Домашняя работа №2.

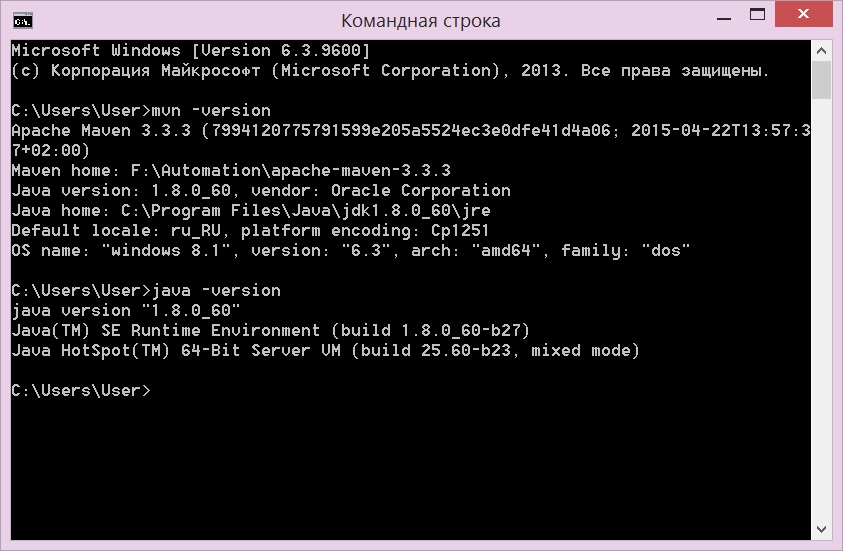
**Задача: собрать и протестировать существующий проект с помощью Jenkins**

**1.Подготовить рабочее окружение (Java/Maven)**

После скачивания и установки Javaи Mavenнеобходимо задать переменные окружения JAVA\_HOMEи MAVEN\_HOME, значения содержат абсолютный путь к Javaи Mavenсоответственно. Для проверки корректности установки используем команды:

**Java –version**

**Mvn -version**



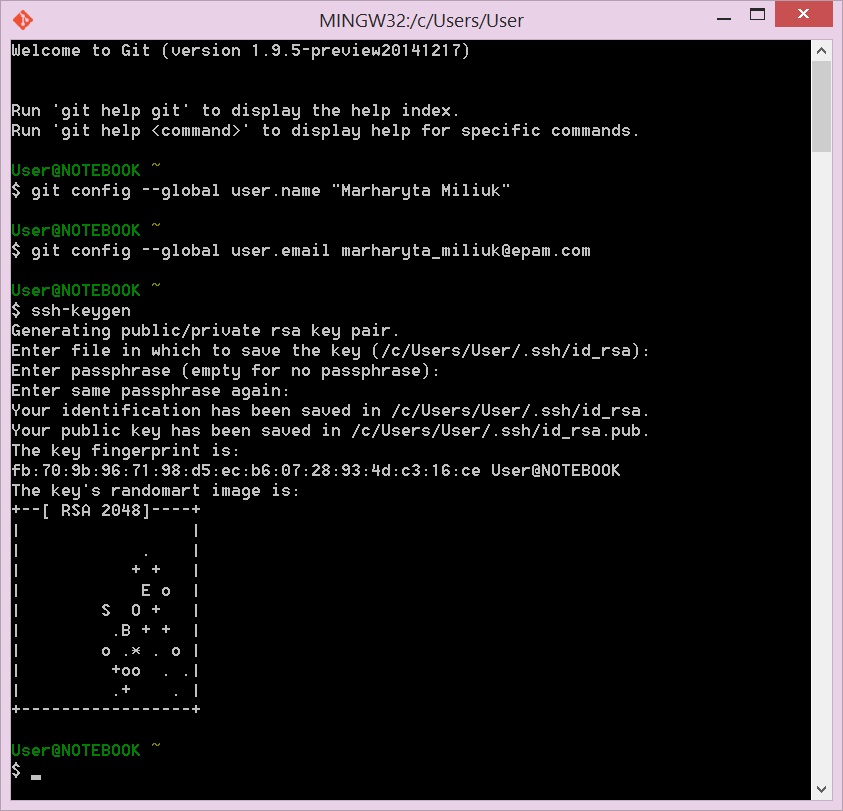
**2.Сгенерировать SSH ключи (если у вас нет существующих)**

