**Jenkins tutorial**

<https://www.youtube.com/watch?v=O6BJEptOjjs> for maven run through git-hub/cmd

<https://www.youtube.com/watch?v=pzbrVVy6ul4> for pipeline

**jenkins**

**password** - 2e2af3a9127a49b99ac5787a7b5fcf45

**my laptop** - cef31a527da24b97aaf06da21175efb7

**password location** - C:\Users\ksanj\.jenkins\secrets\initialAdminPassword

**username-** admin

**password** - 123456789

jenkins war command - to see the steps to create local host

open cmd at jenkins war download location and type --- **java -jar jenkins.war**

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

**After installation of jenkins**

1. manage jenkins**-->**plugins-->available plugin**-->**search maven**-->**select Maven Integration
2. **Steps to run maven program through CMD/Maven project**

**step 1** - create new project

**step 2** - go to General-->advanced-->Use custom workspace-->paste pom.xml path in Directory

**step 3** - Build steps-->Execute windows batch command--> cd space path of pom.xml & mvn test in next line

**“or”**

Build steps-->Invoke top level maven command--> and type clean test

**eg:-** cd C:\Users\Dell\eclipse-Sanjay\Ehome

mvn test

**step 4** - Apply & save -🡪 build now

1. **Steps to run maven program through Git-hub path**

**step** **1** - create new project-->freestyle project

**step** **2** - select Git-->paste url-->give credential

**step 3** - check branch specifier -->master is seleted or (select the other branch in which pom.xml is located)

**step 4** - Build steps-->Invoke top level maven target-->

Goals-->clean test

Advanced-->pom.xml in (POM)

**Step – check C) below**

to generate email notifications click on post buid-🡪Editable email notification-🡪enter mail id of project recipient list-🡪content type-🡪 Html-🡪 status

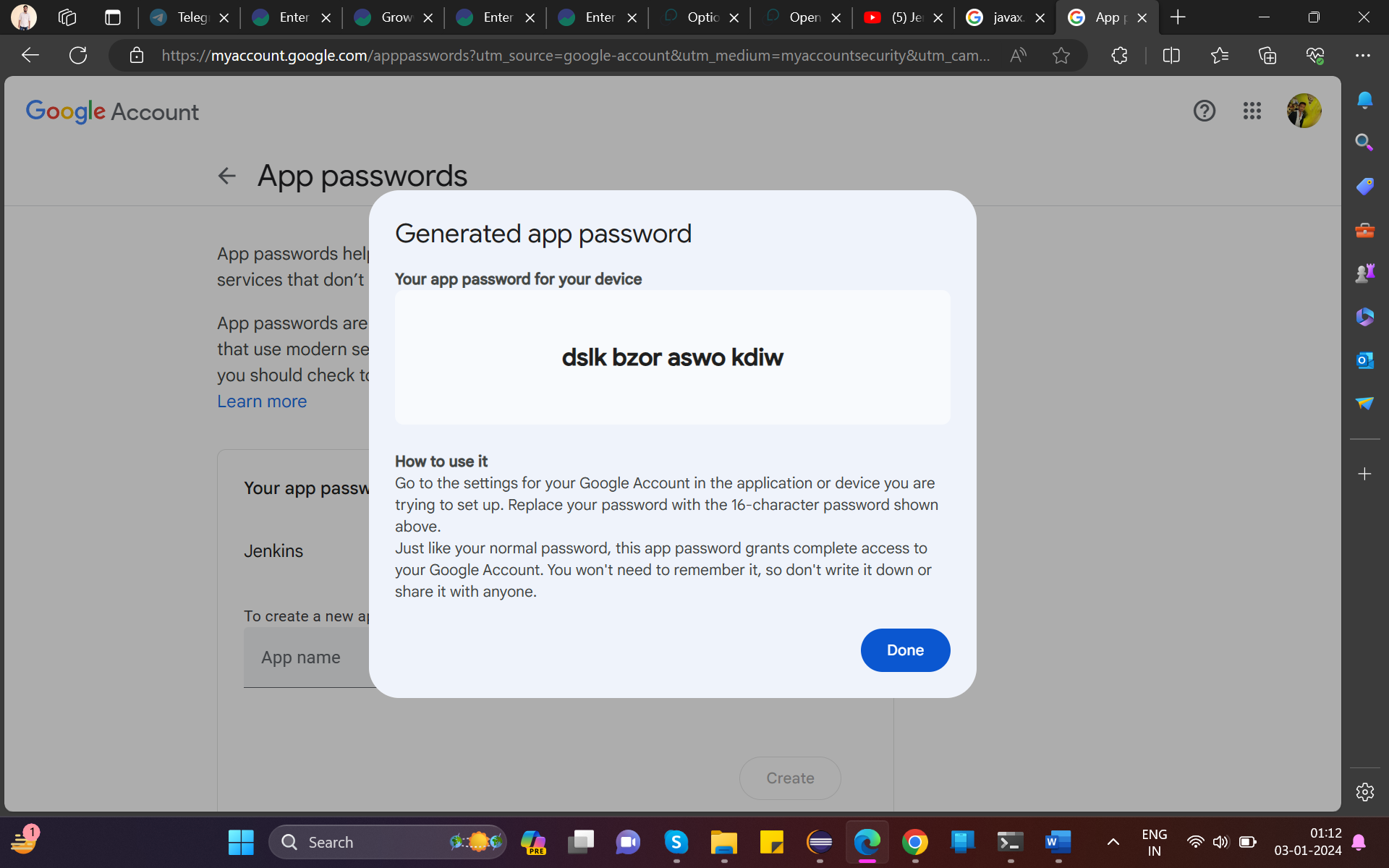
<https://www.youtube.com/watch?v=OdqnzsAHp5I&list=PL5fOKT7XR42Po4zh8nHCozrcsMboLNXjF&index=13>

