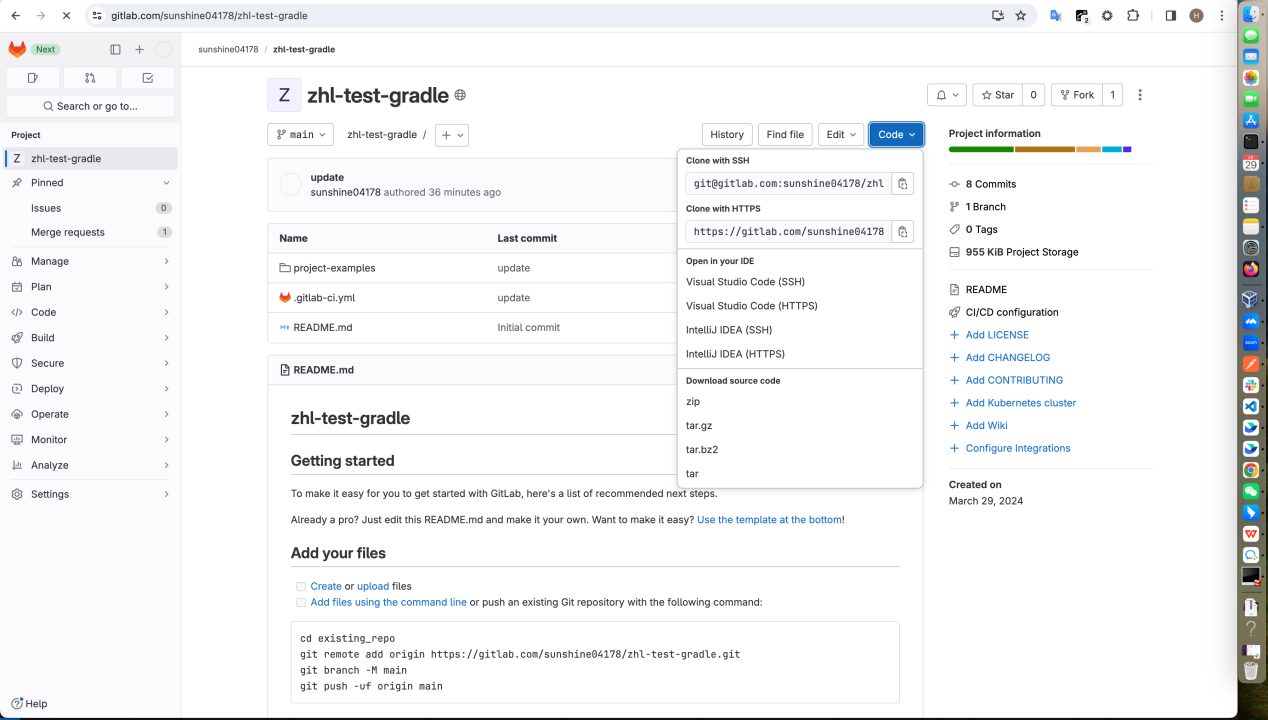
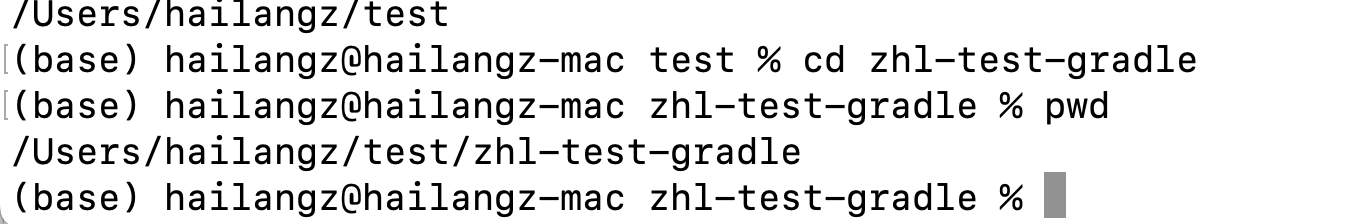
1. create gitlab project



1. clone that project

git clone https://gitlab.com/sunshine04178/zhl-test-gradle.git

(you should use you own project)



