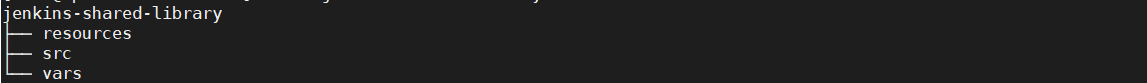
**Jenkins Shared Library**

**Pre-requisites:**

* Git
* Install Jenkins

Create a Directory structure for Jenkins Shared Library as shown below



Keep groovy files in **vars** Directory

As shown below



Content in **gitCheckout.groovy** file:

vi gitCheckout.groovy

def call(Map stageParams) {

checkout([

$class: 'GitSCM',

branches: [[name: stageParams.branch ]],

userRemoteConfigs: [[ url: stageParams.url ]]

])

}

Content in **dockerBuild.groovy** file:

vi dockerBuild.groovy

def call(String project, String hubUser) {

sh "docker image build -t ${hubUser}/${project}:beta-${env.BRANCH\_NAME}-${env.BUILD\_NUMBER} ."

withCredentials([usernamePassword(

credentialsId: "docker",

usernameVariable: "USER",

passwordVariable: "PASS"

)]) {

sh "docker login -u '$USER' -p '$PASS'"

}

sh "docker image push ${hubUser}/${project}:beta-${env.BRANCH\_NAME}-${env.BUILD\_NUMBER}"

}

Content in **dockerCleanup.groovy** file:

vi dockerCleanup.groovy

def call(String project) {

sh "docker stack rm ${project}-beta-${env.BRANCH\_NAME}-${env.BUILD\_NUMBER}"

sh "docker system prune -f"