**step 5** - Apply & save.

**C) Steps to create gmail msg and notification through Jenkins**

1. First install email extension plugin in manage Jenkins
2. Manage Jenkins-🡪configure global settings-🡪Email Notification(at last bottom)
3. SMTP server -🡪 enter smtp.gmail.com(it is for gmail only)
4. Click on advanced -🡪 select **Use SMTP Authentication -🡪** enter gmail username & password**(password should be app passcode created in google privacy settings-🡪 2 step verification 🡪 App passwords)**
5. Click on **Use SSL**
6. Enter SMTP port as 465
7. Select **Test configuration by sending test e-mail**
8. Enter the mail id to whom you want to send
9. Apply & save

**Gmail(ksanjay815@gmail.com) app passcode for Jenkins–** dslkbzoraswokdiw



1. After testing by clicking -🡪 Test email configuration
2. Go to **Extended E-mail Notification**
3. enter SMTP server as -🡪 smtp.gmail.com & port as 465
4. select credential(if added or add same gmail id and passcode as added above **4.** Point)🡪 select use SSL
5. Default content type -🡪 select text/html
6. Enter mail id in default recipient
7. Click on **default triggers** dropdown & select the option
8. Apply & save
9. Select **Editable email notification** in post build steps

**Step 14, 15, 16 can be done in post build actions -🡪 Editable email notifications**

**D)**Using **GitHub hook trigger for GITScm** so that whenever new code is pushed to github it will test it.

**Step1** – select **GitHub hook trigger for GITScm polling** and click on save

Step2 – Go to github repo-🡪setings-🡪webhooks-🡪add webhooks-🡪enter payload url (<http://localhost:8080/>) this link is not preferred.

So create another link using ngrok.exe (download ngrok)

Step3- open ngrok in cmd -🡪to check version 🡪 ngrok -–version

{<https://dashboard.ngrok.com/get-started/setup/windows> ngrok id created by ksanjay mail id}

Signup ngrok-🡪go to Your **Auth token** and copy the command and token from **command line** and paste it in cmd (already opened )

Step4 -type -🡪 ngrok http 8080

Step5 – copy the link from **forwarding http/https** and paste it in -🡪 github payload url and ends with **/github-webhook/**🡪

[https://3a3d-139-5-250-11.ngrok-free.app**/github-webhook/**](https://6990-139-5-250-11.ngrok-free.app/github-webhook/)

**Note: The above link** <https://3a3d-139-5-250-11.ngrok-free.app> will change every time you run **ngrok** and you have to **enter this link in payload url in github everytime to use the hooks** to **auto build in Jenkins** you have to **run the ngrok and Jenkins in cmd always**, otherwise it will not run auto **build now** in **Jenkins.**

<https://www.youtube.com/watch?v=bjelFjcY_o8&list=PLJ2HDHmFbqXKY_f88-hHfMOaOo0xknAsT&index=3&t=1330s>

**E) Build periodically** in build triggers

**step1-** select build periodically --🡪 enter the code as per the time you want to set

eg:- \* \* \* \* \* like 5 star means daily run at 12:00 am

30 20 \* \* \* it means 8.30 pm daily

30 20 \* \* SUN it means 8.30 pm every Sunday

15 08 \* \* \* it means 8.15 am daily

for more examples go to - <https://crontab.guru/examples.html> --> and click on examples-🡪 and select as per yours

**F) Poll SCM** in build triggers

**step1-** select Poll SCM --🡪 enter the code as per the time you want to set

eg:- \* \* \* \* \* like 5 star means polling at every 1 minute for changes in github path to run

\*/2 \* \* \* \* it means polling at every 2 minute

\*/3 \* \* \* \* polling at every 3 minute

\*/5 \* \* \* \* polling at every 5 minute

for more examples go to - <https://crontab.guru/examples.html> --> and click on examples-🡪 and select as per yours

1. **Create Upstream/Downstream pipeline**

For parameterized pipeline we have to install **parameterized trigger plugin** in jenkins

[**https://www.youtube.com/watch?v=\_1EgoteXRoc&list=PL6flErFppaj35spJjPy41-lruDjw2kRV-&index=12**](https://www.youtube.com/watch?v=_1EgoteXRoc&list=PL6flErFppaj35spJjPy41-lruDjw2kRV-&index=12)

Pipeline is used to do continuous integration & continuous deployment of the project

eg:- we created 3 jenkins project in dashboard

1st one is for API Testing

2nd one is for GUI Testing

3rd one is for Performance Testing

Then we will create a pipeline project so that after completion of each test next test will be started.

