**JENKINS**

If you opt LTS version following are the commands to work on macOS.

install: brew install jenkins-lts

start: brew services start jenkins-lts

restart: brew services restart jenkins-lts

update: brew upgrade jenkins-lts

If you install jenkins.war file run it using:

**java -jar jenkins.war**

**Initial Password**

/Users/akshayreddygangula/.jenkins/secrets/initialAdminPassword

5e72aaa7e77047b9aa5cc6a3693cf57b

localhost:8080 is to connect with **jenkins.**

**Jenkins URL:** [**http://localhost:8080/**](http://localhost:8080/)

Username: Akshay

Password: axayreddy

**Installed Plugin**

* Role-based Authorization Strategy

**To restrict a full access to a user**

Manage Jenkins -> Security

* **Security Realm** (Authentication)
* **Authorization**

**Under Authorization**

* Select ***Matrix-based security*** for granular control.
* Install ***Role-based Authorization Strategy*** plugin and get an extra feature ***Manage and assign Role*** under *Security.*

Watch 11th video

**Jenkins Job**

Create new item

* Under ***General***
  + **Description** (supports HTML)
  + **Discards Old builds** (Discards builds based on *Days* or *No. of builds)*
  + **GitHub Project** (paste the URL)
  + **This project is parametrized**
  + **Throttle Builds** (No. of builds in time interval (minute, hour, day, week, year)
  + **Execute concurrent builds if necessary** (Allowing multiple builds to run simultaneously)
* Under ***VCS***
  + **Git** or **None**
* Build ***Triggers***
  + **Trigger builds remotely (e.g., from scripts)**
  + **Build after other projects are built**
  + **Build periodically** (format: MINUTE HOUR DOM MONTH DOW)
  + **GitHub hook trigger for GITScm polling**
  + **Poll SCM**
* Build ***Environment***
  + **Delete workspace before build starts**
  + **Use secret text(s) or file(s)**
  + **Add timestamps to the Console Output**
  + **Inspect build log for published build scans**
  + **Terminate a build if it's stuck**
  + **With Ant**
* ***Build Steps***
  + *Add build step*
* ***Post-build Actions***
  + *Add post-build action*

Create a Free-style job in Jenkins, and a Java program on desktop and in the **Build Steps**

1. put the address of the Java file.
2. javac HelloWorld.java
3. java HelloWorld (This will be automatically generated)

* We can run the job outside the Jenkins by configuring **Build Triggers**.
* Create a random token like: jclkdshgfo9refjceosfh
* Change the URL: JENKINS\_URL/job/Hello-World/build?token=TOKEN\_NAME
  + Replace JENKINS\_URL and TOKEN\_NAME
  + localhost:8080/job/Hello-World/build?token=jclkdshgfo9refjceosfh

***Create Job1, Job2, Job3***

* In Job1 go to **Build other Projects** (Select Job2) -> **Trigger only if build is stable**
* In Job3 go to **Build Trigger** -> Build after other projects are built (Select Job2)