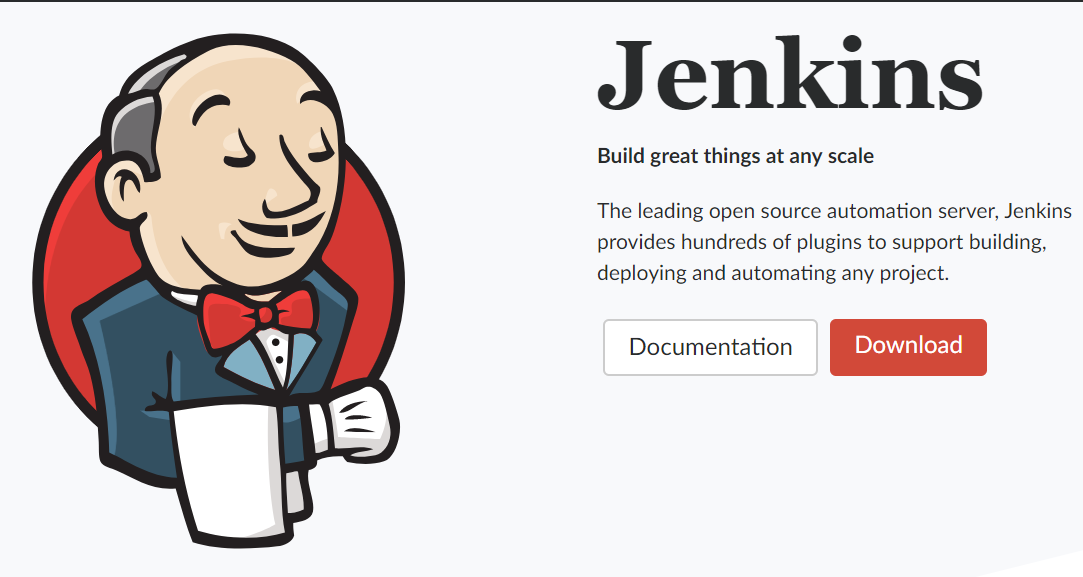
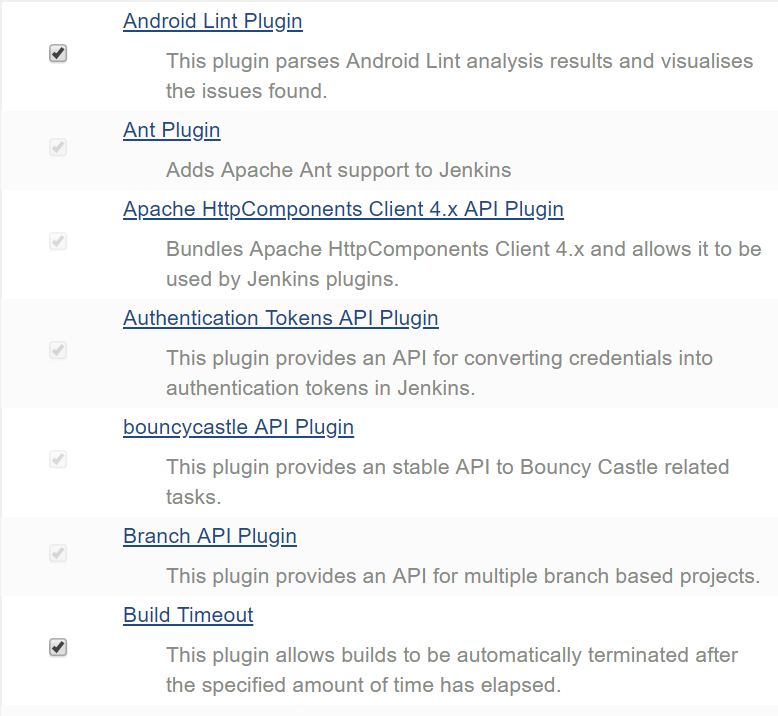
Steps to create framework

