Sangeeth K Sivakumar
Mobile - Cloud - Microservice - Integration

DIY Series : Automation - iOS - Jenkins with Fastlane

If you are reading this post, then you know - Jenkins is one of the popular and widely used opensource automation tools available today. In a typical software organization setup, usually the so called 'Dev-Ops' team configures/ handles this tool. Developers often just consume it for the build purpose.

But what if - You as a developer just want to try out setting up your own Jenkins server in your local machine for your personal/ hobby project? Lets dive into hands-on.

For demonstration, we will use iOS project here with a Mac. But once you get the idea, you can pretty much repeat the same for Android, Java based microservice projects etc.

Step1: Easy Installation with HomeBrew

1. open terminal and run

```
brew install jenkins-lts
```

2. start the service

```
brew services start jenkins- lts
```

Get familiar with the commands to start, stop or restart server.. for future purposes. If you new to brew and require details, refer the below link and go through.**https://www.jenkins.io/download/lts/macos/**

3. Once everything is done, hit the browser ***http://localhost:8080*** *to verify. Now, stop the server with*

```
brew services stop jenkins-lts
```

Step2: Highly Recommended - Change default Jenkins server port

Port 8080 is default port for most of the applications that runs on localhost. We don't want to block it and later face issue with some other application. Instead choose a different uncommon port - for eg. **8087** (you can choose any other also)

To change port, locate& cd to the brew's jenkins installation folder in your system. Mostly, the default path looks like this **/opt/homebrew/Cellar/jenkins-lts/2.xxx.x** But if you are not sure, run

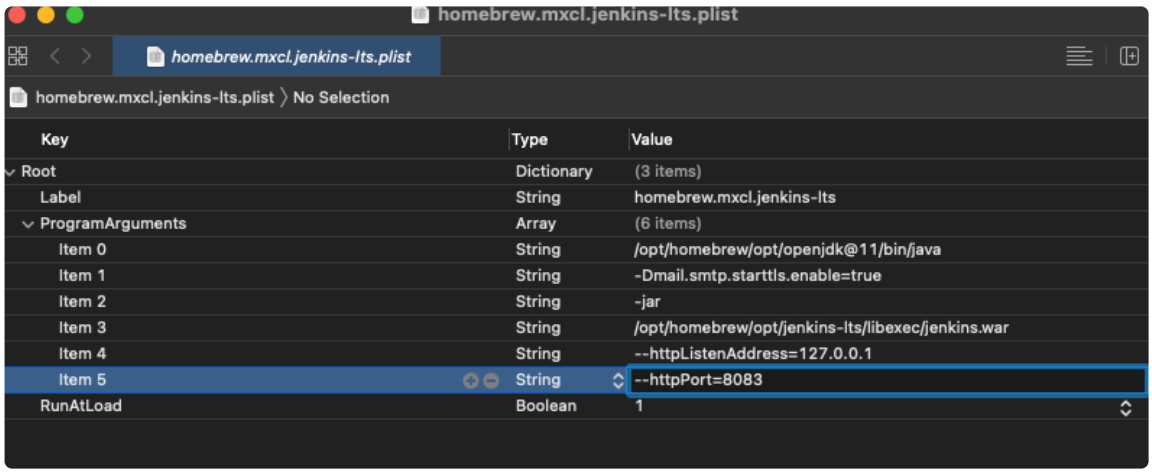
```
which brew
```

in the terminal to find the homebrew location and go to **/Cellar/jenkins-lts/2.xxx.x** from there.

In my case,

```
cd /opt/homebrew/Cellar/jenkins-lts/2.277.4/
```

Now, open **homebrew.mxcl.jenkins-lts.plist** and change the port as show below.



