```
版本控制系统      Gitlab

持续集成工具      Jenkins

部署工具          Ansible
```

本文通过`Jenkins + Ansible + Gitlab`实现自动化部署。

- 角色划分：

```
Jenkins + Ansible   192.168.11.11

test host           192.168.11.12

gitlab              192.168.11.13
```

Jenkins server 搭建 Ansible 环境。

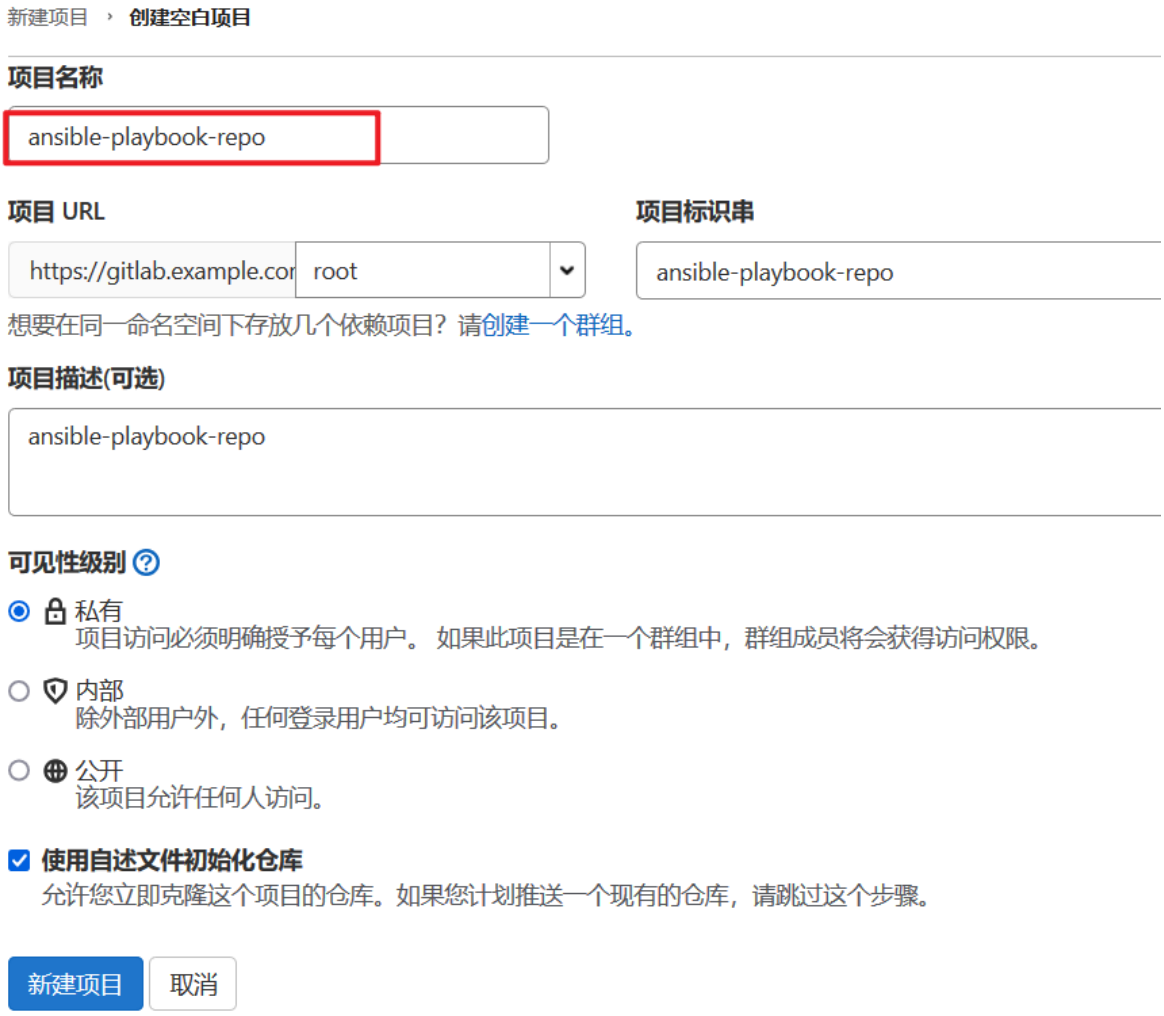
```
curl -o /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo

curl -o /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

yum install -y ansible

ansible --version
```

Freestyle Job自动化部署

使用freestyle job自动化部署静态网页。

- gitlab创建`ansible-playbook-repo`项目：

点击右上方`+` → `New porject`，`Project name`输入ansible-playbook-repo，`Visibility Level`选择默认的`Private`即可，最后点击`Create project`创建项目。



- clone`ansible-playbook-repo`项目：

任选一台其它机器，

```
# yum install -y git

# echo '192.168.11.13 gitlab.example.com' >> /etc/hosts

# mkdir /home/repo && cd /home/repo

# git config --global user.name "admin"

# git config --global user.email "admin@example.com"

# git config --global http.sslVerify false

# git clone https://gitlab.example.com/root/ansible-playbook-repo.git

Cloning into 'ansible-playbook-repo'...
Username for 'https://gitlab.example.com': root
Password for 'https://root@gitlab.example.com':
warning: You appear to have cloned an empty repository.
```

- 编写ansible-playbook：

`files`

```
# cd ansible-playbook-repo/

# mkdir -pv nginx_playbooks/{inventory,roles/nginx/{files,tasks,templates}}

# cd nginx_playbooks
# vim deploy.yml

- hosts: nginx
  remote_user: root
  gather_facts: True
  roles:
    - nginx

# cd inventory

# vim dev

[nginx]
test.example.com

[nginx:vars]
server_name=test.example.com
port=80
user=root
worker_processes=1
max_open_file=65505
root=/www

# cp dev prod

# cd ../roles/nginx

# vim files/health_check.sh

#!/bin/bash
URL=$1
curl -Is http://$URL > /dev/null && echo "The remote side is healthy" ||
echo "The remote side is failed, please check"

# echo "This is my first website" > index.html
```

templates

```
# vim templates/nginx.conf.j2

user {{ user }};
worker_processes  {{ worker_processes }};
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;
worker_rlimit_nofile 65535;

events {
    use epoll;
    worker_connections {{ max_open_file }};
    multi_accept on;
}

```

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request"
'
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    server_tokens       off;
    sendfile            on;
    send_timeout        3m;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    #gzip on;

    server {
        listen {{ port }} default_server;
        server_name {{ server_name }};

        location / {
            root {{ root }};
            index index.html index.htm;
        }

        error_page   404  /404.html;
        location = /404.html {
            root   html;
        }
    }
}
```

tasks

```
# vim tasks/main.yml

- name: Disable system firewalld
  service: name=firewalld state=stopped enabled=no

- name: Disable SELINUX
  selinux: state=disabled

- name: Wget ali source
  get_url: url=http://mirrors.aliyun.com/repo/Centos-7.repo
dest=/etc/yum.repos.d/

- name: Wget nginx yum source
  get_url: url=http://mirrors.aliyun.com/repo/epel-7.repo
dest=/etc/yum.repos.d/

- name: Yum install nginx
  yum: name=nginx state=latest
```

```
17
18 - name: Write then nginx config file
19   template: src=nginx.conf.j2 dest=/etc/nginx/nginx.conf
20
21 - name: Create nginx root folder
22   file: "name={{ root }} state=directory owner={{ user }} group={{ user }}
   mode=0755"
23
24 - name: Copy index.html to remote
25   copy: "remote_src=no src=index.html dest={{ root }}/index.html mode=0755"
26
27 - name: Start nginx service
28   service: name=nginx state=started enabled=yes
29
30 - name: Copy health_check.sh to remote
31   copy: "remote_src=no src=health_check.sh dest=/tmp/health_check.sh
   mode=0755"
32
33 - name: Run the health_check.sh
34   shell: "sh /tmp/health_check.sh {{ server_name }}"
35   register: health_status
36 #注册一个参数，将脚本的输出赋给这个参数名
37 - debug: msg="{{ health_status.stdout }}"
38 #将参数内容输出到ansible的日志当中
```

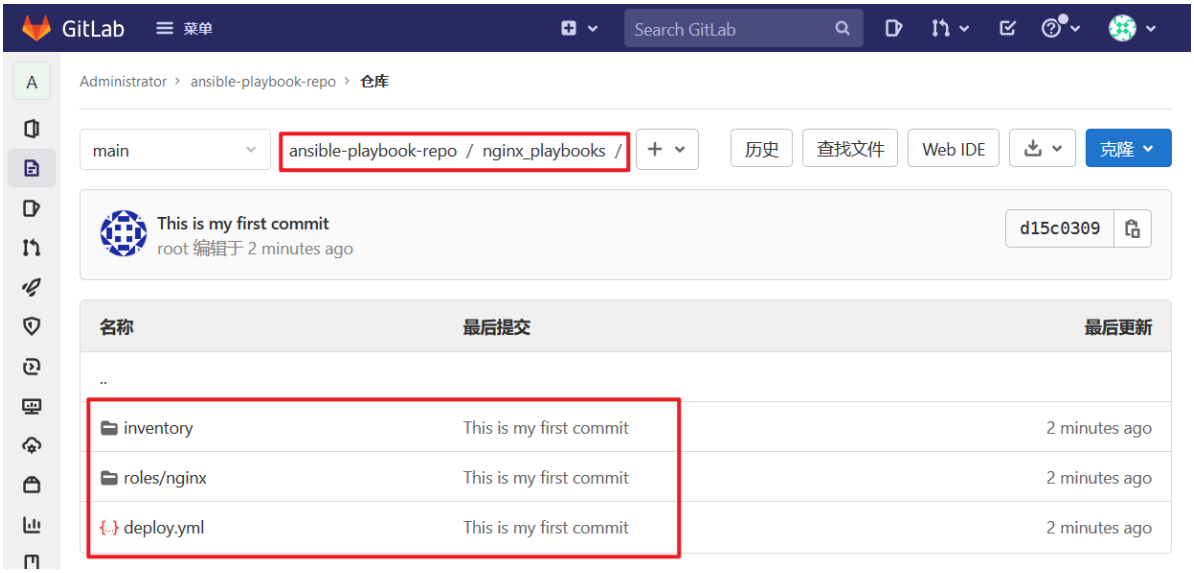
- 将ansible-playbook提交到gitlab：