После установки Gitпереходим в GitBash, настраиваем пользовательские данные

**$gitconfig –global user.name “UserName”**

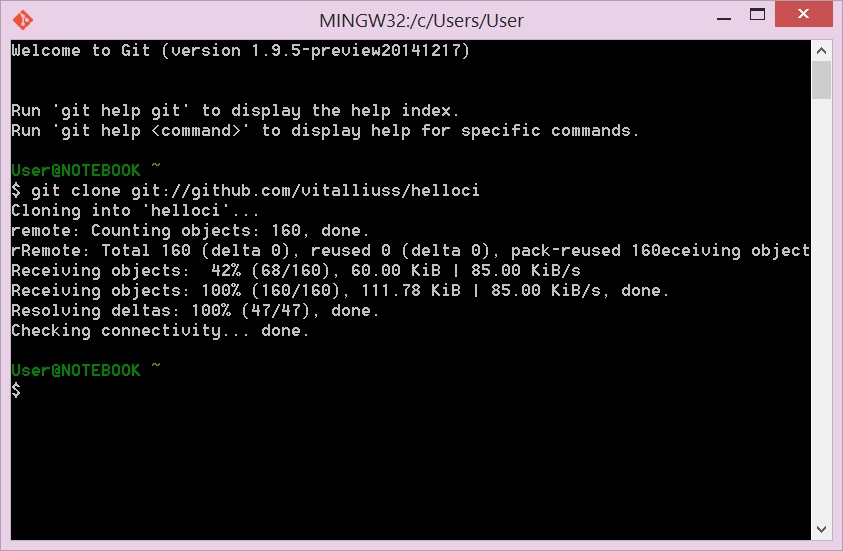
**$gitconfig –global user.email “UserEmail”**

**$ssh-keygen**– автоматически генерирует пару ssh-ключей в указанную либо дефолтовую директорию, создает папку .ssh, в которую помещает готовую пару ключей



**3.Склонировать существующий тестовый проект (используется команда gitclone, рекоммендуетсяgit протокол)**

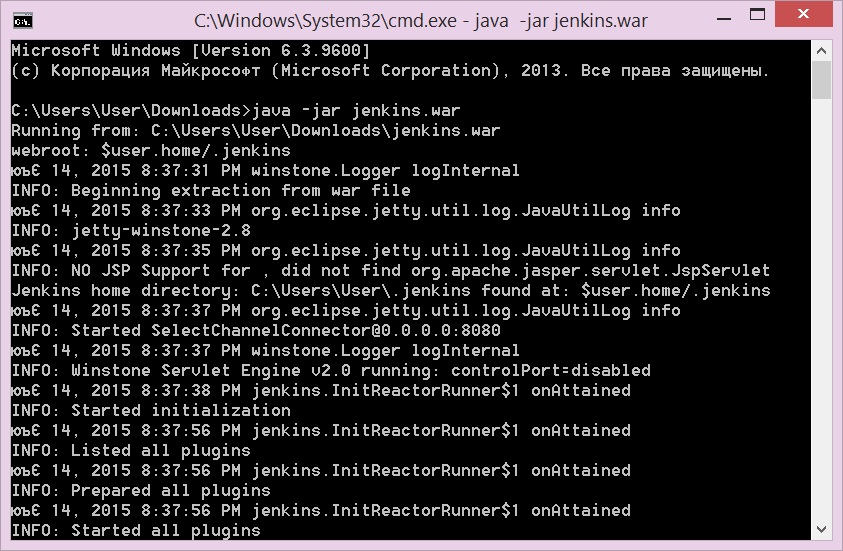
**$gitclonegit://link\_to\_the\_repo** – клонирует репозиторий в текущую директорию с использованием git-протокола