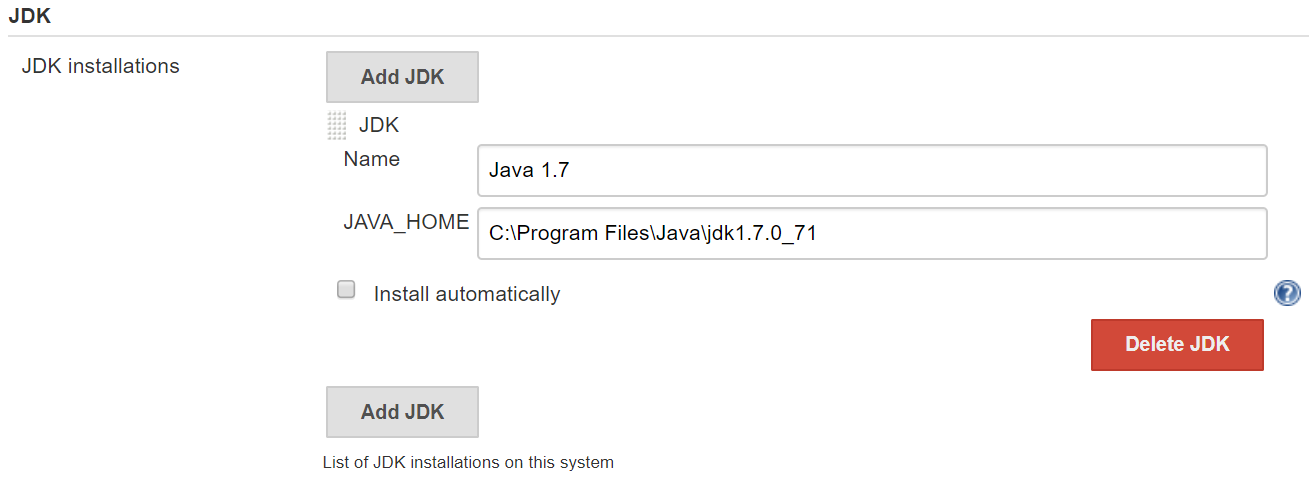
1. Download Jenkins



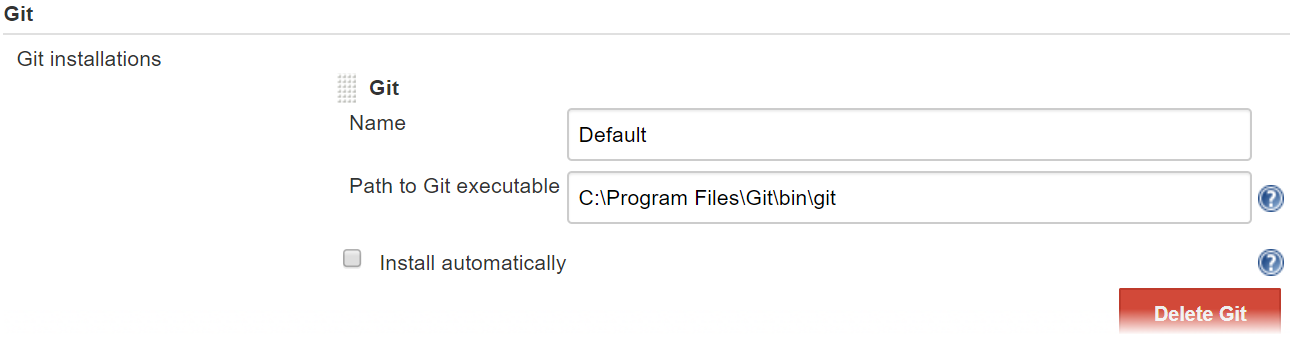
1. Execute the downloaded EXE
2. Open localhost:8080 (or whichever port was chosen)
3. Go do Manage Jenkins -> Manage Plugins
4. Click on the “Installed” tab and verify the following plugins are present
   1. Git
   2. Gradle
   3. Android Lint
   4. Google Play Android Publisher