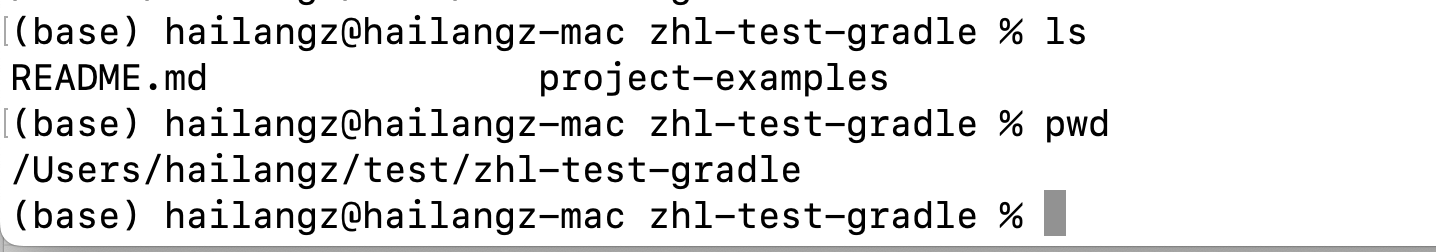
1. download project-examples

<https://github.com/jfrog/project-examples/archive/refs/heads/master.zip>

unzip project-examples-master.zip

mv project-examples-master /Users/hailangz/test/zhl-test-gradle/

project-examples

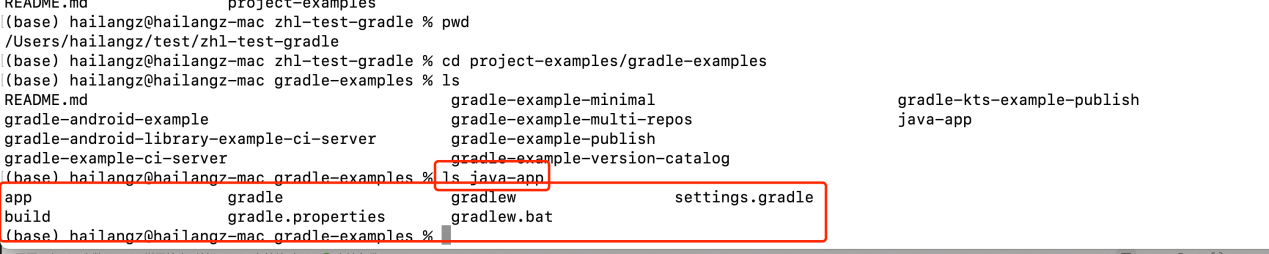


4.init gradle project

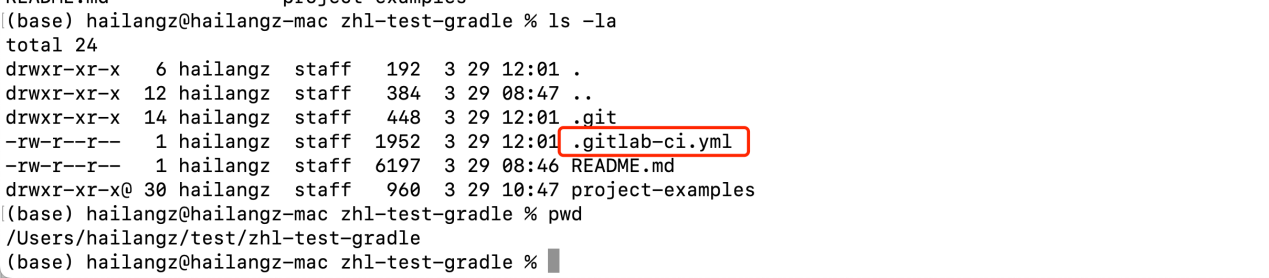
cd project-examples/gradle-examples

mkdir java-app

gradle init  --type java-application



1. create .gitlab-ci.yml



before\_script:

 # Download JFrog CLI

 -  curl -fL https://install-cli.jfrog.io | sh

 # configure artifactory server

 - jf config add $JFROG\_PLATFORM\_SERVER\_ID --url=$JFROG\_PLATFORM\_URL --user=$JFROG\_PLATFORM\_USER --password=$JFROG\_PLATFORM\_PASS

  # configure gradle, set upload and download repos

 - jf gradlec --repo-resolve $JFROG\_PLATFORM\_REPO --repo-deploy $JFROG\_PLATFORM\_REPO

 - jf config show

 # configure gradle, set upload and download repos

 - jf mvnc --repo-resolve-releases $JFROG\_PLATFORM\_REPO --repo-deploy-releases $JFROG\_PLATFORM\_REPO  --repo-resolve-snapshots $JFROG\_PLATFORM\_REPO --repo-deploy-snapshots $JFROG\_PLATFORM\_REPO

 - jf c use $JFROG\_PLATFORM\_SERVER\_ID

stages:

