1. Установка и настройка SDK

Необходимо поставить SDK(на данный момент установлена SDK 23.0.1)

<https://dl.google.com/android/android-sdk_r23.0.1-linux.tgz>

Установка и настройка компонентов и библиотек ведется на CentOS(необходимо устанавливать компоненты SDK именно на той OS на которой вы будете вести сборку, иначе возникнут ошибки разного рода - КРИТИЧНО):

Cd /opt/android-sdk../tools/ - переходим в папку с SDK в tools

./android – данная команда вызовет визуальное меню, где можно выбрать компоненты для SDK.

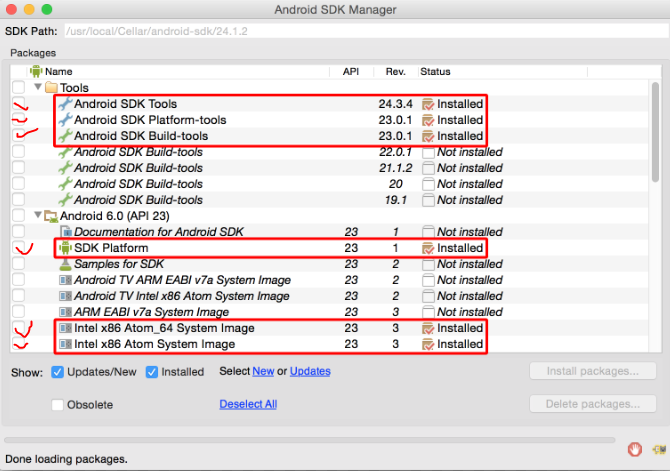
Необходимо доставить-

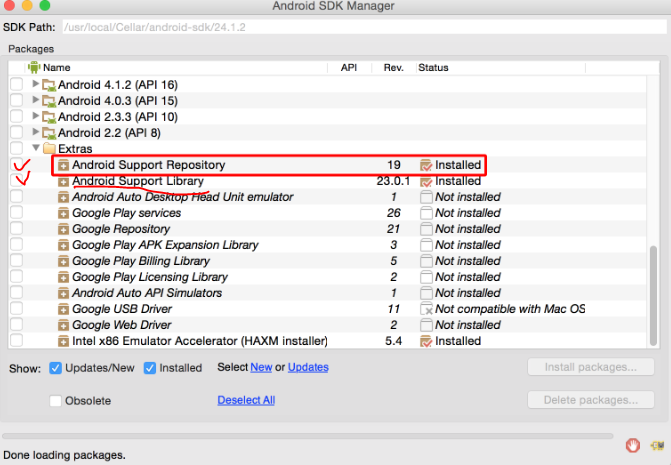
1.       **Android Support Repository**, если такой нету, значит переименовали в **Local Maven repository for Support libraries**

1. **Android SDK Platform-tools**
2. **Android SDK Build-tools**

4.       Скачать **API**

Пример установки компонентов в картинках –





1. **Устанавливаем plugins в Jenkins**
   1. Создаем проект и называем его(пример myc-pipeline-android)
   2. Необходимо поставить плагины, для сборки Android:

[Android Lint Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Android+Lint+Plugin) – для проверки кода на наличие ошибок.

[Checkstyle Plug-in](http://wiki.jenkins-ci.org/x/GYCGAQ)– для проверки кода на наличие ошибок.

[FindBugs Plug-in](http://wiki.jenkins-ci.org/x/GYAs)– для проверки кода на наличие ошибок.

[PMD Plug-in](http://wiki.jenkins-ci.org/x/GAAHAQ)– для проверки кода на наличие ошибок.

[Git plugin](http://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin) – для выкачивания исходников сборки из GIT.

