默认30秒，即30000
      connection-timeout: 120000
      connection-test-query: SELECT 1 FROM DUAL

#url相关
image_path: /home/images/vm/tea_attach/
url: http://122.9.48.80:30080/tea_attach/     #写自己华为云负载均衡ELB的IP

dubbo:
  address: nacos://192.168.1.13:8848
  namespace: linux                         #nacos命名空间（project）id

#打包测试
[root@test tarena-tp-attach]# mvn clean package -D'maven.test.skip'=true
...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for tp-attach 1.0.0-SNAPSHOT:
[INFO]
[INFO] tp-attach ......................................... SUCCESS [  0.179 s]
[INFO] attach-common ..................................... SUCCESS [  2.156 s]
[INFO] attach-po ......................................... SUCCESS [  0.437 s]
[INFO] attach-server ..................................... SUCCESS [  0.006 s]
[INFO] attach-server-protocol ............................ SUCCESS [  0.712 s]
[INFO] attach-server-dao-api ............................. SUCCESS [  0.763 s]
[INFO] attach-server-dao-impl ............................ SUCCESS [  0.058 s]
[INFO] attach-server-domain .............................. SUCCESS [  0.865 s]
[INFO] attach-server-infrastructure ...................... SUCCESS [  0.402 s]
[INFO] server-client-api ................................. SUCCESS [  0.295 s]
[INFO] server-web-adapter ................................ SUCCESS [  0.420 s]
[INFO] attach-server-main ................................ SUCCESS [ 21.315 s]
[INFO] attach-admin ...................................... SUCCESS [  0.005 s]
[INFO] attach-admin-protocol ............................. SUCCESS [  0.378 s]
[INFO] attach-admin-dao-api .............................. SUCCESS [  0.245 s]
[INFO] attach-admin-dao-impl ............................. SUCCESS [  0.029 s]
```

```
[INFO] attach-admin-domain ................................ SUCCESS [  0.605 s]
[INFO] attach-admin-infrastructure ........................ SUCCESS [  0.250 s]
[INFO] admin-web-adapter .................................. SUCCESS [  0.647 s]
[INFO] attach-admin-main .................................. SUCCESS [  0.526 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  31.074 s
[INFO] Finished at: 2023-04-16T02:27:41+08:00
[INFO] ------------------------------------------------------------------------

查看结果并启动jar测试
[root@test tarena-tp-attach]# ls attach-server/attach-server-main/target/attach-server-main-1.0.0-SNAPSHOT.jar
[root@test tarena-tp-attach]# java -Dfile.encoding=utf-8 -jar attach-server/attach-server-main/target/attach-server-main-1.0.0-SNAPSHOT.jar -Xmx128M -Xms128M -Xmn64m -XX:MaxMetaspaceSize=128M -XX:MetaspaceSize=128M --server.port=30093 --spring.profiles.active=vm

#清理测试文件
[root@test tarena-tp-attach]# mvn clean
```

4、验证中心组件

```
#编写配置文件
[root@test tarena-tp-attach]# cd /root/tea/code/backend/tarena-passport/
[root@test tarena-passport]# vim passport-provider/src/main/resources/application-vm.yml
jwt:
  rsa:
    enabled: true
    private-key: |
      -----BEGIN PRIVATE KEY-----
      MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBANAn5giVT7oy/drc
      o0evOFJUYSnHLbYtXG4MUuEUM2QdGX9d1q3y5k9orwX3wrt4WP4gIkQdB0EzQApp
      WZMUbbVsibVWkVSTQ5aInPgjLjfaDlcP5U4Z1hh3QiPZ01i3AWYY8HS+2SBIlRLH
      S5X0MjZkdbOVEPODU73JNSjFKz4XAgMBAAECgYBwXblt1LxPNYuYBBcYcVwkBWzW
      ErF4cJA+z+RFoJQFbTgAa3WiPUdagpZI0HCMpvDTbiFRR6JM7g+aDzjNju+RUCW9
      0iaDrB5vKyh3INMzcdCfm4btGectxb/ZCYFkXPBvqtqzK/H9ANk/Xlq1KFw0O2x2
      fgV7swYuiZ3kSYLbEQJBAPx2gXPkpraqEPmt//ce7AP5fHxC8Td2JFROsRfyweJP
      0dUdiLDgMxwpFZIhidU1sJBWX62Vqv6Jh/MvoMBhttsCQQDTEnqy9P+VAS1huYkl
      XwWjbZKxHEIaxW+Bg/ombXLhGudpy+chHFy4QEgA1Puda7kxgcTMhRj1i1EliytB
      ecR1AkEA9STJPNS4wURQKXG4y6f9+zoFk8+Y1IbmberfcWaqt2tUUEFWWpHJbz3B
      kVz5rstsgCsuyqo7GOBI323PBR+c4wJBAJxA3dyFSn+AM5xYwZKM2Zu2jhXXGYjA
      CGAU16aC97x1MkM704rLeEheLe+PvAU5rgtcSdgt3+BGlnf4orkB79ECQE2uNOTP
      enyfRVLgN0j3ZLSyOSqw+kqGsoU2jpcxFCGy0RGk3z1fHVccHfspqCA7AMj9jK5O
      3EULSLOmkXQHPLQ=
      -----END PRIVATE KEY-----
    expiration: 31536000
    public-key: |
      -----BEGIN PUBLIC KEY-----
      MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDQJ+YIlU+6Mv3a3KNHrzhSVGEp
      xy22LVxuDFLhFDNkHRl/Xdat8uZPaK8F98K7eFj+ICJEHQdBM0AKaVmTFG21bIm1
      VpFUk0OWiJz4Iy432g5XD+VOGdYYd0Ij2dNYtwFmGPB0vtkgSJUSx0uV9DI2ZHWz
      lRDzg1O9yTUoxSs+FwIDAQAB
      -----END PUBLIC KEY-----

spring:
  datasource:
    url: jdbc:mysql://192.168.1.12:3306/cs_mall_passport?useSSL=false&useUnicode=true&allowPublicKeyRetrieval=true&characterEncoding=utf-8&serverTimezone=Asia/Shanghai&allowMultiQueries=true
    username: cs_mall_user
    password: Taren123
  redis:
    host: 192.168.1.147    #写自己redis负载均衡的地址
    port: 6379
    password: "Taren123"
    database: 9
    jedis:
      pool:
        max-active: 8
        min-idle: 5
        max-idle: 8
        max-wait: PT15M
        time-between-eviction-runs: PT15M

  cloud:
    nacos:
      discovery:
        server-addr: 192.168.1.13:8848
        namespace: linux                #nacos命名空间（project）id
        enabled: true
        register-enabled: true
  application:
    name: passport-server
mybatis:
  mapper-locations: classpath:mapper/*.xml
logging:
  level:
    com.tarena.passport: trace
  file:
    name: logs/passport.log

#打包测试
[root@test tarena-passport]# mvn clean package -D'maven.test.skip'=true
...
[INFO] passport-bom ....................................... SUCCESS [  0.228 s]
[INFO] passport-protocol .................................. SUCCESS [  2.154 s]
[INFO] passport-autoconfigure ............................. SUCCESS [03:57 min]
[INFO] passport-common .................................... SUCCESS [  0.766 s]
[INFO] passport-domain .................................... SUCCESS [  0.456 s]
[INFO] passport-adaptor ................................... SUCCESS [  0.520 s]
[INFO] passport-instruction ............................... SUCCESS [  0.320 s]
[INFO] passport-sdk ....................................... SUCCESS [  0.372 s]
[INFO] passport-main ...................................... SUCCESS [  0.262 s]
[INFO] passport-provider .................................. SUCCESS [  7.911 s]
[INFO] tarena-passport .................................... SUCCESS [  0.001 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:11 min
[INFO] Finished at: 2023-07-10T01:39:06+08:00
[INFO] ------------------------------------------------------------------------

查看结果并启动jar测试
[root@test tarena-passport]# ls passport-provider/target/passport-provider-1.0-SNAPSHOT.jar
[root@test tarena-passport]# java -Dfile.encoding=utf-8 -jar passport-provider/target/passport-provider-1.0-SNAPSHOT.jar -Xmx128M -Xms128M -Xmn64m -XX:MaxMetaspaceSize=128M -XX:MetaspaceSize=128M --server.port=30094 --spring.profiles.active=vm

#清理测试文件
[root@test tarena-passport]# mvn clean
```

二、安装jenkins

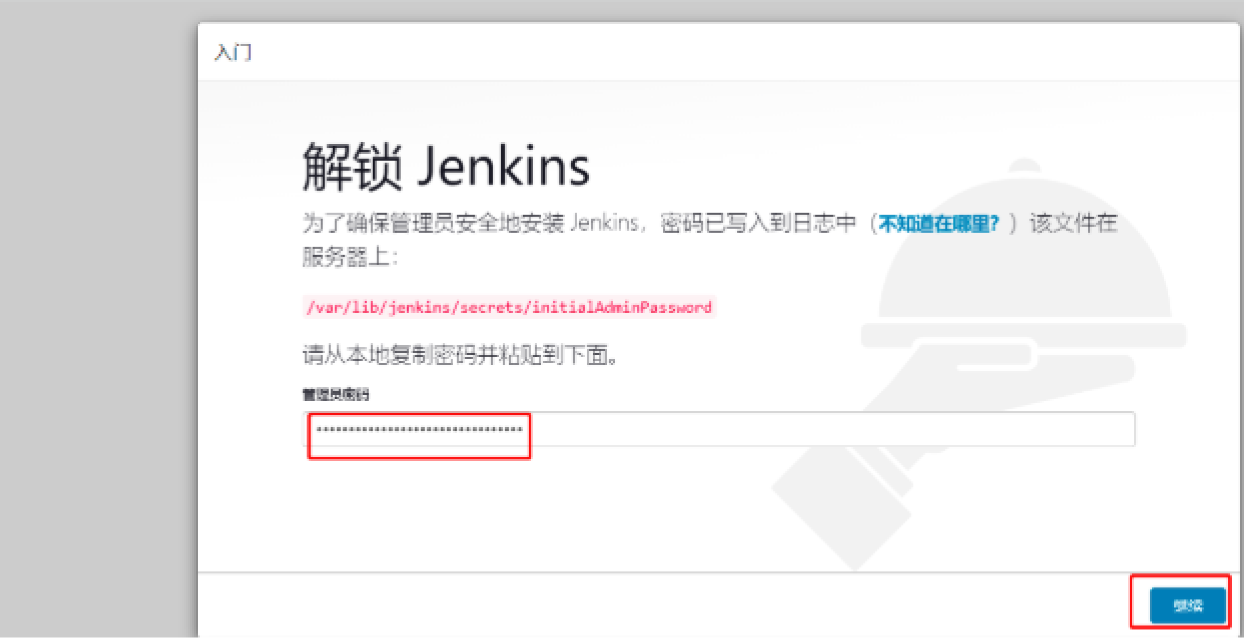
P1、安装jenkins

1、实现自动化部署

```
[root@test ~]# scp 192.168.1.252:/root/jenkins-2.440.1-1.1.noarch.rpm /root/
[root@test tarena-passport]# cd /root/
[root@test ~]# yum -y install jenkins-2.440.1-1.1.noarch.rpm
[root@test ~]# systemctl enable --now jenkins
[root@test ~]# ss -antlp | grep 8080
LISTEN 0      50                *:8080            *:*     users:(("java",pid=48056,fd=9))
```

2、jenkins初始化

给jenkins直接绑定一个公网IP，浏览器访问test主机的公网ip：公网ip:8080



入门

解锁 Jenkins

为了确保管理员安全地安装 Jenkins，密码已写入到日志中（不知道在哪里？）该文件在服务器上：

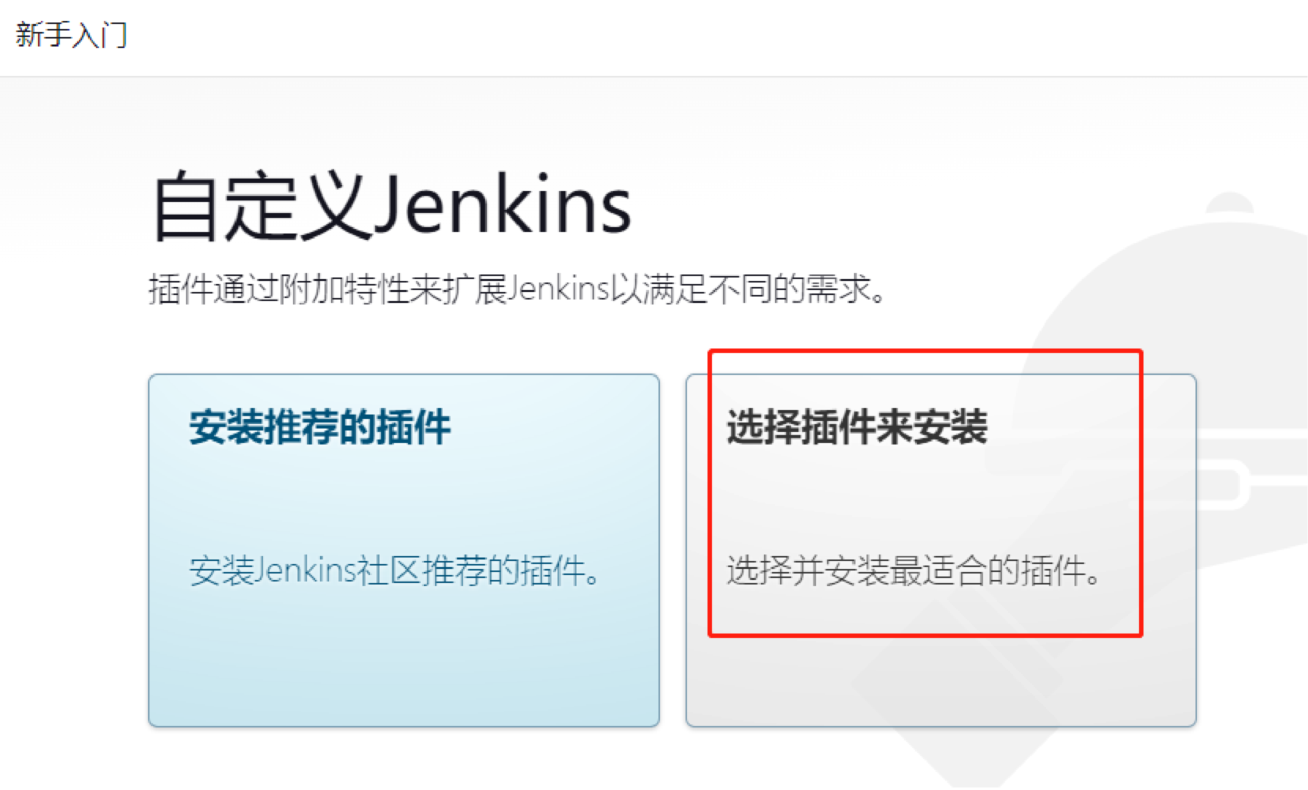
/var/lib/jenkins/secrets/initialAdminPassword

请从本地复制密码并粘贴到下面。

管理员密码

································

