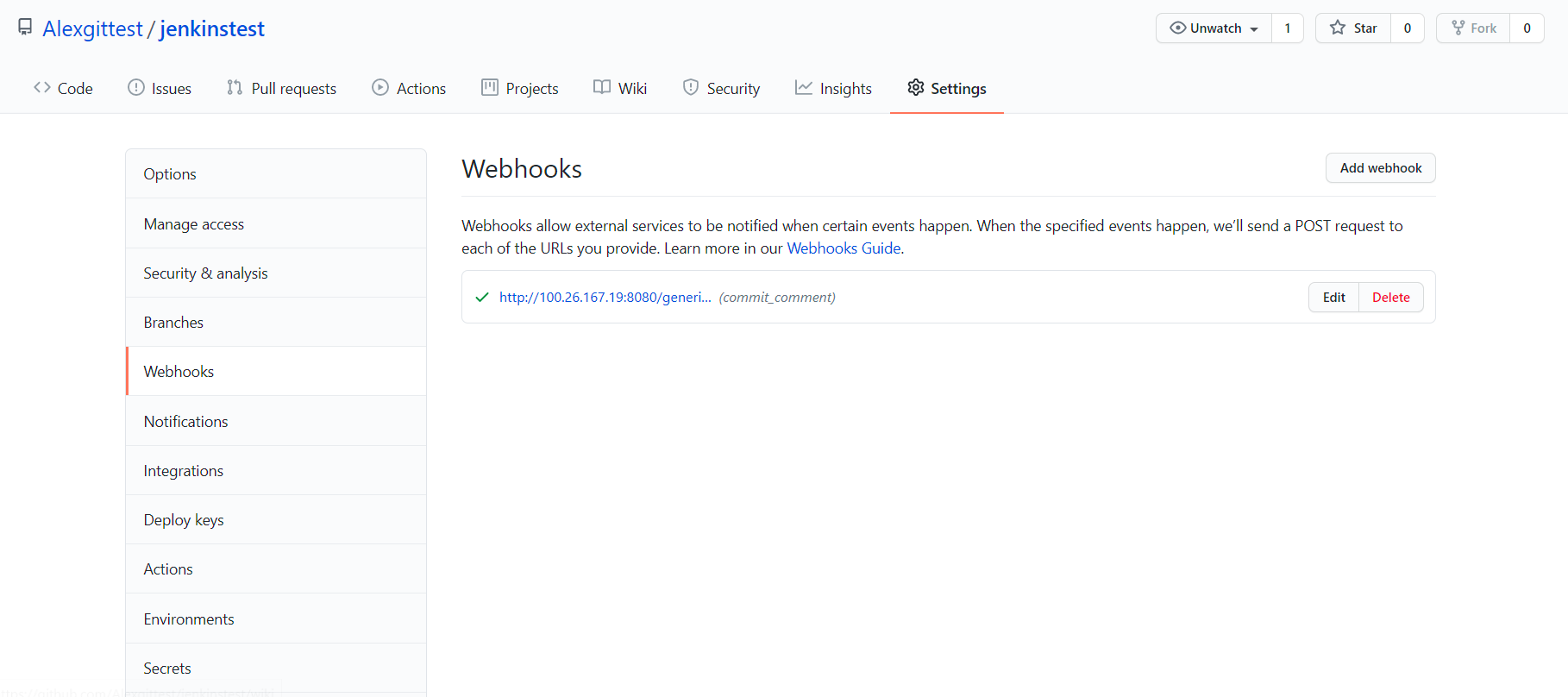
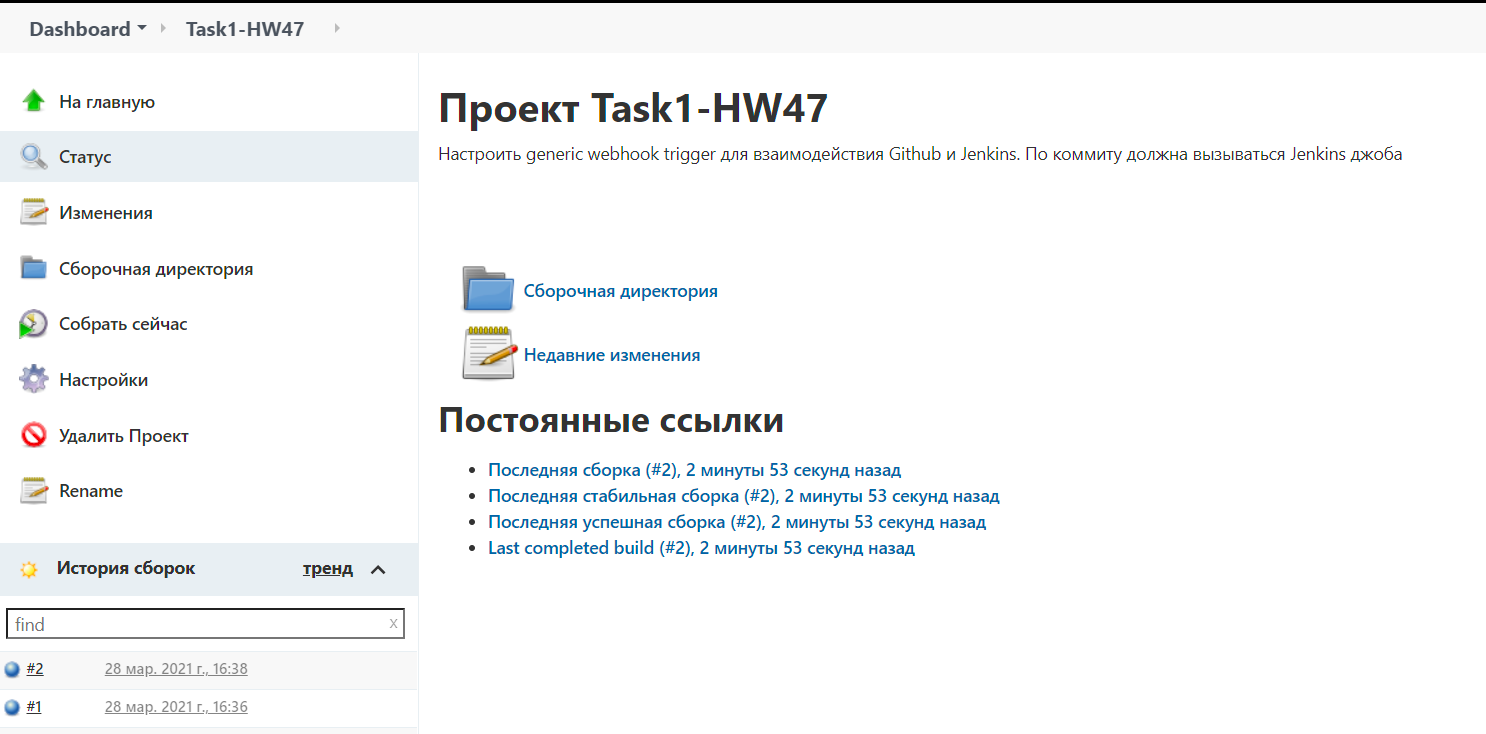
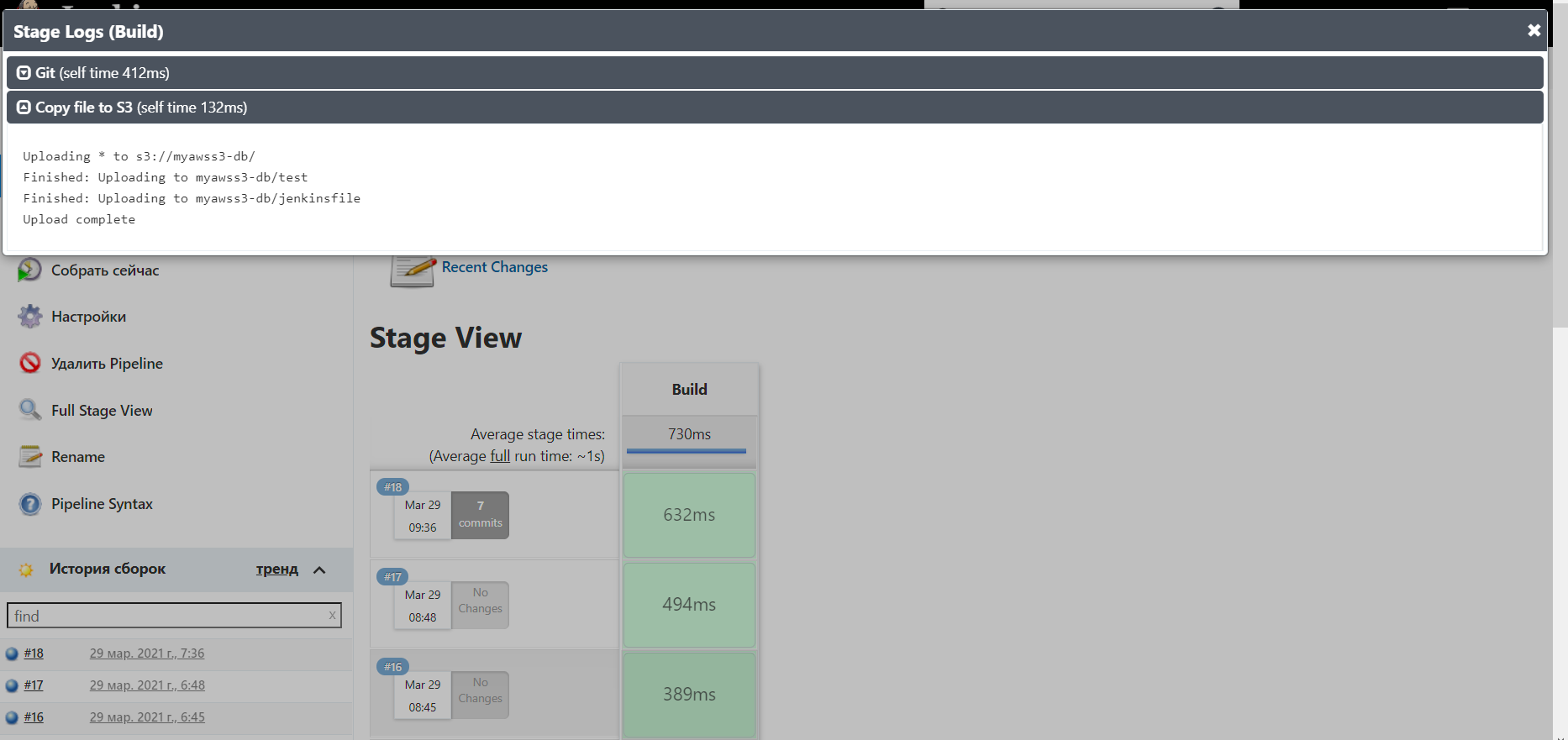
1. Настроить generic webhook trigger для взаимодействия Github и Jenkins. По коммиту должна вызываться Jenkins джоба