```
1  # pwd
2  /home/repo/ansible-playbook-repo
3
4
5  # tree nginx_playbooks/
6  nginx_playbooks/
7  ├── deploy.yml
8  ├── inventory
9  |   ├── dev
10 |   └── prod
11 └── roles
12     └── nginx
13         ├── files
14         |   ├── health_check.sh
15         |   └── index.html
16         ├── tasks
17         |   └── main.yml
18         └── templates
19             └── nginx.conf.j2
20
21 6 directories, 7 files
```

```
1  # git add .
2
3  # git commit -m "This is my first commit"
4
5  [master (root-commit) 475fd34] This is my first commit
6   7 files changed, 107 insertions(+)
7   create mode 100644 nginx_playbooks/deploy.yml
```

```
 create mode 100644 nginx_playbooks/inventory/dev
 create mode 100644 nginx_playbooks/inventory/prod
 create mode 100644 nginx_playbooks/roles/nginx/files/health_check.sh
 create mode 100644 nginx_playbooks/roles/nginx/files/index.html
 create mode 100644 nginx_playbooks/roles/nginx/tasks/main.yml
 create mode 100644 nginx_playbooks/roles/nginx/templates/nginx.conf.j2

# git push origin main

Username for 'https://gitlab.example.com': root
Password for 'https://root@gitlab.example.com':
Counting objects: 16, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 2.17 KiB | 0 bytes/s, done.
Total 15 (delta 0), reused 0 (delta 0)
To https://gitlab.example.com/root/ansible-playbook-repo.git
   e421f8c..d15c030  main -> main
```

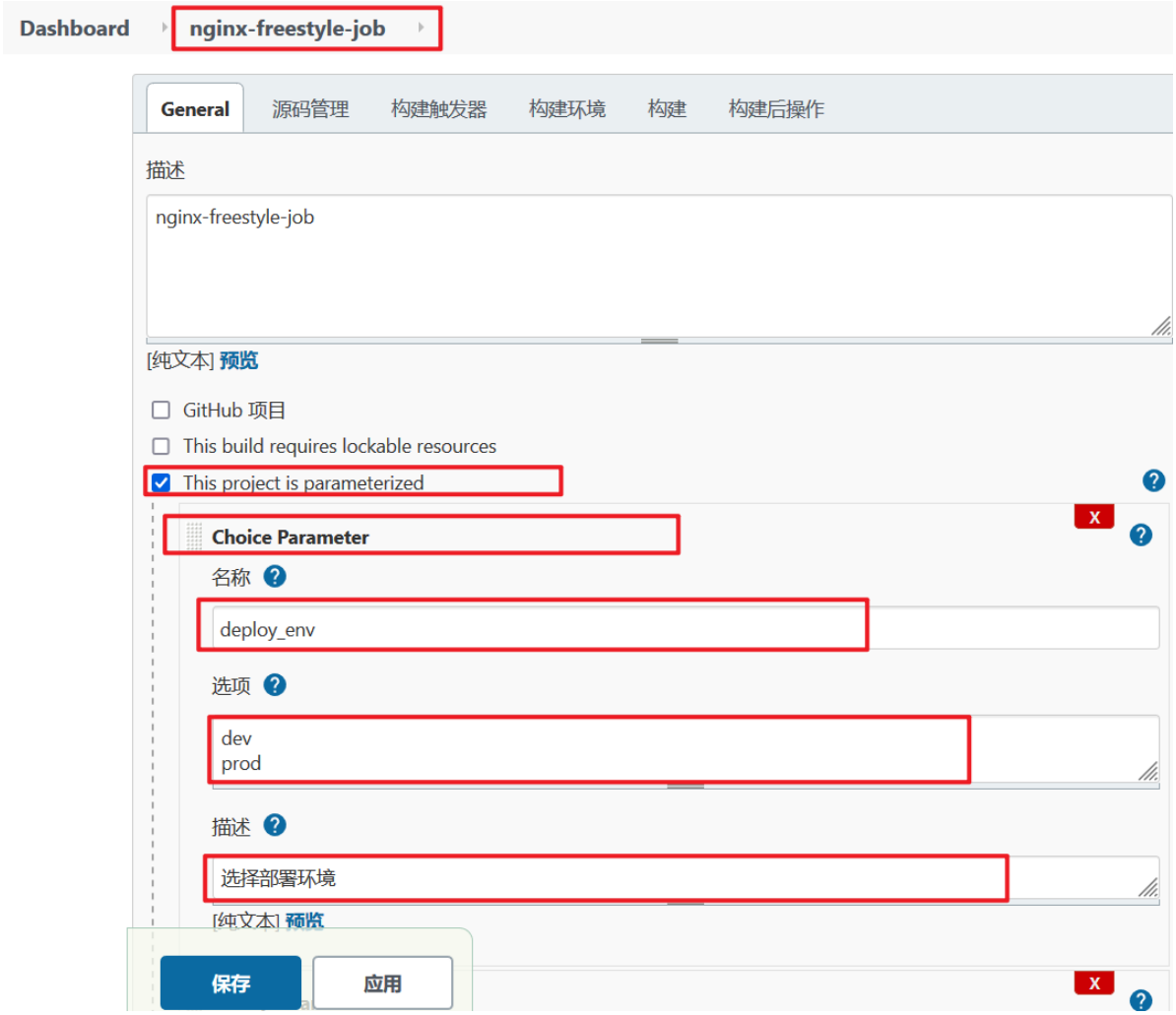
刷新gitlab后台，查看ansible-playbook-repo项目

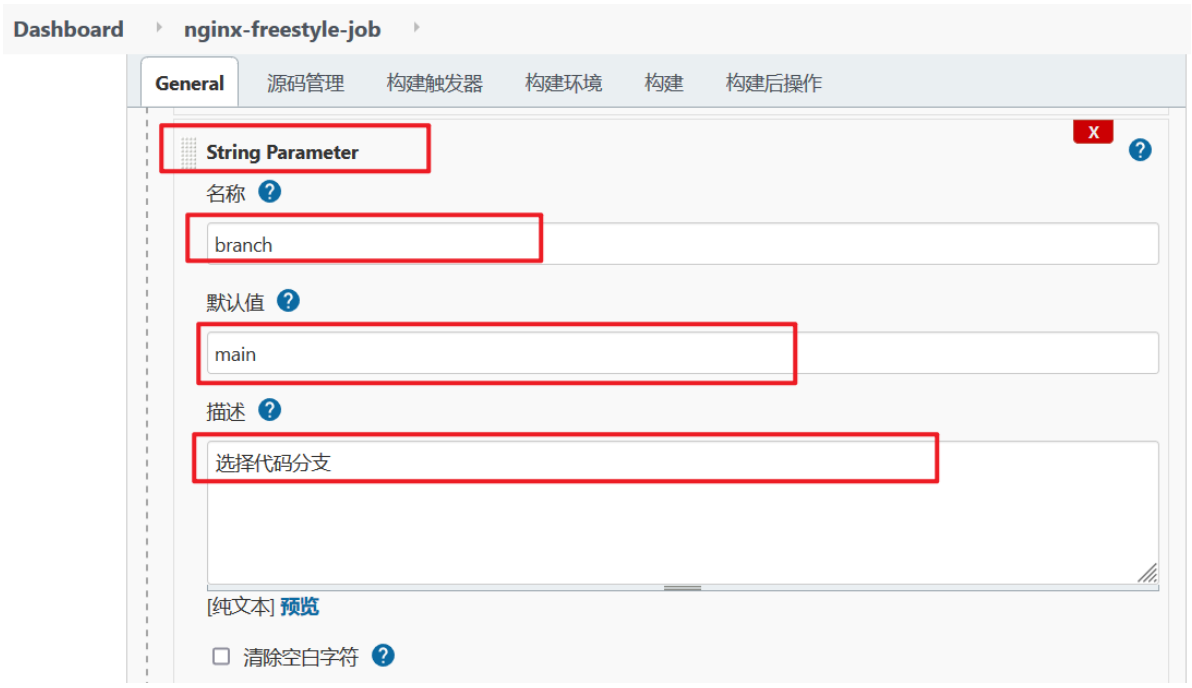


- freestyle job构建与自动化部署：

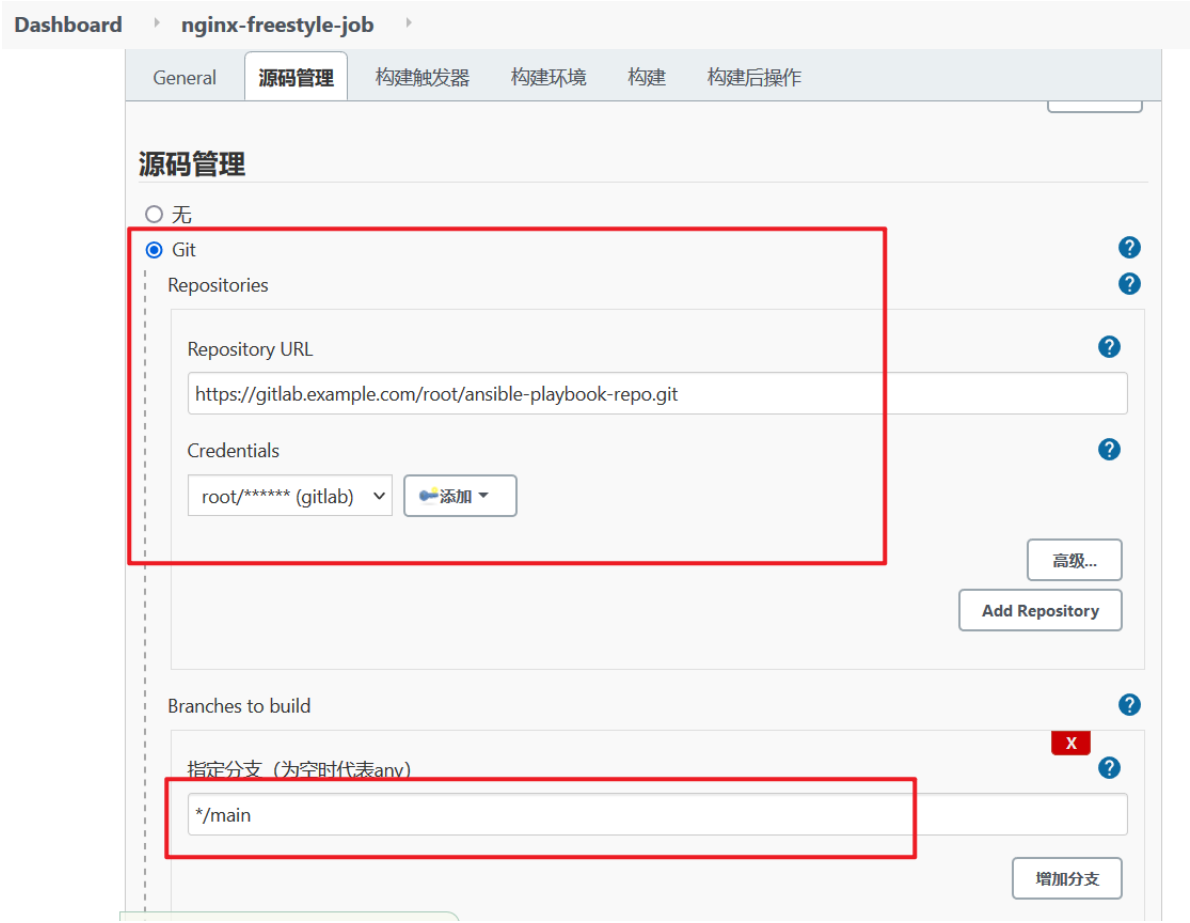
点击新建任务，任务名称输入nginx-freestyle-job，选择自由风格的软件项目。

添加描述信息：This is my first nginx freestyle job，勾选参数化构建过程，添加选项参数和文本参数



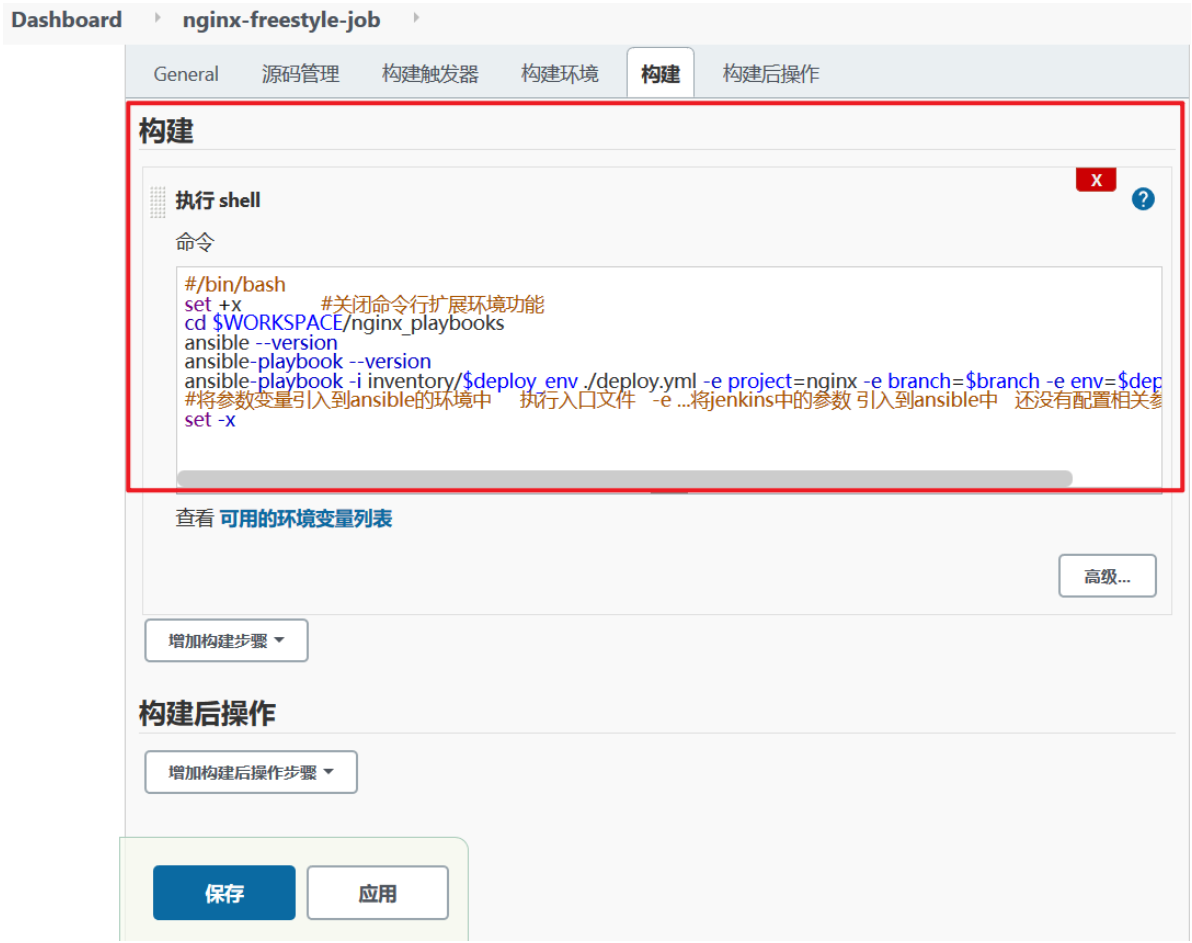


接着`源码管理`这里勾选`Git`，写入`ansible-playbook-repo`这个项目地址，然后选择之前添加的`git`凭据。



接着 `构建` → `增加构建步骤` → `执行 shell`，写入下面内容