```
#获取Jenkins初始密码
[root@test ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
197027e9419f4943aae3add096deb4b0
```



新手入门

自定义Jenkins

插件通过附加特性来扩展Jenkins以满足不同的需求。

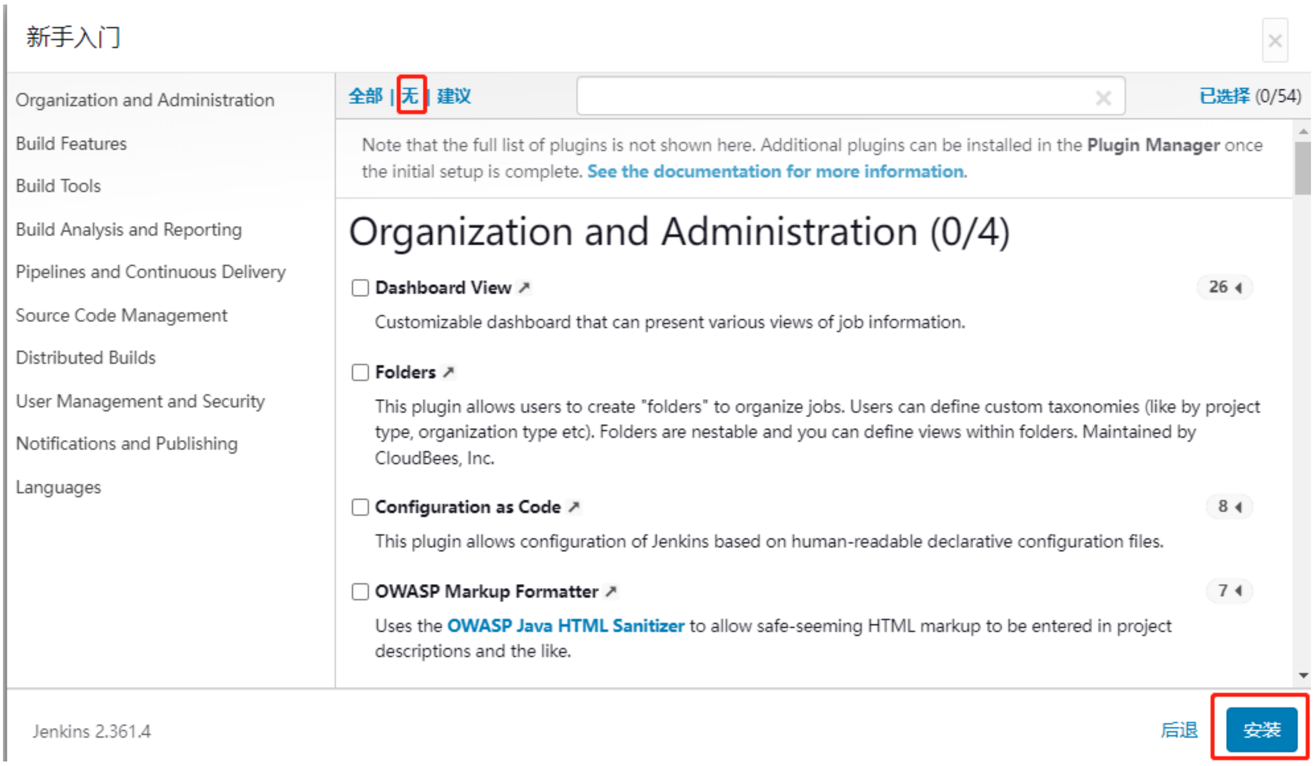
安装推荐的插件

安装Jenkins社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

选择无，暂时不安装插件



新手入门

Organization and Administration
Build Features
Build Tools
Build Analysis and Reporting
Pipelines and Continuous Delivery
Source Code Management
Distributed Builds
User Management and Security
Notifications and Publishing
Languages

全部 | 无 | 建议

已选择 (0/54)

Note that the full list of plugins is not shown here. Additional plugins can be installed in the **Plugin Manager** once the initial setup is complete. **See the documentation for more information.**

Organization and Administration (0/4)

□ **Dashboard View** ↗ 26

Customizable dashboard that can present various views of job information.

□ **Folders** ↗

This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.

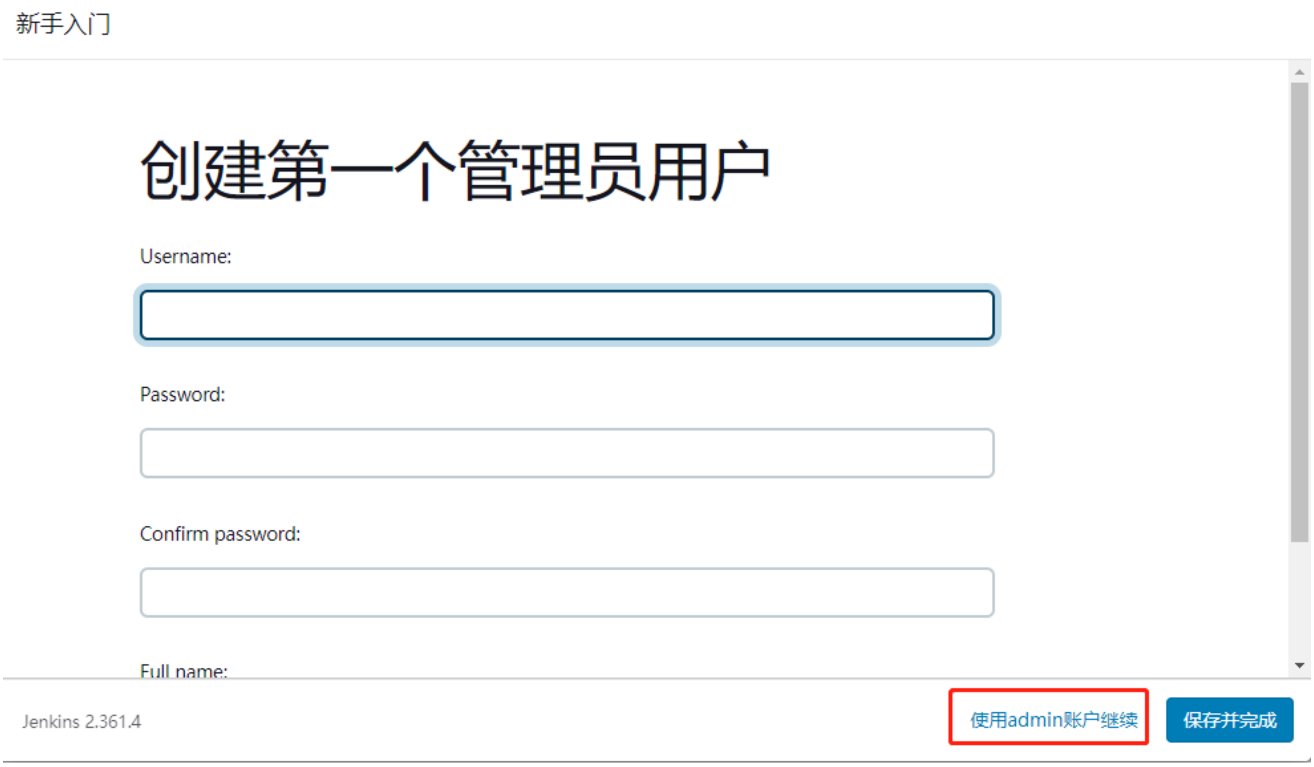
□ **Configuration as Code** ↗ 8

This plugin allows configuration of Jenkins based on human-readable declarative configuration files.

□ **OWASP Markup Formatter** ↗ 7

Uses the **OWASP Java HTML Sanitizer** to allow safe-seeming HTML markup to be entered in project descriptions and the like.

Jenkins 2.361.4



新手入门

创建第一个管理员用户

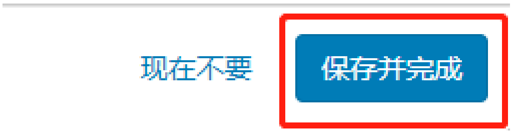
Username:

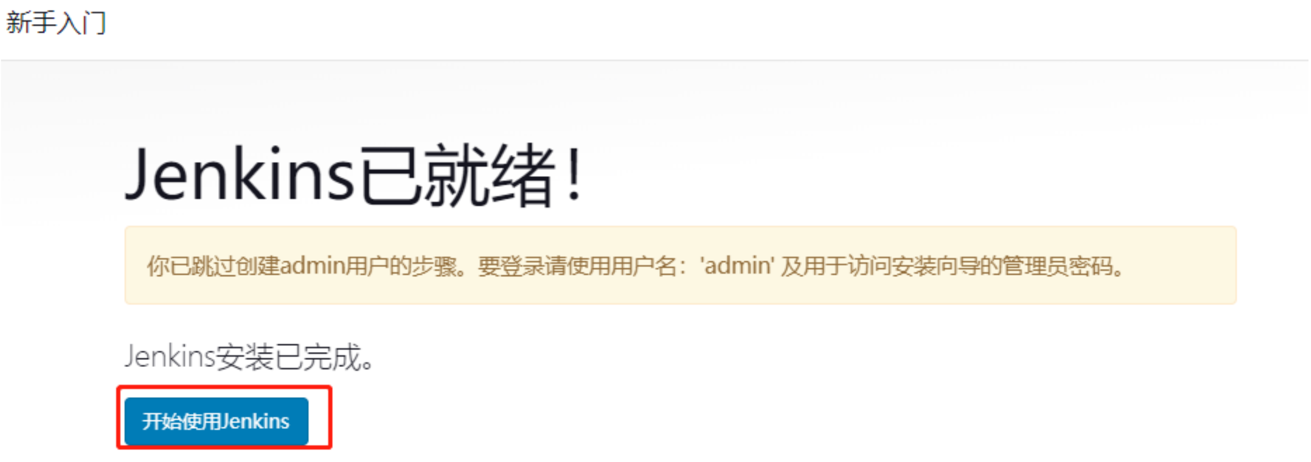
Password:

Confirm password:

Full name:

Jenkins 2.361.4





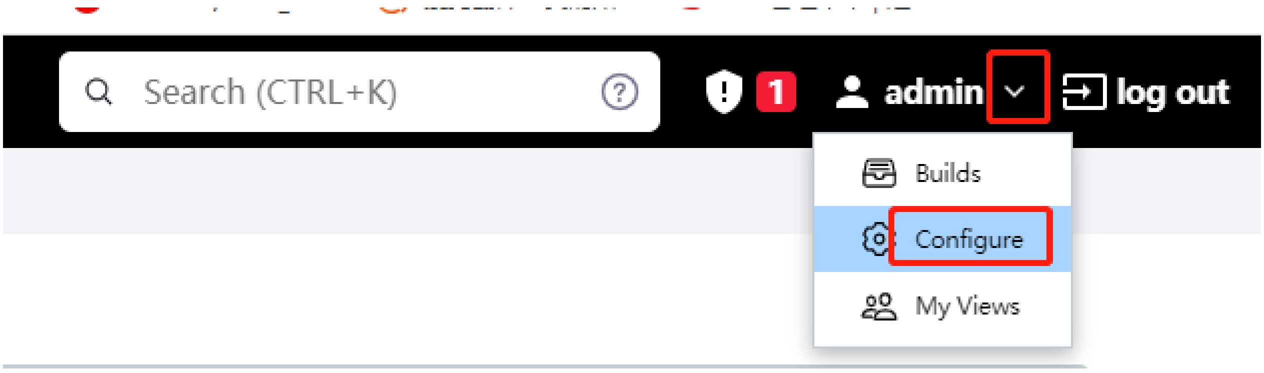
新手入门

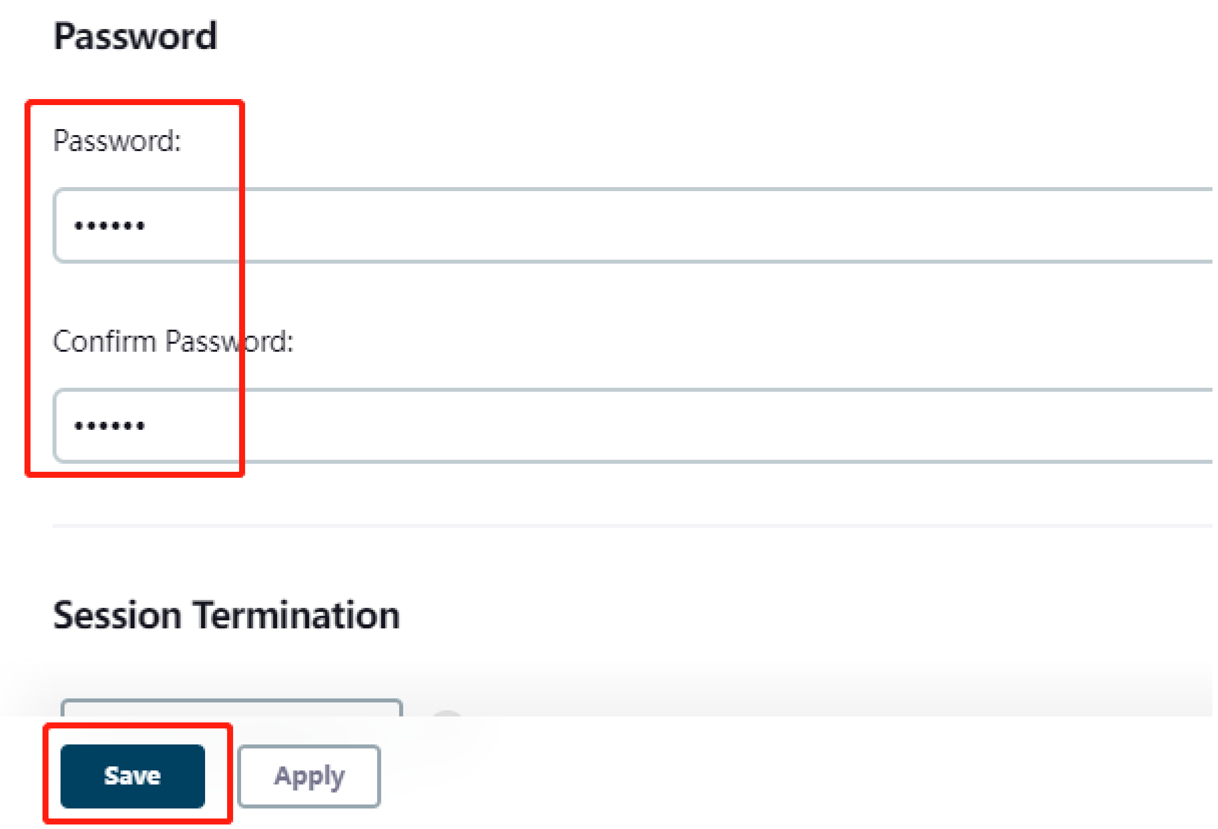
Jenkins已就绪!

你已跳过创建admin用户的步骤。要登录请使用用户名：'admin' 及用于访问安装向导的管理员密码。

Jenkins安装已完成。

更改密码为123456，点击save保存





Password

Password:

••••••

Confirm Password:

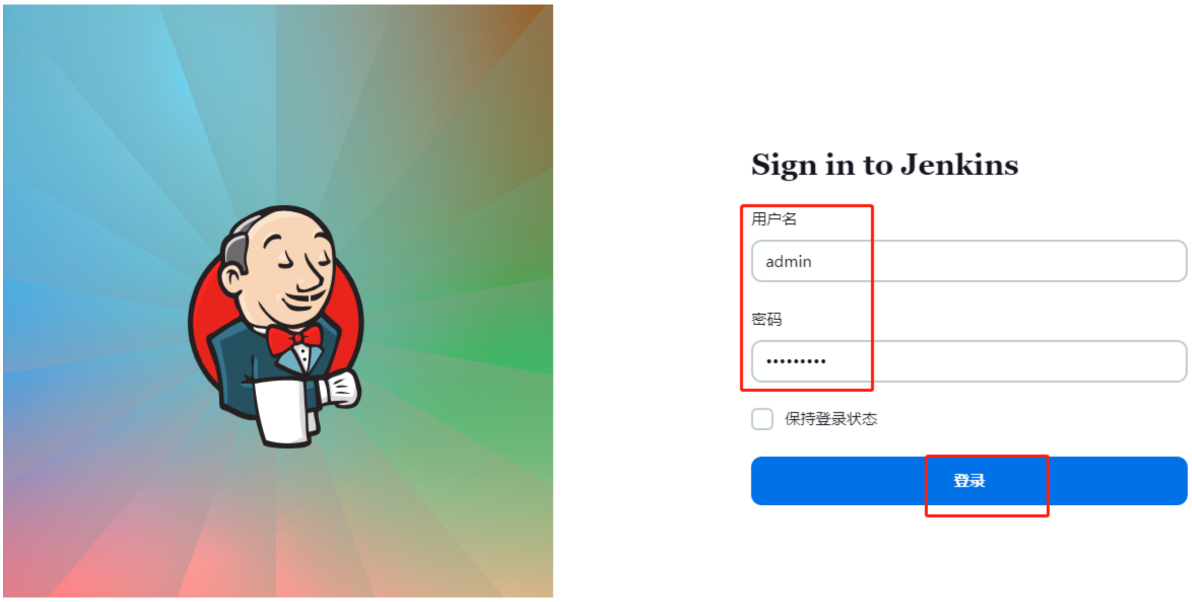
••••••

Session Termination

3、登录测试

重新使用用户admin，密码123456登录jenkins