**4.Hастроить на своей или виртуальной машине Jenkins**

С официального сайта скачиваем .warфайл с Jenkins

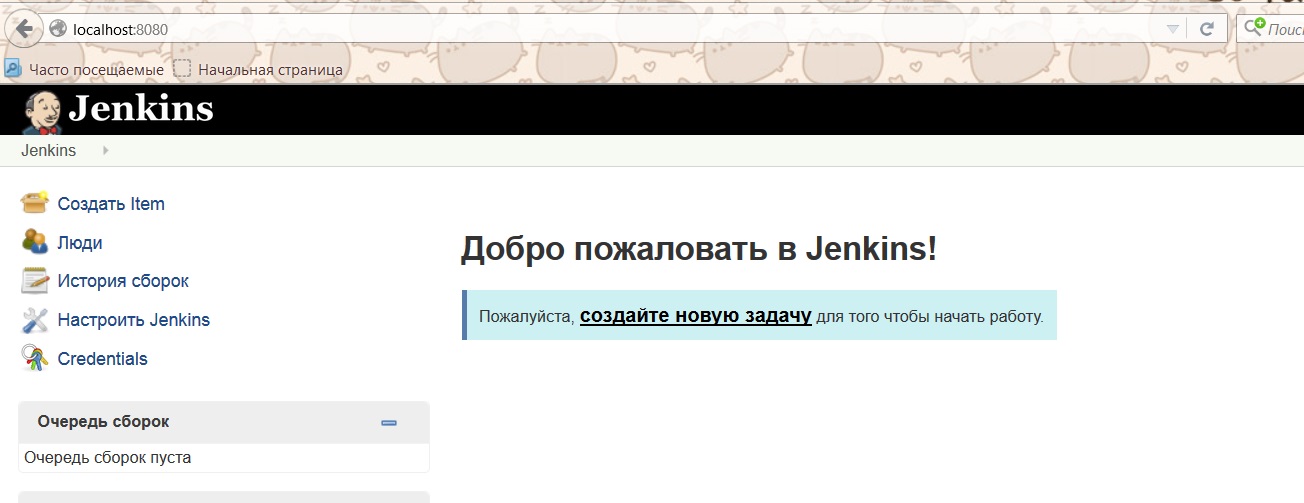
С помощью команды **java –jarJenkins.war**запускаем Jenkinsлокально под портом 8080



В адресную строку браузера вводим localhost:8080

Переходим в «Настроить Jenkins» >«Конфигурирование системы»

Прописываем имена и значения переменных окружения для Java (JAVA\_HOME) и Maven (MAVEN\_HOME)