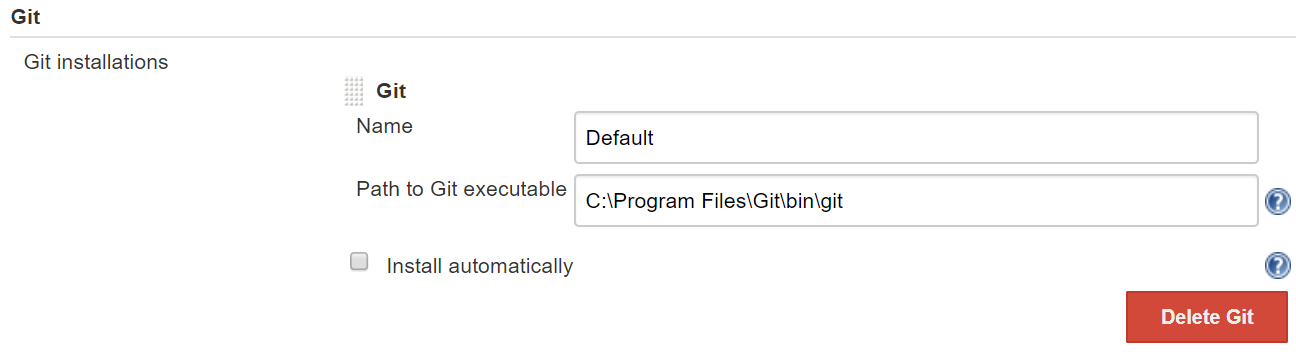
1. If they are not present, click the “Available” tab and download them
2. Go back and click Global Tool Configuration
3. Click “Add JDK” and call it Java 1.7



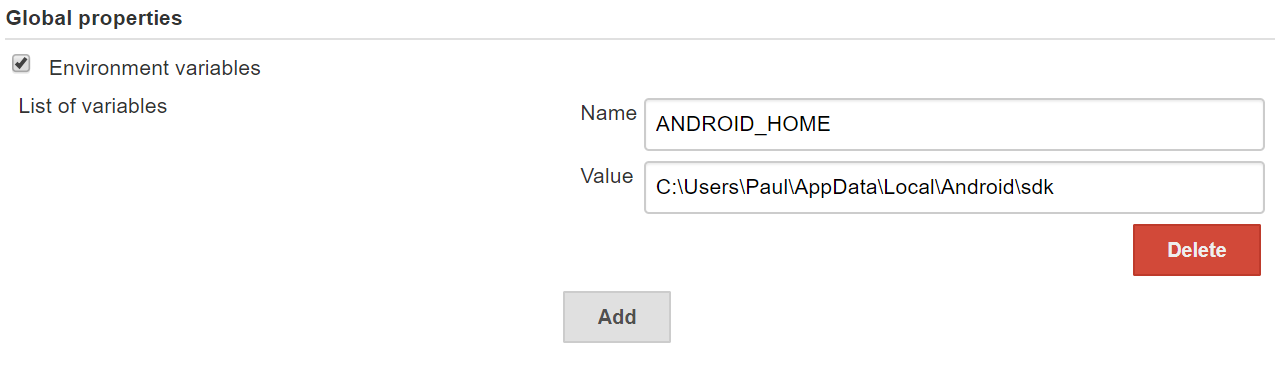
1. Fill in the JAVA\_HOME with your java path
2. Click “Add Git”