```
#离线部署Jenkins插件
[root@test ~]# ls /var/lib/jenkins/plugins/            #空目录，没有任何插件
[root@test ~]# tar -xPpf jenkins_plugins.tar.gz -C /    #解压jenkins插件，保留文件归属
[root@test ~]# systemctl restart jenkins.service       #重启Jenkins加载插件
[root@test ~]# ss -antlp | grep 8080
LISTEN 0      50                *:8080          *:*     users:(("java",pid=18962,fd=9))
浏览器刷新jenkins页面，登录jenkins
```



Sign in to Jenkins

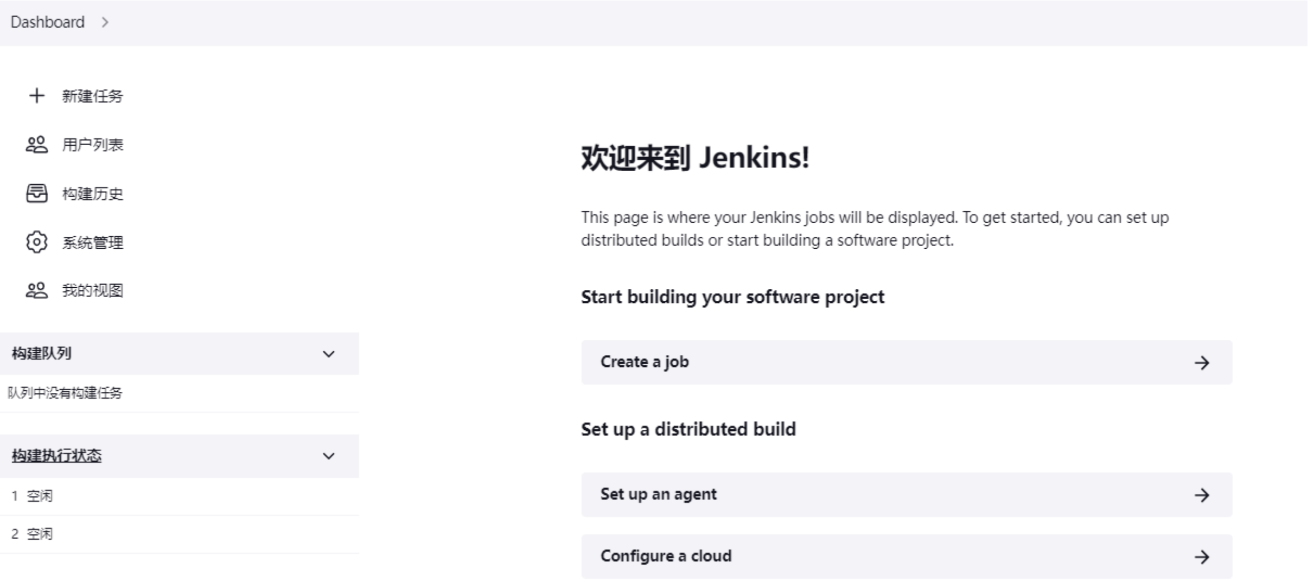
用户名

admin

密码

•••••••••

□ 保持登录状态



Dashboard >

＋ 新建任务

用户列表

构建历史

系统管理

我的视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

欢迎来到 Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

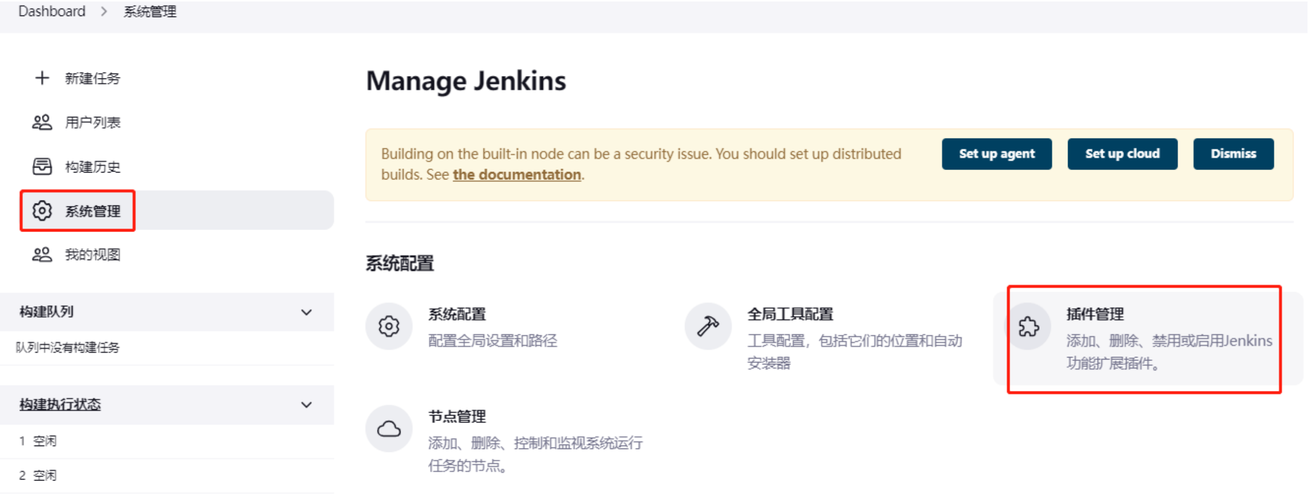
Set up a distributed build

Set up an agent →

Configure a cloud →

4、对接gitee

需要安装gitee的插件才能去gitee拉取代码



Dashboard > 系统管理

＋ 新建任务

用户列表

构建历史

系统管理

我的视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Set up agent | Set up cloud | Dismiss

系统配置

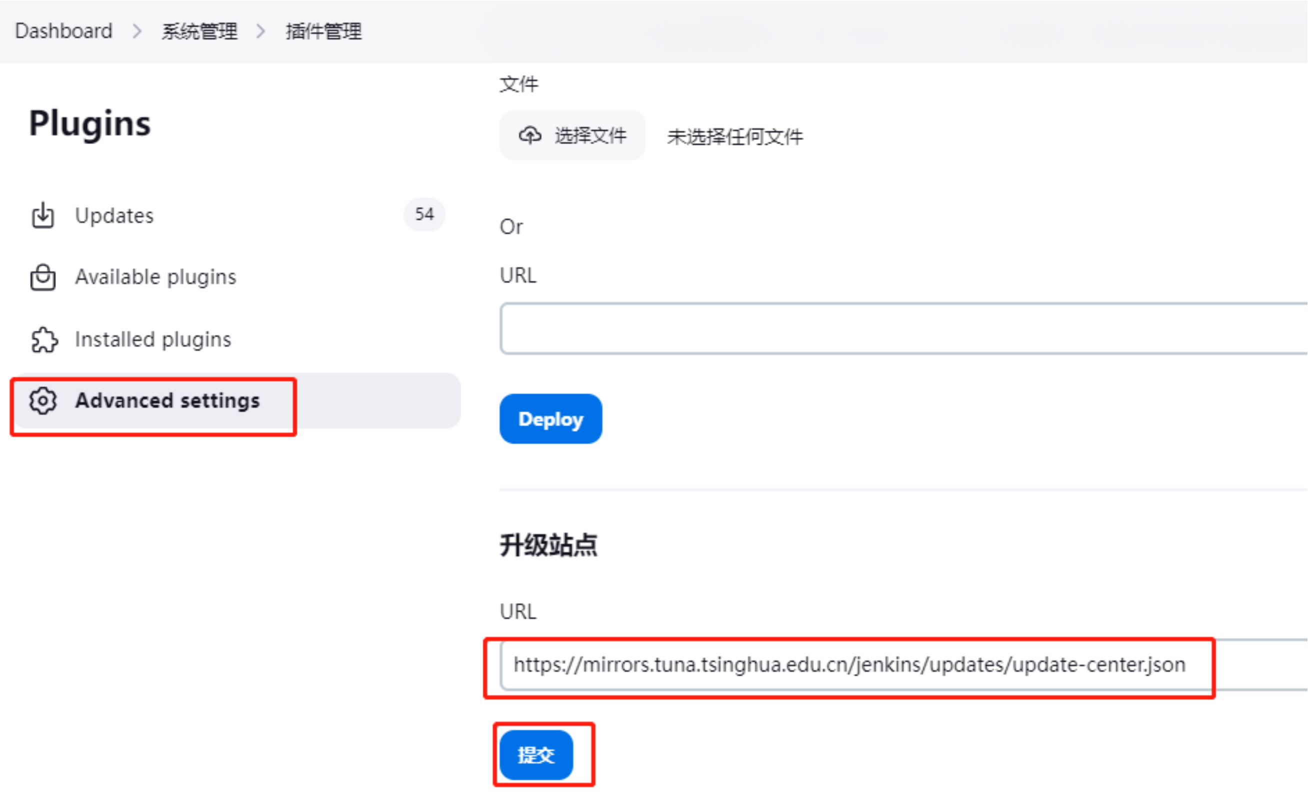
系统配置
配置全局设置和路径

全局工具配置
工具配置，包括它们的位置和自动安装器

插件管理
添加、删除、禁用或启用Jenkins功能扩展插件。

节点管理
添加、删除、控制和监视系统运行任务的节点。

点击Advanced Settings，使用国内站点下载gitee插件：https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json



Dashboard > 系统管理 > 插件管理

Plugins

- Updates 54
- Available plugins
- Installed plugins
- Advanced settings

文件

选择文件 未选择任何文件

Or

URL

Deploy

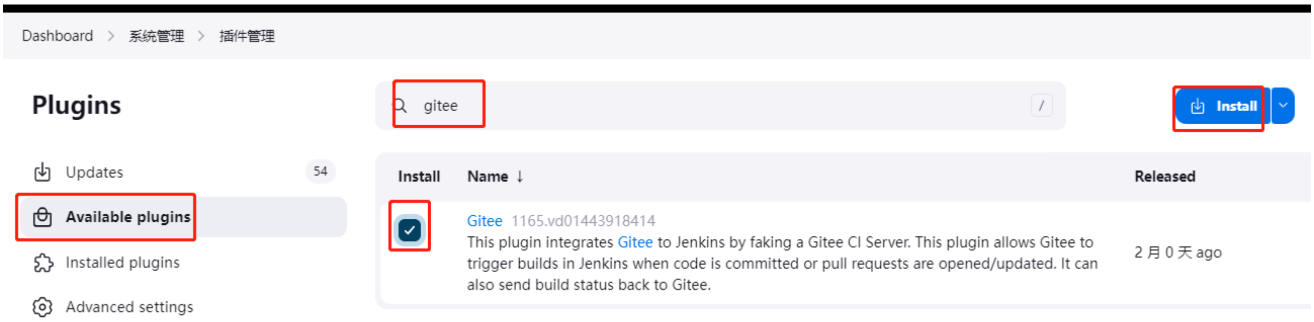
升级站点

URL

https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json

提交

点击Available plugins



Dashboard > 系统管理 > 插件管理

Plugins

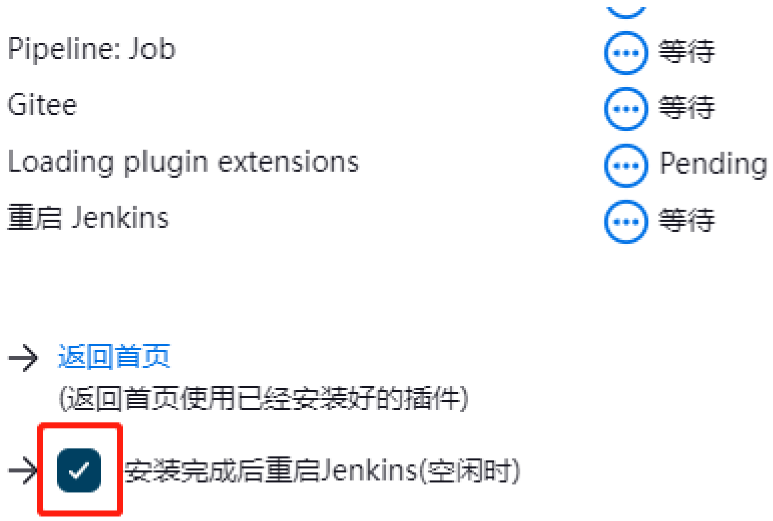
- Updates 54
- Available plugins
- Installed plugins
- Advanced settings

gitee

Install

Install	Name ↓	Released
☑	Gitee 1165.vd01443918414 This plugin integrates Gitee to Jenkins by faking a Gitee CI Server. This plugin allows Gitee to trigger builds in Jenkins when code is committed or pull requests are opened/updated. It can also send build status back to Gitee.	2 月 0 天 ago

安装成功后，选择重启即可，会自动刷新

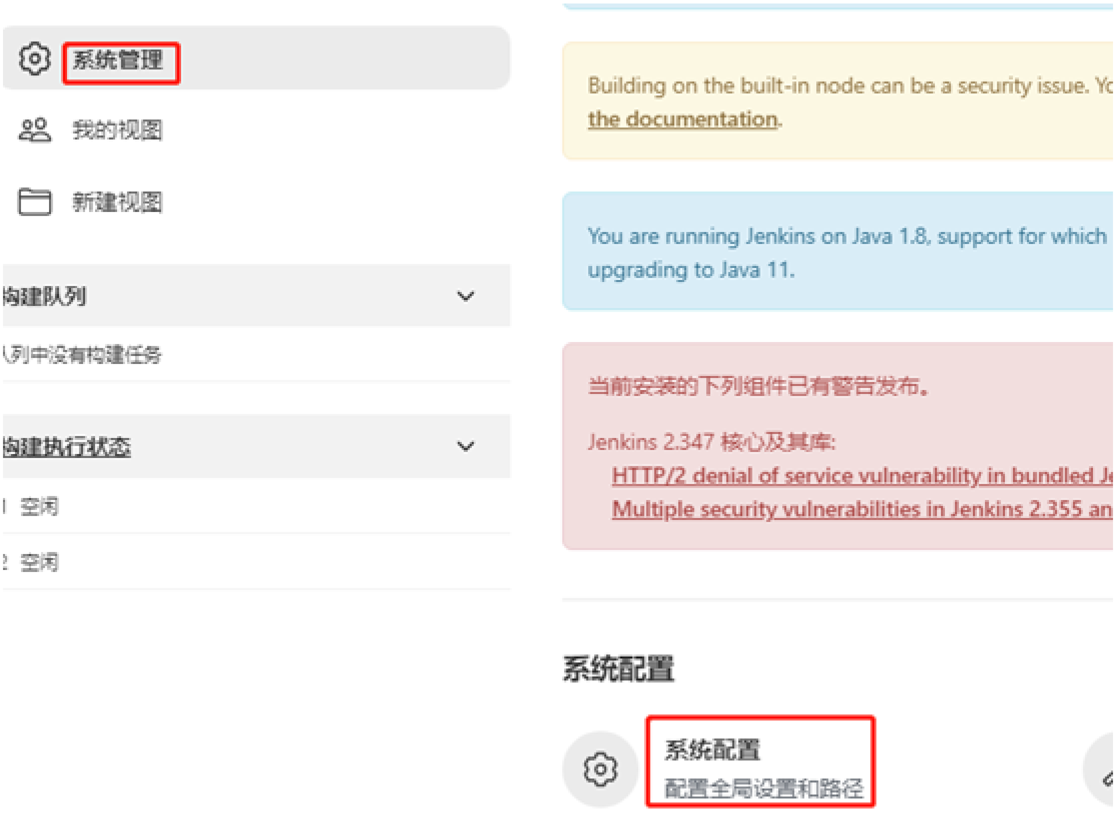