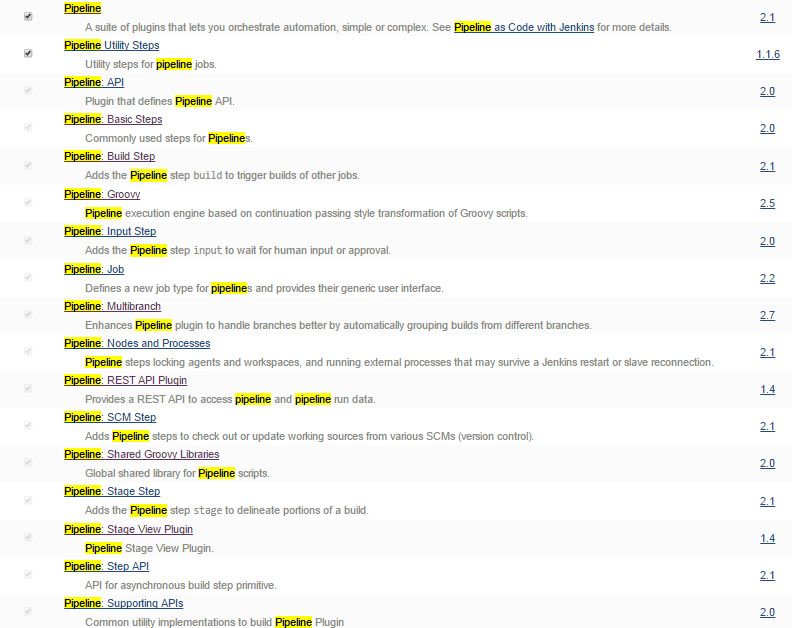
[JUnit Plugin](http://wiki.jenkins-ci.org/display/JENKINS/JUnit+Plugin) – для самописных тестов.

[Artifactory Plugin](http://wiki.jenkins-ci.org/display/JENKINS/Artifactory+Plugin) – для добавления сборки(.apk) в локальный Artifactory.

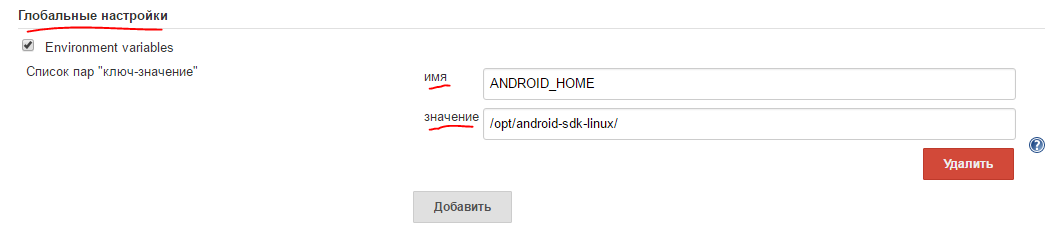
[Gradle plugin](http://wiki.jenkins-ci.org/display/JENKINS/Gradle+Plugin) – собственно через этот плагин мы и будем собирать нашу сборку, т.к. весь код написан на Gradle.

Для полноценной работы pipeline необходимо доставить кучу плагинов:

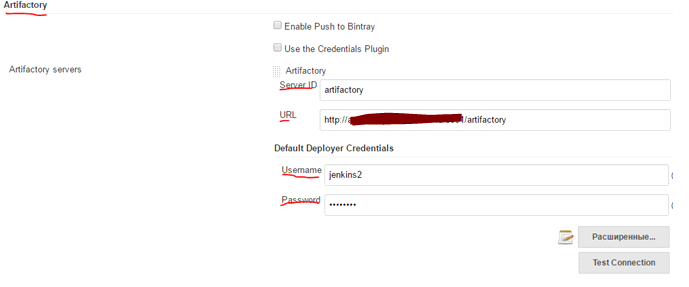
[Build Pipeline Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Build+Pipeline+Plugin) – для конвеерной сборки.