1. Написать Jenkins джобу, которая вызывается при коммите в репу и забирает оттуда файлы и закидывает их на S3 бакет(должно быть поле Exclude)



pipeline {

agent any

triggers {

GenericTrigger (causeString: 'Generic Cause', regexpFilterExpression: '', regexpFilterText: '', token: 'task3-hw47', tokenCredentialId: '')

}

stages {

stage('Build') {

steps {

git branch: 'main', url: 'https://github.com/Alexgittest/jenkinstest.git'

withAWS(region:'us-east-1',credentials:'Aleksandr\_AWS') {

s3Upload acl: 'Private', bucket: 'myawss3-db', cacheControl: '', includePathPattern:'\*',excludePathPattern:'\*.git', metadatas: [''], redirectLocation: '', sseAlgorithm: '', tags: '', workingDir: ''

}

}

}

}

}

1. Поднять базу данных и при коммите в репу(в репе должен передаваться SELECT запрос) дергается Jenkins джоба, которая забирает этот запрос, делает бэкап базы и закидывает на S3 и выполняет этот запрос и закидывает результат на S3(можно использовать базу, которую проходили в курсе)

pipeline {

agent any

triggers {

GenericTrigger (causeString: 'Generic Cause', regexpFilterExpression: '', regexpFilterText: '', token: 'task3-hw47', tokenCredentialId: '')