Pipeline: Job	等待
Gitee	等待
Loading plugin extensions	Pending
重启 Jenkins	等待

→ 返回首页
(返回首页使用已经安装好的插件)

→ ☑ 安装完成后重启Jenkins(空闲时)

配置jenkins连接gitee拉取代码



系统管理

我的视图

新建视图

构建队列

列中没有构建任务

构建执行状态

1 空闲

2 空闲

Building on the built-in node can be a security issue. Yo
the documentation.

You are running Jenkins on Java 1.8, support for which
upgrading to Java 11.

当前安装的下列组件已有警告发布。

Jenkins 2.347 核心及其库:
HTTP/2 denial of service vulnerability in bundled Je
Multiple security vulnerabilities in Jenkins 2.355 an

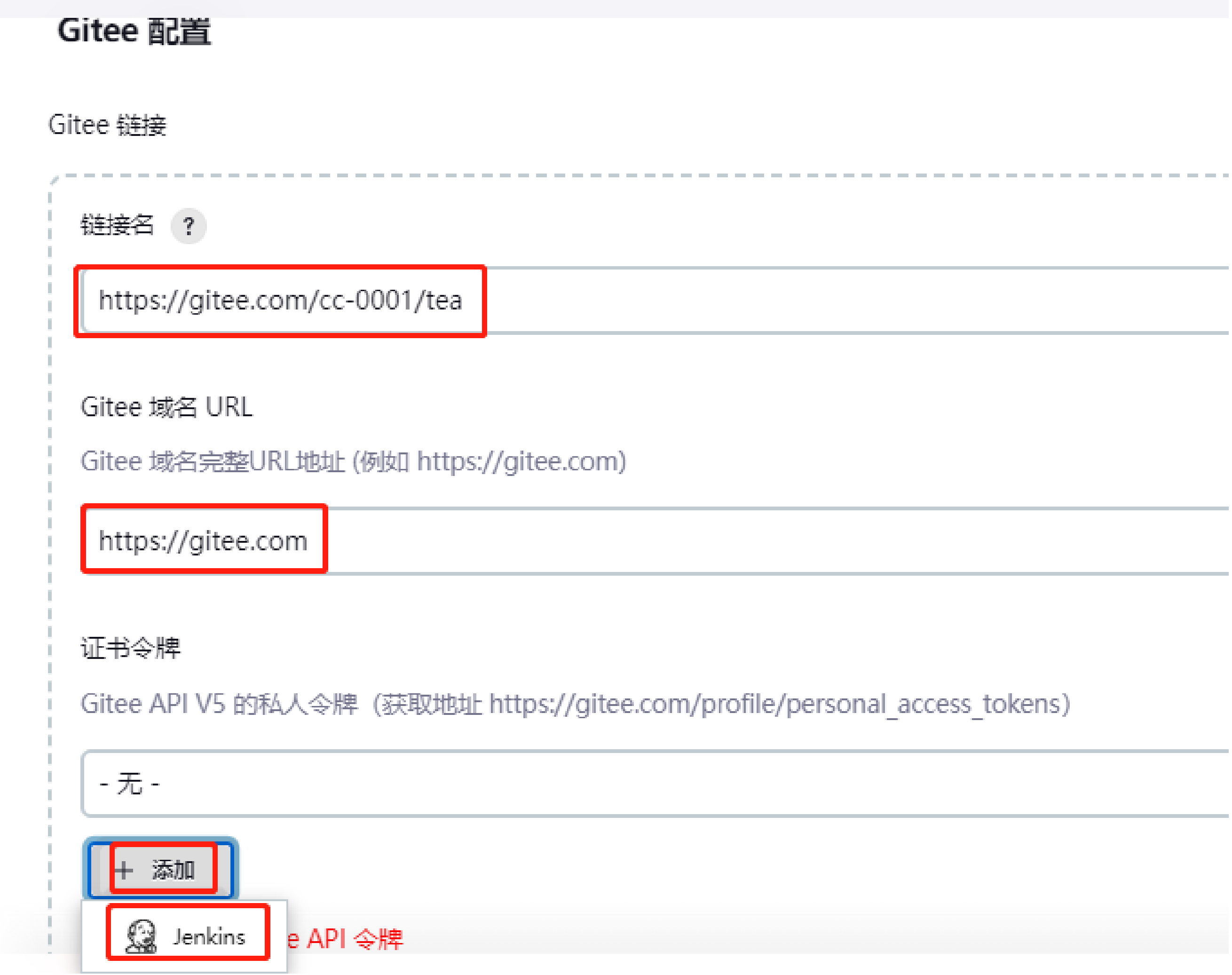
系统配置

系统配置
配置全局设置和路径

链接名：https://gitee.com/cc-0001/tea ， gitee拉取的项目

域名：https://gitee.com，gitee的地址

证书令牌id为：b27648705bc56d43e322af70b7b0ac97



Gitee 配置

Gitee 链接

链接名 ?

https://gitee.com/cc-0001/tea

Gitee 域名 URL

Gitee 域名完整URL地址 (例如 https://gitee.com)

https://gitee.com

证书令牌

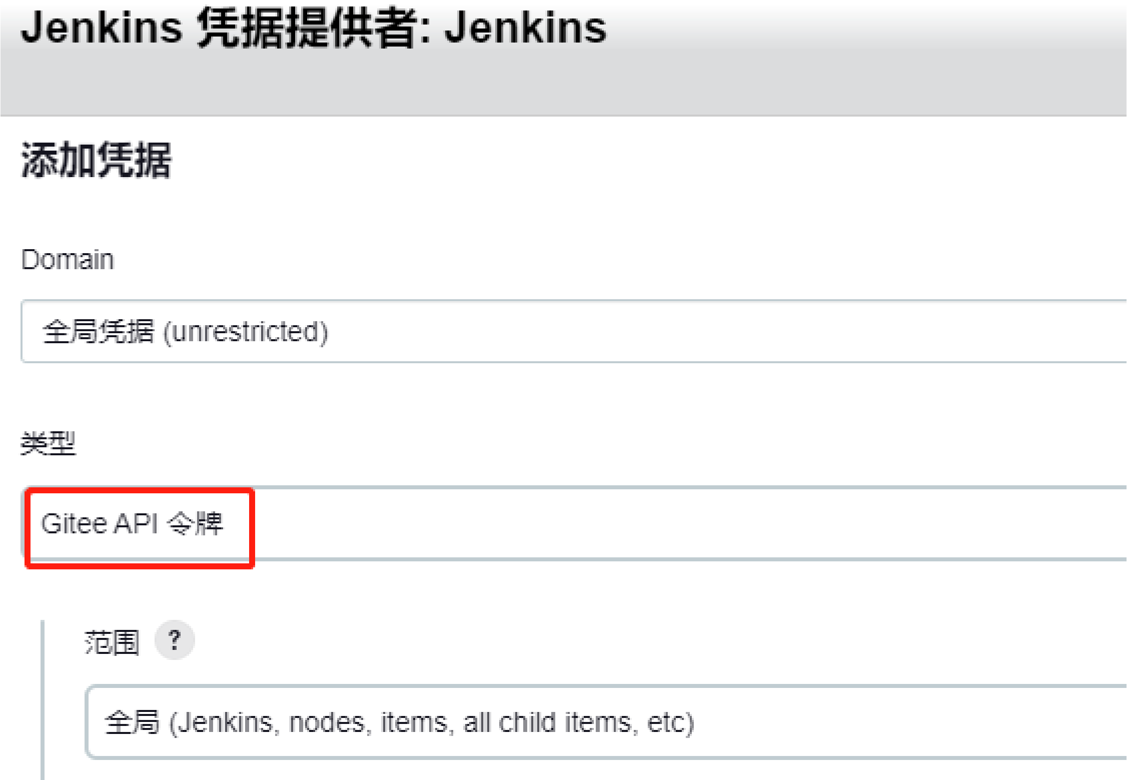
Gitee API V5 的私人令牌（获取地址 https://gitee.com/profile/personal_access_tokens)

- 无 -

＋ 添加

Jenkins

e API 令牌



Jenkins 凭据提供者: Jenkins

添加凭据

Domain

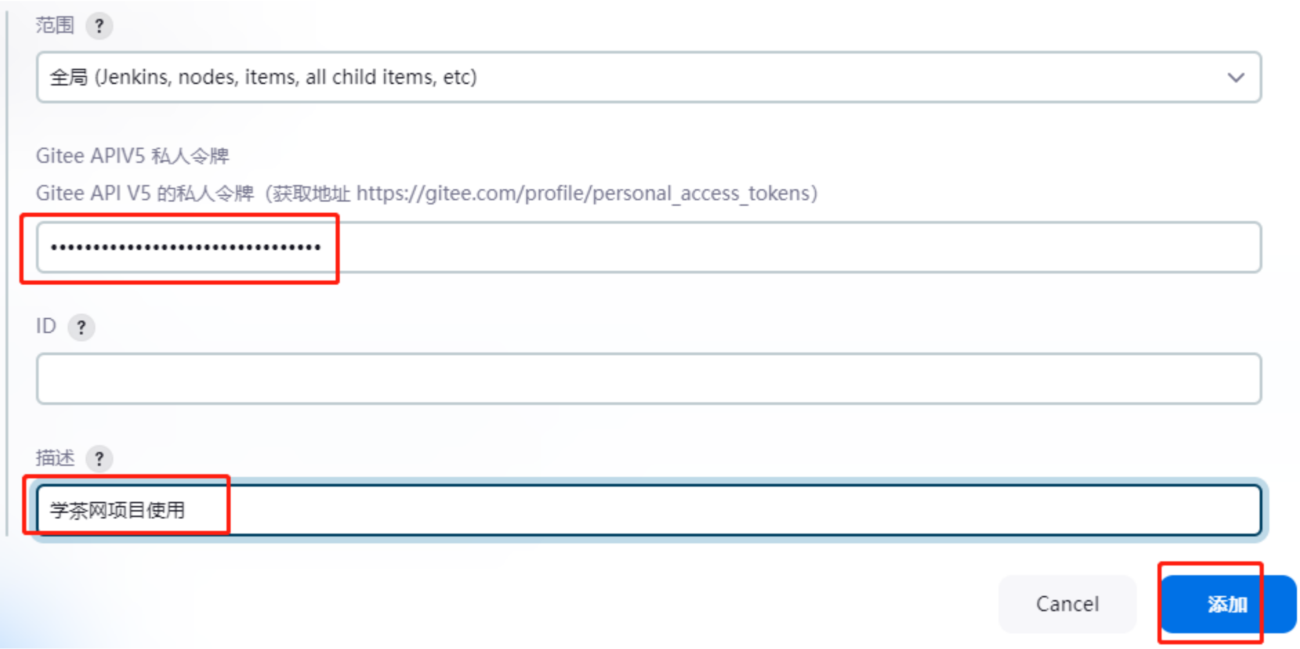
全局凭据 (unrestricted)

类型

Gitee API 令牌

范围 ?

全局 (Jenkins, nodes, items, all child items, etc)



范围 ?

全局 (Jenkins, nodes, items, all child items, etc)

Gitee APIV5 私人令牌

Gitee API V5 的私人令牌（获取地址 https://gitee.com/profile/personal_access_tokens）

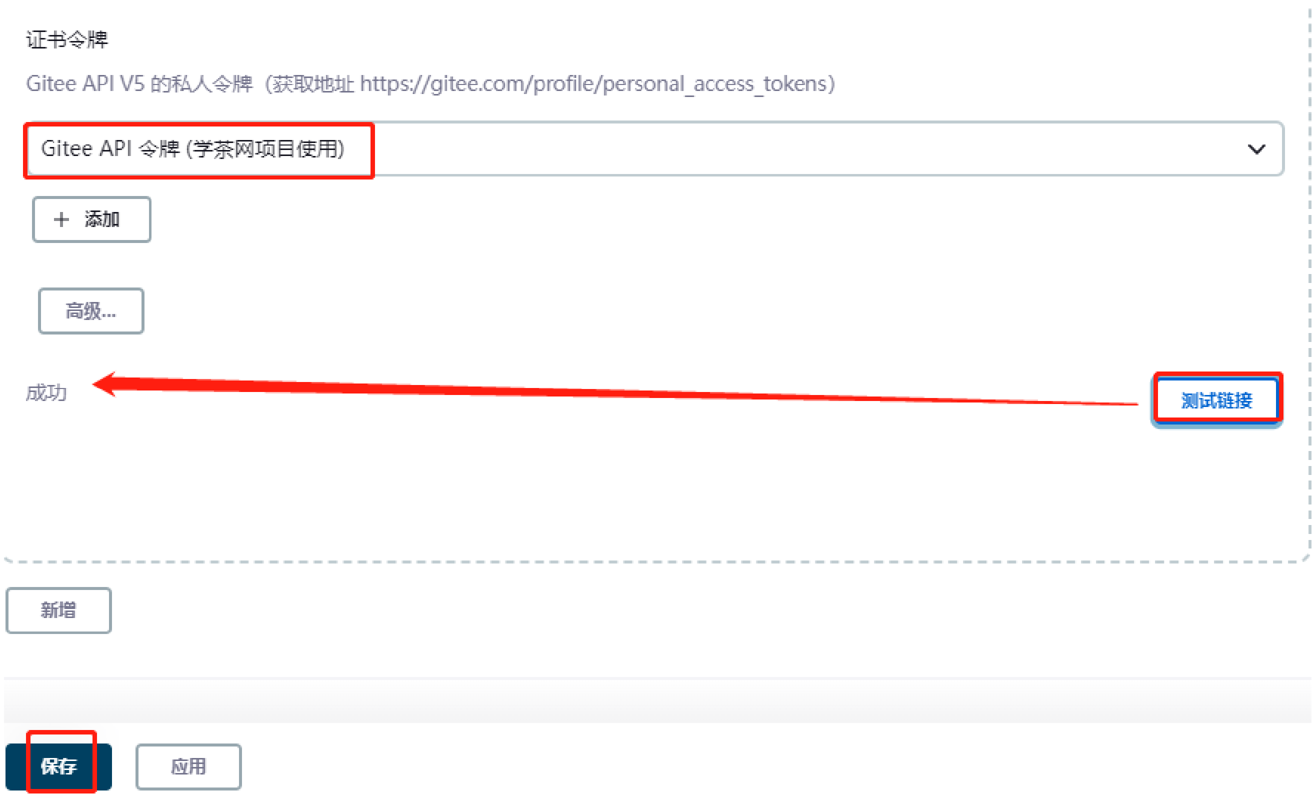
••••••••••••••••••••••••••••••••

ID ?

描述 ?

学茶网项目使用

选中令牌，点击测试连接，成功即可



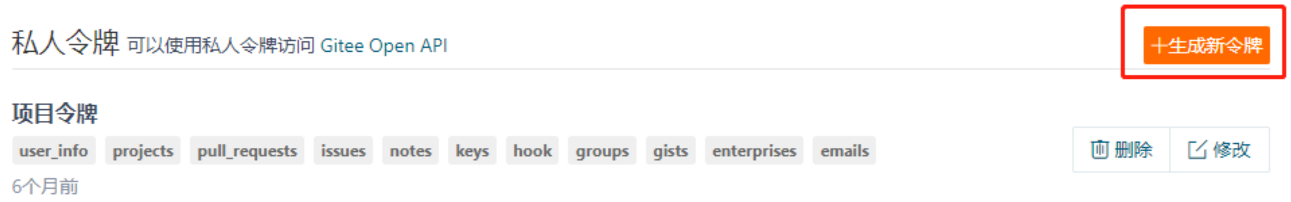
证书令牌

Gitee API V5 的私人令牌（获取地址 https://gitee.com/profile/personal_access_tokens）

Gitee API 令牌 (学茶网项目使用)

注：扩展（实验中不需要操作，知晓即可）

获取证书令牌：https://gitee.com/profile/personal_access_tokens，浏览器输入以上地址

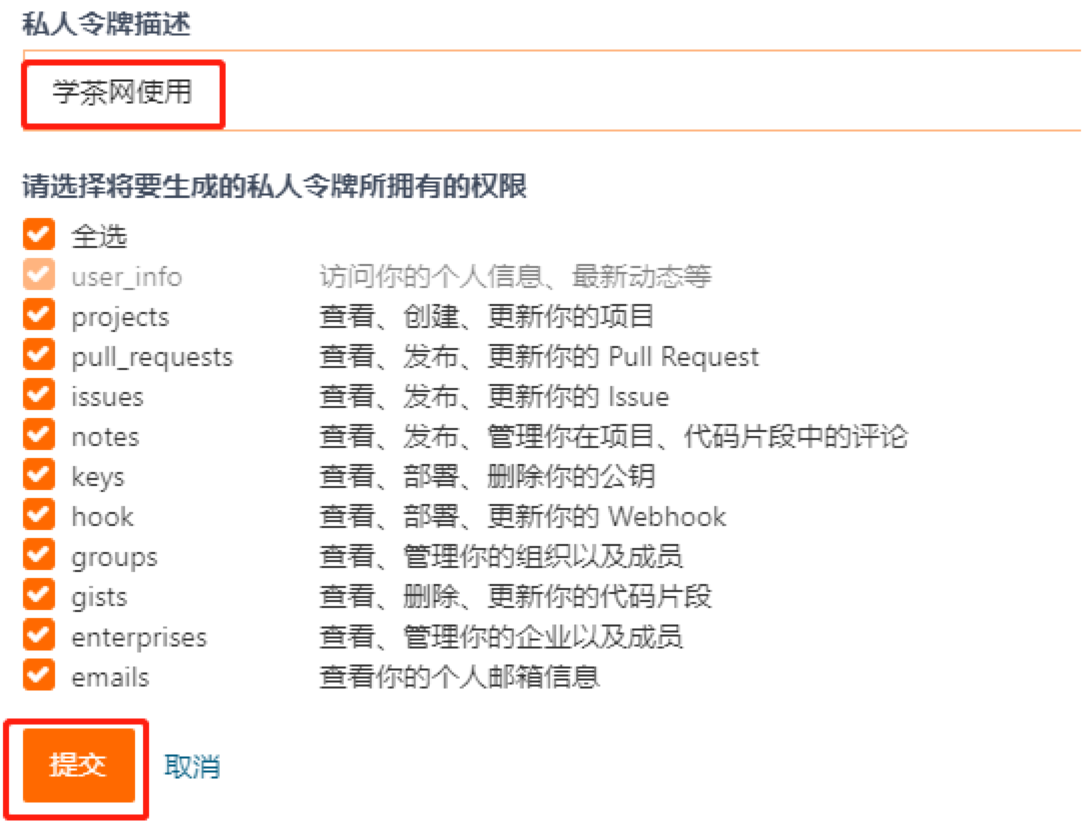


私人令牌 可以使用私人令牌访问 Gitee Open API

项目令牌

user_info projects pull_requests issues notes keys hook groups gists enterprises emails

6个月前

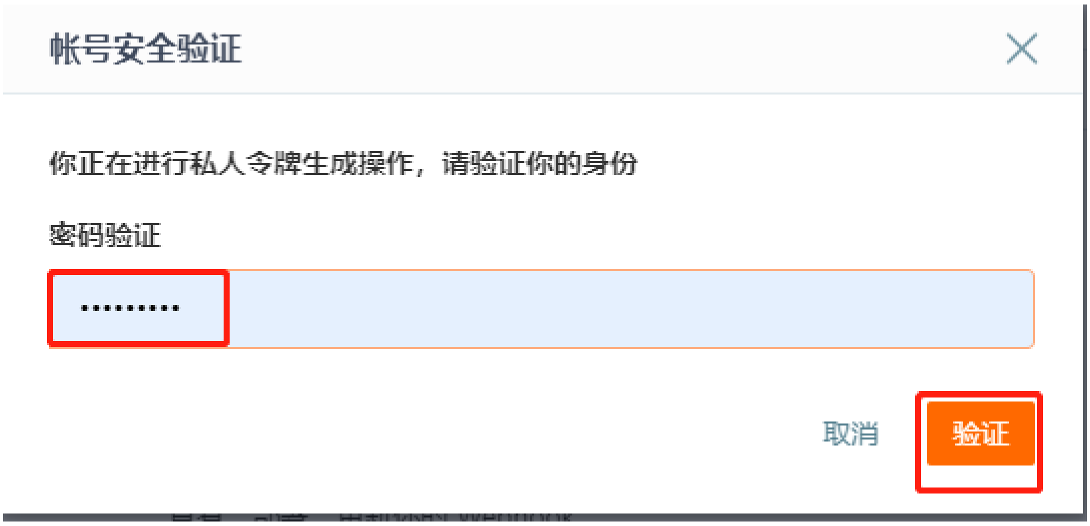


私人令牌描述

学茶网使用

请选择将要生成的私人令牌所拥有的权限

☑ 全选
☑ user_info 访问你的个人信息、最新动态等
☑ projects 查看、创建、更新你的项目