1. **Настройка plugins в Jenkins**
   * 1. Указываем путь до SDK

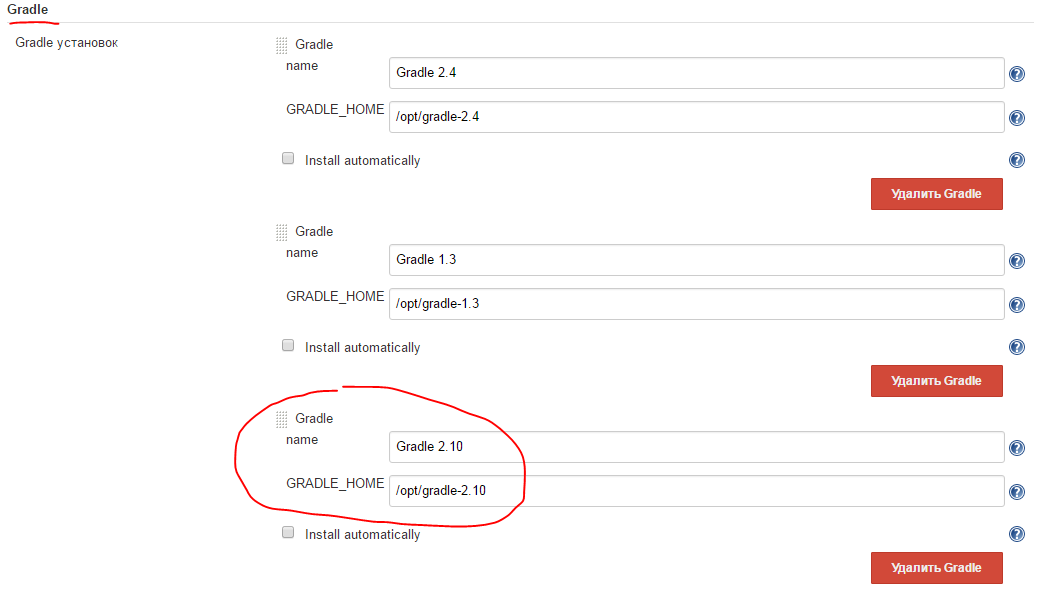


* + 1. Конфигурируем Artifactory



* 1. Конфигурируем Gradle

НО, перед этим необходимо скачать нужную версию самого Gradle(на данный момент 2.10) <https://downloads.gradle.org/distributions/gradle-2.10-all.zip> и положить на сервер.



1. **Пишем код Pipeline для сборки**

Выглядит он так:



Полный код:

node { //Можно указать на каком узле собираться данному проекту(например: node master {) и сборка будет происходить только на данной ноде.

catchError { //отлавливает “сломанный” билд

stage 'build' //Для разделение сборки на части - “сборка”

checkout([$class: 'GitSCM', branches: [[name: '\*/jenkins-1']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[url: 'ssh://git@NAME:PORT/myc/android.git']]]) //указываем наш git

def gradle = tool name: 'Gradle 2.10', type: 'hudson.plugins.gradle.GradleInstallation' //указываем версию gradle для сборки проекта

env.ANDROID\_HOME = '/opt/android-sdk-linux/' //указываем дом. Директорию android для сборки.

sh "cd MyCredit && ${gradle}/bin/gradle check assemble" //указываем путь до проекта(cd MyCredit), потом до gradle(${gradle}/bin/gradle) , затем указываем, таски(check assemble)

echo 'build' // конец сборки

stage 'check' //новая часть сборки “проверка”

step([$class: 'LintPublisher', canComputeNew: false, defaultEncoding: '', healthy: '', pattern: '', unHealthy: '']) //указываем, что используем для проверки Lint

step([$class: 'CheckStylePublisher', canComputeNew: false, defaultEncoding: '', healthy: '', pattern: '\*\*/checkstyle.xml', unHealthy: '']) //указываем, что используем для проверки Checkstyle

step([$class: 'FindBugsPublisher', canComputeNew: false, defaultEncoding: '', excludePattern: '', healthy: '', includePattern: '', pattern: '', unHealthy: '']) //указываем, что используем для проверки FindBugs

step([$class: 'PmdPublisher', canComputeNew: false, defaultEncoding: '', healthy: '', pattern: '', unHealthy: '']) //указываем, что используем для проверки PMD

echo 'check' //конец “проверки”

stage 'uploadArtifactory' //upload сборки на artifactory(в нашем случае .apk)

def server = Artifactory.newServer('http://NAME:PORT/artifactory/', 'login', 'password') //указываем путь до нашего артифактори, далее указываем логин и пасс

def uploadSpec = """{

"files": [

{

"pattern": "MyCredit/app/build/outputs/apk/\*.apk", //указываем путь до наших артефактов

"target": "gradle-local/" //указываем репозиторий, куда положить наши артефакты

}

]

}"""

def buildInfo = server.upload(uploadSpec) //данная строчка служит для заливки(Upload) артефактов

server.publishBuildInfo(buildInfo) //данная строчка отобразит информация в Jenkins на артифактори

echo 'uploadArtifactory' //конец деплоя артефактов

} //данная скобка обозначает конец catchError

step([$class: 'Mailer', notifyEveryUnstableBuild: true, recipients: 'NAME@DOMAIN, sendToIndividuals: true]) //отправляет на мыло сообщение о неудачной сборке.

}