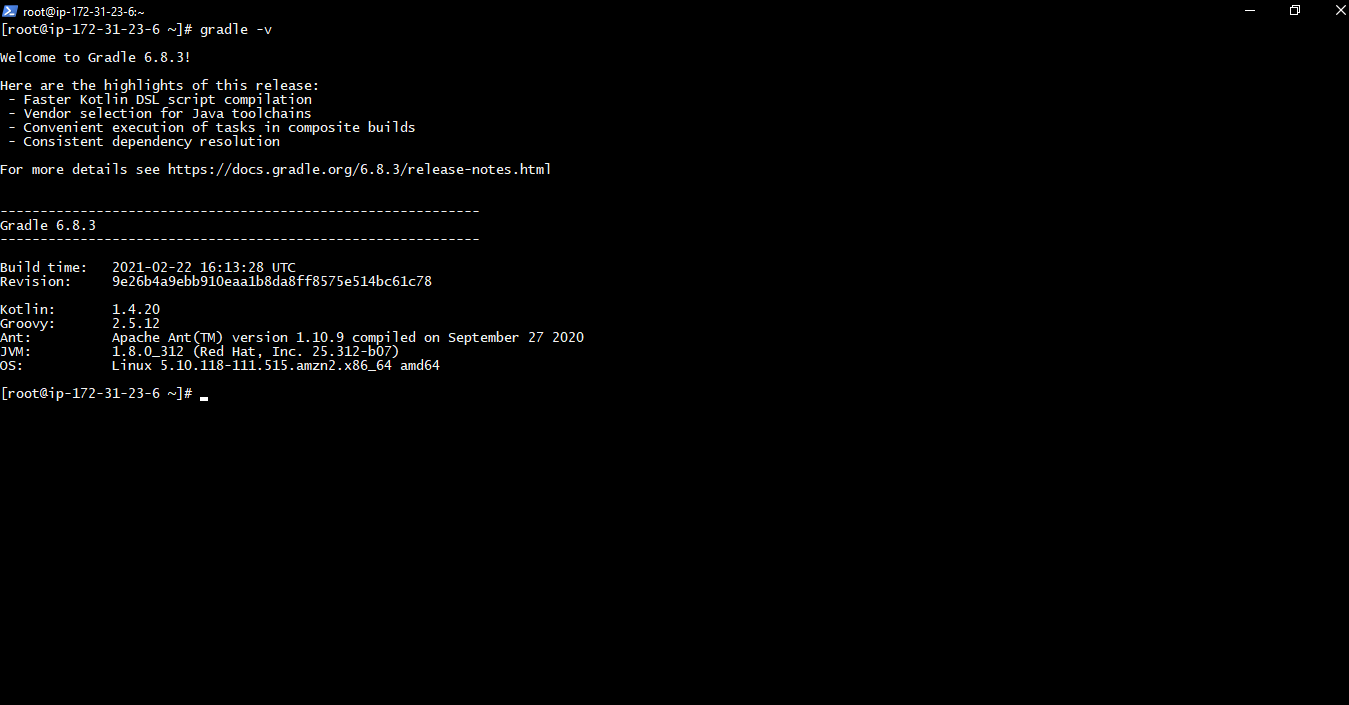
**Integration of Android with Jenkins**

1. **Install Jenkins Dependencies**
2. **Install Gradle**
3. **Configure Gradle**
4. **Fixed it with**
5. **wget** [**https://services.gradle.org/distributions/gradle-6.8.3-bin.zip**](https://services.gradle.org/distributions/gradle-6.8.3-bin.zip)

* **unzip -d /opt/gradle gradle-6.8.3-bin.zip**
* **nano .bash\_profile**
* **GRADLE=** /opt/gradle/gradle-6.8.3/bin (Save it)
* **gradle -v**



Run Jenkins on Browser

[**http://publicIp:8080**](http://publicIp:8080)

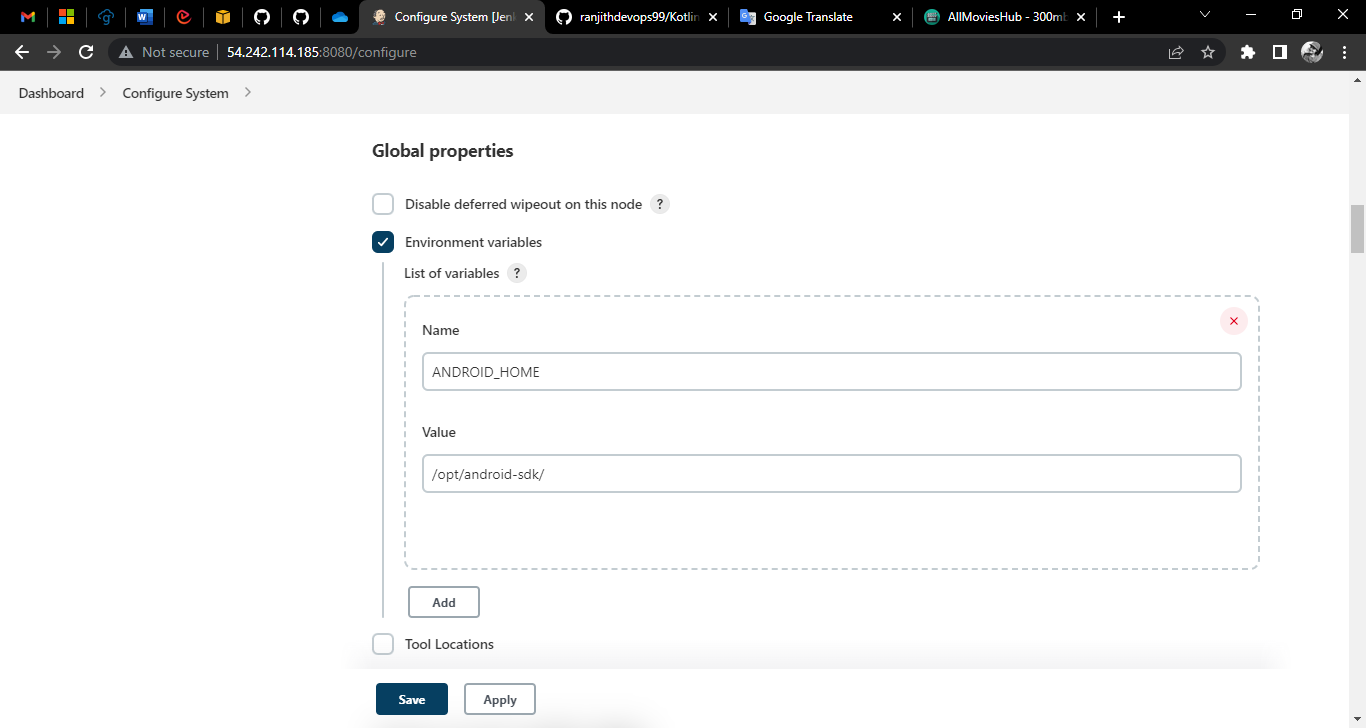
**Create an Android Pipeline**

**For that, we need to install android sdk**

* **cd /opt**
* **wget** [**https://dl.google.com/android/repository/sdk-tools-linux-4333796.zip**](https://dl.google.com/android/repository/sdk-tools-linux-4333796.zip)
* **unzip sdk-tools-linux-4333796.zip -d android-sdk**
* **export ANDROID\_SDK\_ROOT=/opt/android-sdk**
* **echo $ANDROID\_SDK\_ROOT**
* **sudo chmod 777 android-sdk/**
* **cd /opt/android-sdk/tools/bin**
* **/opt/android-sdk/tools/bin/sdkmanager --update**
* **/opt/android-sdk/tools/bin/sdkmanager --licenses**