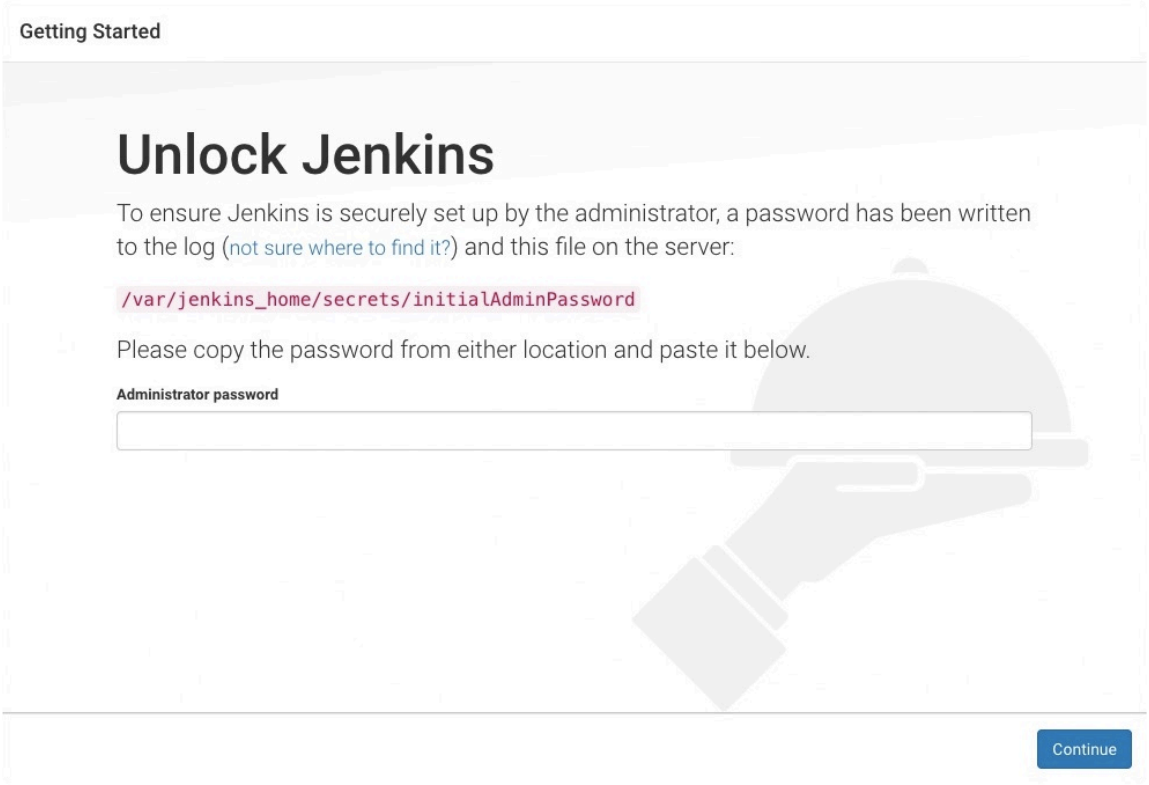
Save, close and go to terminal and run again,

```
brew services start jenkins-lts
```

Thats it! Open browser and hit the url **http://localhost:8087**

Step3: Jenkins Initial Setup

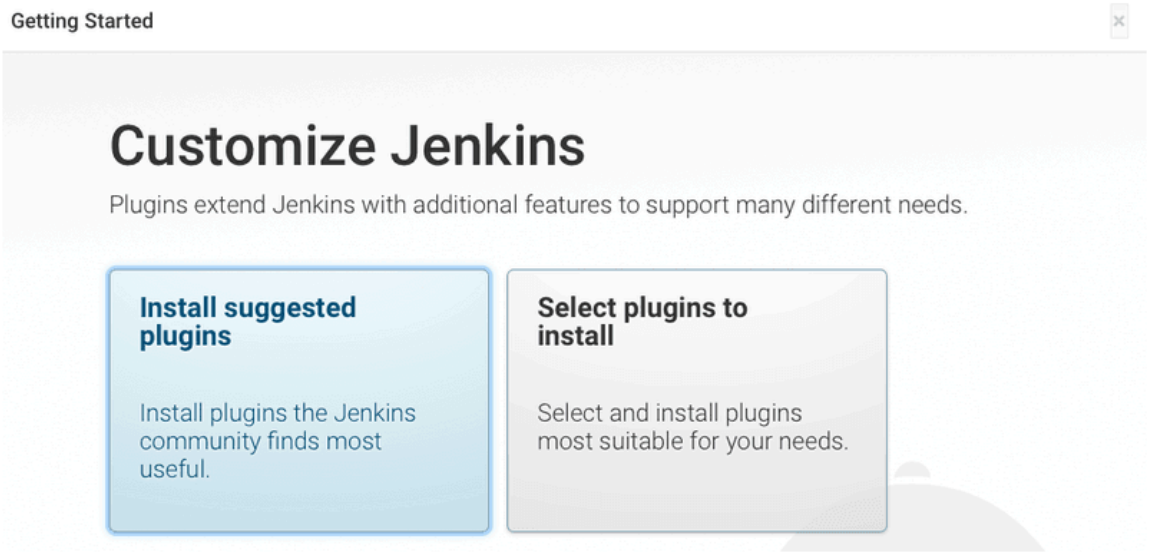
When you run Jenkins for first time, you will see the wizard like below