```
#/bin/bash
set +x              #关闭命令行扩展环境功能
cd $WORKSPACE/nginx_playbooks
ansible --version
ansible-playbook --version
ansible-playbook -i inventory/$deploy_env ./deploy.yml -e project=nginx -e branch=$branch -e env=$deploy_env
#将参数变量引入到ansible的环境中      执行入口文件    -e ...将jenkins中的参数 引入到ansible中    还没有配置相关参数
set -x
```



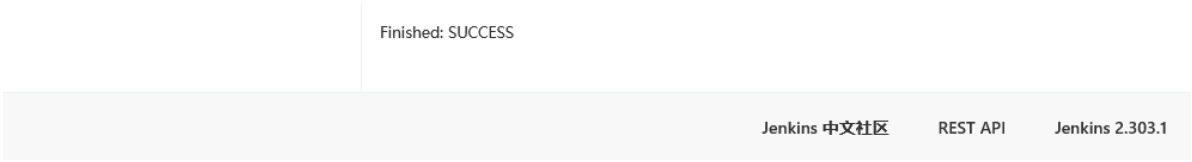
点击`应用`，再点击`保存`。接着点击`Build with Parameters`，选择对应的`选项参数`，选择好后点击`开始构建`。

等待构建完成，查看控制台输出

返回到工程
状态集
变更记录
控制台输出
文本方式查看
编辑编译信息
Parameters
Git Build Data
上一次构建

控制台输出

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/nginx-freestyle-job
The recommended git tool is: NONE
using credential bf29c7f3-2159-44e7-a9ee-b94168cea9c8
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/nginx-freestyle-job/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://gitlab.example.com/root/ansible-playbook-repo.git # timeout=10
Fetching upstream changes from https://gitlab.example.com/root/ansible-playbook-repo.git
 > git --version # timeout=10
 > git --version # 'git version 1.8.3.1'
using GIT_ASKPASS to set credentials gitlab
 > git fetch --tags --progress https://gitlab.example.com/root/ansible-playbook-repo.git +refs/heads
/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision d15c0309855a695e17c8dddd2bcdae8963aff693 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f d15c0309855a695e17c8dddd2bcdae8963aff693 # timeout=10
Commit message: "This is my first commit"
 > git rev-list --no-walk d15c0309855a695e17c8dddd2bcdae8963aff693 # timeout=10
[nginx-freestyle-job] $ /bin/sh -xe /tmp/jenkins2208466978448268258.sh
+ set +x
ansible 2.9.25
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins
/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible
  python version = 2.7.5 (default, Oct 14 2020, 14:45:30) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]
ansible-playbook 2.9.25
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins
/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible-playbook
  python version = 2.7.5 (default, Oct 14 2020, 14:45:30) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

PLAY [nginx] *******************************************************************

TASK [Gathering Facts] *********************************************************
ok: [test.example.com]

TASK [nginx : Disable system firewalld] ****************************************
ok: [test.example.com]

TASK [nginx : Disable SELINUX] *************************************************
ok: [test.example.com]

TASK [nginx : Wget ali source] *************************************************
changed: [test.example.com]

TASK [Wget nginx yum source] ***************************************************
changed: [test.example.com]

TASK [Yum install nginx] *******************************************************
changed: [test.example.com]

TASK [Write then nginx config file] ********************************************
changed: [test.example.com]

TASK [Create nginx root folder] ************************************************
changed: [test.example.com]

TASK [nginx : Copy index.html to remote] ***************************************
changed: [test.example.com]

TASK [Start nginx service] *****************************************************
changed: [test.example.com]

TASK [nginx : Copy health_check.sh to remote] ***********************************
changed: [test.example.com]

TASK [nginx : Run the health_check.sh] *****************************************
changed: [test.example.com]

TASK [nginx : debug] ***********************************************************
ok: [test.example.com] => {
    "msg": "The remote side is healthy"
}

PLAY RECAP *********************************************************************
test.example.com           : ok=13   changed=9    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```



打开浏览器，访问

使用freestyle job自动化部署静态网页的过程到此成功结束。

Pipeline Job自动化部署

使用pipeline job自动化部署wordpress。

在上面克隆过 `ansible-playbook-repo` 项目的机器上操作。

- 编写ansible-playbook：

files

```
# cd /home/repo/ansible-playbook-repo/

# mkdir -pv wordpress_playbooks/{inventory,roles/wordpress/{files,tasks,templates}}
# cd wordpress_playbooks

# vim deploy.yml

- hosts: wordpress
  remote_user: root
  gather_facts: True
  vars:
    backup_to: "{{root}}_{{branch}}_{{ansible_date_time.epoch}}"

  roles:
    - wordpress

# vim inventory/dev

[wordpress]
test.example.com
```

```

[wordpress:vars]
server_name=test.example.com
port=8080
user=root
worker_processes=1
max_open_file=65505
root=/data/www
gitlab_user='root'              #gitlab账号及密码
gitlab_pass='admin.123'

# vim inventory/prod

[wordpress]
test.example.com

[wordpress:vars]
server_name=test.example.com
port=80
user=root
worker_processes=1
max_open_file=65505
root=/data/www
gitlab_user='root'
gitlab_pass='admin.123'

# vim roles/wordpress/files/health_check.sh

#!/bin/bash
URL=$1
PORT=$2
curl -Is http://$URL:$PORT/info.php > /dev/null && echo "The remote side is healthy" || echo "The remote side is failed, please check"

# vim roles/wordpress/files/www.conf
```

```
[www]
user = apache
group = apache
listen = /var/run/php-fpm/php-fpm.sock
listen.owner = apache
listen.group = apache
listen.allowed_clients = 127.0.0.1
pm = dynamic
pm.max_children = 50
pm.start_servers = 5
pm.min_spare_servers = 5
pm.max_spare_servers = 35
slowlog = /var/log/php-fpm/www-slow.log
php_admin_value[error_log] = /var/log/php-fpm/www-error.log
php_admin_flag[log_errors] = on
php_value[session.save_handler] = files
php_value[session.save_path]    = /var/lib/php/session
php_value[soap.wsdl_cache_dir]  = /var/lib/php/wsdlcache
```

templates

```
# vim roles/wordpress/templates/nginx.conf.j2

user {{ user }};
worker_processes  {{ worker_processes }};
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;
worker_rlimit_nofile 65535;

events {
    use epoll;
    worker_connections {{ max_open_file }};
    multi_accept on;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request"
'
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    server_tokens       off;
    sendfile            on;
    send_timeout        3m;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    #gzip on;

    server {
        listen {{ port }} default_server;
        server_name {{ server_name }};
        root {{ root }};

        location / {
            index  index.html index.htm index.php;
        }

        location ~ \.php$ {
            try_files $uri =404;
            fastcgi_pass unix:/var/run/php-fpm/php-fpm.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
            include fastcgi_params;
        }
    }
}
```

tasks

```
# vim roles/wordpress/tasks/main.yml

- name: Disable system firewalld
  service: name=firewalld state=stopped enabled=no

- name: Disable SELINUX
  selinux: state=disabled

- name: Wget ali source
  get_url: url=http://mirrors.aliyun.com/repo/Centos-7.repo
dest=/etc/yum.repos.d/

- name: Wget nginx yum source
  get_url: url=http://mirrors.aliyun.com/repo/epel-7.repo
dest=/etc/yum.repos.d/

- name: Setup webtatic yum source for php-fpm
  yum: name=https://mirror.webtatic.com/yum/el7/webtatic-release.rpm

- name: Yum install nginx
  yum: name=nginx state=latest

- name: Write then nginx config file
  template: src=nginx.conf.j2 dest=/etc/nginx/nginx.conf

- name: Create nginx root folder
  file: "name={{ root }} state=directory owner={{ user }} group={{ user }}
mode=0755 recurse=yes"

- name: Start nginx service
  service: name=nginx state=started enabled=yes

- name: Setup php-fpm
  command: "yum install -y php72w php72w-fpm php72w-common php72w-mysql
php72w-gd php72w-xml php72w-mbstring php72w-mcrypt warn=False"

#- name: Start php-fpm service
#  service: name=php-fpm state=started enabled=yes

- name: Copy www.conf to remote
  copy: "remote_src=no src=www.conf dest=/etc/php-fpm.d/www.conf mode=0755
owner={{ user }} group={{ user }} force=yes"

- name: Restart php-fpm service
  service: name=php-fpm state=restarted

- name: Copy health_check.sh to remote
  copy: "remote_src=no src=health_check.sh dest=/tmp/health_check.sh
mode=0755 force=yes"

- name: Run the health_check.sh
  shell: "sh /tmp/health_check.sh {{ server_name }} {{ port }}"
  register: health_status

- debug: msg="{{ health_status.stdout }}"

- name: Setup mariadb(mysql)
  command: "yum install -y mariadb mariadb-server warn=False"
```

