# **Задание 1**

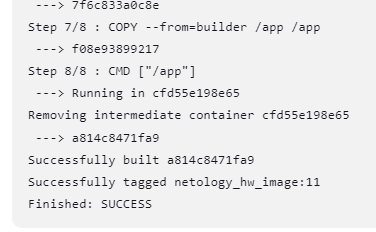
Что нужно сделать:

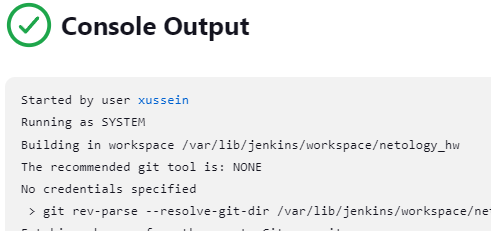
1. Установите себе jenkins по инструкции из лекции или любым другим способом из официальной документации. Использовать Docker в этом задании нежелательно.
2. Установите на машину с jenkins [golang](https://golang.org/doc/install).
3. Используя свой аккаунт на GitHub, сделайте себе форк [репозитория](https://github.com/netology-code/sdvps-materials.git). В этом же репозитории находится [дополнительный материал для выполнения ДЗ](https://github.com/netology-code/sdvps-materials/blob/main/CICD/8.2-hw.md).
4. Создайте в jenkins Freestyle Project, подключите получившийся репозиторий к нему и произведите запуск тестов и сборку проекта go test . и docker build ..

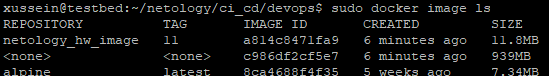
В качестве ответа пришлите скриншоты с настройками проекта и результатами выполнения сборки.

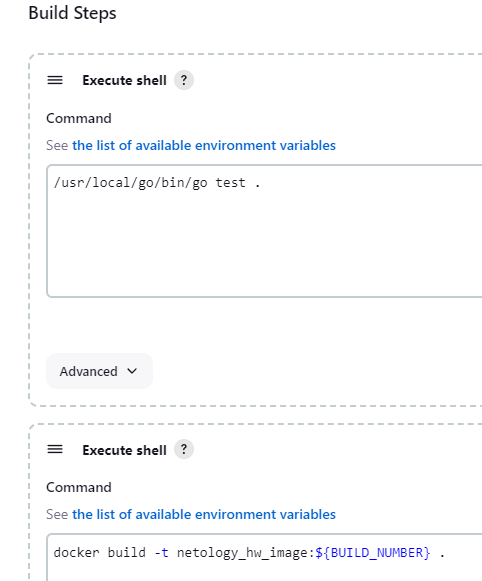
# **Решение 1**

ВНЕЗАПНО, взлетело только с 11й попытки:









# **Задание 2**

Что нужно сделать:

1. Создайте новый проект pipeline.
2. Перепишите сборку из задания 1 на declarative в виде кода.

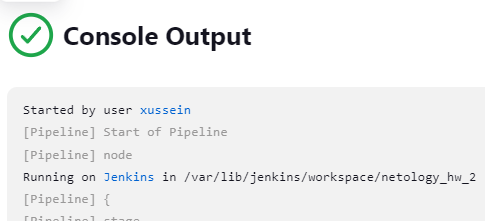
В качестве ответа пришлите скриншоты с настройками проекта и результатами выполнения сборки.

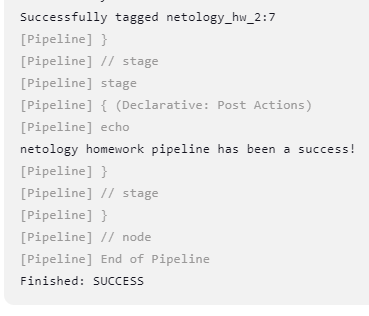
# **Решение 2**

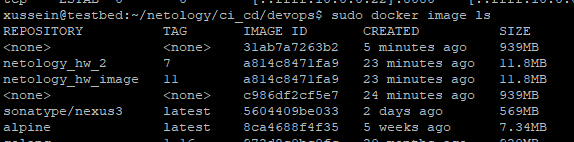
pipeline code:

1. pipeline {
2. agent any
3. stages {
4. stage('Checkout') {
5. steps {
6. git branch: 'main', url: 'https://github.com/sadxussein/sdvps-materials'
7. }
8. }
9. stage('go test') {
10. steps {
11. sh '/usr/local/go/bin/go test .'
12. }
13. }
14. stage('docker build') {
15. steps {
16. sh 'docker build -t netology\_hw\_2:${BUILD\_NUMBER} .'
17. }
18. }
19. }
20. post {
21. // Действия после завершения pipeline
22. success {
23. echo 'netology homework pipeline has been a success!'
24. }
25. failure {
26. echo 'netology homework pipeline has been a failure! Just like you are.'
27. }
28. }
29. }

Всего с 7й попытки:







# **Задание 3**

Что нужно сделать:

1. Установите на машину Nexus.
2. Создайте raw-hosted репозиторий.
3. Измените pipeline так, чтобы вместо Docker-образа собирался бинарный go-файл. Команду можно скопировать из Dockerfile.
4. Загрузите файл в репозиторий с помощью jenkins.

