I Can Win

1. Установите *Jenkins*

done

2. Создайте задачу, которая будет делать следующее:

2.1. Клонировать проект *https://github.com/vitalliuss/helloci*

В *Global Tool Configuration* прописываем расположение нашего гита и его версию.

Нажимаем на *New Item.* Указываем имя проекта и тип, нажимаем *Save*. Далее в *Source Code Management* выбираем *Git* и в нем прописываем адрес репозитория. В *Branches to build* можно указать спецификатор ветки – в данном случае укажем главную ветвь \*/*master*. *Save*. С помощью *Build Now* клонируем проект. Проверить можно в *Workspace*.

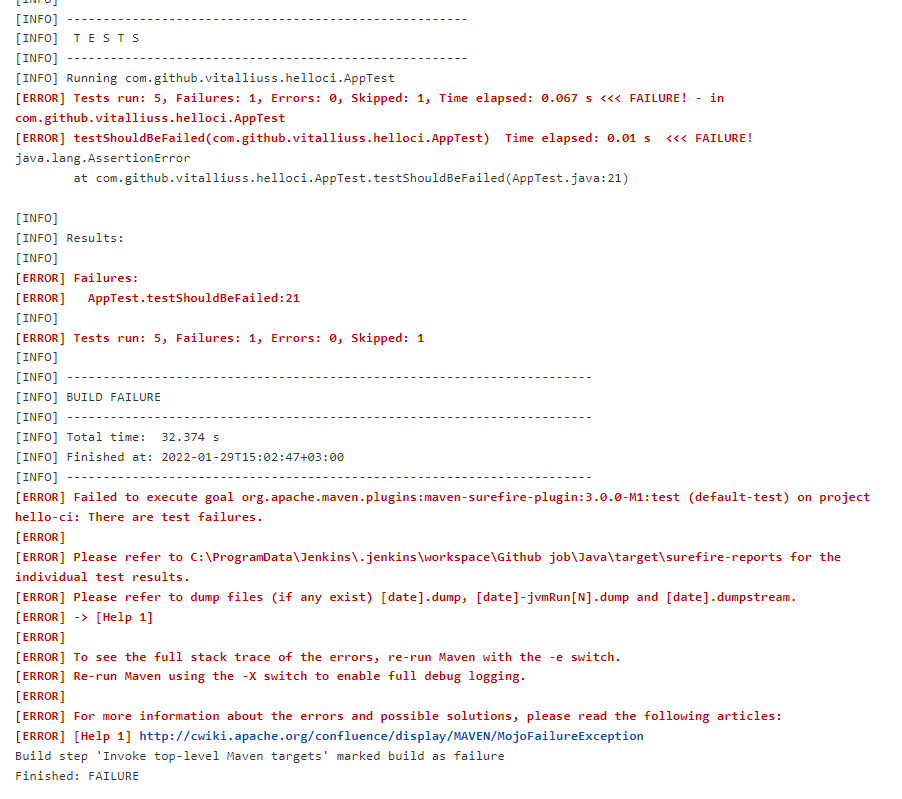


2.2. Запускать тесты из проекта в директории Java с помощью цели *mvn test*

В *Global Tool Configuration* прописываем расположение нашего мавена и его версию.

В настройках созданной джобы в пункте 2.1. во вкладке *Build* указываем версию мавена и цель – *test.* В *Advanced* указываем *pom.xml* *Save.*

После запуска джобы результат можно посмотреть в *Console Output*:



3. Настройте билд триггеры таким образом, чтобы задача выполнялась раз в 5 минут

В джобе из пункта 2, в настройках, во вкладке *Build Triggers* выбираем чек-бокс *Build periodically* и прописываем \*/5 \* \* \* \*.

Bring It On

1. Установите *Jenkins*

done

2. Создайте задачу которая будет делать следующее:

2.1. Клонировать проект *https://github.com/vitalliuss/helloci*

2.2. Запускать тесты из проекта в директории Java с помощью цели *mvn test*

done

3. Настройте билд триггеры:

3.1. Запуск тестов не позднее чем через 5 минут после коммита в *git*

В настройках джобы, во вкладке *Build Triggers* выбираем чек-бокс *Poll SCM* и прописываем H/5 \* \* \* \*.