**5. Создать job, который будет компилировать проект**

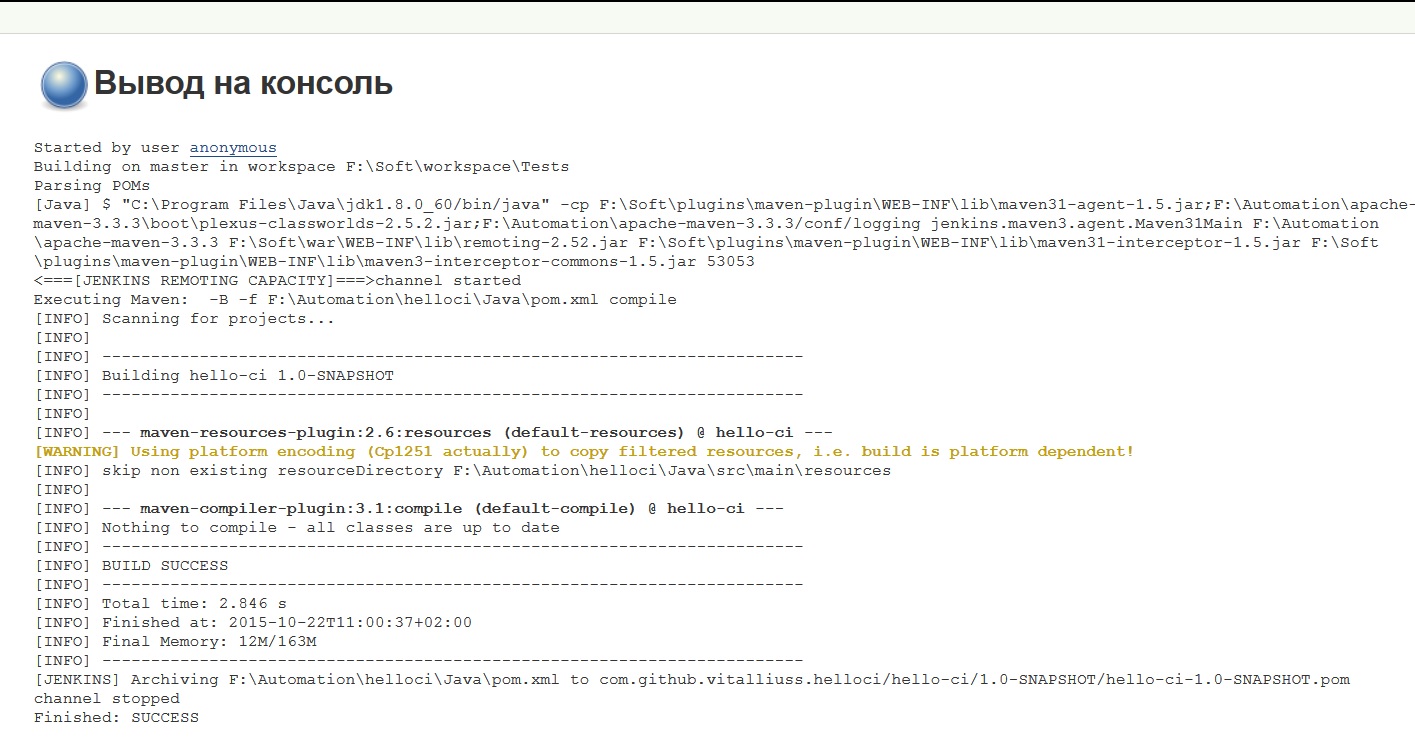
ЗаходимвJenkins> Создать Item

Вводим любое имя Item’а, выбираем опцию «Создать проект maven”

Заходим в настройки созданной джобы

В блоке «Сборка» в поле «Корневой POM» прописываем путь к pom.xml, лежащему в папке проекта (\helloci\Java\pom.xml), в поле goal добавляем «compile», сохраняем все изменения

Нажимаем кнопку «Собрать сейчас»



**Задача повышенной сложности №1: выполнить запуск тестов проекта helloci и опубликовать результаты в jenkins.**

ЗаходимвJenkins> Создать Item

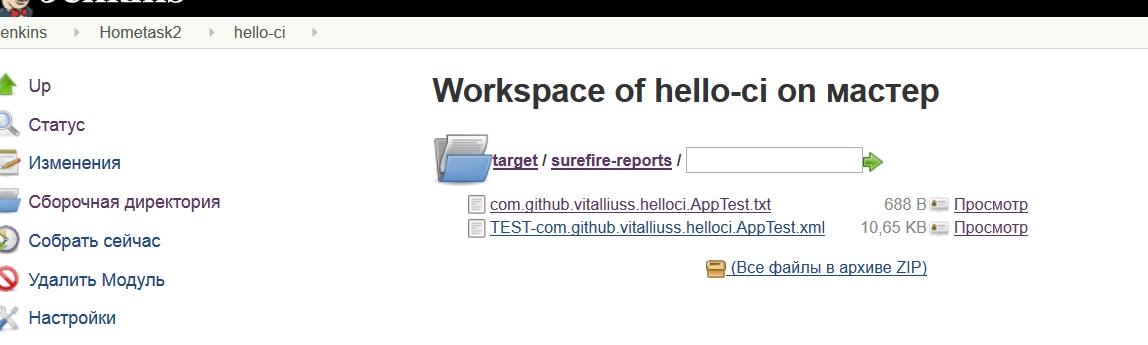
Вводим любое имя Item’а, выбираем опцию «Создать проект maven”

Заходим в настройки созданной джобы

В блоке «Сборка» в поле «Корневой POM» прописываем путь к pom.xml, лежащему в папке проекта (\helloci\Java\pom.xml), в поле goal добавляем «test», сохраняем все изменения

Нажимаем кнопку «Собрать сейчас»

Переходим в папку «Модули», выбираем наш проект (helloci), переходим в папку target > surefire-reports. В ней находятся 2 файла с результатами тестирования: xml и txt.



Содержимое txt файла:

-------------------------------------------------------------------------------

Test set: com.github.vitalliuss.helloci.AppTest

-------------------------------------------------------------------------------

Tests run: 5, Failures: 1, Errors: 0, Skipped: 1, Time elapsed: 0.109 sec <<< FAILURE! - in com.github.vitalliuss.helloci.AppTest

testShouldBeFailed(com.github.vitalliuss.helloci.AppTest) Time elapsed: 0.016 sec <<< FAILURE!

java.lang.AssertionError: null

at org.junit.Assert.fail(Assert.java:86)

at org.junit.Assert.assertTrue(Assert.java:41)

at org.junit.Assert.assertTrue(Assert.java:52)

at com.github.vitalliuss.helloci.AppTest.testShouldBeFailed(AppTest.java:21)

**Задача повышенной сложности №4: сделать все предыдущие задачи на linux машине.**

1. **Подготовить рабочее окружение (Java/.NET, Maven/MSBuild)**

Установка Java.