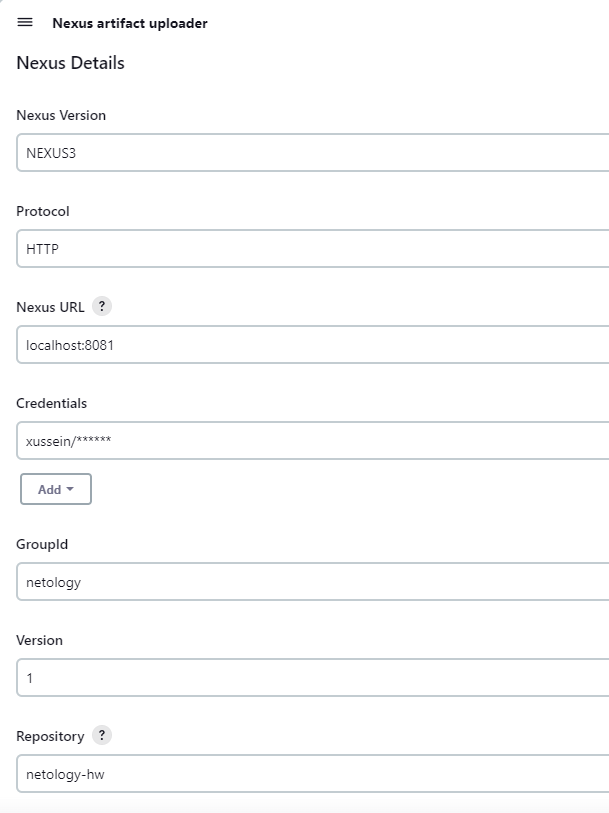
В качестве ответа пришлите скриншоты с настройками проекта и результатами выполнения сборки.

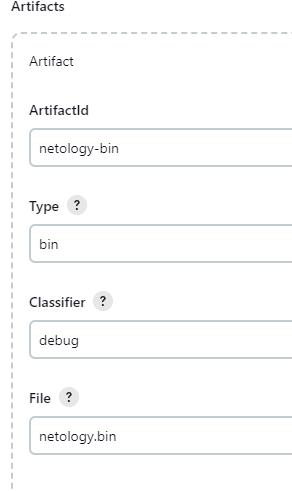
# **Решение 3**

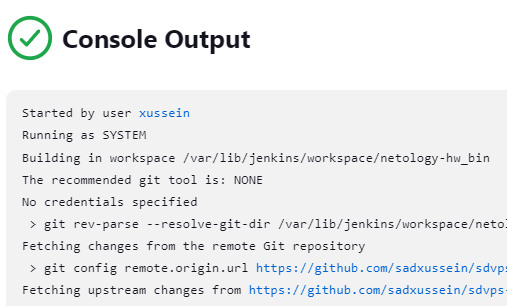
Изначально я пытался описать через declarative, но nexus artifact uploader уперся и наотрез отказался принимать credentialsId, видимо я где-то там таки накосячил… Почему-то на credentialsId java уходит в nullPointer, хотя id настоящий.

В итоге решил просто через UI (мануал уже утром покурю):

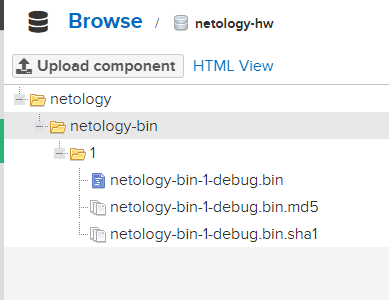












Код был такой, вопрос почему не сработал?

1. pipeline {
2. agent any
3. environment {
4. NEXUS\_URL = 'http://localhost:8081'
5. NEXUS\_REPOSITORY = 'netology-hw'
6. BINARY\_NAME = 'netology.bin'
7. BUILD = "${BUILD\_NUMBER}"
8. }
9. stages {
10. stage('Checkout') {
11. steps {
12. git branch: 'main', url: 'https://github.com/sadxussein/sdvps-materials'
13. }
14. }
15. stage('go build') {
16. steps {
17. sh '/usr/local/go/bin/go build -o $BINARY\_NAME .'
18. }
19. }
20. stage('Push to Nexus') {
21. steps {
22. script {
23. nexusArtifactUploader(
24. nexusVersion: 'nexus3',
25. protocol: 'http',
26. nexusUrl: NEXUS\_URL,
27. groupId: '',
28. version: BUILD,
29. repository: NEXUS\_REPOSITORY,
30. credentialsId: '176d7fdb-4278-4425-9b1e-121bfb353fec',
31. artifacts: [
32. [artifactId: BINARY\_NAME,
33. classifier: '',
34. file: BINARY\_NAME,
35. type: 'bin']
36. ]
37. )
38. }
39. }
40. }
41. }
42. post {
43. success {
44. echo 'netology homework pipeline has been a success!'
45. }
46. failure {
47. echo 'netology homework pipeline has been a failure! Just like you are.'
48. }
49. }
50. }

# **Задание 4**

Придумайте способ версионировать приложение, чтобы каждый следующий запуск сборки присваивал имени файла новую версию. Таким образом, в репозитории Nexus будет храниться история релизов.

Подсказка: используйте переменную BUILD\_NUMBER.

В качестве ответа пришлите скриншоты с настройками проекта и результатами выполнения сборки.

# **Решение 4**

В предыдущих заданиях я уже использовал эту возможность:

1. stage('docker build') {
2. steps {
3. sh 'docker build -t netology\_hw\_2:${BUILD\_NUMBER} .'
4. }
5. }