☑ pull_requests 查看、发布、更新你的 Pull Request
☑ issues 查看、发布、更新你的 Issue
☑ notes 查看、发布、管理你在项目、代码片段中的评论
☑ keys 查看、部署、删除你的公钥
☑ hook 查看、部署、更新你的 Webhook
☑ groups 查看、管理你的组织以及成员
☑ gists 查看、删除、更新你的代码片段
☑ enterprises 查看、管理你的企业以及成员
☑ emails 查看你的个人邮箱信息



帐号安全验证

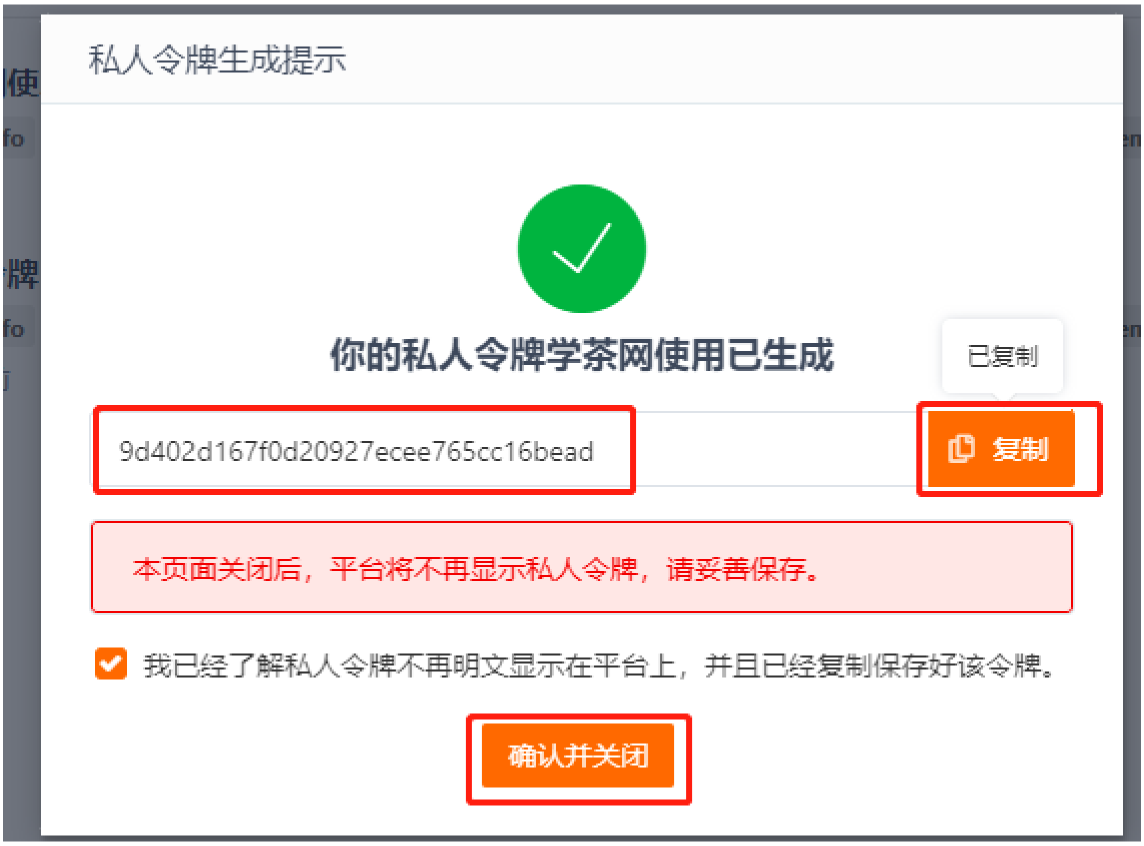
你正在进行私人令牌生成操作，请验证你的身份

密码验证

••••••••••

取消 验证

私人令牌：每次令牌都不一样



私人令牌生成提示

你的私人令牌学茶网使用已生成

已复制

9d402d167f0d20927ecee765cc16bead

复制

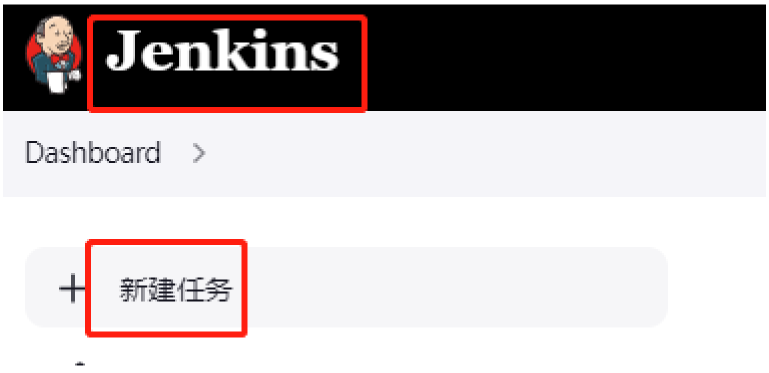
本页面关闭后，平台将不再显示私人令牌，请妥善保存。

☑ 我已经了解私人令牌不再明文显示在平台上，并且已经复制保存好该令牌。

确认并关闭

5、构建项目

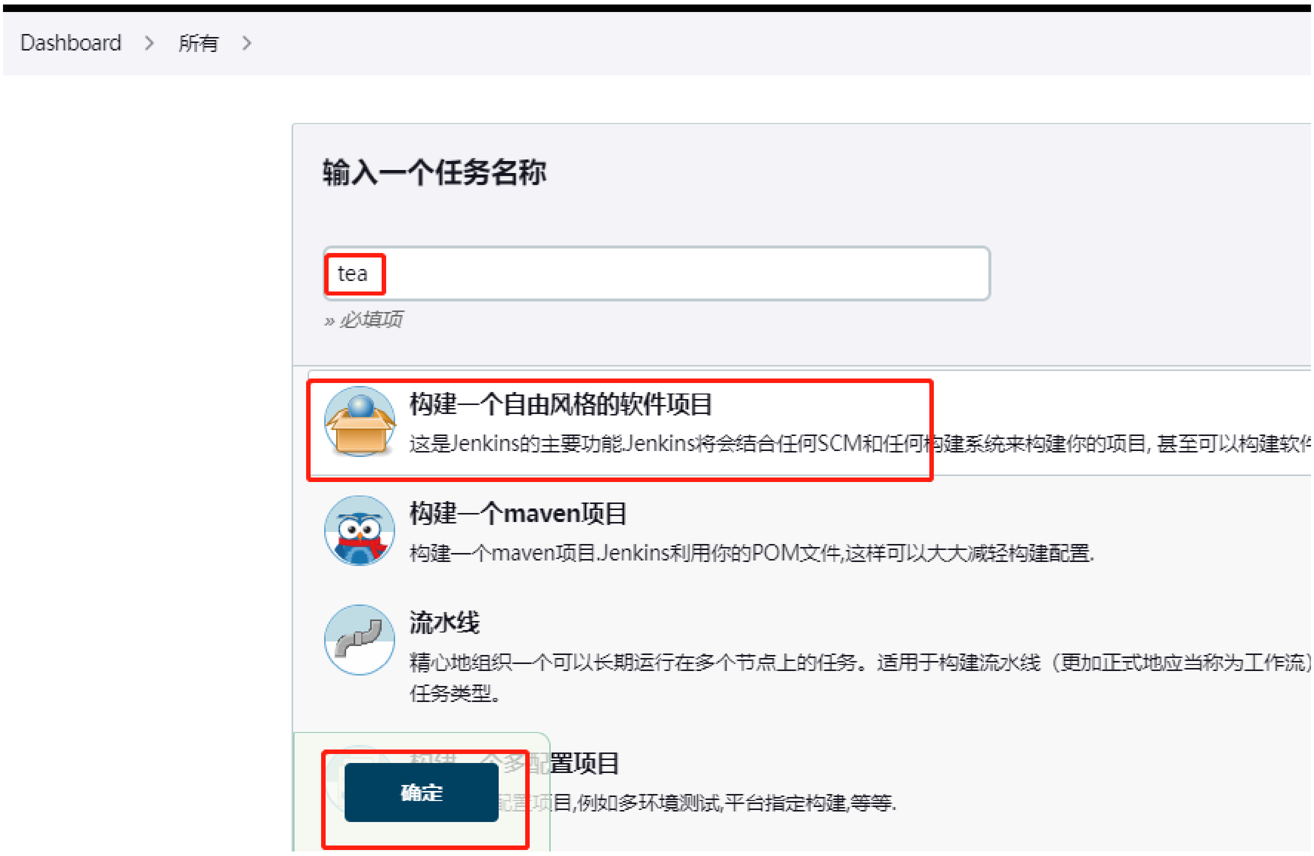
构建一个自由风格的项目，拉取代码



Jenkins

Dashboard >

+ 新建任务



Dashboard > 所有 >

输入一个任务名称

tea

» 必填项

构建一个自由风格的软件项目

这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件

构建一个maven项目

构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.

流水线

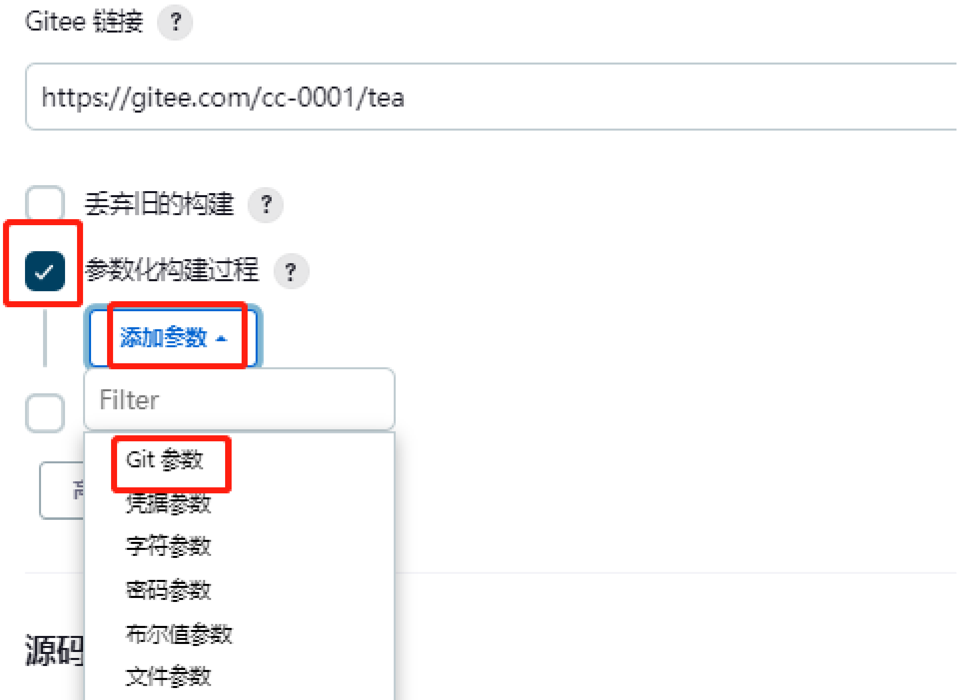
精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当称为工作流）任务类型。

构建一个多配置项目

配置项目,例如多环境测试,平台指定构建,等等.

设置参数和源码管理

选择【参数化构建过程】或英文【This project is parameterized】，点击【添加参数】的下拉列表，选择【Git 参数】



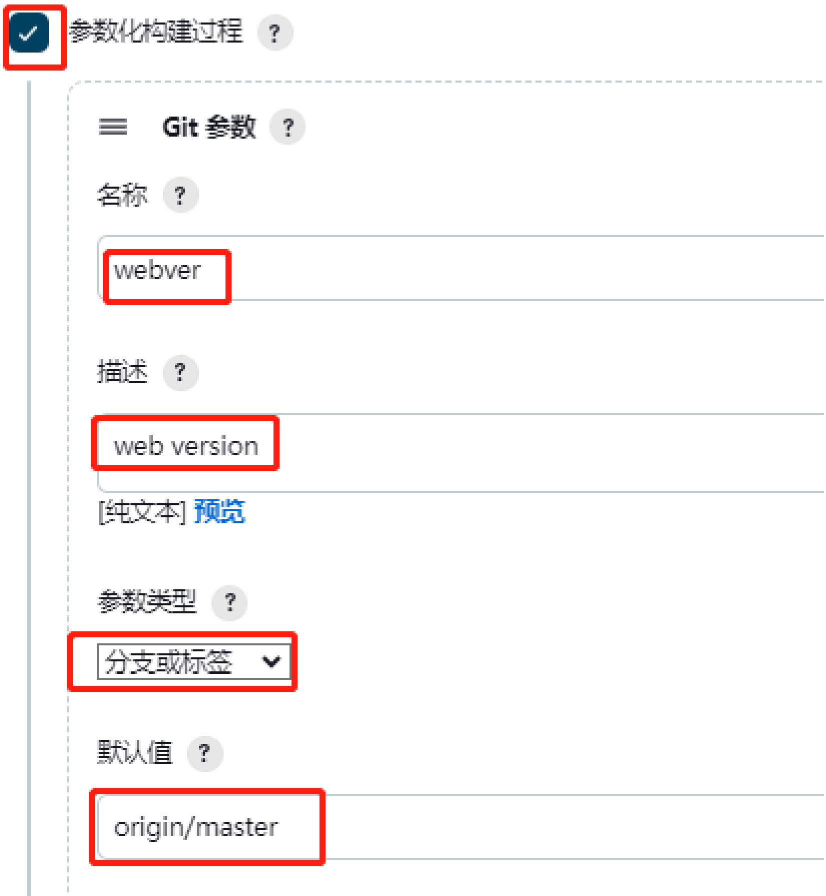
Gitee 链接 ?

https://gitee.com/cc-0001/tea

□ 丢弃旧的构建 ?

☑ 参数化构建过程 ?

定义变量【名称】可以任意，编写变量信息【描述】，根据提示填写【默认值】，选择【Git】，填写git仓库的url路径【Repository URL】



☑ 参数化构建过程 ?

Git 参数 ?

名称 ?

webver

描述 ?

web version

[纯文本] 预览

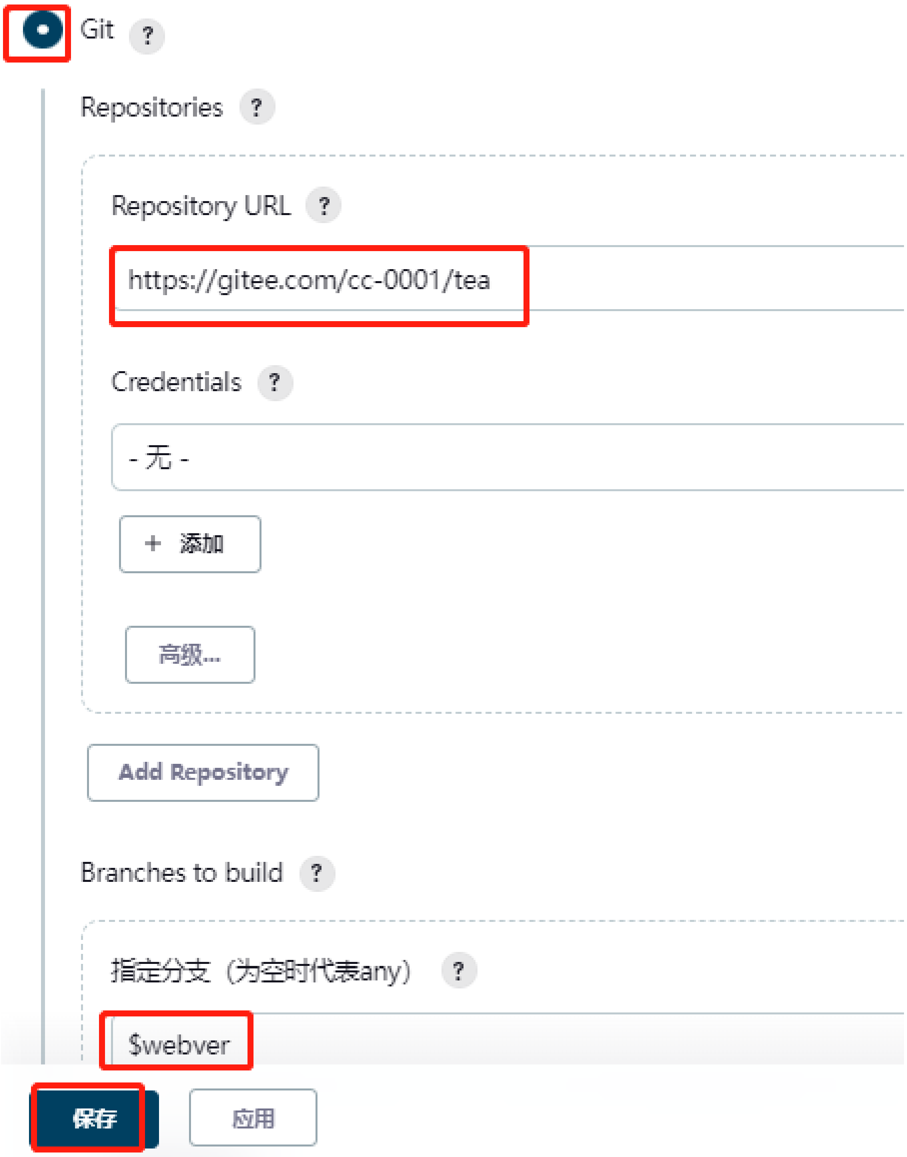
参数类型 ?

分支或标签

默认值 ?

origin/master

git获取项目地址：https://gitee.com/cc-0001/tea；分支为$webver



● Git ?

Repositories ?

Repository URL ?

https://gitee.com/cc-0001/tea

Credentials ?

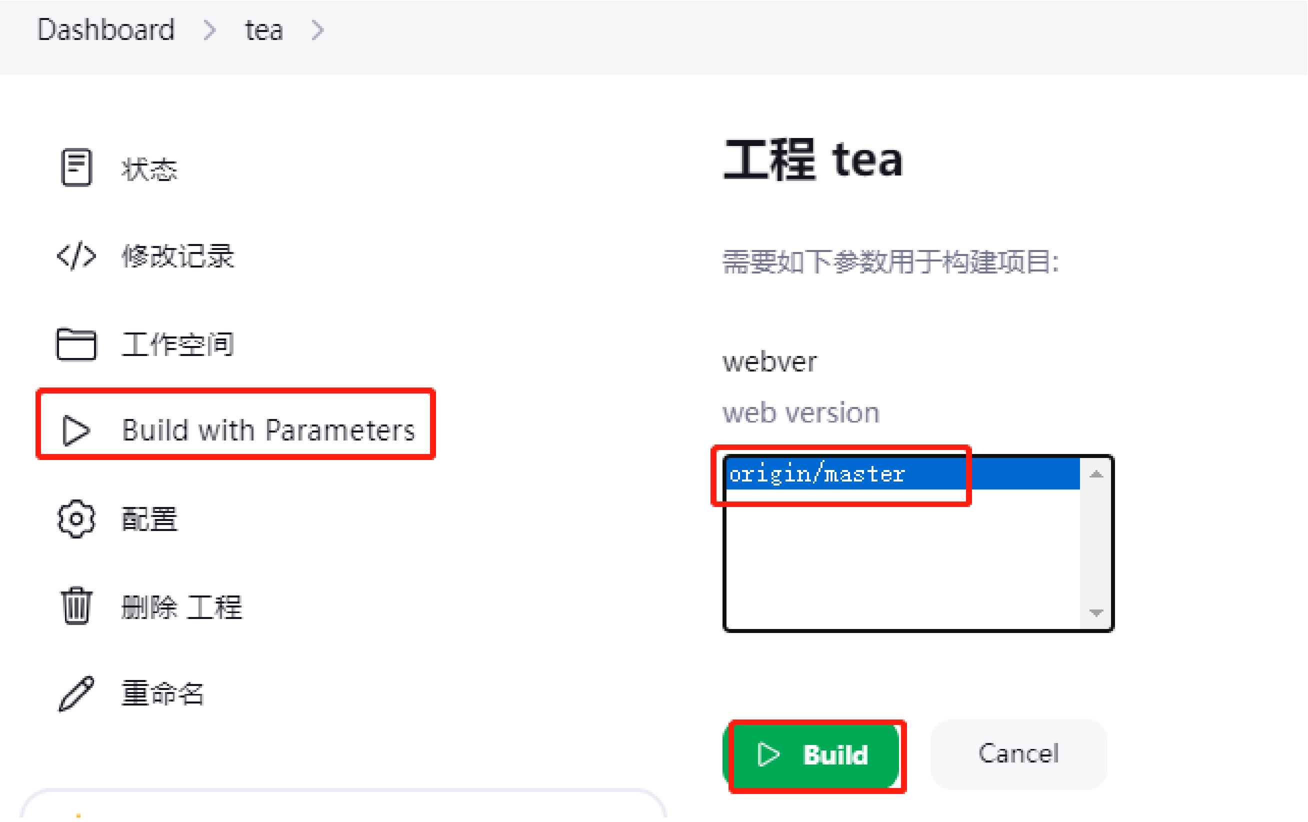
- 无 -

Branches to build ?

指定分支（为空时代表any） ?

$webver

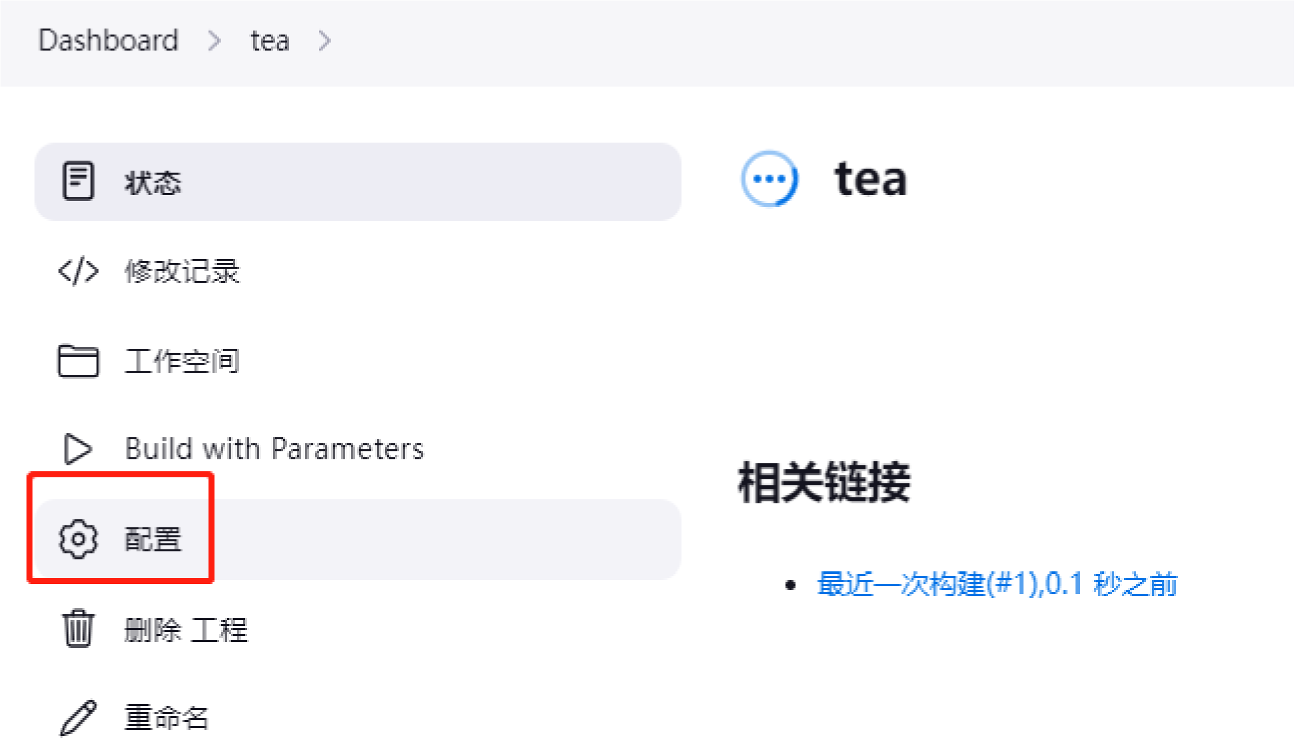
1、测试结果

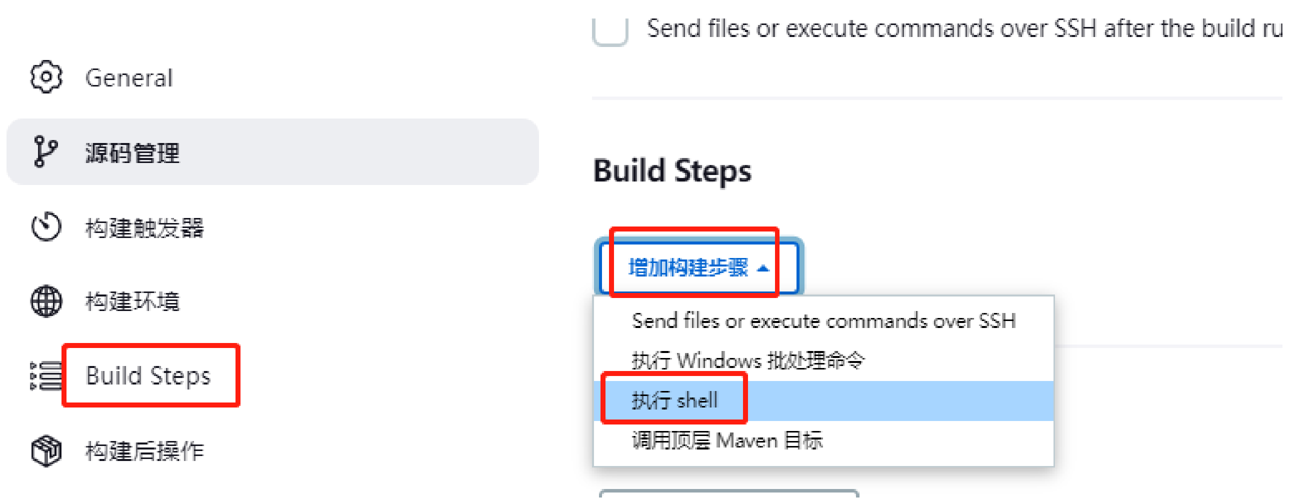


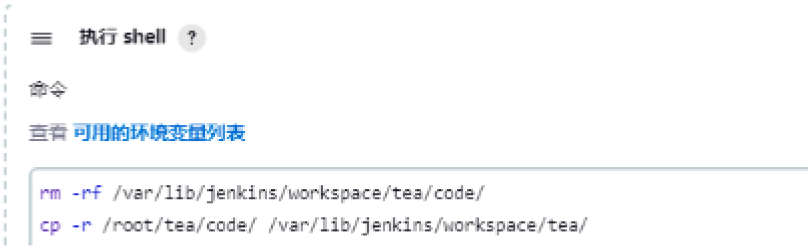
```
[root@test ~]# ls /var/lib/jenkins/workspace/tea/       #可以看到结果
README.txt  code  id.txt  sql  teaimg.tar.gz
```

但是此时下载下来的项目代码，yaml文件的配置和现有的实验环境不符（如，redis地址，nacos命名空间id等），需要解决此问题，才不会影响后续mvn打包封装

替换更改好的code目录到/var/lib/jenkins/workspace/tea/