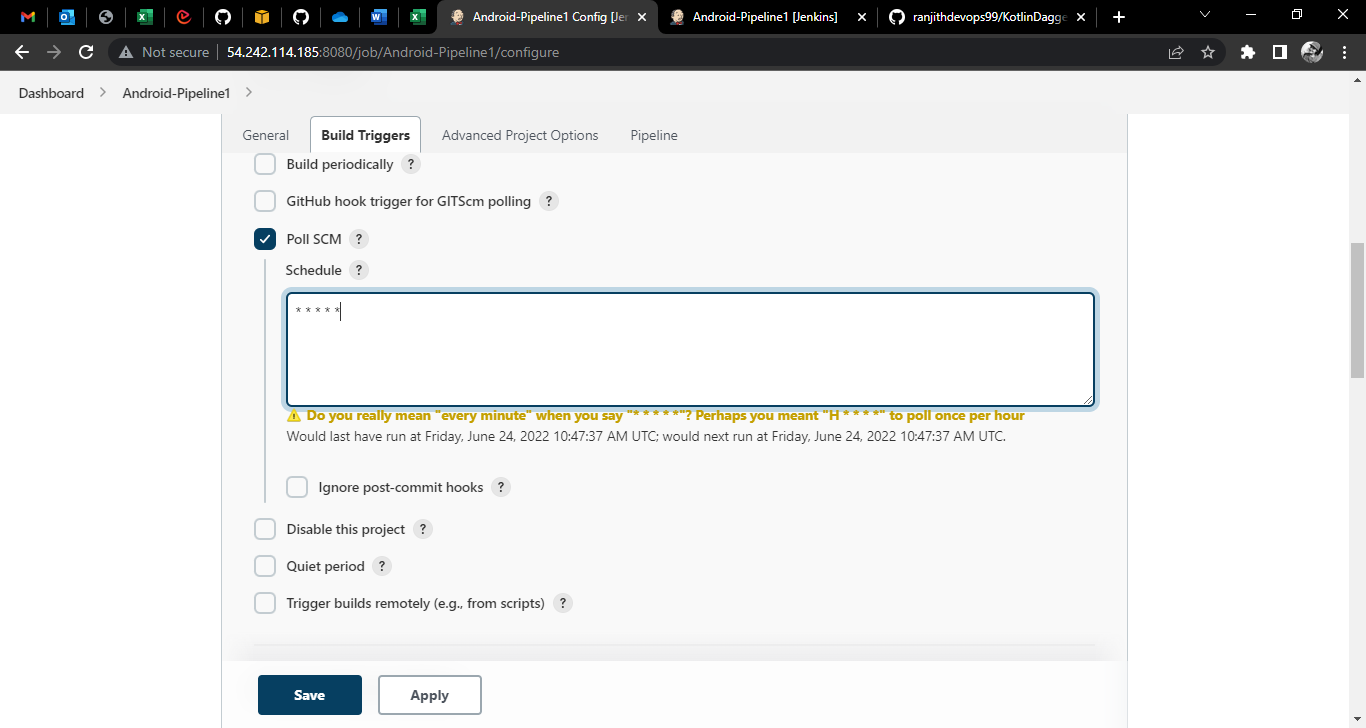
We have to set the android SDK path environment variable path in both Jenkins configuration and Linux Machine

**Jenkins Server -> Manage Jenkins -> Configure System -> Global Properties -> Environment variables**

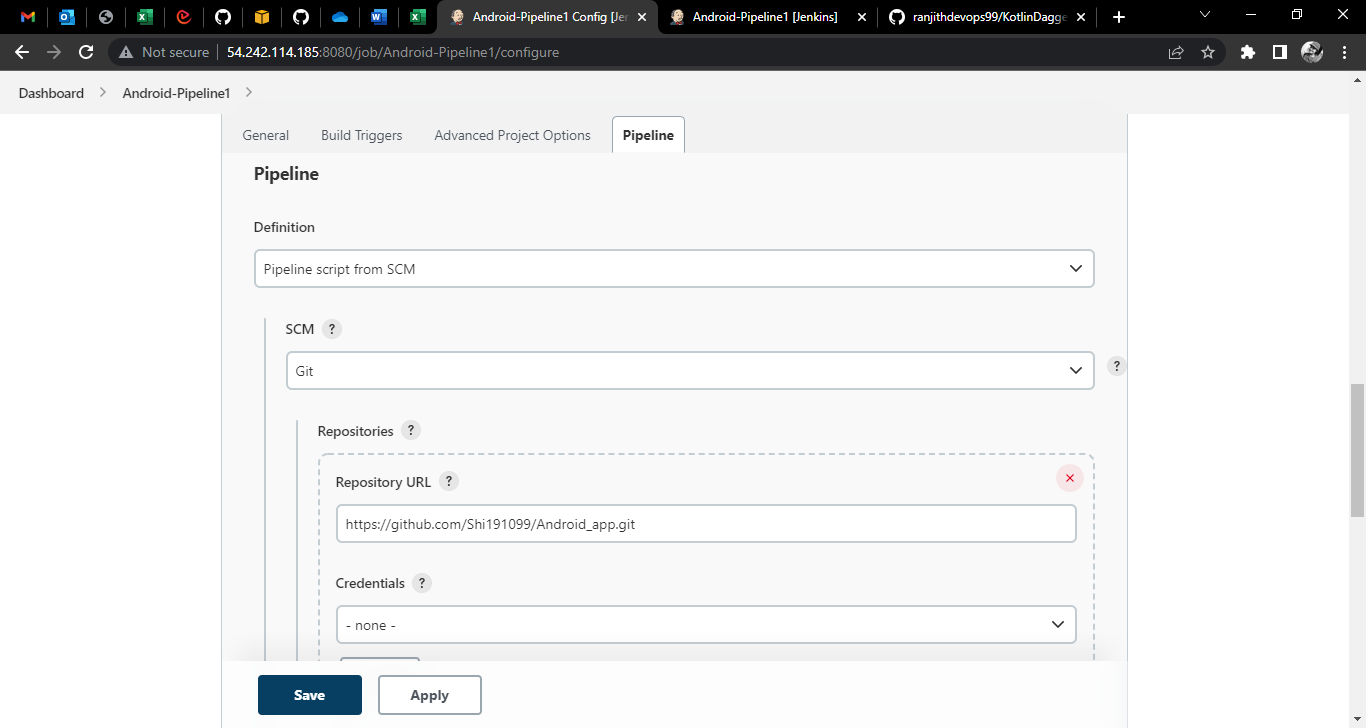
**Restart Jenkins**, and **Linux Machine**