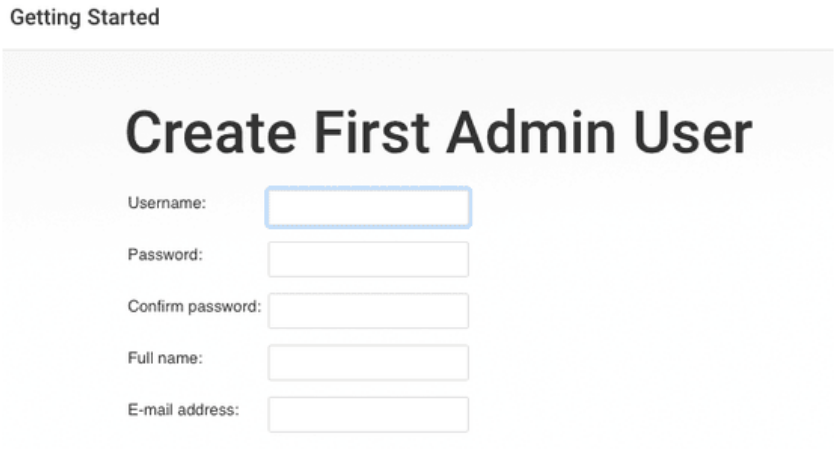
When the above screen appears, Copy the initial password using below command and paste to continue.

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Next - Go with suggested plugin



Setup intial user credentials



Note: In the final page, you must give the instance url matching the **port number** customized above. In our example, **http://localhost:8087** & **NOT** http://localhost:8080

Our initial setup is ready.

Step4: Add a new pipeline Job.

Let's create our first simple pipeline job for an iOS app code from hosted in the github repository.

A pipeline can have one single stage or multiple stages. I prefer to breakdown into multiple stages to visualise the process and identify any potential issues. In our example, lets go with 3 steps