  - build

gitlab-ci-gradle-build: #This job name will be used as the build info repository name in artifactory

 stage: build

 image: gradle:8.0.2

 script:

   - cd project-examples/gradle-examples/java-app

   # upload the built artifact to artifactory

   - jf gradle clean artifactoryPublish --build-name=$CI\_JOB\_NAME --build-number=$CI\_JOB\_ID

   # Collect environment variables for the build

   - jf rt bce $CI\_JOB\_NAME $CI\_JOB\_ID

   # Collect VCS details from git and add them to the build

   - jf rt bag $CI\_JOB\_NAME $CI\_JOB\_ID

   # Publish build info

   - jf rt bp $CI\_JOB\_NAME $CI\_JOB\_ID

   # Scan with xray

  #  - jf bs $CI\_JOB\_NAME $CI\_JOB\_ID

java-build:

  # select the most appropriate image for your project

  image: maven:3.8-openjdk-11

  stage: build

  script:

    - cd project-examples/maven-examples/maven-example

    # upload the built artifact to artifactory

    - jf mvn deploy --build-name=$CI\_JOB\_NAME --build-number=$CI\_JOB\_ID

   # Collect environment variables for the build

    - jf rt bce $CI\_JOB\_NAME $CI\_JOB\_ID

   # Collect VCS details from git and add them to the build

    - jf rt bag $CI\_JOB\_NAME $CI\_JOB\_ID

   # Publish build info

    - jf rt bp $CI\_JOB\_NAME $CI\_JOB\_ID

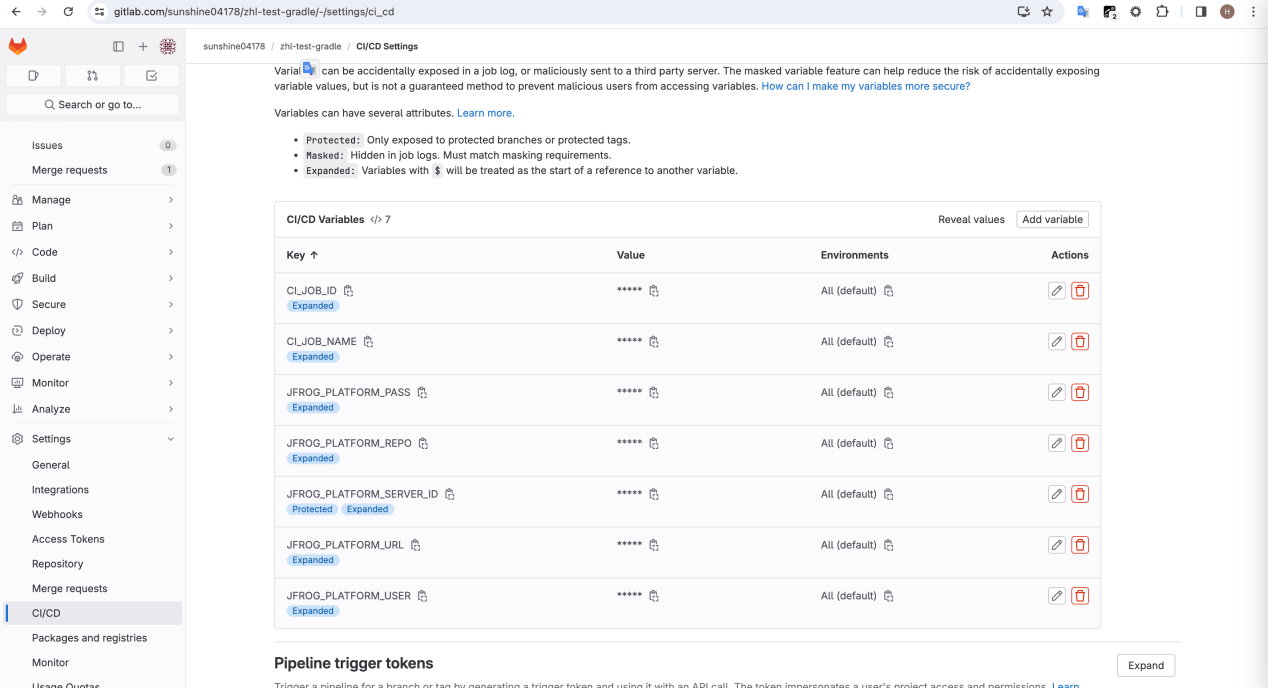
   # Scan with xray

    # - jf bs $CI\_JOB\_NAME $CI\_JOB\_ID

1. git push project

git push

1. config Variables



1. run ci