Now create a Pipeline,

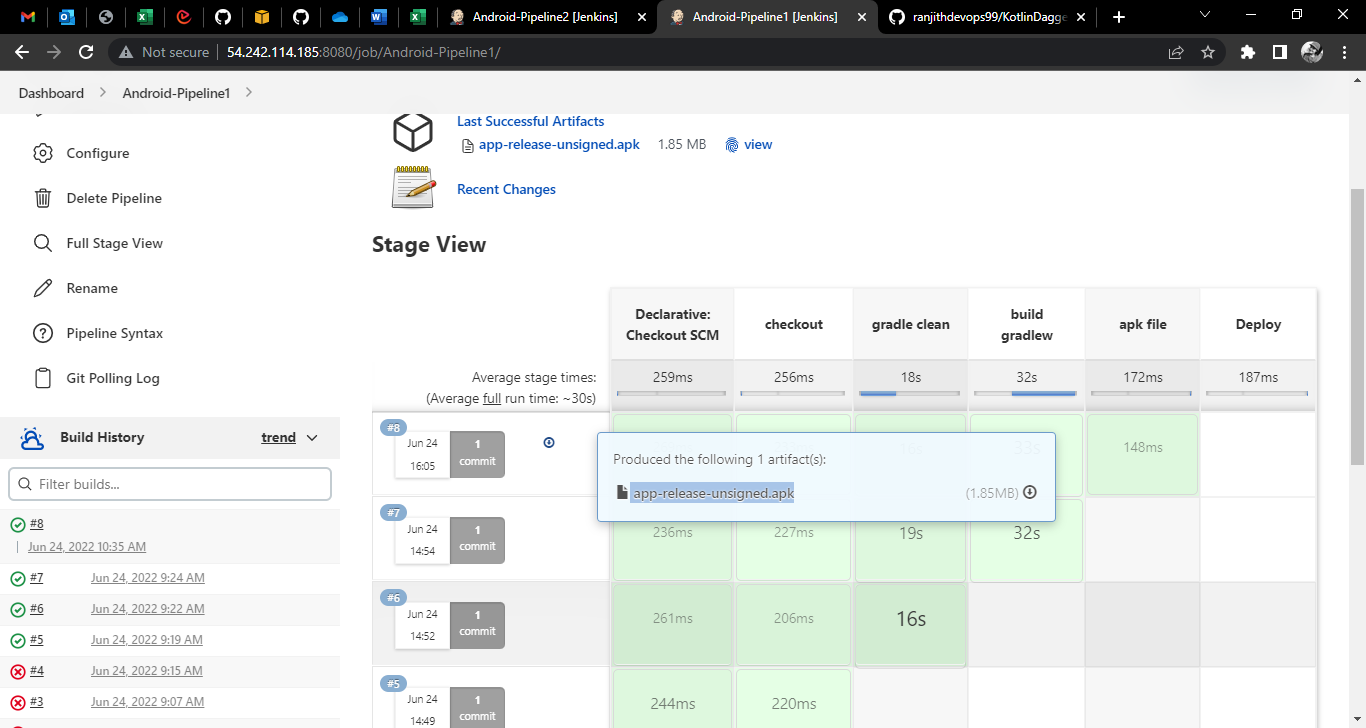
* Give a Name
* Select Pipeline Job
* Goto Build Triggers (when Developer update changes it will automatically trigger)



* GOTO Pipeline
* Select Pipeline script from SCM
* Select Git
* Give Repository URL
* If repo is private then give credentials
* Give branch to build
* Give script path as **Jenkins File**
* **Apply and Save**