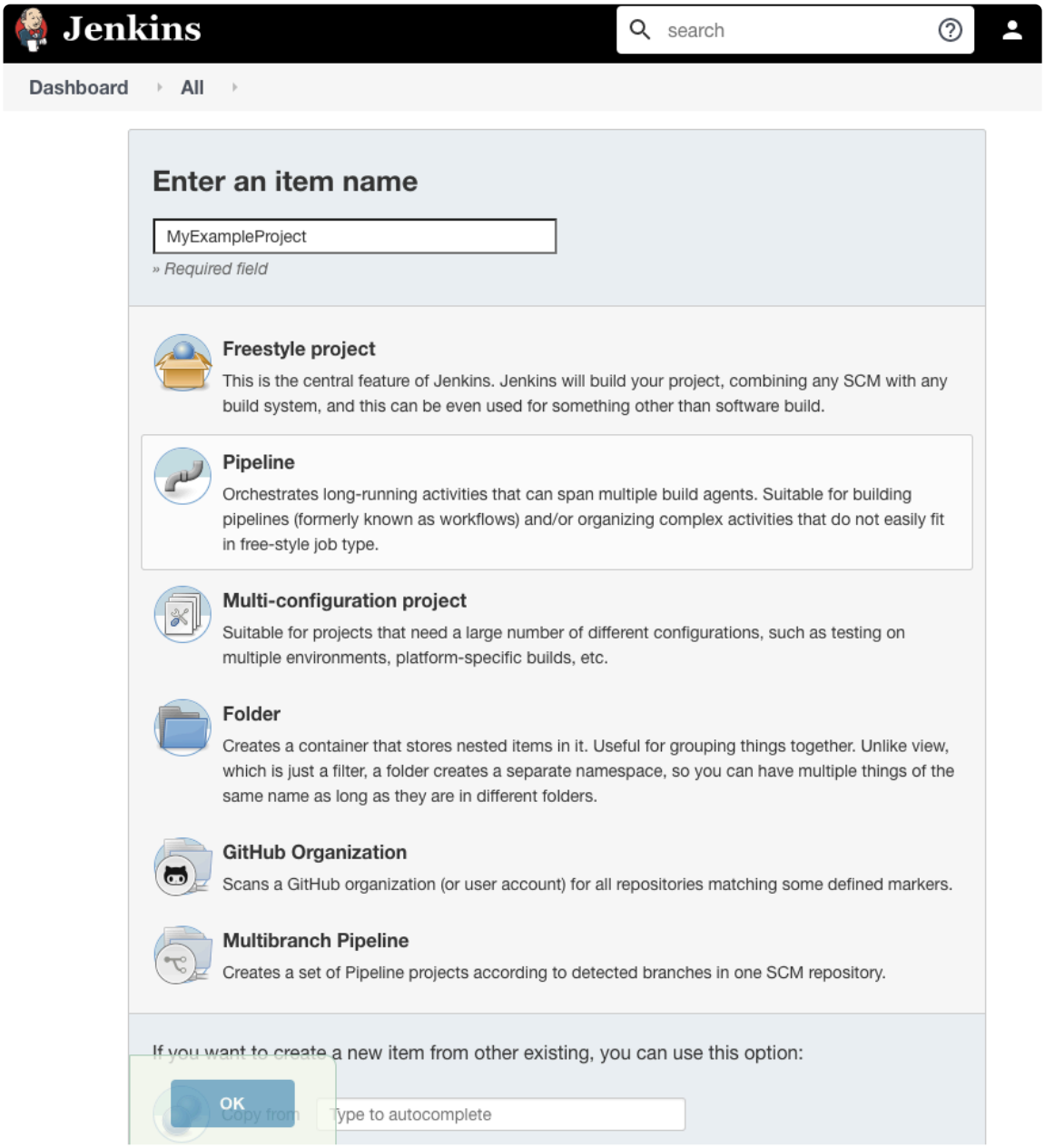
Checkout - Checks out the code from git repo to local machine where jenkins server is running.

Build - Build, Archive & Generate IPA from for the IOS App. You could replace it with Android APK build / Maven Java build etc

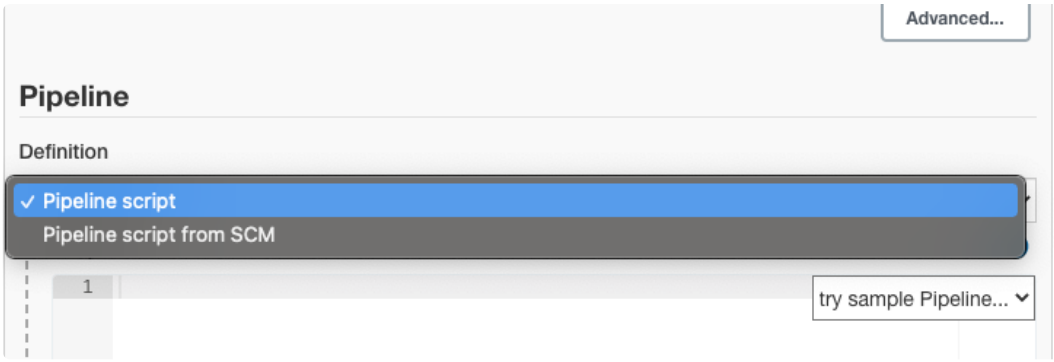
Deploy - Deploy the generated IPA to the test distribution server from which user can download the app to the device. You could replace it with distribution or deployment of microservices to VPS, cloud server etc. More Stages can also be added such as **test, code coverage, validation etc.**

Note: For the sake of simplicity and abstraction, **fastlane** tool for IOS is used for build generation and deployment. I am not covering the steps here as fastlane itself is a separate topic to be discussed and learnt.

- To create a pipeline, **click on 'New Item'** from Jenkins dashboard.



- Give a name of your choice - '**MyExampleProject**' and choose '**Pipeline**', select '**ok**' at bottom to redirect to the new job 'configuration page'.
- You can visit the same configure page next time from the left side menu when you open the created job from the dashboard list.
- There are plenty of things you can do with the pipeline, but lets skip those options as it is and jump to the Pipeline scripting option.