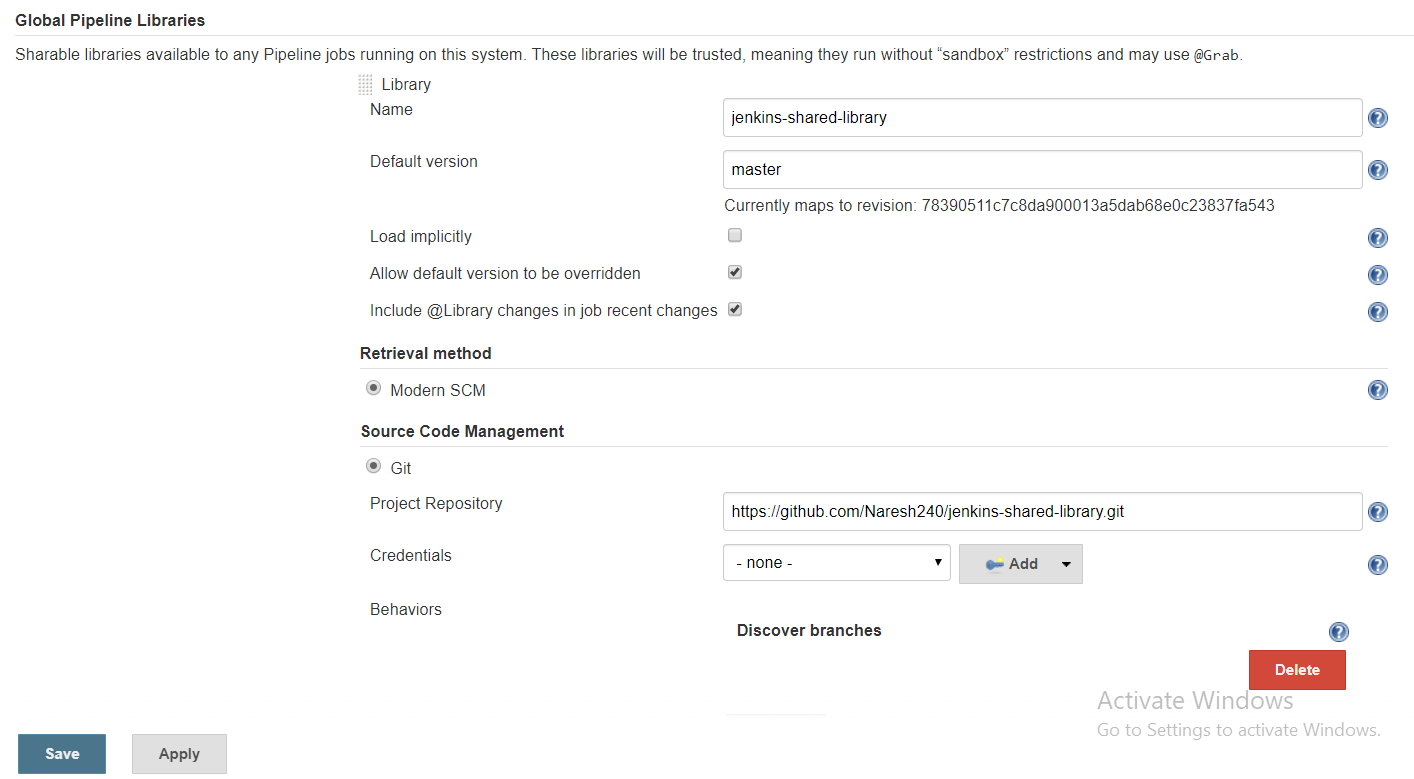
}

Now push to **Github** to configure this with Jenkins as a shared library

Integrate Shared Library with Jenkins

Go to Jenkins 🡪 Manage Jenkins 🡪 Configure System

Goto **Global Pipeline Libraries** and add SCM (where you keep Jenkins shared library repository)



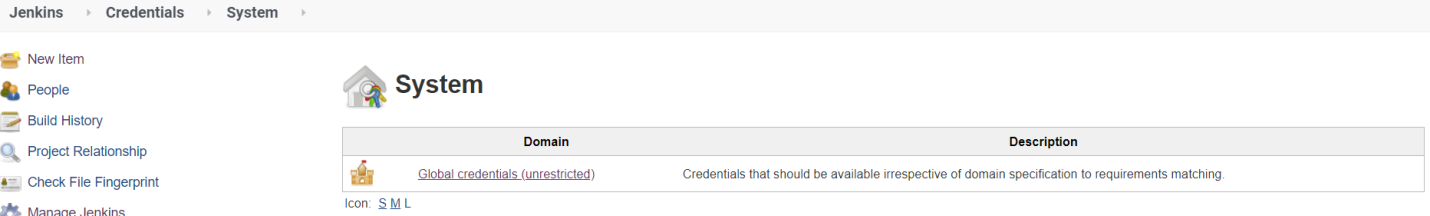
Give details as shown in above image and save it.

Create docker credentials with docker

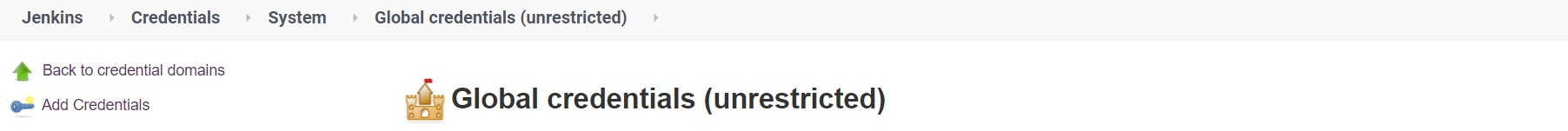
Go to Jenkins 🡪 Credentials



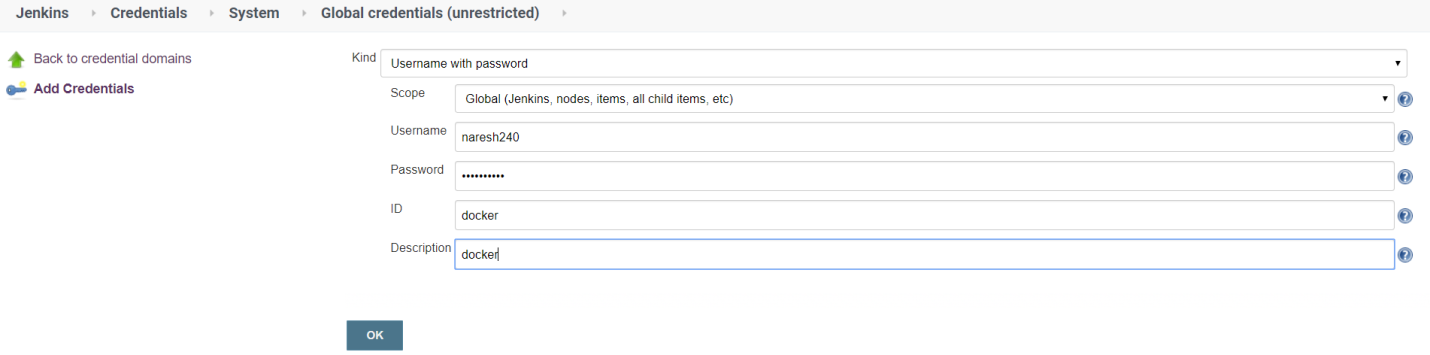
Click on Jenkins



Click on Global credentials (unrestricted)