Build It

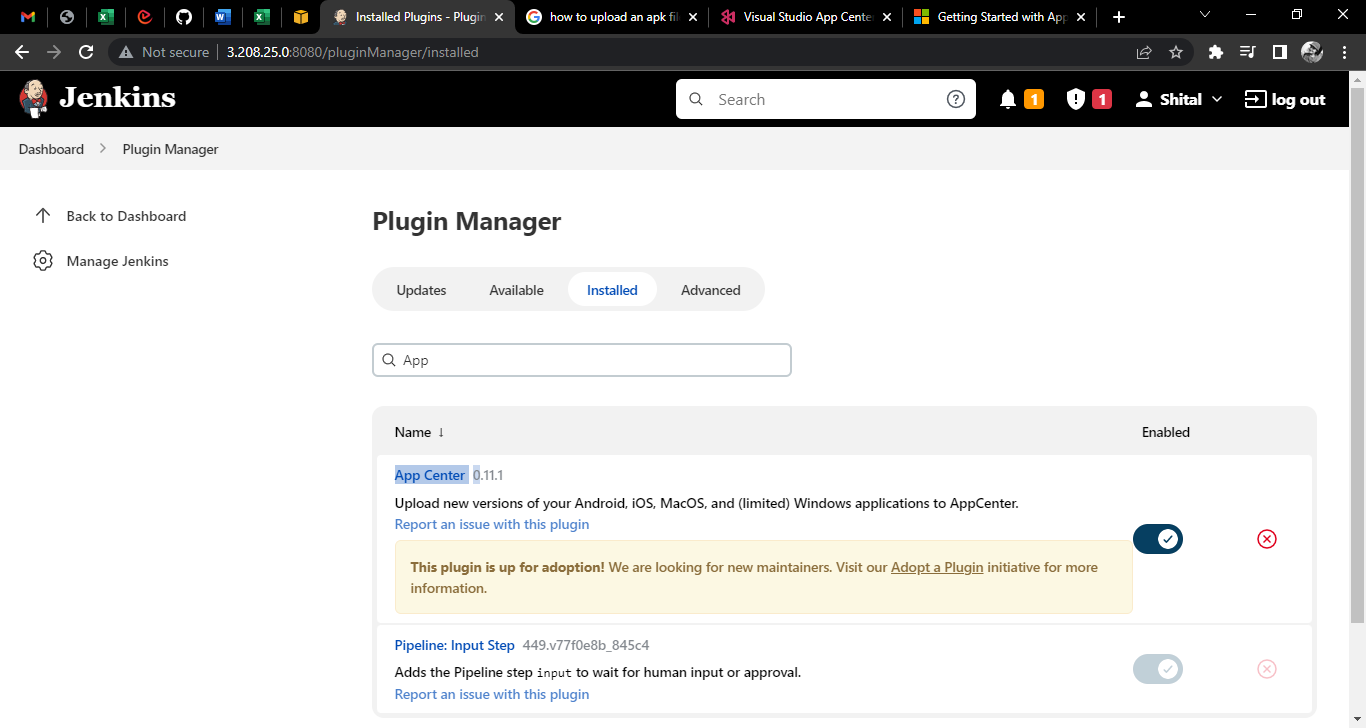


Here I got .apk file, now have to deploy on App Center,

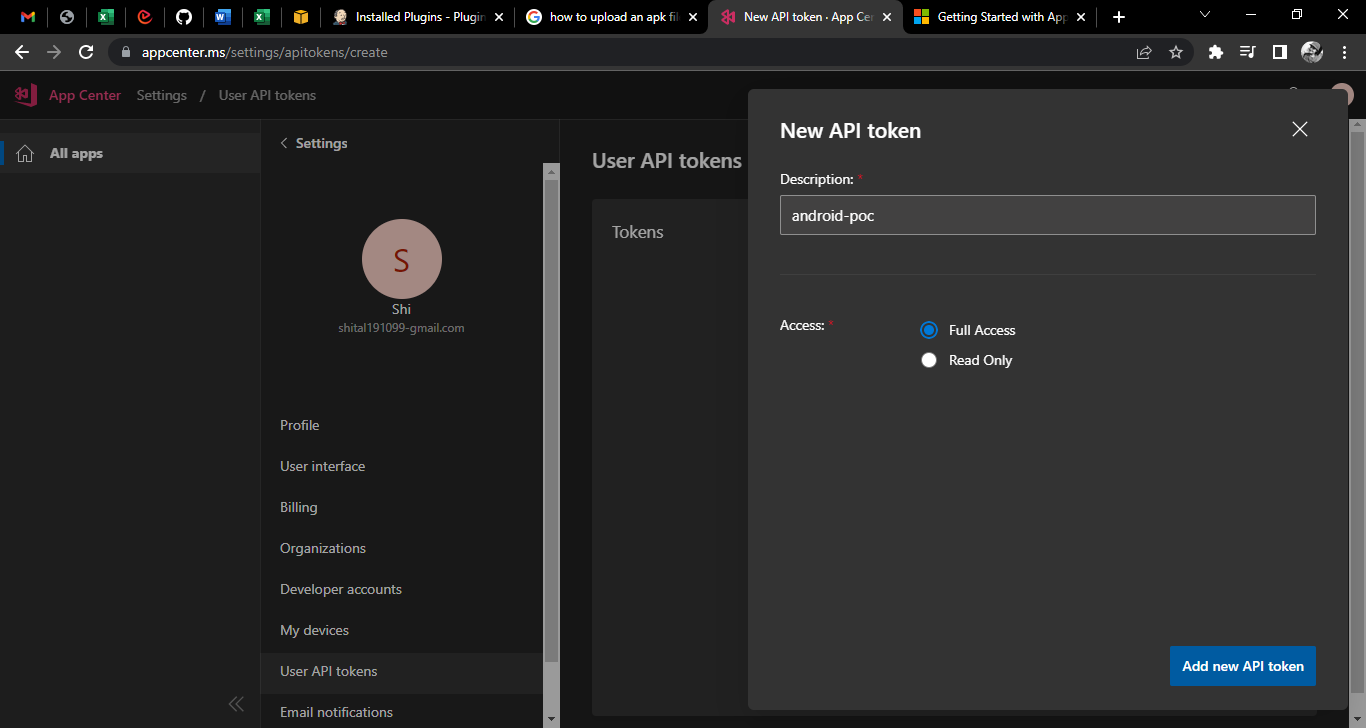
Now Install SonarQube and set Configuration,

