**Перед уствановкой Дженкнис нужно установить java - пакет: openjdk-8-jreheadless**

**SSH ПОДКЛЮЧЕНИЕ**

Плагин: **Publish over SSH**

создать ключ SSH

ssh-keygen -t rsa -C "jenkins" -m PEM -P "" -f /var/lib/jenkins/.ssh/id\_rsa

скопировать на удаленный сервер

ssh-copy-id -i /var/lib/jenkins/.ssh/id\_rsa ubuntu@remoteserverip

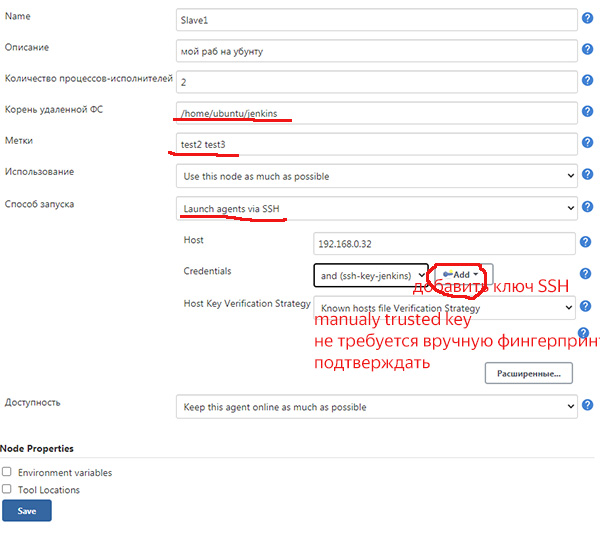
добавляются SSH сервера в настройках дженкинс/конф системы, после этого можно послать билд по SSH (добавляются в настройках задачи внизу «послесборочные операции/send build artifacts over SSH»)

**ДОБАВЛЕНИЕ SLAVE**

скачать два плагина:

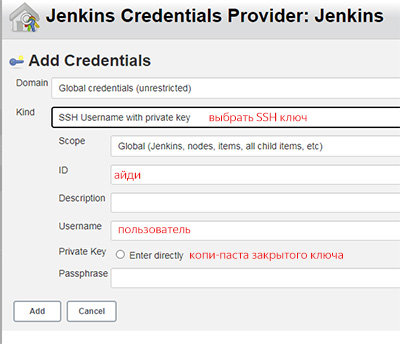
SSH slaves (SSH Build Agents) добавляет агенты через SSH

SSH Agent сохраняет SSH-key и имя пользователя в credentials

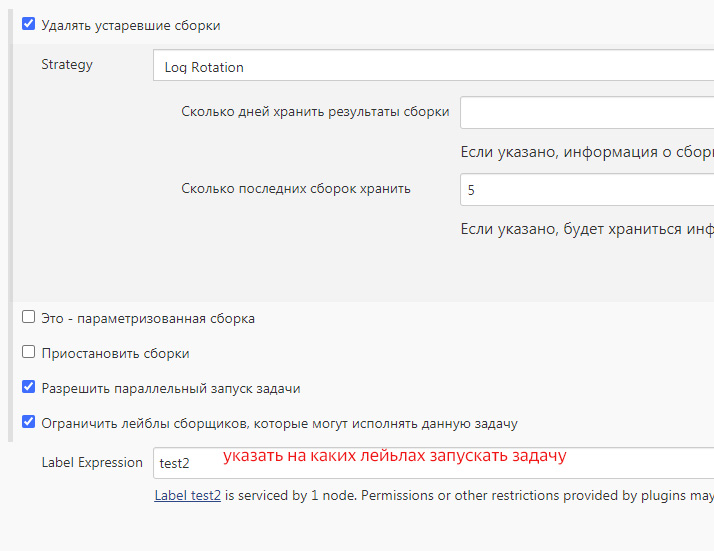
В настройках Дженкинс «**управление средами сборки** (manage nodes)» **новый узел**

Дать права на папку /home/ubuntu/jenkins

Добавление SSH ключа



Указание на каких слейвах запускать задачу, в настройках задачи:



**Удалённое и локальное управление через Jenkins CLI Client**

Скачать из настроек/Jenkins CLI файл  [jenkins-cli.jar](http://192.168.0.140:8080/jnlpJars/jenkins-cli.jar) в терминале через wget

*Вход по паролю:*

java -jar jenkins-cli.jar –auth пользователь:пароль -s http://localhost:8080 who-am-i

*Вход через токен:*

Залогинится под нужным юзеров на сайт, в настройках/упр. Пользователями/настройка юзера/add new token задать имя/generate и скопровать в буфер

1155870c12fd85fccfad175de0caba814c

java -jar jenkins-cli.jar –auth пользователь:токен -s http://localhost:8080 who-am-i