```
53
54 - name: Backup current www folder
55   shell: "mv {{ root }} {{ backup_to }}"
56
57 - name: Close git ssl verifycation
58   shell: "git config --global http.sslVerify false"
59
60 - name: Clone Wordpress repo to remote
61   git: "repo=https://{{ gitlab_user | urlencode }}:{{ gitlab_pass |
   urlencode }}@gitlab.example.com/root/wordpress-project.git dest={{ root }}
   version={{ branch }}"
62   when: project == 'wordpress'
63
64 - name: Change www folder permission
65   file: "name={{ root }} state=directory owner={{ user }} group={{ user }}
   mode=0755 recurse=yes"
66
67 - name: Restart nginx service
68   service: name=nginx state=restarted
```

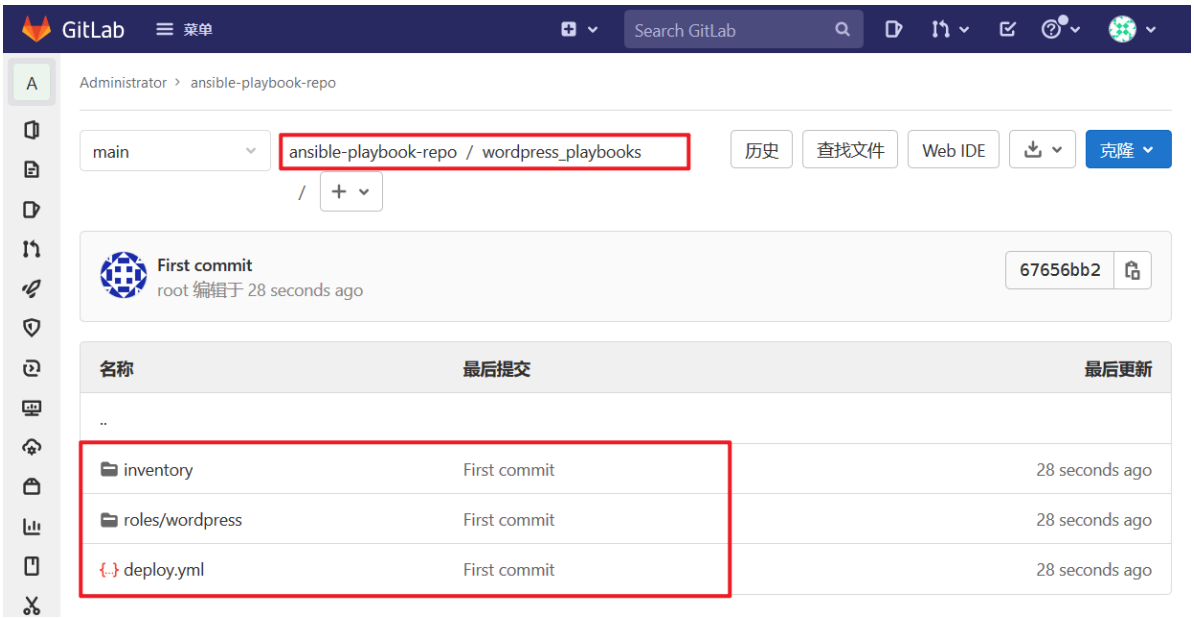
- 将ansible-playbook提交到gitlab：

```
1  # pwd
2  /home/repo/ansible-playbook-repo
3
4
5  # tree wordpress_playbooks/
6  wordpress_playbooks/
7  ├── deploy.yml
8  ├── inventory
9  |   ├── dev
10 |   └── prod
11 └── roles
12     └── wordpress
13         ├── files
14         |   ├── health_check.sh
15         |   ├── info.php
16         |   └── www.conf
17         ├── tasks
18         |   └── main.yml
19         └── templates
20             └── nginx.conf.j2
21
22 6 directories, 8 files
```

```
1  # git add .
2
3  # git commit -m "First commit"
4
5  [master dfeb3df] First commit
6   8 files changed, 550 insertions(+)
7   create mode 100644 wordpress_playbooks/deploy.yml
8   create mode 100644 wordpress_playbooks/inventory/dev
9   create mode 100644 wordpress_playbooks/inventory/prod
10  create mode 100644
    wordpress_playbooks/roles/wordpress/files/health_check.sh
```

```
 create mode 100644 wordpress_playbooks/roles/wordpress/files/info.php
 create mode 100644 wordpress_playbooks/roles/wordpress/files/www.conf
 create mode 100644 wordpress_playbooks/roles/wordpress/tasks/main.yml
 create mode 100644
wordpress_playbooks/roles/wordpress/templates/nginx.conf.j2

# git push origin main

Username for 'https://gitlab.example.com': root
Password for 'https://root@gitlab.example.com':
Counting objects: 18, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (17/17), 8.27 KiB | 0 bytes/s, done.
Total 17 (delta 1), reused 0 (delta 0)
To https://gitlab.example.com/root/ansible-playbook-repo.git
   fc8b669..dfeb3df  master -> master
```

刷新gitlab后台，查看ansible-playbook-repo项目



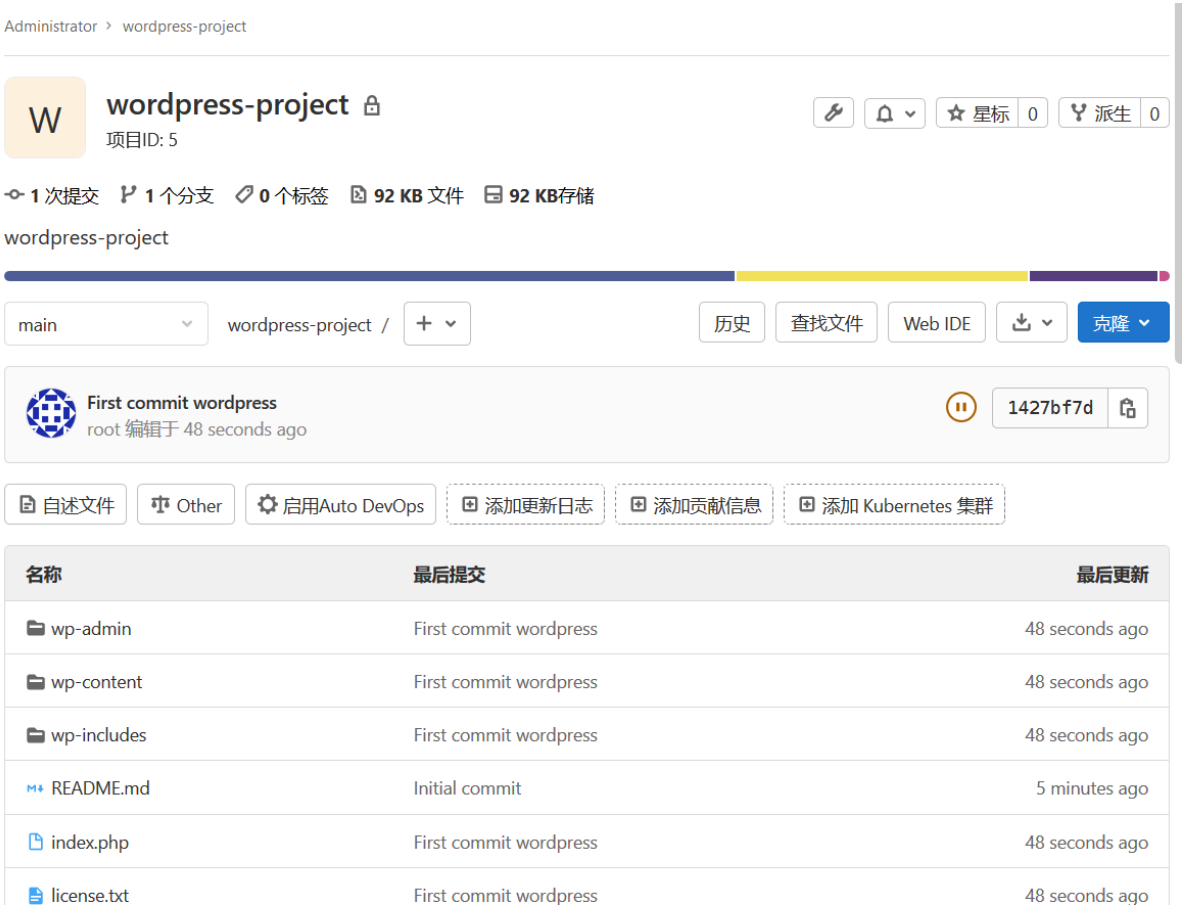
- gitlab创建`wordpress-project`项目：

点击右上方`+` → `New porject`，`Project name`输入wordpress-project，`Visibility Level`选择默认的`Private`即可，最后点击`Create project`创建项目。

- 下载workpress源码，提交到gitlab：