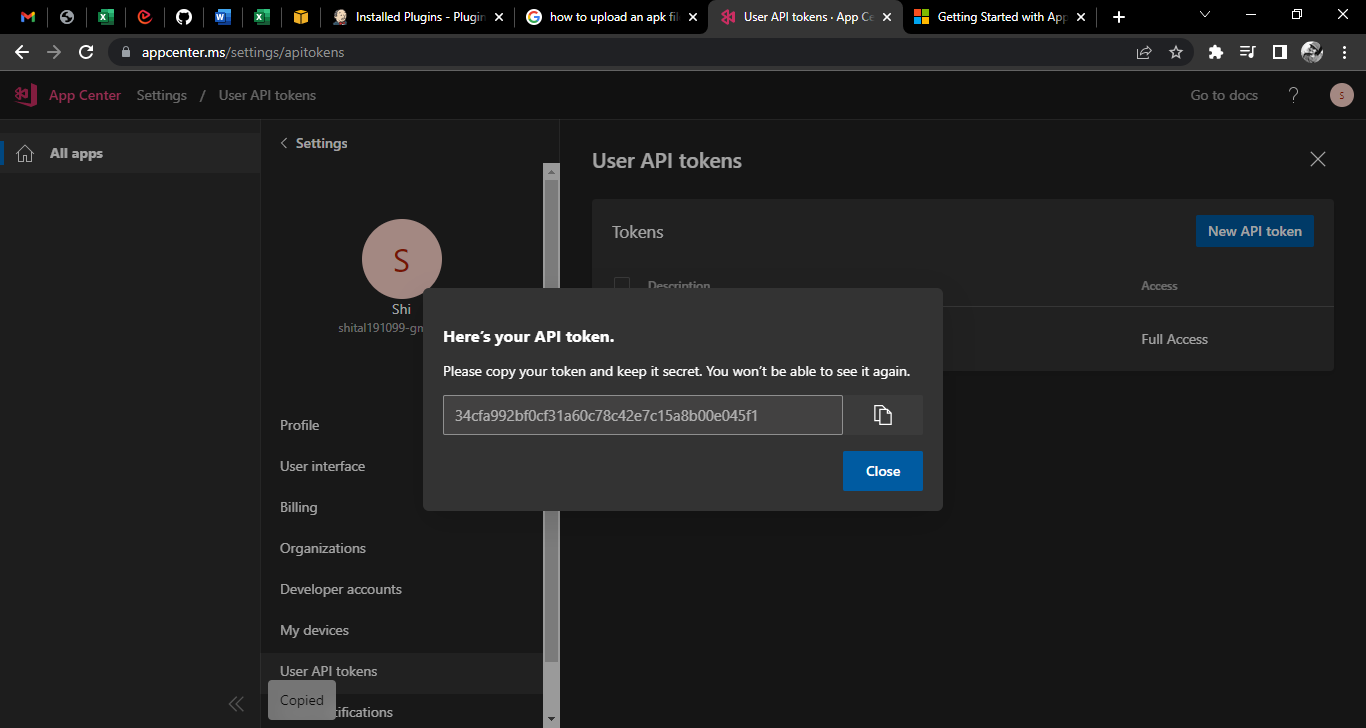
Install plugin App Center

**Manage Jenkins --> Manage Plugins --> Available --> app Center**

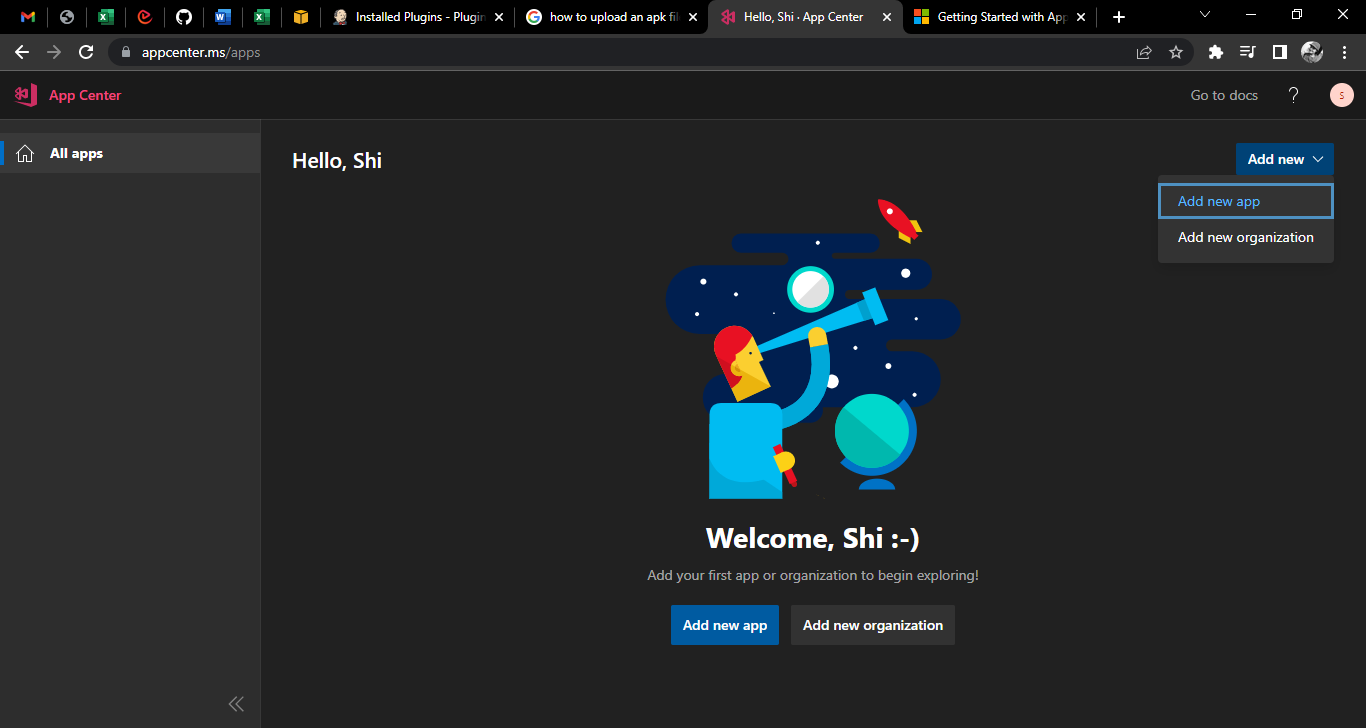


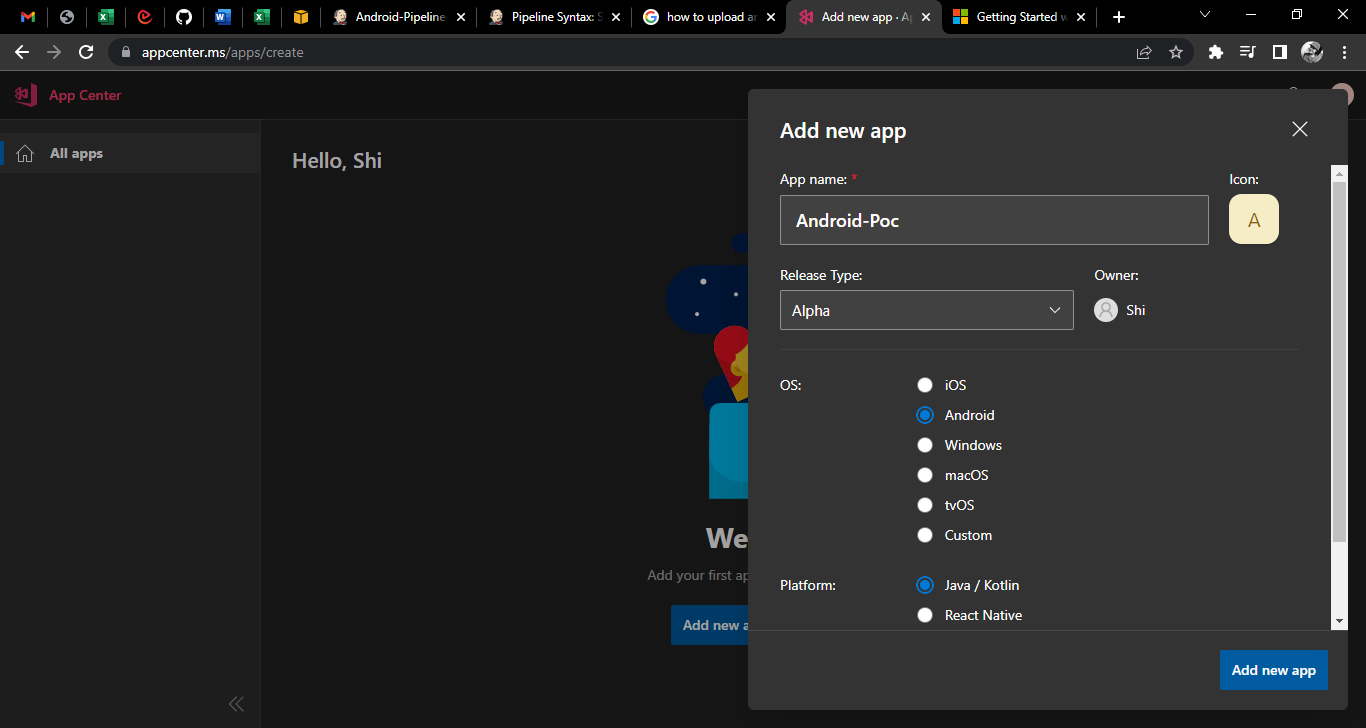
Login to app center --> Go to Accounts --> Copy user name for Plugin --> Generate a Token