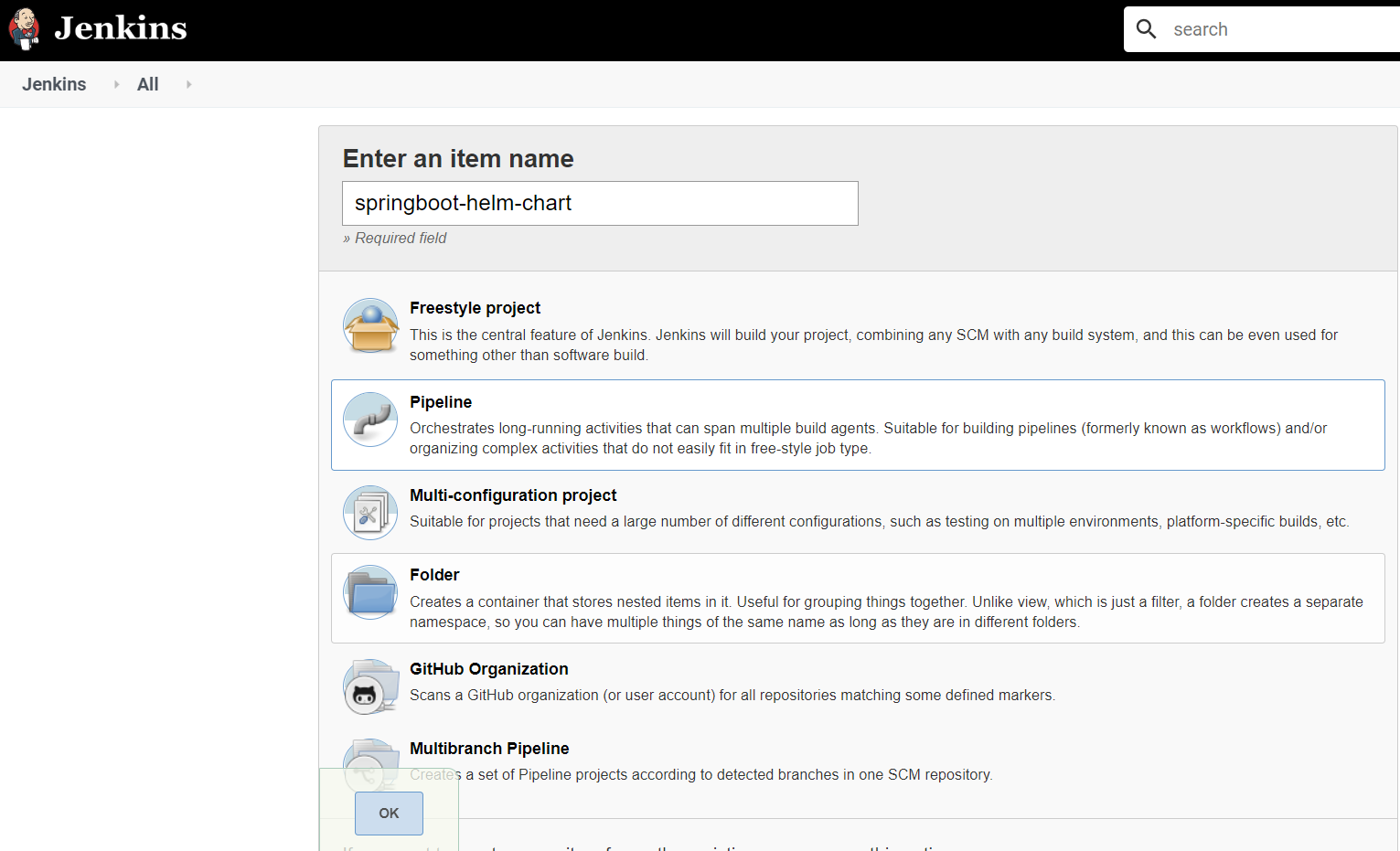
Click on Add Credentials



Give user name, password of your **dockerhub** and also mention ID as **docker** only, because in our groovy files we mentioned **CredentialsId** as **docker**.