*вход через переменную:*

создать переменные, стандартной командой export

export JENKINS\_USER\_ID=юзер

export JENKINS\_API\_TOKEN=токен

теперь можно запускать без указания логина/пароля:

java -jar jenkins-cli.jar -s http://localhost:8080 who-am-i

таже команда с другого терминала при скачанном jenkins-cli.jar с ip адресом

список команд в настройках/JENKINS CLI

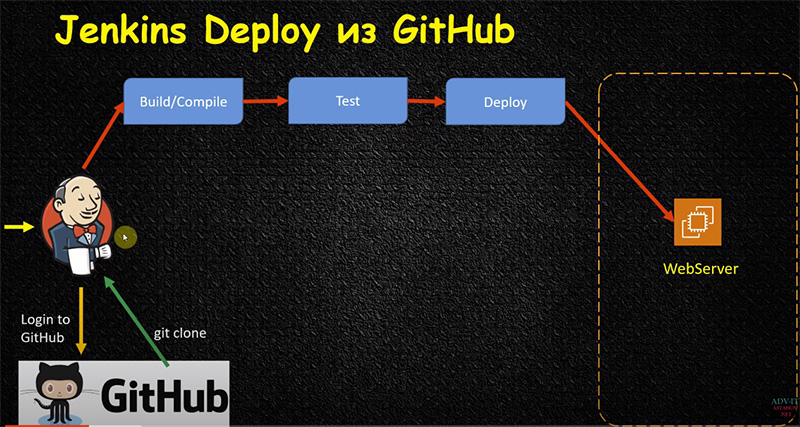
*например скопировать джоб в XML файл командой get-job :*

java -jar jenkins-cli.jar -auth логин:пароль -s http://192.168.0.140:8080/ get-job MyJob-1 > myjob.xml

*можно отредактировать и отправить в назад командой create\_job :*

java -jar jenkins-cli.jar -auth логин:пароль -s http://192.168.0.140:8080/ create-job JobFromXML < myjob.xml

**Deployment из GitHub**

****

Пушим из GIT в Github файл который надо задеплоить через дженкинс

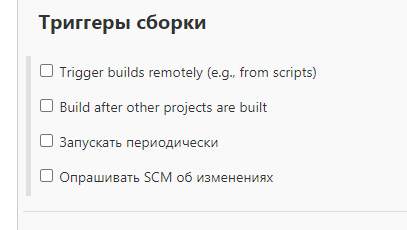
В настройках джоба в разделе source code (надо устнавить плагин git plugin) точка GIT

Вставляем repository URL

В GitHub добавляем публичный ключ jenkins, в самом jenkins добавляем закрытый ключ

Теперь джоб будет идти в GitHub и клонировать весь указанный репозиторий

**Автоматизация запуска Build Job - Jenkins Build Triggers**



1.*Trigger builds remotely* – по запросу из ссылки, ввести токен для генерации ссылки

 JENKINS\_URL/job/Auto%20trigger%20JOB/build?token=TOKEN\_NAME

Подставить genkins\_url и token\_name

2. *Build after other projects are built* –

Указываем после какого джоба запуститься этот джоб

3. *запуск по расписанию*

Пример:

H/15 \* \* \* \* - запускать каждые 15 мин

4. *опрашивать изменения в source code management –*

Установить время проверки, при изменении в коммите гитхаб, произойдет запуск джоба

**Автоматизация запуска Build из GitHub - Jenkins trigger from GitHub, Jenkins webhook**

Запуск джоба сразу при изменении в репозитори ГитХаба без таймеров

Установить плагин GitHub

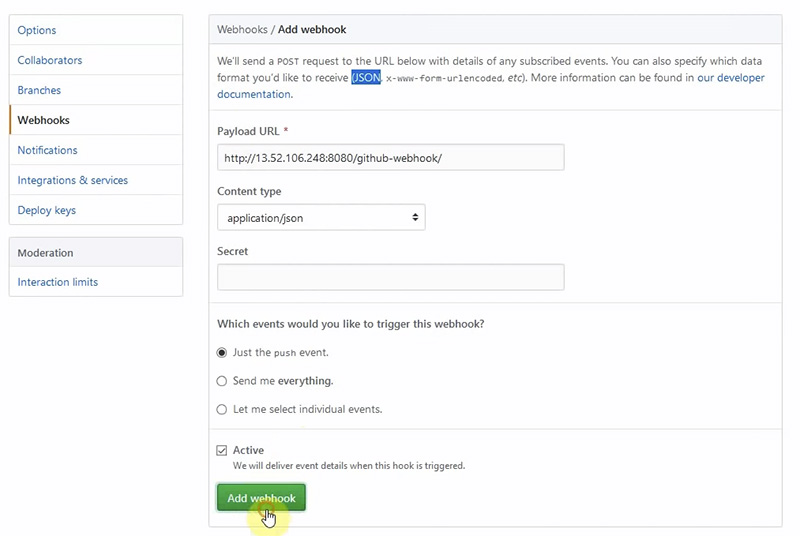
В настройках джоба поставить ссылку на ГитХаб



В триггерах сборки установить галку

GitHub hook trigger for GITScm polling

В настройках на GitHub:



Payload URL должен быть виден в интернет

Теперь при изменении в репозитории GitHub будет произведен запуск джоба в jenkins