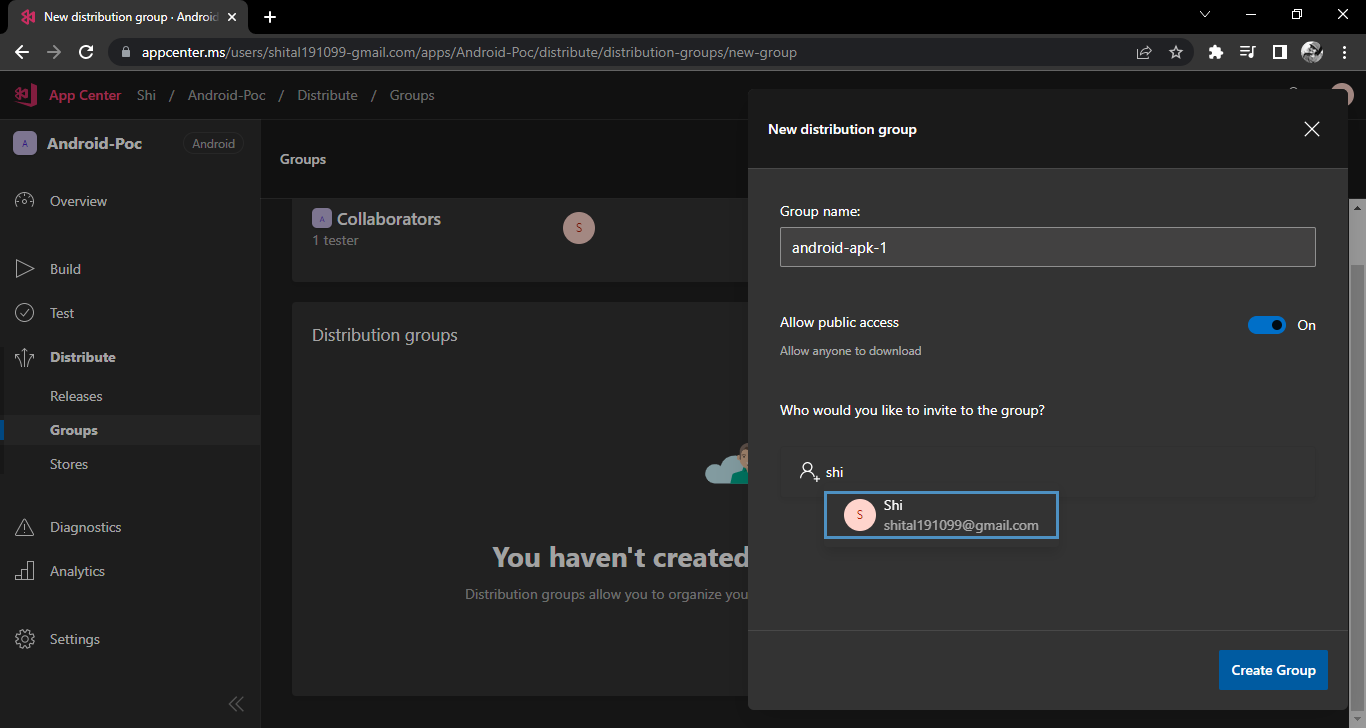
Add New --> Add New App





Now Create a Group

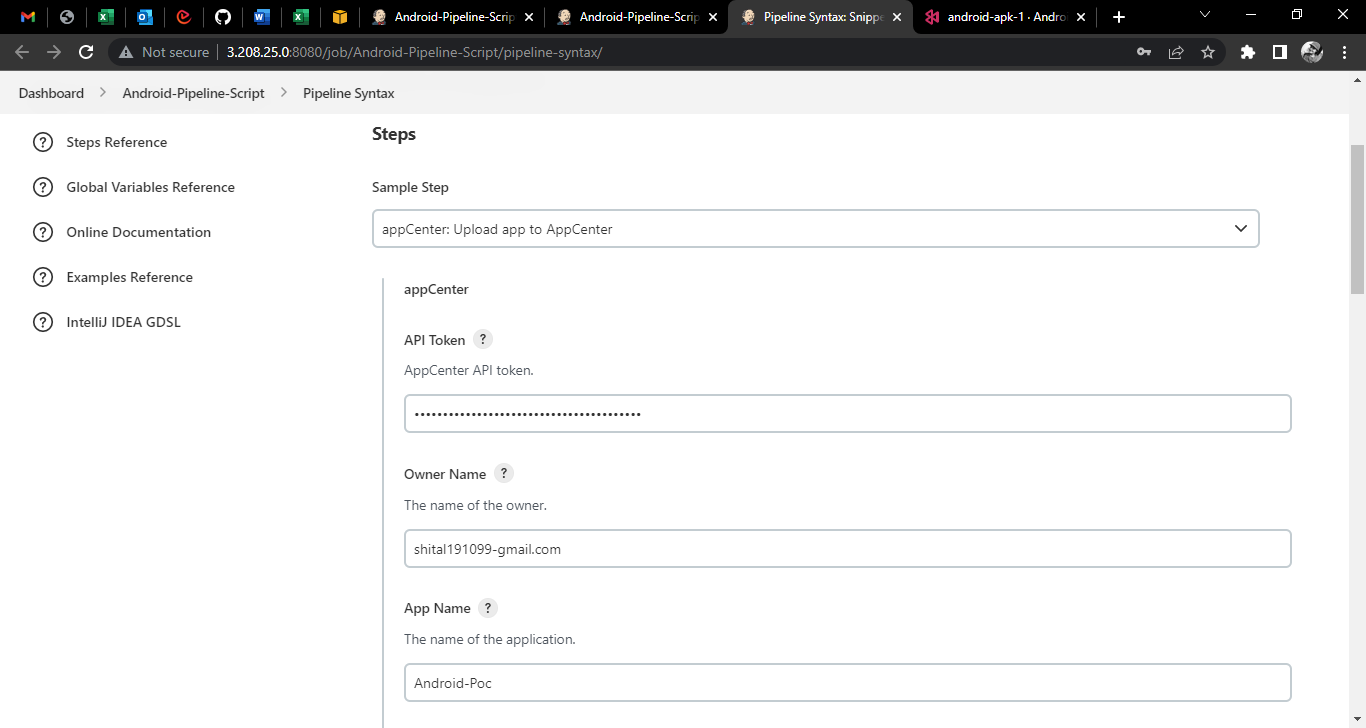
**App --> Distributes --> Groups --> Add Group**