}

stages {

stage('build') {

steps {

git branch: 'main', url: 'https://github.com/Alexgittest/jenkinstest.git'

sh 'mysqldump -h 18.185.92.39 --no-tablespaces -u jenkins --password=\'password\' shop > shop\_db\_dump\_`date +%d\_%m\_%Y\_%T`.sql'

sh '''last\_select="`tail -n 1 select\_file`"

mysql -h 18.185.92.39 -u jenkins --password=\'password\' shop -e "$last\_select" > select\_result\_`date +%d\_%m\_%Y\_%T`.sql'''

withAWS(region:'us-east-1',credentials:'Aleksandr\_AWS') {

s3Upload acl: 'Private', bucket: 'myawss3-db', cacheControl: '', includePathPattern:'\*.sql',excludePathPattern:'\*.git', metadatas: [''], redirectLocation: '', sseAlgorithm: '', tags: '', workingDir: ''

}

}

}

}

}

1. У нас есть любой python code(лучше взять свой код, когда мы писали flask приложение). При пуше в репу мы забираем этот код(должны быть настроены git credentials), запускаем pytests и дальше собираем Docker container и закидываем его на Docker Hub(лучше в ecr в aws, но там ограничения по фри тир, нужно посмотреть).

pipeline {

triggers {

GenericTrigger (causeString: 'Generic Cause', regexpFilterExpression: '', regexpFilterText: '', token: 'task4-hw47', tokenCredentialId: '')

}

agent any

stages {

stage('Hello') {

steps {

git branch: 'main', url: 'https://github.com/Alexgittest/Flaskapp.git'

docker.withRegistry('https://hub.docker.com', 'dockerhub') {

def Myflaskapp = docker.build("Aleks:flask-app")

Myflaskapp.push()

}

}

}

}

}

Вариант2

pipeline {

triggers {

GenericTrigger (causeString: 'Generic Cause', regexpFilterExpression: '', regexpFilterText: '', token: 'task4-hw47', tokenCredentialId: '')

}

agent any

stages {

stage('Hello') {

steps {

git branch: 'main', url: 'https://github.com/Alexgittest/Flaskapp.git'

sh ‘’’ docker build -t alexandrkorol/flaskapp:v1 . ‘’’

sh ‘’’ docker login -u alexandrkorol -p password ‘’’

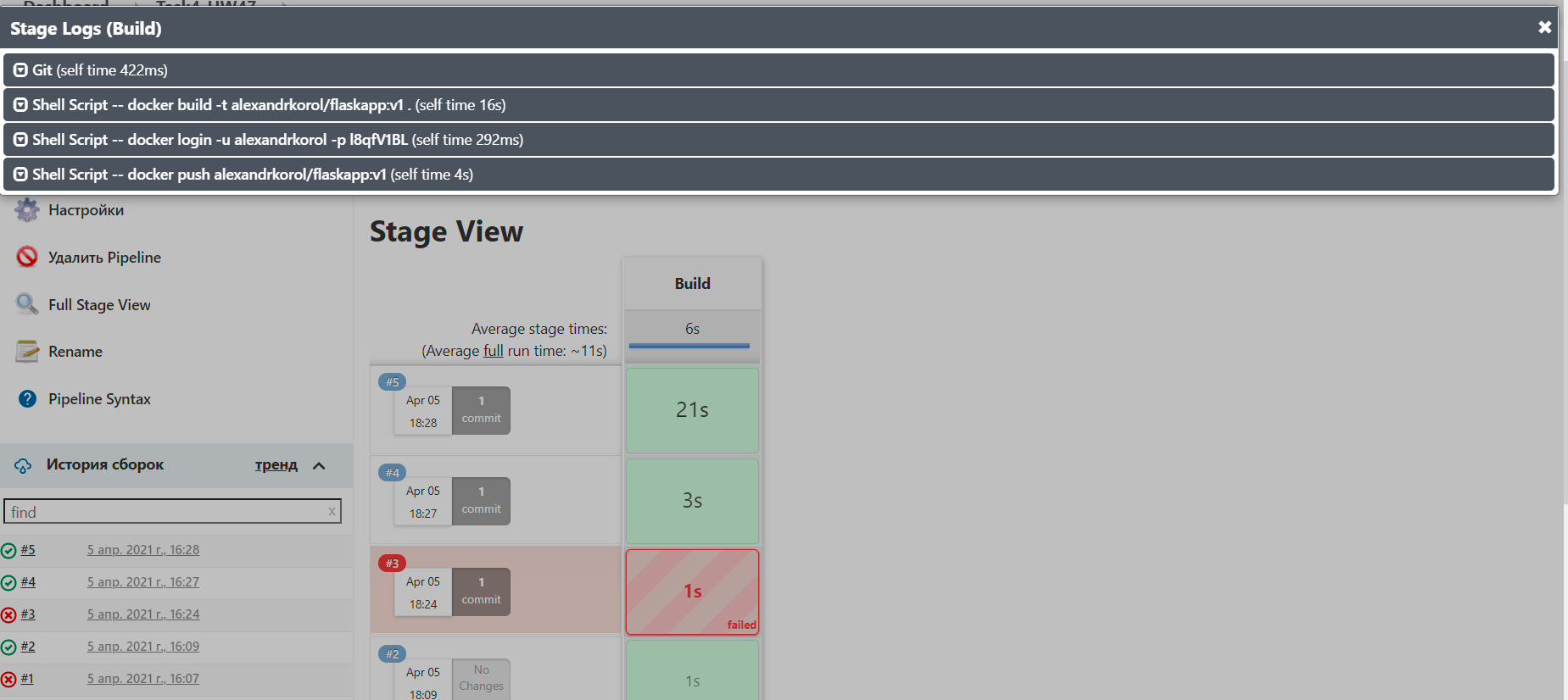
sh ‘’’ docker push alexandrkorol/flaskapp:v1 ‘’’