- Make sure '**Pipeline script**' is chosen in the definition dropdown.
- Now add the groovy script for different stages of your own or copy paste from below and save. [Groovy script for our example explained below]

Example Pipeline Script Explained:

```
pipeline {
    agent any
    environment {
        // Fastlane Environment Variables
        PATH = "$HOME/.rbenv/versions/2.7.3/bin:" +"$PATH"
        LC_ALL="en_US.UTF-8"
        LANG="en_US.UTF-8"
    }

    stages {
        stage('Checkout') {
            steps {
               checkout([$class: 'GitSCM', branches: [[name: '*/master']], ex
               sh '''
               pwd'''
            }
       }
       stage('Build') {
        steps {
            script {
                if (fileExists('FastLaneTestProject.ipa')) {
                    echo 'deleting existing ipa'
                  sh 'rm ./FastLaneTestProject.ipa'
              }
          }
          sh '''#!/bin/bash
          fastlane buildOnly
          '''
      }
```

```
    }
    stage('Deploy IPA') {
      steps {
          echo 'deploying to deployGate'
          sh '''#!/bin/bash
          fastlane myDeployGate deployGateKey:<KEY>
          '''
      }
  }
  }
  }
```

There are few important things here..

- **agent** - indicates the build machine in which your job runs. Since we are running in local with no master slave, just keep it to 'any'

environment - If you want to set any environment variables to be set for your script to be loaded. Jenkins uses the system default global PATH and not refer the user PATH defined in your bash_profile file. So, you might want to do something like this..

```
PATH = "$HOME/.rbenv/versions/2.7.3/bin:" +"$PATH"
```

THIS IS TO APPEND OUR USER DEFINED CUSTOM PATH TO THE DEFAULT PATH JENKINS REFERS FROM SYSTEM. IF YOU FACE 'COMMAND NOT FOUND' KIND OF ERROR WHILE RUNNING THE JOB, REVISIT AND CHECK THIS SECTION.

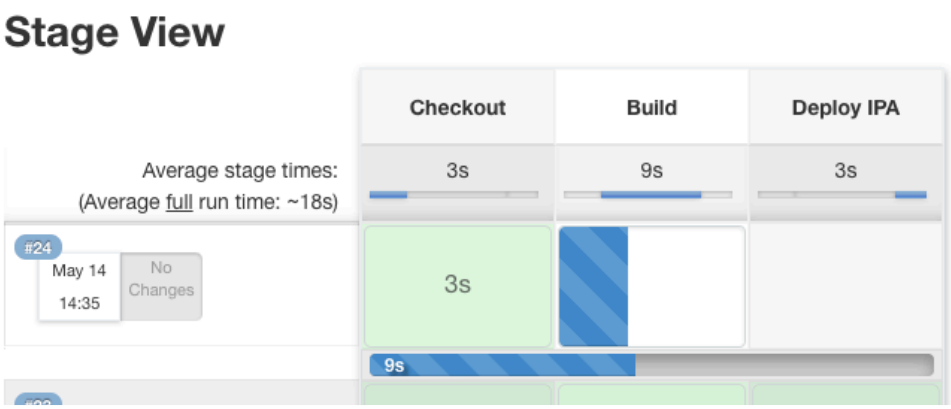
- **stages -** Here only we define our actual steps or stages of pipeline. In our case, its Checkout, Build & Deploy. You can write shell script with in the main groovy stage script using..

```
sh '''
echo 'my first command'
# add more
'''
```

- **Pipeline Syntax -** If you are not familiar with the groovy syntax to perform different operation, here is our saviour. In the pipeline configure page, below the Pipeline script such as checkouts, shell scripts etc, use visual aids to generate the script code.

Step5: All Done, Ready to build

As the configuration is done, lets get it tested. Go back to the created Job's landing page. From the left pane menu -> Click in Build. In a second or two, the jobs will starts with three different stages as show below.



YOU CAN CLICK ON THE BUILD HISTORY TO CHECK THE LOGS AS WELL.

Step6: Next ..

Start playing around different other options in the Jenkins like auto trigger, Git polling, Multibranch pipeline etc. You can start adding new stages, explore plugins, add child nodes etc.

Happy Jenking!!!

FRIDAY, MAY 14, 2021