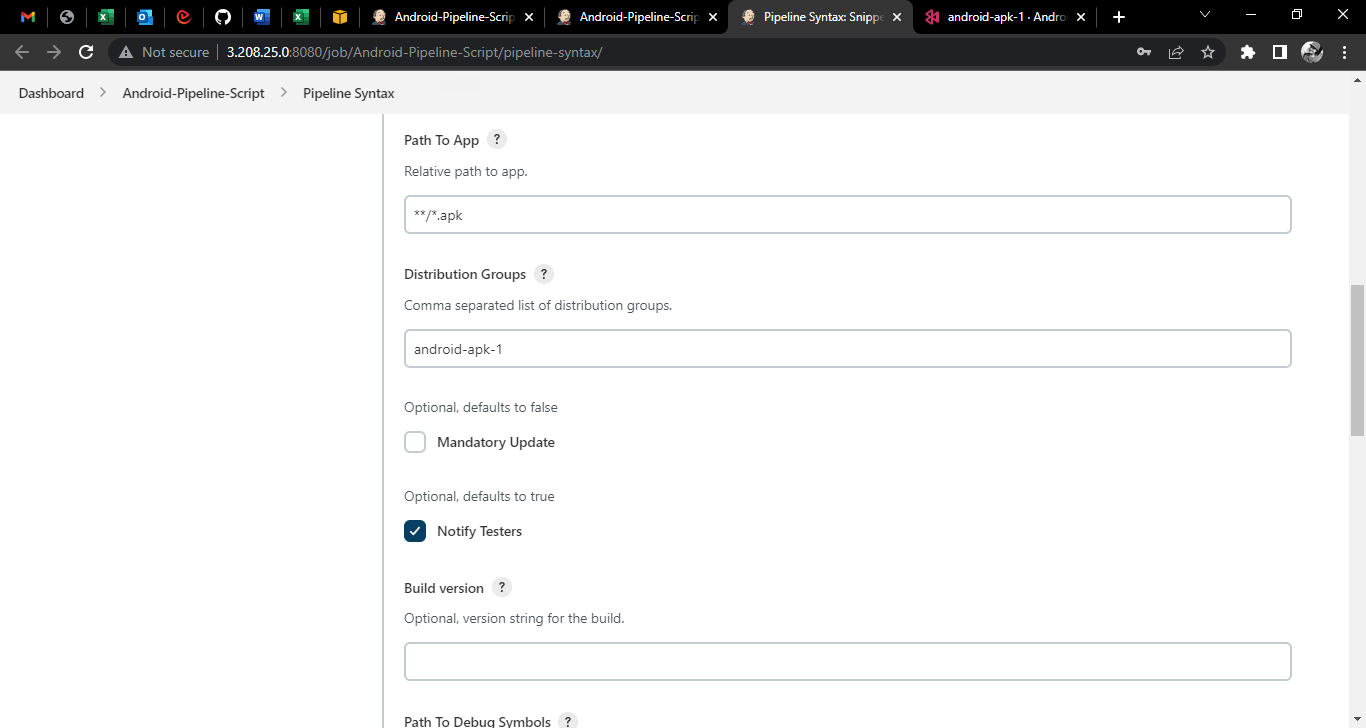


Then Go to Pipeline and click on **Pipeline Syntax**

Select AppCenter --> Upload app to **app Cente**r

**API Token ---> Owner Name ---> App Name ---> Group Name**



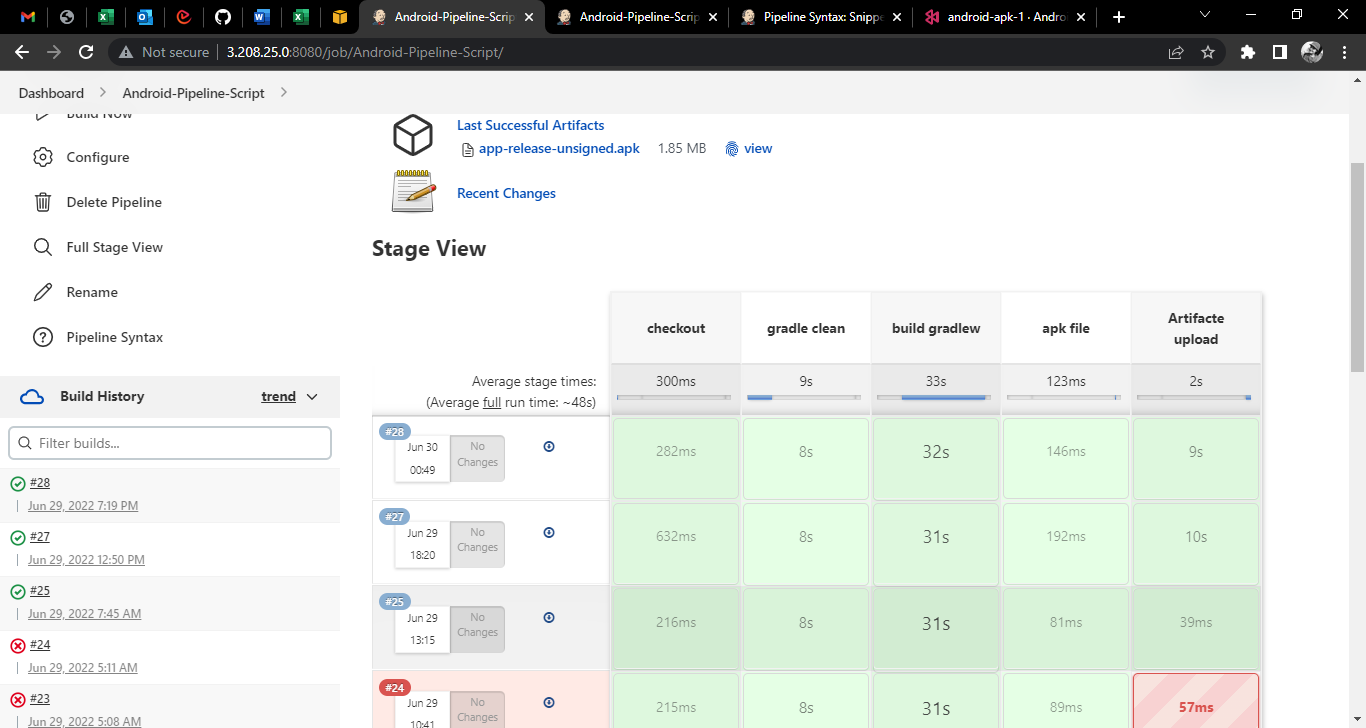


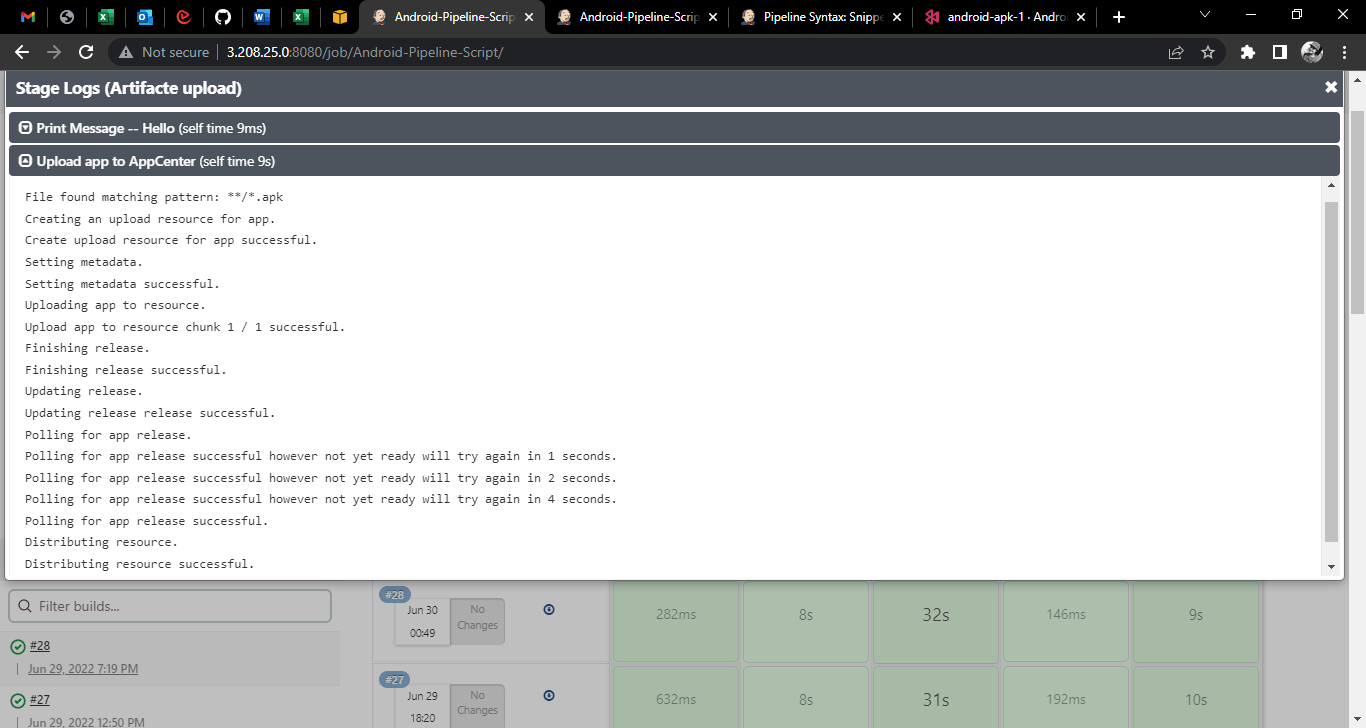
**Generate Pipeline Script**

|  |
| --- |
| **Jenkins File** **pipeline {**  **agent any**    **stages {**  **stage('checkout') {**  **steps {**  **checkout([$class: 'GitSCM', branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Shi191099/Android\_app.git']]])**  **}**  **}**  **stage('gradle clean'){**  **steps {**  **sh './gradlew clean'**  **}**  **}**  **stage('build gradlew') {**  **steps {**  **sh './gradlew assembleRelease'**  **}**  **}**  **stage('apk file') {**  **steps {**  **archiveArtifacts