**Step1 –** Go to project 1 -🡪select post build action --🡪 build others-🡪select project 2--🡪 triggered only if build is stable

**Step2 -**  Go to project 2 -🡪select post build action --🡪 build others-🡪select project 3--🡪 triggered only if build is stable

**Step3** – go to dashboard -🡪 click on + symbol --🡪 select build pipeline view-🡪 create-🡪 go to initial job-🡪 select project 1--🡪 go to no of build -🡪 select 1 or 2 or 3(keep it as per your wish)

**Step4 –** apply & save

**Step5 –** Click on run

1. **For Jenkins Pipeline**

Pipeline is used to do continuous integration & continuous deployment of the project

eg:- we created 3 jenkins project in dashboard

1st one is for API Testing

2nd one is for GUI Testing

3rd one is for Performance Testing

Then we will create a pipeline project so that after completion of each test next test will be started.

[**https://www.youtube.com/watch?v=AqITZLJ5eZ4&list=PL6flErFppaj35spJjPy41-lruDjw2kRV-&index=13**](https://www.youtube.com/watch?v=AqITZLJ5eZ4&list=PL6flErFppaj35spJjPy41-lruDjw2kRV-&index=13)pipeline setup link

Setup-🡪 manage jenkins-->plugins-->available plugin-->search pipeline-->select pipeline Maven Integration

**step 1** - create new project-->pipeline

1. **Pipeline Script for creating stages and sending email(write syntax in post build)**

pipeline {

agent any

stages {

stage('build') {

steps {

echo 'Hello build'

}

}

stage('test') {

steps {

echo 'Hello test'

}

}

stage('deploy') {

steps {

echo 'Hello deploy'

}

}

}

Post(

Always{

}

emailext body: 'Plz find the results', subject: 'Test results', to: 'sanjaykumar.marolix@gmail.com'

}

}

1. **Steps for creating pipeline script from SCM**

<https://www.youtube.com/watch?v=AqITZLJ5eZ4&list=PL6flErFppaj35spJjPy41-lruDjw2kRV-&index=13>

**Step1 –** select pipeline script from SCM -🡪 select git-🡪 paste git url -🡪 select credential -🡪 check branch -🡪 check script path as Jenkinsfile-🡪 apply & save

1. **Steps for creating pipeline through github(to clone the project)** [**https://www.youtube.com/watch?v=DCY530UL0as**](https://www.youtube.com/watch?v=DCY530UL0as)

Step 2 – click on pipeline syntax-🡪 sample steps-🡪git:Git-🡪paste github url-🡪check branch-🡪credentials-🡪click on generate pipeline script

Copy the generated script and paste in the **pipeline script**

pipeline {

agent any

stages{

**stage("git"){**

**steps{**

**git credentialsId: 'b33a4149-8c8b-438c-aff2-8fbeada656a5', url: 'https://github.com/ksanjay815/Ehome.git'**

**}**

**}**

}

}

Step 3 – apply & save -🡪 build now

**Note :** In The above script the stage { } part is to clone git-hub project to local repo only.

1. **Steps for creating pipeline through github(to clone the project and run through maven command)**

**To run maven test we have to add one more stage for maven command.(as mentioned below)**

[**https://www.youtube.com/watch?v=kO7wrtY2b3E**](https://www.youtube.com/watch?v=kO7wrtY2b3E)

**sh** is used for **linux** computer & **bat** is used for **windows** computer

pipeline {

agent any

environment{

PATH= "/opt/apache-maven-3.9.6/bin/:$PATH"

}

stages{

stage("git clone"){

steps{

git credentialsId: 'b33a4149-8c8b-438c-aff2-8fbeada656a5', url: 'https://github.com/ksanjay815/Ehome.git'

}

}

stage("build code"){

steps{

sh "mvn clean install"

}

}

}

}

**Example 2**

pipeline {

agent any

tools{

maven "MAVEN\_HOME"

} stages{

stage("git clone"){

steps{

git credentialsId: 'b33a4149-8c8b-438c-aff2-8fbeada656a5', url: 'https://github.com/ksanjay815/Ehome.git'

}

}

stage("build code"){

steps{

bat "mvn clean compile test"

}

}

}

}