```
rm -rf /var/lib/jenkins/workspace/tea/code/
cp -r /root/tea/code/ /var/lib/jenkins/workspace/tea/
```

```
rm -rf /var/lib/jenkins/workspace/tea/code/
cp -r /root/tea/code/ /var/lib/jenkins/workspace/tea/
```

复制粘贴如上指令，保存

更改启动jenkins服务的用户，否则执行拷贝时没有权限

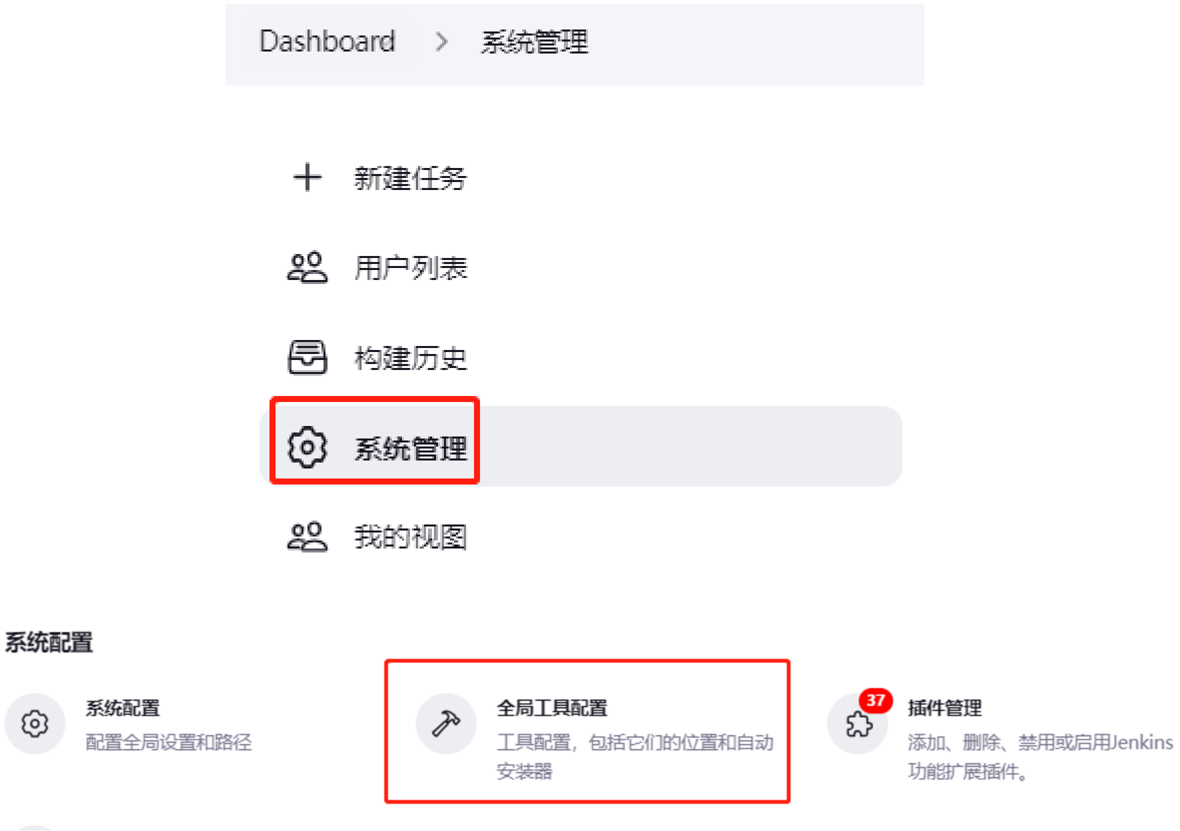
```
[root@test ~]# vim /lib/systemd/system/jenkins.service
 34 User=root
[root@test ~]# chown -R root /var/lib/jenkins/
[root@test ~]# systemctl daemon-reload      #重新加载 systemd 的配置文件
[root@test ~]# systemctl restart jenkins
```

登录jenkins，再次构建即可

6、部署测试

1、后端JAR包部署

配置maven环境，使jenkins调用maven打包



Dashboard > 系统管理

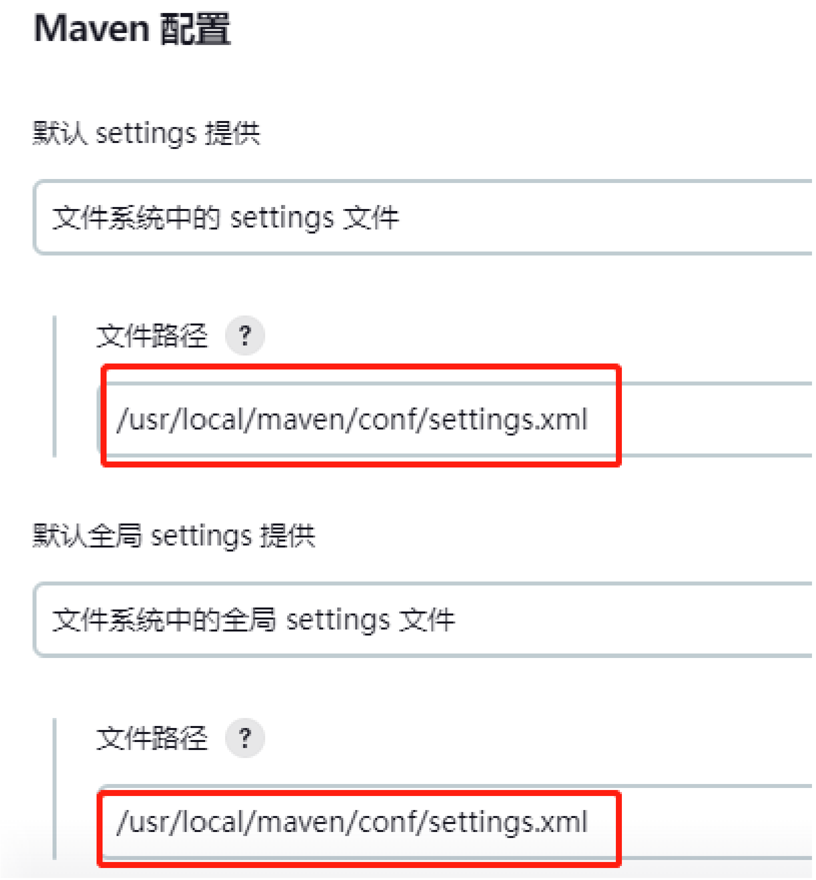
系统配置

系统配置
配置全局设置和路径

全局工具配置
工具配置，包括它们的位置和自动安装器

37
插件管理
添加、删除、禁用或启用Jenkins功能扩展插件。

使用maven的配置文件，定义maven配置文件/usr/local/maven/conf/settings.xml



Maven 配置

默认 settings 提供

文件系统中的 settings 文件

文件路径 ?

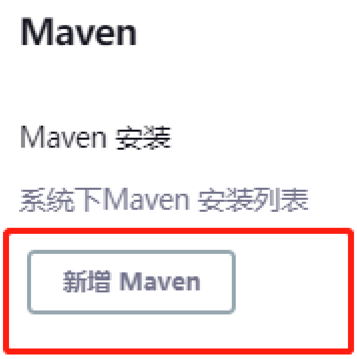
/usr/local/maven/conf/settings.xml

默认全局 settings 提供

文件系统中的全局 settings 文件

文件路径 ?

/usr/local/maven/conf/settings.xml

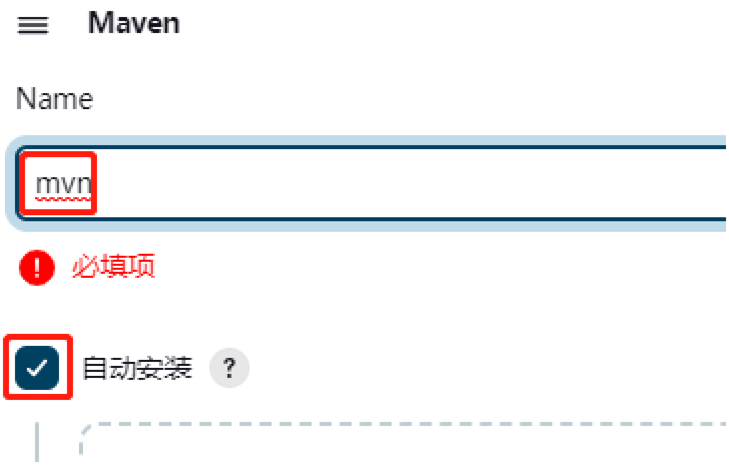


Maven

Maven 安装

系统下Maven 安装列表

设置名字：mvn，取消自动安装



Maven

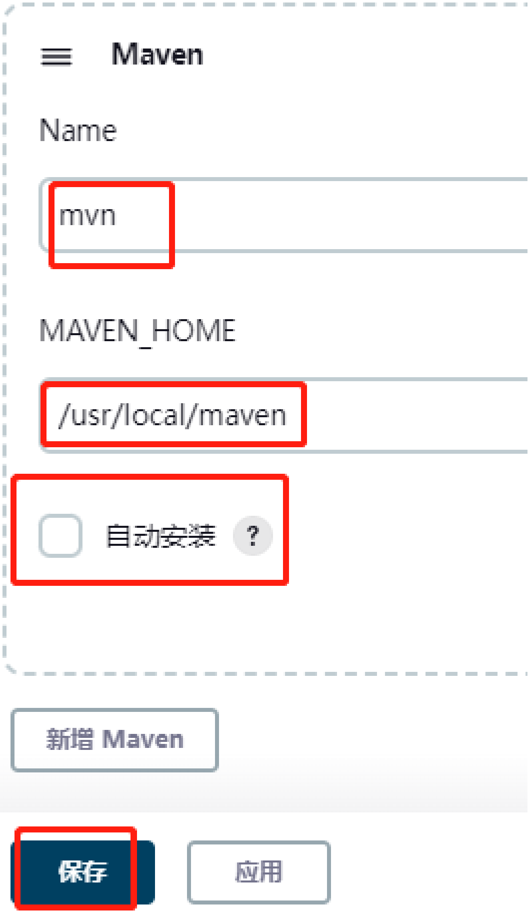
Name

mvn

必填项

☑ 自动安装

maven家目录/usr/local/maven，最终结果显示



Maven

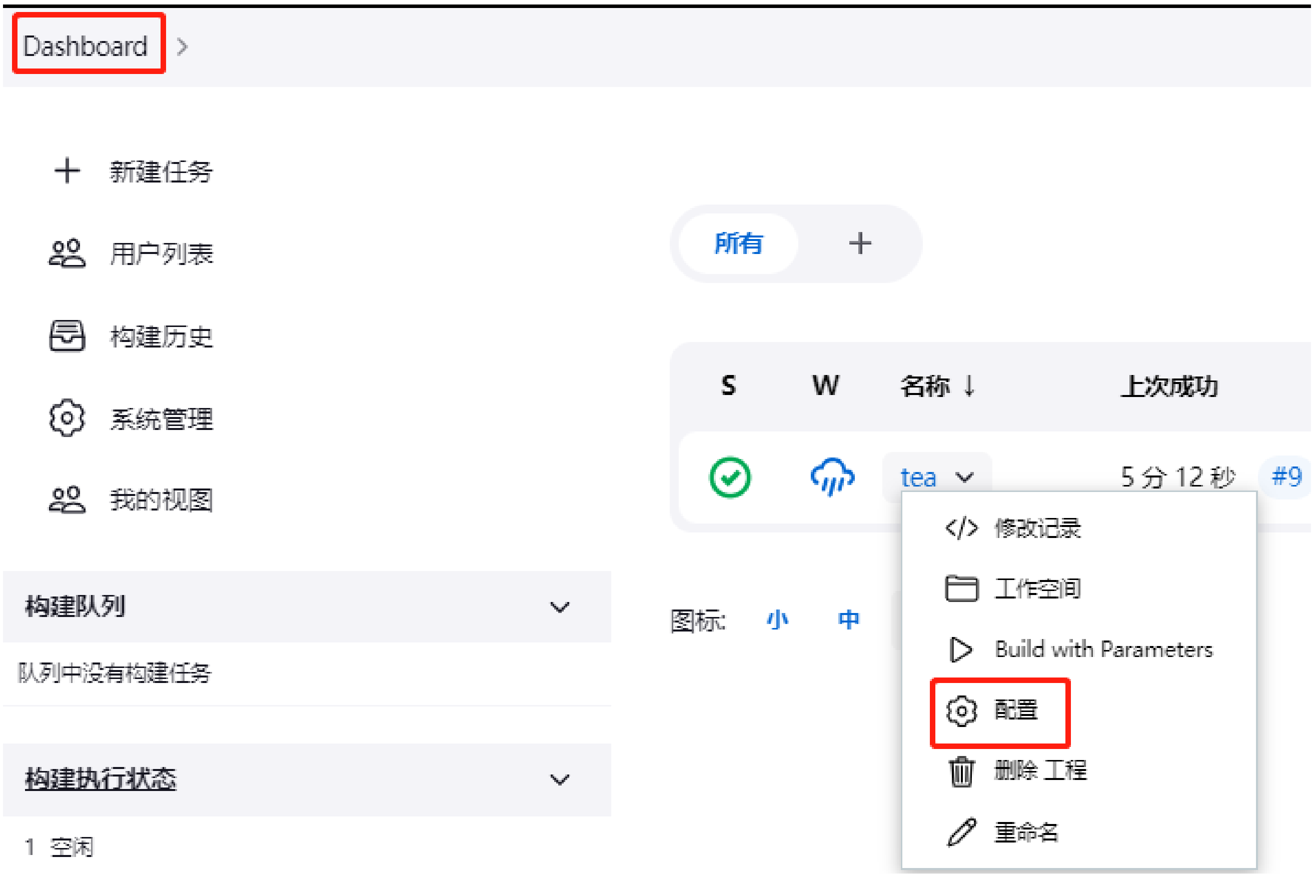
Name

mvn

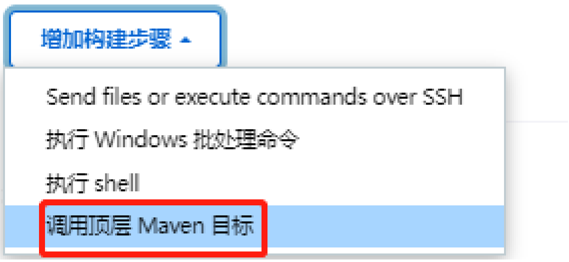
MAVEN_HOME

/usr/local/maven

□ 自动安装

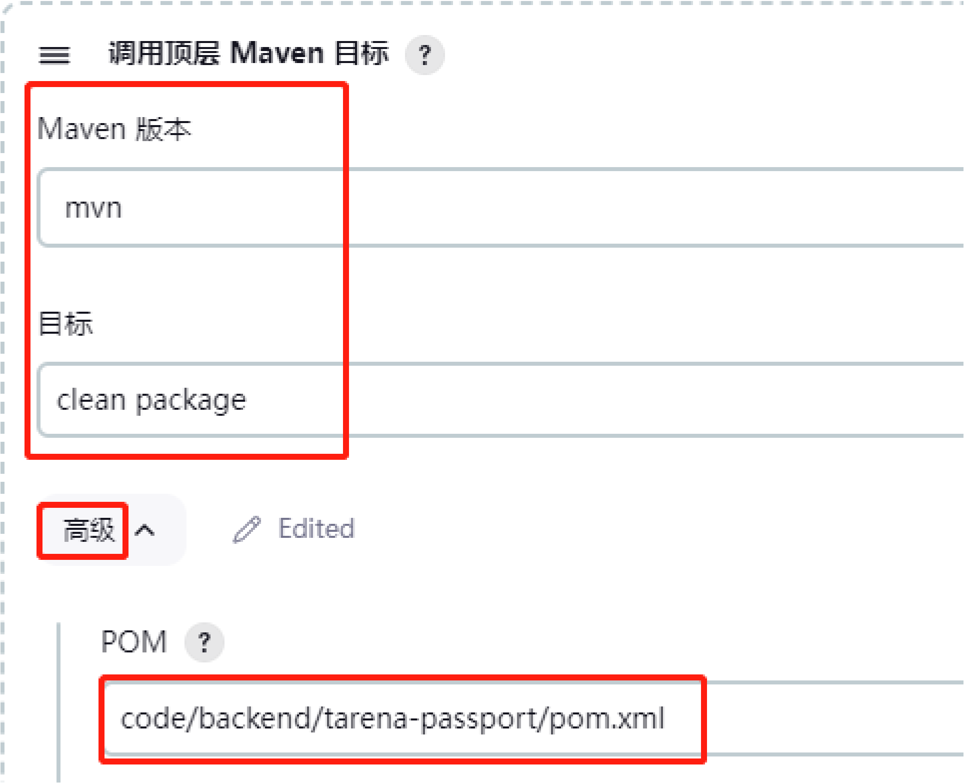


7、配置mvn打包jar包



目标：clean package

POM：code/backend/tarena-passport/pom.xml



调用顶层 Maven 目标

Maven 版本

mvn

目标

clean package

高级

Edited

POM

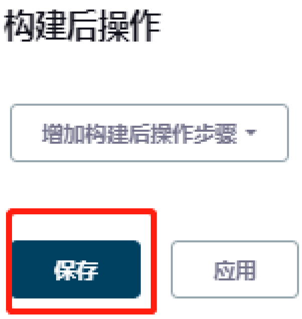
code/backend/tarena-passport/pom.xml

继续增加2个构建步骤，按照以上步骤继续使用maven打包把除了passport之外的jar也要打包；步骤和打包passport一样，只是pom的路径换成以下：

code/backend/tarena-tp-attach/pom.xml

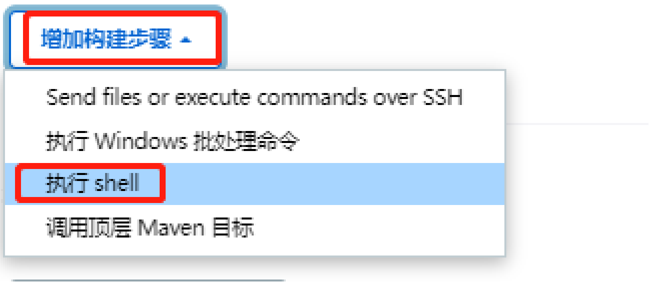
code/backend/tarena-tp-tea/pom.xml

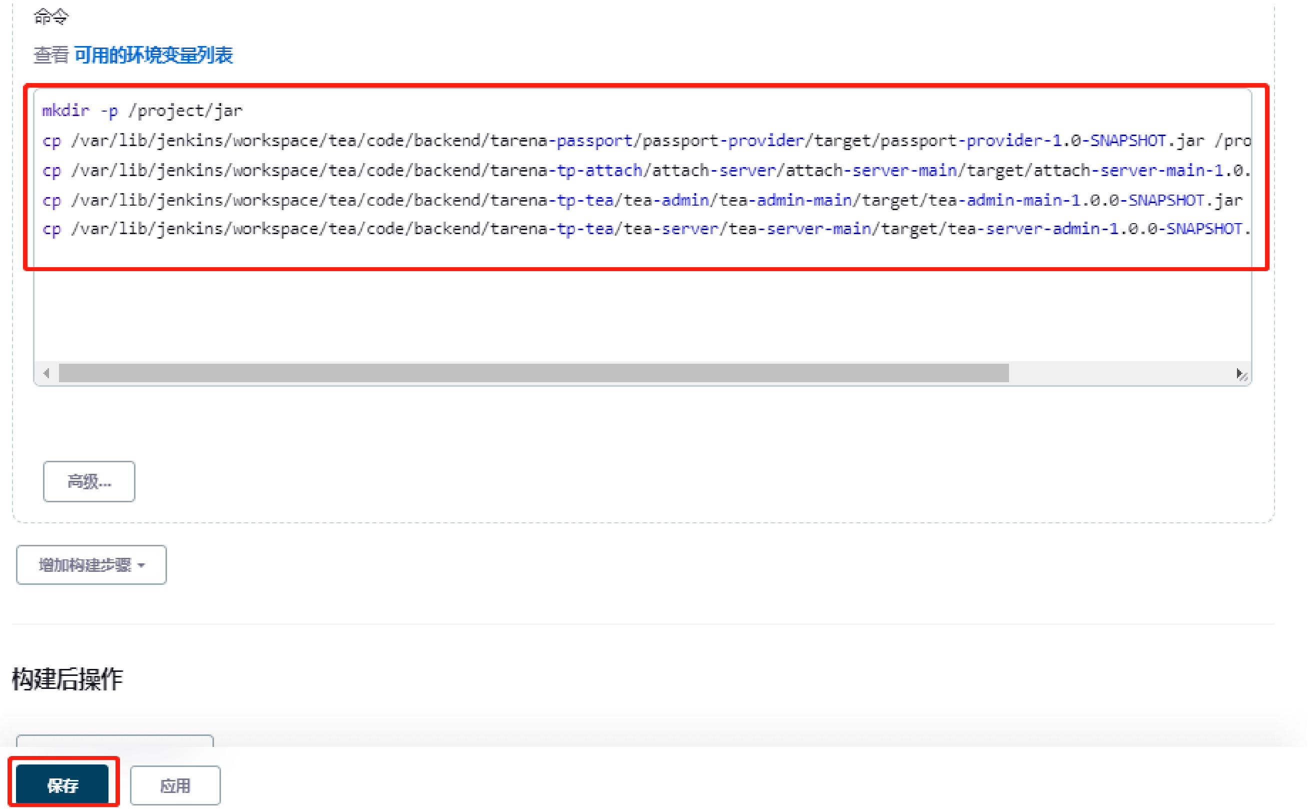
点击保存，测试



此时虽然jenkins已经能够使用maven打包，但是这些jar包，是分散存储到不同的目录下面的，管理比较麻烦，可以把jar包集中放到一个目录下管理，在这里统一放到/project/jar下面管理

最后增加构建shell的步骤，拷贝jar到test主机的/project/jar