artifacts: '\*\*/\*.apk', fingerprint: true, onlyIfSuccessful: true**  **}**  **}**  **stage('Artifacte upload') {**  **steps {**  **echo 'Hello'**  **appCenter apiToken: '34cfa992bf0cf31a60c78c42e7c15a8b00e045f1',**  **appName: 'Android-Poc',**  **distributionGroups: 'android-apk-1',**  **mandatoryUpdate: false,**  **notifyTesters: true,**  **ownerName: 'shital191099-gmail.com',**  **pathToApp: '\*\*/\*.apk'**  **}**  **}**  **}**  **}** |

**Apply and Save**

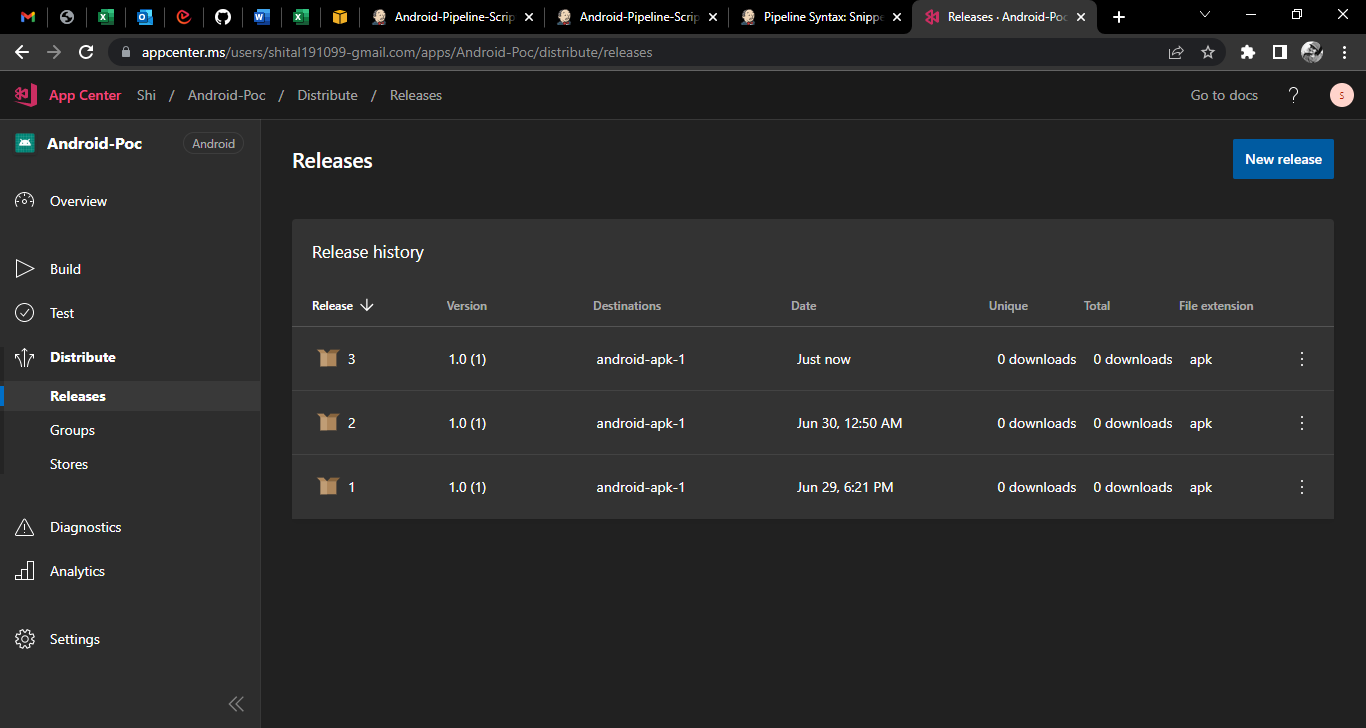
**Build Now**





Now check this apk is uploaded on app Center or not,

Goto **App Center --> Android-Poc --> Distribute --> Release**



Now check on Linux Machine it’s deployed there or not,

So Open Terminal,

**cd /var/lib/jenkins/workspace/Android-Pipeline-Script/app/build/outputs/apk/release**