```
# cd /home/repo/

# git clone https://gitlab.example.com/root/Wordpress-project.git

# tar xf wordpress-5.0.3-zh_CN.tar.gz

# mv wordpress/* wordpress-project/

# echo "<?php phpinfo(); ?>" > wordpress-project/info.php

# git add .

# git commit -m "First commit wordpress"

# git push origin main
Username for 'https://gitlab.example.com': root
Password for 'https://root@gitlab.example.com':
Counting objects: 1899, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (1867/1867), done.
Writing objects: 100% (1898/1898), 10.87 MiB | 7.86 MiB/s, done.
Total 1898 (delta 176), reused 0 (delta 0)
remote: Resolving deltas: 100% (176/176), done.
To https://gitlab.example.com/root/wordpress-project.git
   15556a7..1427bf7  main -> main
```

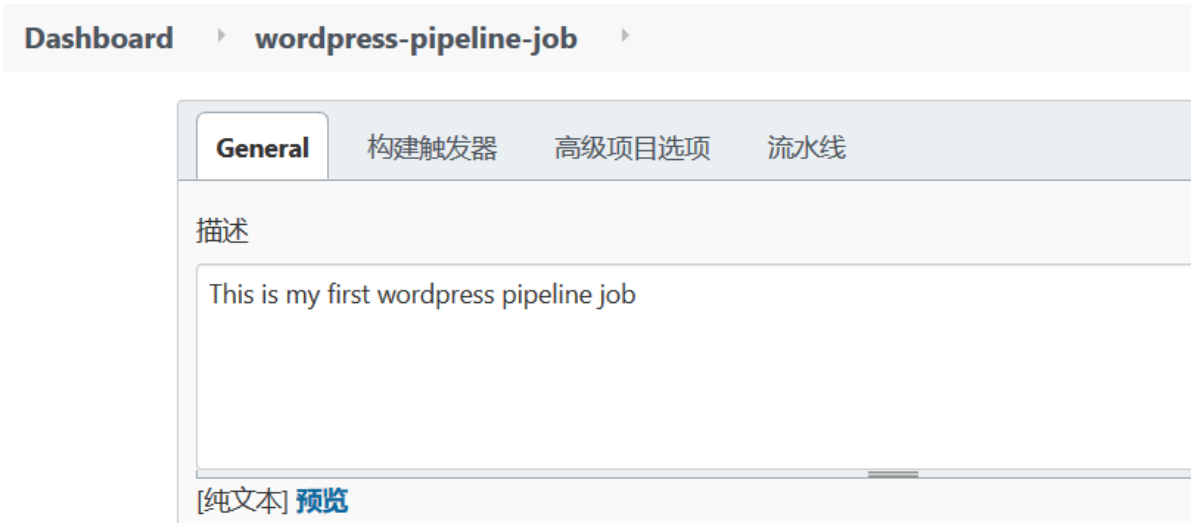
刷新gitlab后台，查看wordpress-project项目



- pipeline job构建与自动化部署：

点击新建任务，任务名称输入wordpress-pipeline-job，选择流水线。

添加描述信息：This is my first wordpress pipeline job，



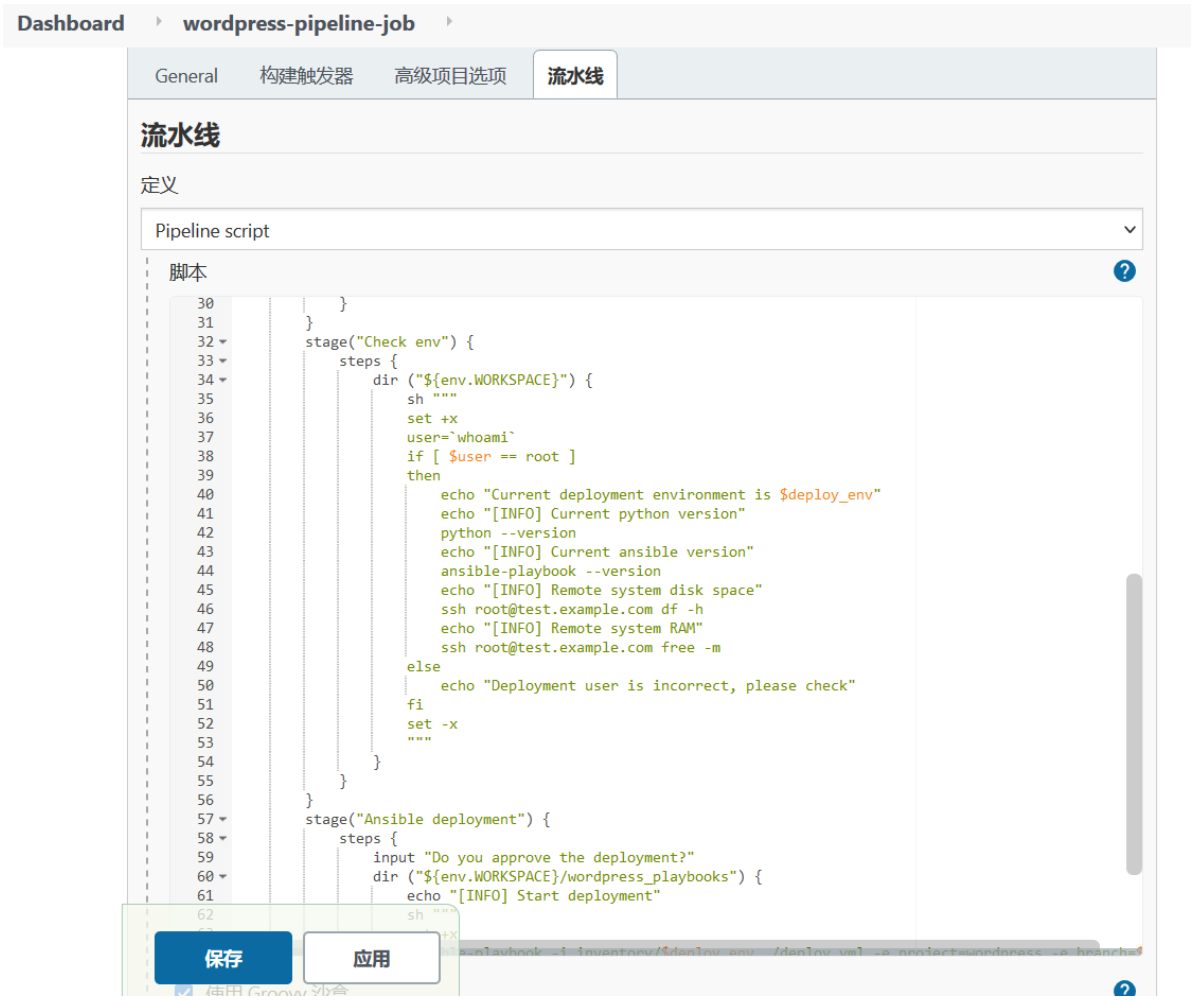
接着定义Pipeline script，写入下面内容

```
#!groovy

pipeline {
    agent {node {label 'master'}}

    environment {
        PATH="/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin"
    }

    parameters {
```

```
        choice (
            choices: 'dev\nprod',
            description: 'choose deploy environment',
            name: 'deploy_env'
        )
        string (
            name: 'branch',
            defaultValue: 'main',
            description: 'build branch'
        )
    }

    stages {
        stage("Pull deploy code") {
            steps {
                sh 'git config --global http.sslVertify false'
                dir ("${env.WORKSPACE}") {
                    git branch: 'main', credentialsId: "00ee2c7c-c475-440c-a88f-dd6c9a49f669", url: 'https://gitlab.example.com/root/ansible-playbook-repo.git'
                }
            }
        }
        stage("Check env") {
            steps {
                dir ("${env.WORKSPACE}") {
                    sh """
                    set +x
                    user=`whoami`
                    if [ $user == root ]
                    then
                        echo "Current deployment environment is $deploy_env"
                        echo "[INFO] Current python version"
                        python --version
                        echo "[INFO] Current ansible version"
                        ansible-playbook --version
                        echo "[INFO] Remote system disk space"
                        ssh root@test.example.com df -h
                        echo "[INFO] Remote system RAM"
                        ssh root@test.example.com free -m
                    else
                        echo "Deployment user is incorrect, please check"
                    fi
                    set -x
                    """
                }
            }
        }
        stage("Ansible deployment") {
            steps {
                input "Do you approve the deployment?"
                dir ("${env.WORKSPACE}/wordpress_playbooks") {
                    echo "[INFO] Start deployment"
                    sh """
                    set +x
                    ansible-playbook -i inventory/$deploy_env ./deploy.yml -e project=wordpress -e branch=$branch -e env=$deploy_env
                    set -x
```

```
                    """
                    echo "[INFO] Deployment finished..."
                }
            }
        }
    }
}
```

上面的credentialsId即`00ee2c7c-c475-440c-a88f-dd6c9a49f669`，是复制凭据页面创建该凭据（gitlab管理员全局凭据）时自动生成的凭据id。



点击`应用`，再点击`保存`。接着点击`立即构建`，等待构建完成，会发现构建失败，原因是第一次构建时`选项参数`没有引入到当前的pipeline job中。

点击`wordpress-pipeline-job`回到该项目中，会发现`立即构建`变成了`Build with Parameters`，选择对应的`选项参数`，选择好后点击`开始构建`。

等待构建完成，查看控制台输出

返回到项目
状态
变更历史
Console Output
View as plain text
编辑构建信息
Parameters
Git Build Data
需要输入而暂停
Thread Dump
暂停（恢复）
回放
流水线步骤
Workspaces
上一次构建

Console Output