```
mkdir -p /project/jar
cp /var/lib/jenkins/workspace/tea/code/backend/tarena-passport/passport-provider/target/passport-provider-1.0-SNAPSHOT.jar /pro
cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-attach/attach-server/attach-server-main/target/attach-server-main-1.0.
cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-tea/tea-admin/tea-admin-main/target/tea-admin-main-1.0.0-SNAPSHOT.jar
cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-tea/tea-server/tea-server-main/target/tea-server-admin-1.0.0-SNAPSHOT.
```

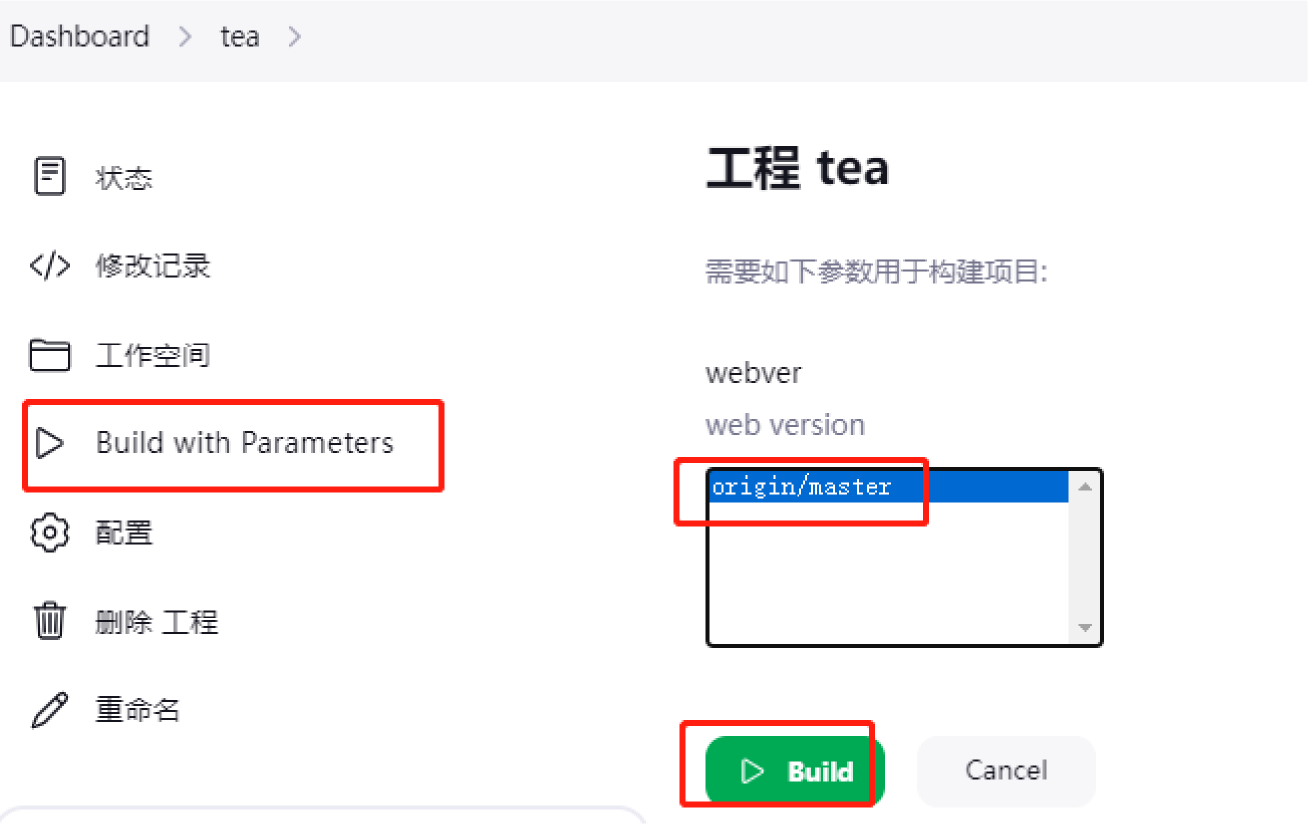
执行的shell命令如下：