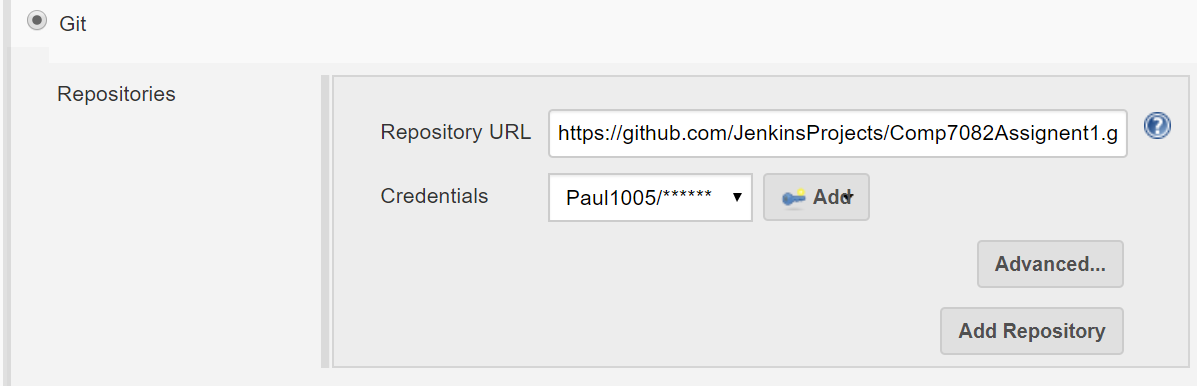
1. Fill in your path to Git executable



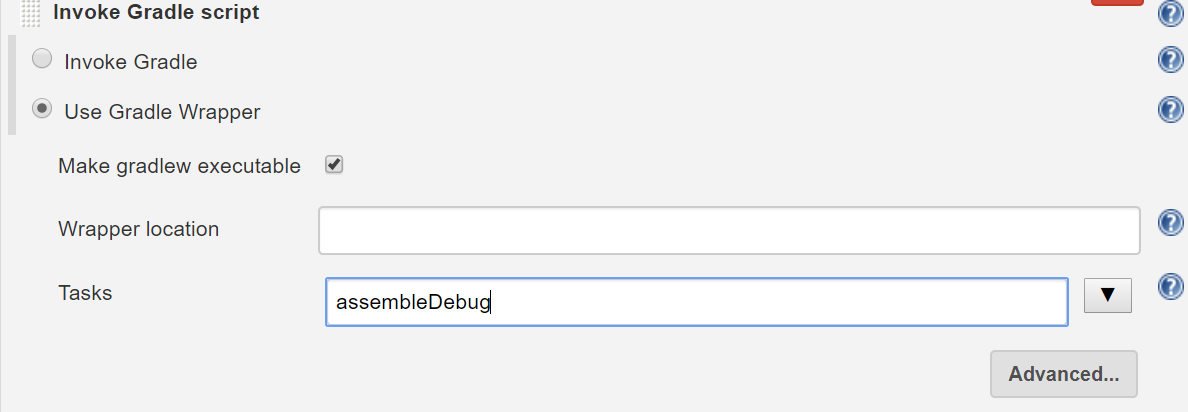
1. Go back and click Configure System
2. Check the environment variables box and add a variable called ANDROID\_HOME and fill in your android sdk path



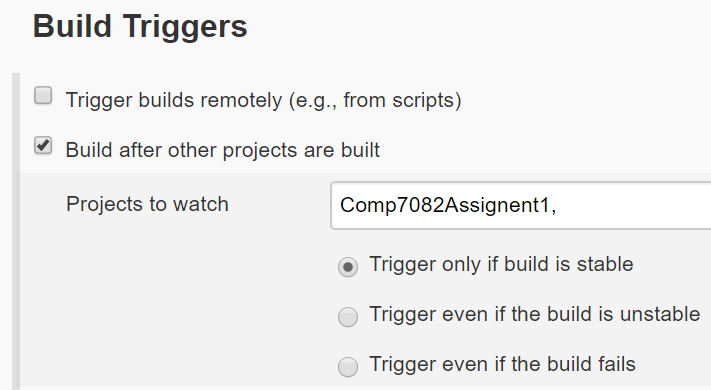
1. Click New Item
2. Select Freestyle project and enter a name, press OK
3. Go to source code management
4. Select Git
5. Open your GitHub repository page and copy the git url
6. Paste the URL in the Repository URL space



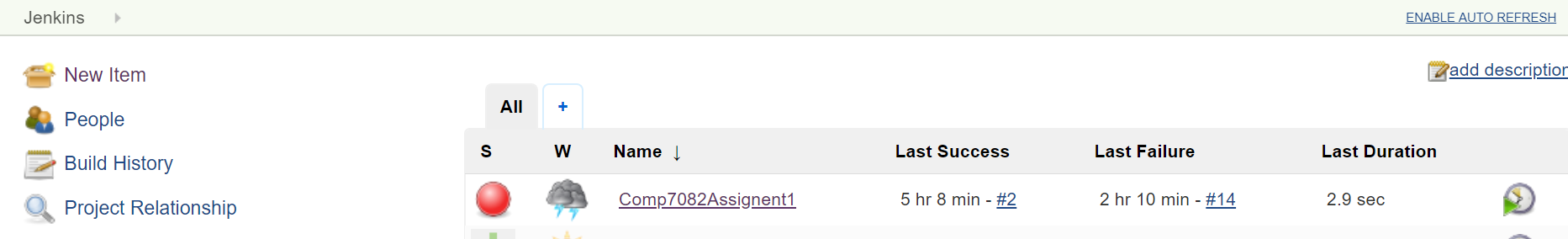
1. Click Add and input your GitHub credentials
2. Select your newly created credentials
3. Go to Build
4. Select Invoke Gradle Script
5. Select Use Gradle Wrapper
6. Check Make gradlew executable
7. Fill in tasks with assembleDebug