```
Started by user admin
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/wordpress-pipeline-job
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Pull deploy code)
[Pipeline] sh
+ git config --global http.sslVertify false
[Pipeline] dir
Running in /var/lib/jenkins/workspace/wordpress-pipeline-job
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
using credential bf29c7f3-2159-44e7-a9ee-b94168cea9c8
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/wordpress-pipeline-job/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://gitlab.example.com/root/ansible-playbook-repo.git # timeout=10
Fetching upstream changes from https://gitlab.example.com/root/ansible-playbook-repo.git
 > git --version # timeout=10
 > git --version # 'git version 1.8.3.1'
using GIT_ASKPASS to set credentials gitlab
 > git fetch --tags --progress https://gitlab.example.com/root/ansible-playbook-repo.git +refs/heads
/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision a1dcd129107c1716593bf8cd9ef8ac0c2808c181 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f a1dcd129107c1716593bf8cd9ef8ac0c2808c181 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main a1dcd129107c1716593bf8cd9ef8ac0c2808c181 # timeout=10
Commit message: "change file"
 > git rev-list --no-walk a1dcd129107c1716593bf8cd9ef8ac0c2808c181 # timeout=10
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Check env)
[Pipeline] dir
Running in /var/lib/jenkins/workspace/wordpress-pipeline-job
[Pipeline] {
[Pipeline] sh
+ set +x
Current deployment environment is dev
[INFO] Current python version
Python 2.7.5
[INFO] Current ansible version
ansible-playbook 2.9.25
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins
/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible-playbook
  python version = 2.7.5 (default, Oct 14 2020, 14:45:30) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]
[INFO] Remote system disk space
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                475M     0  475M   0% /dev
tmpfs                   487M     0  487M   0% /dev/shm
tmpfs                   487M  7.7M  479M   2% /run
tmpfs                   487M     0  487M   0% /sys/fs/cgroup
/dev/mapper/centos-root   27G  2.0G   26G   8% /
/dev/sr0                4.4G  4.4G     0 100% /media/cdrom
/dev/sda1               1014M  138M  877M  14% /boot
tmpfs                   98M     0   98M   0% /run/user/0
[INFO] Remote system RAM
              total        used        free      shared  buff/cache   available
Mem:            972         224         267           7         479         588
Swap:          2047           0        2047
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Ansible deployment)
[Pipeline] input
Do you approve the deployment?
Proceed or Abort
Approved by admin
[Pipeline] dir
```

```
Running in /var/lib/jenkins/workspace/wordpress-pipeline-job/wordpress_playbooks
[Pipeline] {
[Pipeline] echo
[INFO] Start deployment
[Pipeline] sh
+ set +x

PLAY [wordpress] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [test.example.com]

TASK [wordpress : Disable system firewalld] ************************************
ok: [test.example.com]

TASK [wordpress : Disable SELINUX] *********************************************
ok: [test.example.com]

TASK [wordpress : Wget ali source] *********************************************
ok: [test.example.com]

TASK [wordpress : Wget nginx yum source] ***************************************
ok: [test.example.com]

TASK [wordpress : Setup webtatic yum source for php-fpm] ***********************
ok: [test.example.com]

TASK [wordpress : Yum install nginx] *******************************************
ok: [test.example.com]

TASK [wordpress : Write then nginx config file] ********************************
ok: [test.example.com]

TASK [wordpress : Create nginx root folder] ************************************
ok: [test.example.com]

TASK [wordpress : Copy info.php to remote] *************************************
changed: [test.example.com]

TASK [wordpress : Start nginx service] *****************************************
ok: [test.example.com]

TASK [wordpress : Setup php-fpm] ***********************************************
changed: [test.example.com]

TASK [wordpress : Copy www.conf to remote] *************************************
ok: [test.example.com]

TASK [wordpress : Restart php-fpm service] *************************************
changed: [test.example.com]

TASK [wordpress : Copy health_check.sh to remote] ******************************
ok: [test.example.com]

TASK [wordpress : Run the health_check.sh] *************************************
changed: [test.example.com]

TASK [wordpress : debug] *******************************************************
ok: [test.example.com] => {
    "msg": "The remote side is healthy"
}

TASK [wordpress : Setup mariadb(mysql)] ****************************************
changed: [test.example.com]

TASK [wordpress : Backup current www folder] ***********************************
changed: [test.example.com]

TASK [wordpress : Close git ssl verifycation] **********************************
changed: [test.example.com]

TASK [wordpress : Clone Wordpress repo to remote] ******************************
changed: [test.example.com]

TASK [wordpress : Change www folder permission] ********************************
changed: [test.example.com]

TASK [wordpress : Restart nginx service] ***************************************
changed: [test.example.com]

PLAY RECAP *********************************************************************
test.example.com           : ok=23   changed=10   unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

[Pipeline] echo
[INFO] Deployment finished...
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
```

```
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Jenkins 中文社区 REST API Jenkins 2.303.1

- 创建数据库wordpress：　在 test.example.com

```
# systemctl enable --now mariadb.service

# mysql_secure_installation

Enter current password for root (enter for none):                #直接回车
Set root password? [Y/n] y                #设置root密码，我这里是123456
New password:
Re-enter new password:
Remove anonymous users? [Y/n] y              #删除匿名用户
Disallow root login remotely? [Y/n] y                #不允许root用户远程登录
Remove test database and access to it? [Y/n] y                #删除test database及其访问
Reload privilege tables now? [Y/n] y                #重载所有权限
```

```
# mysql -uroot -p123456

MariaDB [(none)]> create database wordpress character set utf8;
```

打开浏览器，访问`test.example.com:8080`（如果`deploy_env`选择的是`prod`，则访问`test.example.com`）。

点击`现在就开始`，填入数据库root用户及其密码，点击`提交`。

接着填入站点标题、管理员用户及其密码等信息，点击`安装WordPress`。

接着使用上面定义的管理员用户及其密码登录wordpress后台管理界面。

使用pipeline job自动化部署wordpress的过程到此成功结束。

Gitlab 触发 jenkins 自动构建

要实现在向 GitLab push 代码时，自动 trigger Jenkins 工程执行构建动作，需要在 GitLab 和 Jenkins 的多个地方做配置：(1)、在 Jenkins 中安装插件；(2)、配置 Jenkins 工程；(3)、配置 GitLab 工程添加 webhook。