3.2. Каждый будний день в полночь

В настройках джобы, во вкладке *Build Triggers* выбираем чек-бокс *Build periodically* и прописываем 0 0 \* \* 1-5.

4. Опубликуйте файл *“Java\target\surefire-reports\com.github.vitalliuss.helloci.AppTest.txt”* как артефакт

В настройках джобы, во вкладке *Post-build Actions* в выпадающем списке *Add post-build action* выбрать *Archive the artifacts* и прописать путь файла и сам файл в появившемся поле. *Save*.

Результат можно увидеть в билде:



Hurt Me Plenty

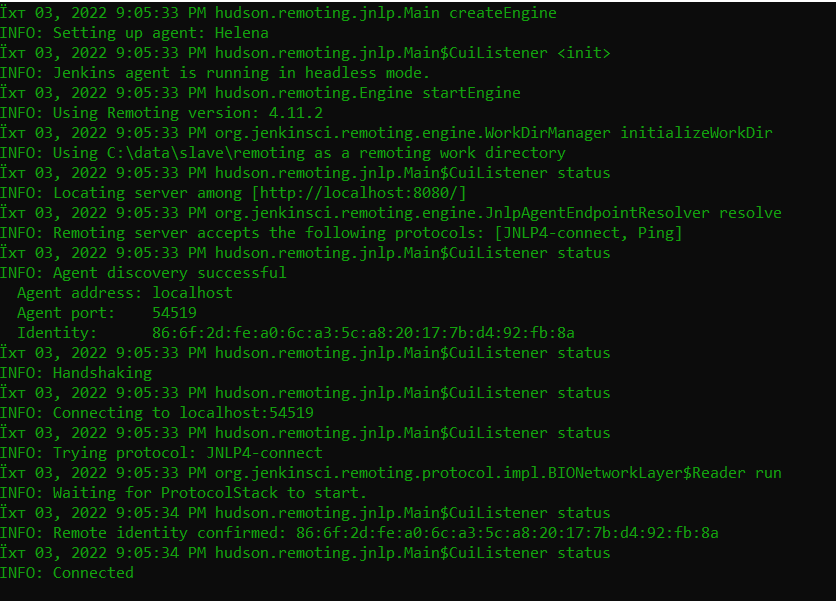
1. Сменить порт сервера на 8081

В Службе останавливаем *Jenkins*. В файле *C:\ProgramData\Jenkins\.jenkins\jenkins.model.JenkinsLocationConfiguration.xml* меняем название порта 8080 на 8081.

2. Создать ноду и настроить сервер так, чтобы джоба выполнялась только на *slave* ноде

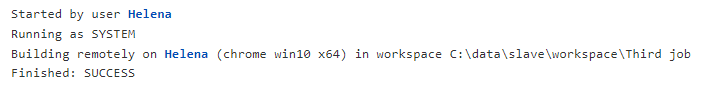
Для начала необходимо выполнить следующее: *Build Executor Status* -> *New Node*. Здесь укажем имя ноды. Создадим у себя на компьютере папку *slave*. В *Remote root directory* пропишем путь к папке *slave*, а в *Launch method* выберем *Launch agent by connecting it to the controller*. В выделенном участке *Run from agent command line* скачиваем и размещаем файл *agent.jar* в папку *slave*, потом копируем всю строку в командную строку и запускаем.

Результатом является:



Далее настроим сервер так, чтобы джоба выполнялась только на *slave* ноде. Для этого зайдем в настройки джобы и в *Restrict where this project can be run* укажем имя ноды в *Label Expression.*

Для проверки нажмем *Build Now* и можно увидеть следующее:



3. Настроить *Job Config History* и *thinBackup*

Для установки *ThinBackup* (выполняет резервное копирование глобальной конфигурации и конкретной конфигурации) и *Job Configuration History* (позволяет отслеживать историю изменений файлов конфигурации *jobs*) необходимо зайти в *Manage Plugins*, добавить сами плагины и указать все необходимые настройки.