```
mkdir -p /project/jar
cp /var/lib/jenkins/workspace/tea/code/backend/tarena-passport/passport-provider/target/passport-provider-1.0-SNAPSHOT.jar /project/jar/

cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-attach/attach-server/attach-server-main/target/attach-server-main-1.0.0-SNAPSHOT.jar /project/jar/

cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-tea/tea-admin/tea-admin-main/target/tea-admin-main-1.0.0-SNAPSHOT.jar  /project/jar/

cp /var/lib/jenkins/workspace/tea/code/backend/tarena-tp-tea/tea-server/tea-server-main/target/tea-server-admin-1.0.0-SNAPSHOT.jar /project/jar/
```



工程 tea

需要如下参数用于构建项目:

webver

web version

origin/master

test主机查看结果，已经有jar包

```
[root@test tea]# ls /project/jar/
attach-server-main-1.0.0-SNAPSHOT.jar  tea-admin-main-1.0.0-SNAPSHOT.jar
passport-provider-1.0-SNAPSHOT.jar     tea-server-admin-1.0.0-SNAPSHOT.jar
```

三、Node部署

P1、前后端页面编译打包

1、node安装

```
Java前端代码打包编译 可以使用Node.js来运行前端构建工具，并使用npm来管理前端依赖。通过配置前端构建工具，
可以将前端代码打包成静态资源文件，并生成合并、压缩、优化后的文件，方便部署到服务器
#Node安装
[root@test tea]# cd /root/
[root@test ~]# tar -xf node-v16.13.0-linux-x64.tar.gz
[root@test ~]# mv node-v16.13.0-linux-x64 /usr/local/node
[root@test ~]# ls /usr/local/node
bin  CHANGELOG.md  include  lib  LICENSE  README.md  share

#配置环境变量
[root@test ~]# vim /etc/bashrc
...
 99 export MAVEN_HOME="/usr/local/maven"
100 export NODE_HOME="/usr/local/node"      #新添加
101 export PATH=${MAVEN_HOME}/bin/:${NODE_HOME}/bin/:$PATH    #更改
[root@test ~]# source /etc/bashrc
```

```
#测试npm命令
[root@test ~]# npm -v
8.1.0
```

2、部署网站页面

1、商品管理页面

```
#测试编译
[root@test ~]# cd /var/lib/jenkins/workspace/tea/code/page/admin-page/
[root@test admin-page]# npm install         #安装此项目的各依赖项，此时，项目才是完整的、可运行的状态
[root@test admin-page]# npm run build:test  #对项目进行打包和编译，根据test的环境进行打包编译
...
  Images and other types of assets omitted.

 DONE  Build complete. The dist directory is ready to be deployed.
 INFO  Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

#创建存放前端页面的目录/project/page
[root@test admin-page]# mkdir -p /project/page

#由于打包编译的页面（无论是商品管理页面还是商品展示页面）都在dist目录下面存放，所以在/project/page再次创建子目录，存放页面，便于区分

#拷贝商品管理页面资源到/project/page/admin-page/
[root@test admin-page]# mkdir /project/page/admin-page
[root@test admin-page]# cp -r dist/ /project/page/admin-page
```

2、商品展示页面

```
#测试编译
[root@test admin-page]# cd ../front-page/
[root@test front-page]# npm install     #安装此项目的各依赖项，此时，项目才是完整的、可运行的状态
...
added 1902 packages in 1m
[root@test front-page]# npm run build:linuxTech
...
  Images and other types of assets omitted.

 DONE  Build complete. The dist directory is ready to be deployed.
 INFO  Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

#拷贝商品展示页面资源到/project/page/front-page/
[root@test front-page]# mkdir /project/page/front-page
[root@test front-page]# cp -r dist/ /project/page/front-page
```

3、共享图片，jar包程序以及前端页面

使用nfs 把目录/project/jar/；/project/page；/home/images/vm/共享出去

```
[root@test ~]# yum -y install nfs-utils
[root@test ~]# vim /etc/exports
/project/jar       *(rw)
/project/page      *(rw)
/home/images/vm    *(rw)
[root@test ~]# systemctl enable --now nfs-server
[root@test ~]# chmod -R 777 /home/images/vm/    #后期需要在此目录中存储新的图片，需要写入的权限
```