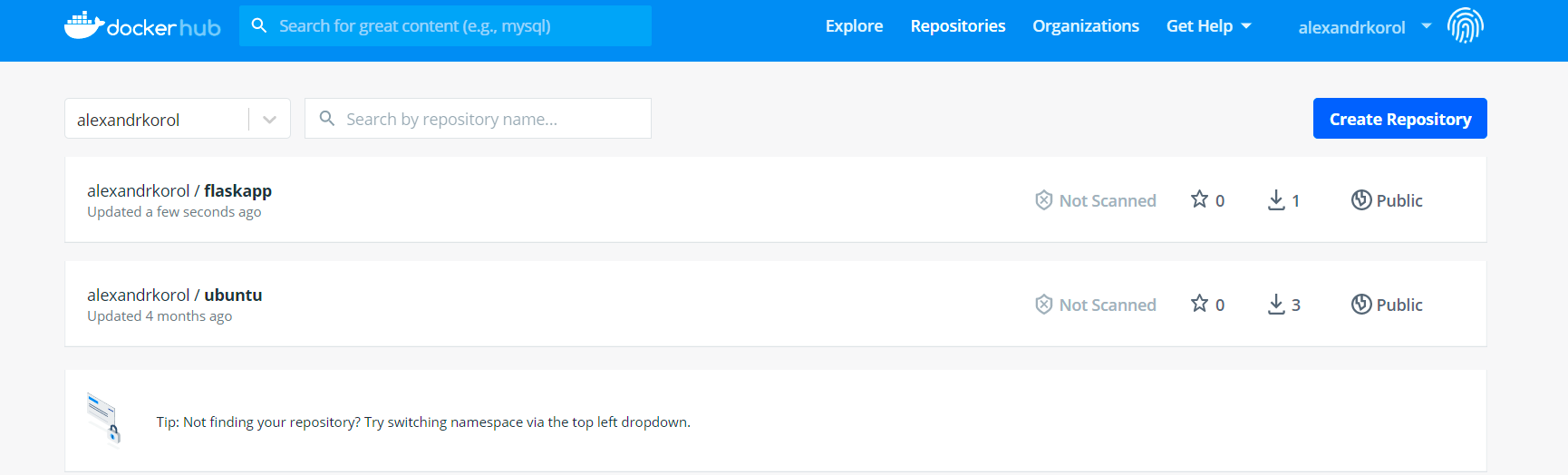
}

}

}

}





1. Выполнить задания через Jenkinsfile