jenkins 安装插件

可用插件 -- 搜索： Gitlab Hook 和 Gitlab 安装 -- 重启 jenkins

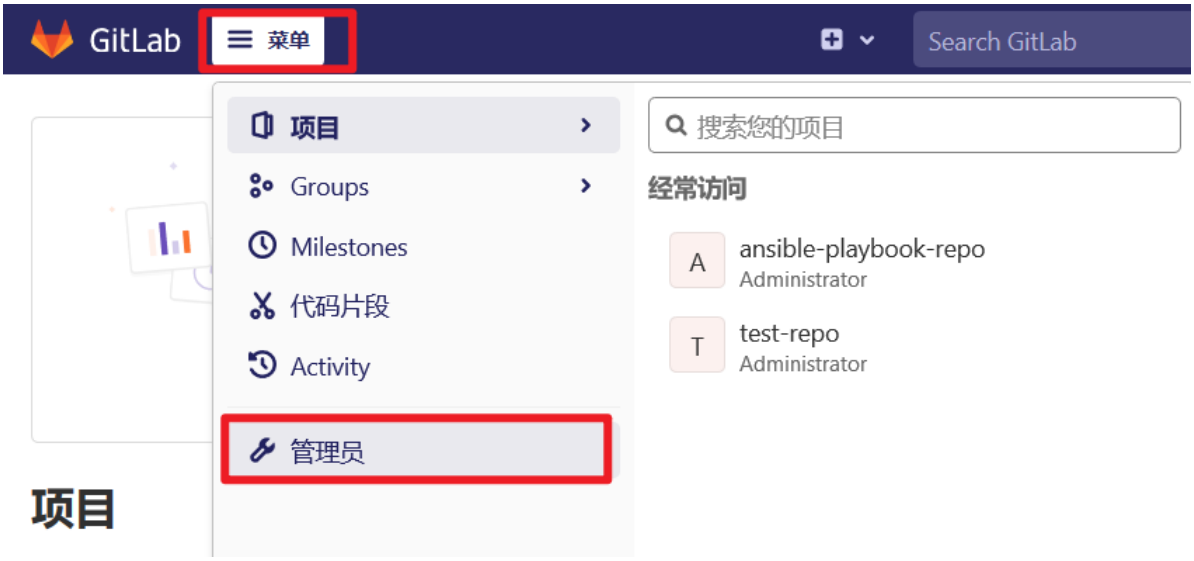
配置 jenkins 工程

配置之前的 Freestyle Job `nginx-freestyle-job`

记录下 URL 和 token ， 配置 gitlab webhook 需要用

配置 gitlab 网络

进入管理员设置



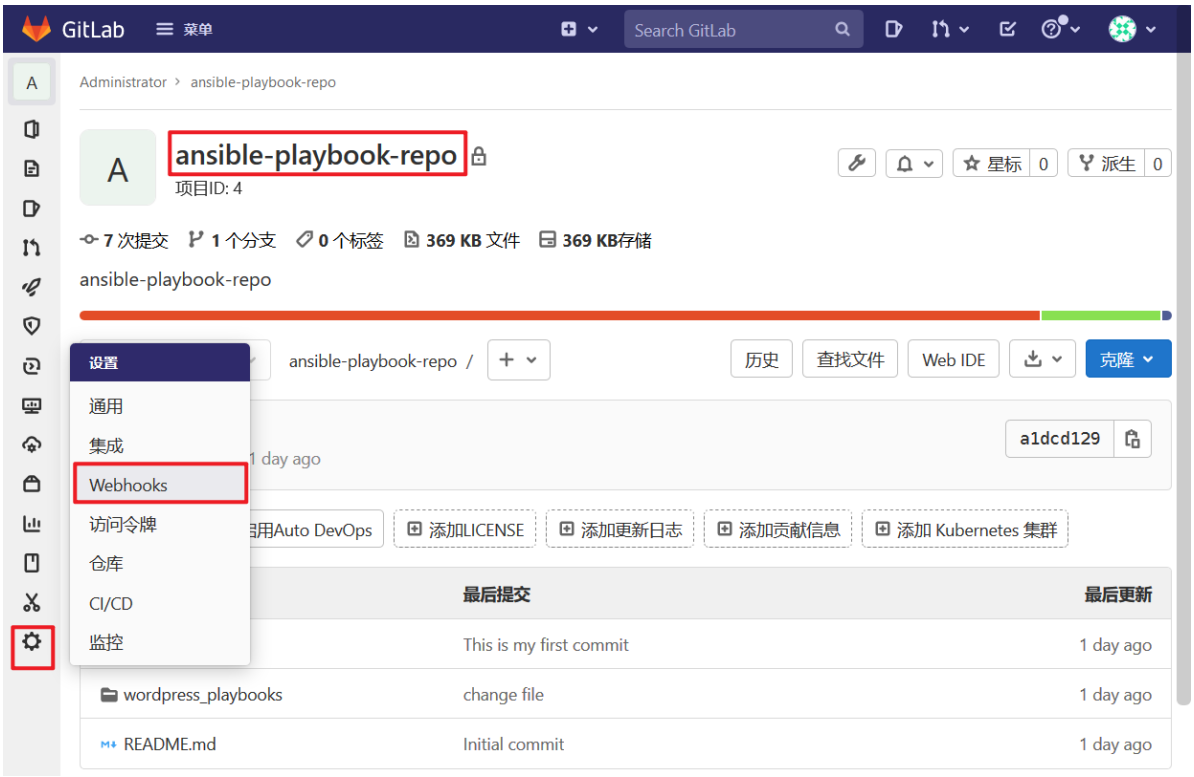
选择网络


GitLab
菜单
管理中心 › 仪表板
实例概览
4
项目
新建项目
查看最新项目 ›
统计
设置
通用
集成
仓库
CI/CD
报告
指标与分析
网络
外观
偏好设置
0
0
0
0
0
0
0
1

配置 gitlab 工程

注意：jenkins 的工程和 gitlalb 的工程师关联绑定的，这样才能触发构建

进入之前创建的 `ansible-playbook-repo` 工程



Webhooks

Webhooks允许您针对某个群组或项目中的事件发送通知到web应用程序。 如需使用webhook, 我们推荐优先使用已有集成。

网址

http://jenkins.example.com:8080/project/nginx-freestyle-job jenkins 工程的 url

URL must be percent-encoded if neccessary.

Secret token

dda6b88c14860ac41e1e253ecd6f5bbb jenkins 生成的 token

Use this token to validate received payloads. It is sent with the request in the X-Gitlab-Token HTTP header.

触发来源

- [x] **推送事件**

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

  URL is triggered by a push to the repository

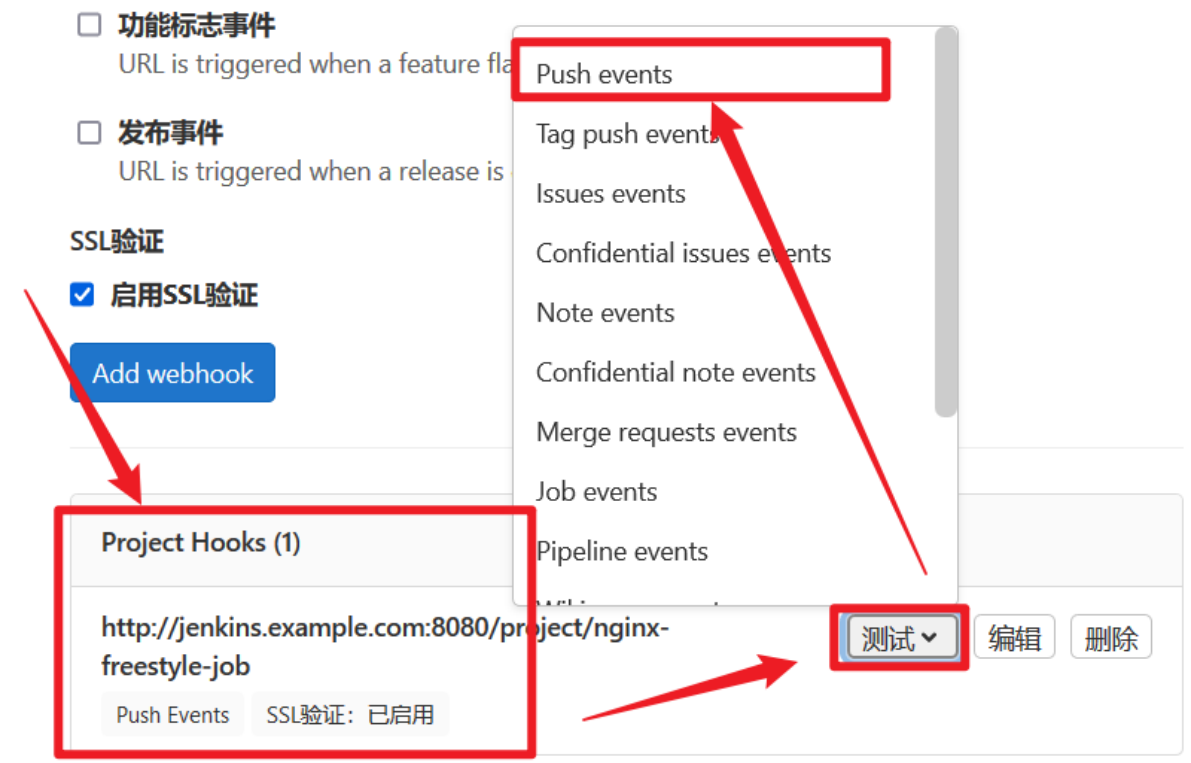
- [ ] **标签推送事件**
  URL is triggered when a new tag is pushed to the repository
- [ ] **评论**
  URL is triggered when someone adds a comment
- [ ] **Confidential comments**
  URL is triggered when someone adds a comment on a confidential issue
- [ ] **议题事件**
  URL is triggered when an issue is created, updated, closed, or reopened
- [ ] **Confidential issues events**
  URL is triggered when a confidential issue is created, updated, closed, or reopened
- [ ] **合并请求事件**
  URL is triggered when a merge request is created, updated, or merged
- [ ] **作业事件**
  URL is triggered when the job status changes
- [ ] **流水线事件**
  URL is triggered when the pipeline status changes
- [ ] **Wiki页面事件**
  URL is triggered when a wiki page is created or updated
- [ ] **部署事件**
  URL is triggered when a deployment starts, finishes, fails, or is canceled
- [ ] **功能标志事件**
  URL is triggered when a feature flag is turned on or off
- [ ] **发布事件**
  URL is triggered when a release is created or updated

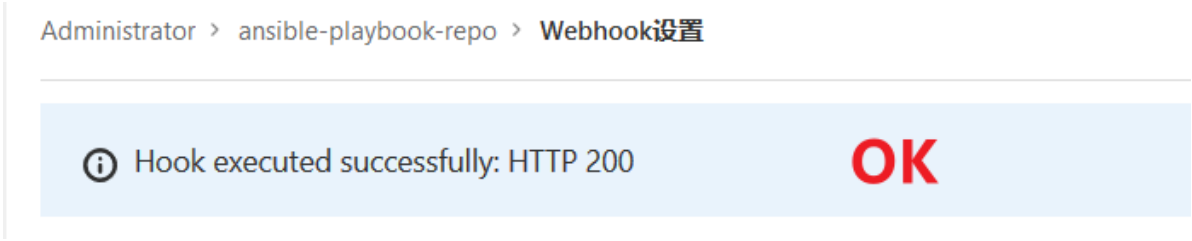
SSL验证

- [x] **启用SSL验证**

Add webhook

测试 webhook





验证触发自动构建

修改 `nginx_playbooks/roles/nginx/files/index.html`

提交推送到 `gitlab`

观察是否触发 `jenkins` 工程自动构建