**$sudo apt-get update**

**$sudo apt-get install default-jre** – устанавливаемjre

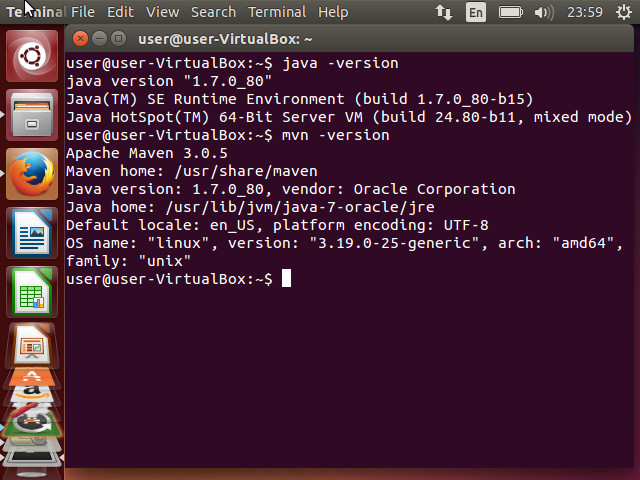
**$sudo apt-get install default-jdk** – устанавливаемjdk

**$gksudogedit /etc/environment** – с помощью редактора открываем файл с переменными окружения. Туда нужно добавить новую переменную JAVA\_HOME со значением, выведенным в консоль после запуска команды $java–version, в моем случае /usr/lib/jvm/java-7-oracle (аналогично для MAVEN\_HOME)

УстановкаMaven:

**$sudo apt-get update**

**$sudo apt-get maven** – устанавливаемMaven

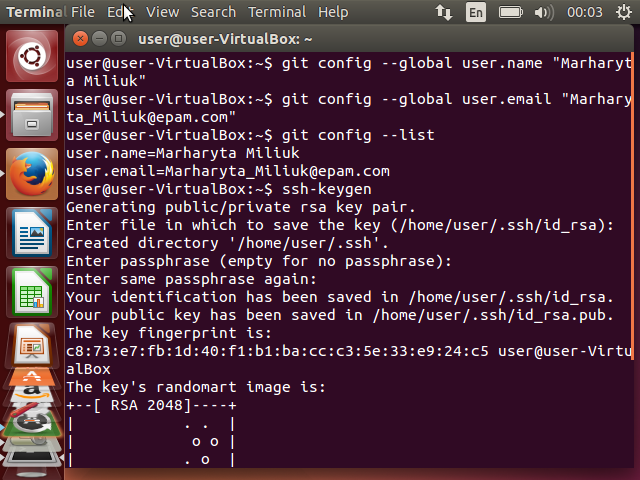


**2.Сгенерировать SSH ключи (если у вас нет существующих)**

**$gitconfig –global user.name “UserName”**

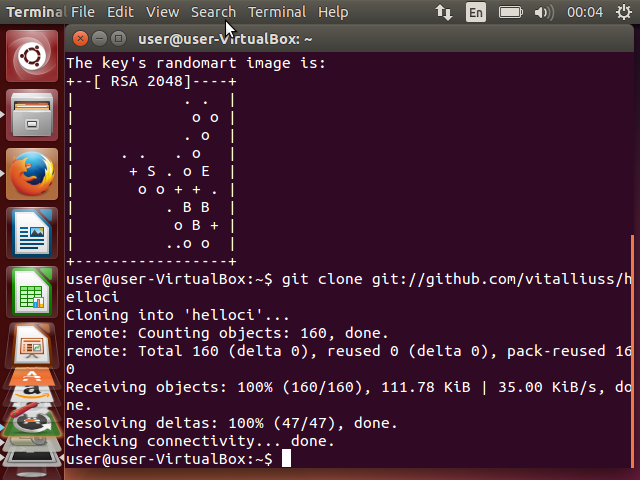
**$gitconfig –global user.email “UserEmail”**

**$ssh-keygen**– автоматически генерирует пару ssh-ключей в указанную либо дефолтовую директорию, создает папку .ssh, в которую помещает готовую пару ключей



**3.Склонировать существующий тестовый проект**

**$gitclonegit://link\_to\_the\_repo** – клонирует репозиторий в текущую директорию с использованием git-протокола



**4.Hастроить на своей или виртуальной машине Jenkins**

**$sudo apt-get update**

**$sudo apt-get install Jenkins** – скачиваетиустанавливаетJenkins

В адресную строку браузера вводим localhost:8080

Переходим в «Настроить Jenkins» >«Конфигурирование системы»

Прописываем имена и значения переменных окружения для Java (JAVA\_HOME) и Maven (MAVEN\_HOME)