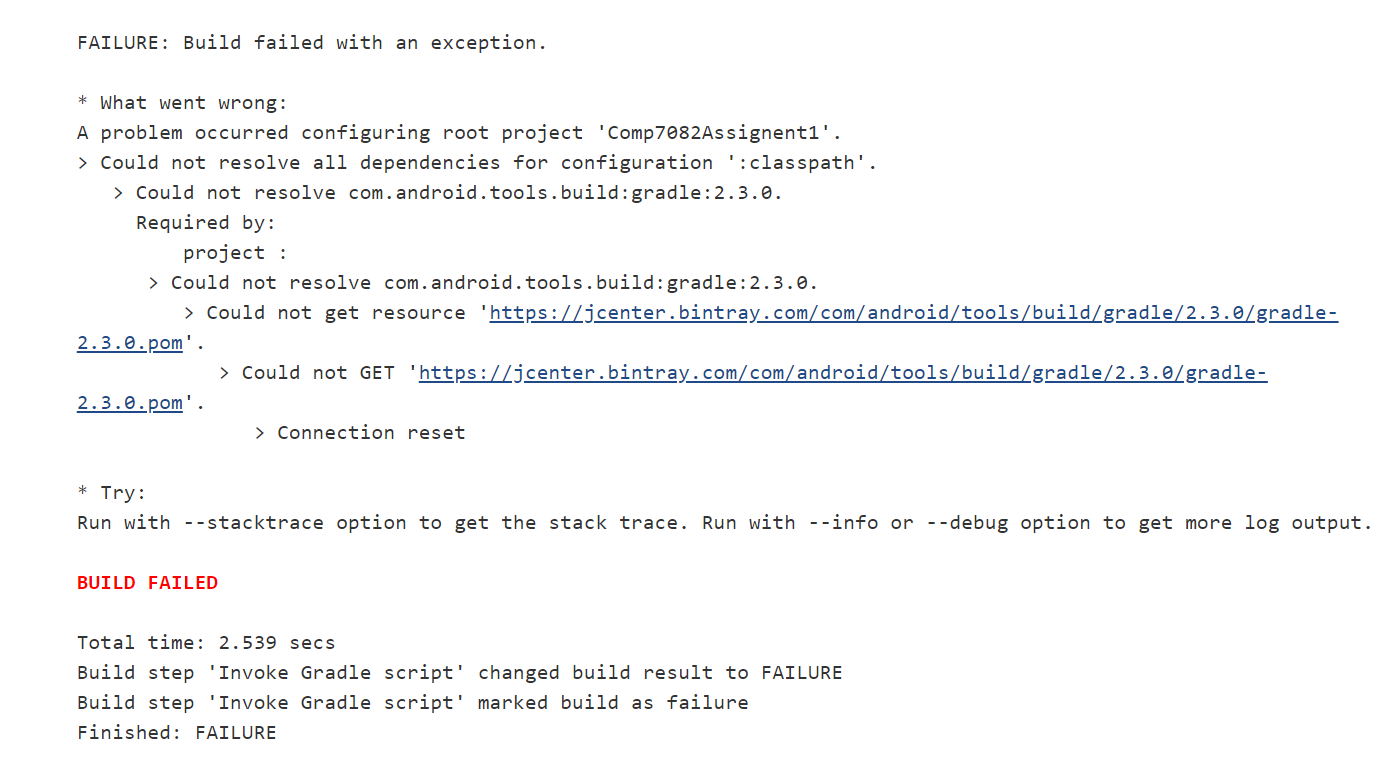
1. Go to Build Triggers
2. Check Build after other projects are built



1. Go back to home page and click on your project



1. Click Build Now
2. Click on your completed build and click console output



If I were to structure this project to support future development with multiple teams, I would do the following:

* Create multiple branches off of master, including
  + A develop branch, then, branching off from this
    - Feature branches for large features
  + A hotfix branch
* Allow each team to fork the repository
* Create multiple testing environments
  + One for each major feature branch,
  + One for development branch
  + One for master branch
* Tests would have to pass on one environment before code is merged
* Build and deployment will be automated