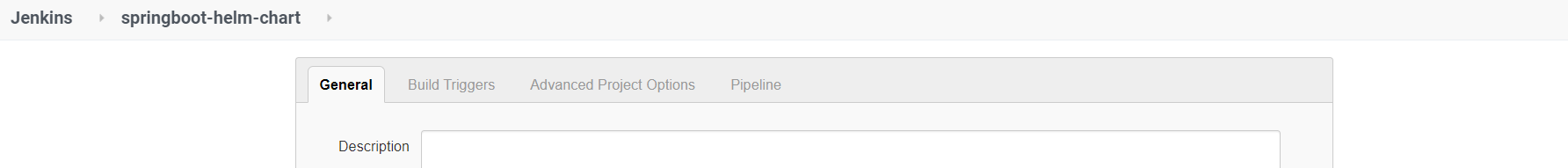
Click on ok

Goto Jenkins Dashboard and create a job as shown below:



Give name for the Jenkins job and select type as pipeline

Click on **OK**

****

Click on pipeline and add below pipeline script

@Library('jenkins-shared-library@master') \_

pipeline {

agent any

tools{

maven 'maven3'

}

stages {

stage('Git Checkout') {

steps {

gitCheckout(

branch: "master",

url: "https://github.com/Naresh240/springboot-helm-chart.git"

)

}

}

stage('Build Maven'){

steps {

dir('demoweb') {

sh 'mvn clean package'

}

}

}

stage("Docker Build and Push") {

steps {

dir('demoweb') {

dockerBuild ( "springboot-helm-chart", "naresh240" )

}

}

}

stage("Docker CleanUP") {

steps {

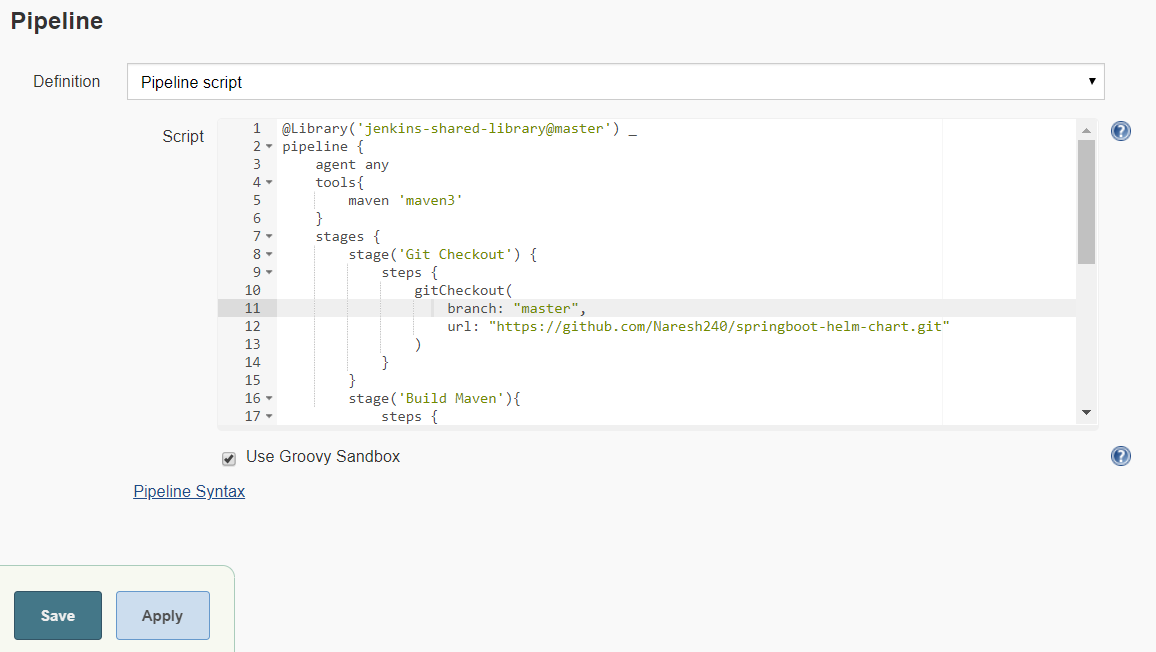
dockerCleanup ( "springboot-helm-chart", "naresh240" )

}

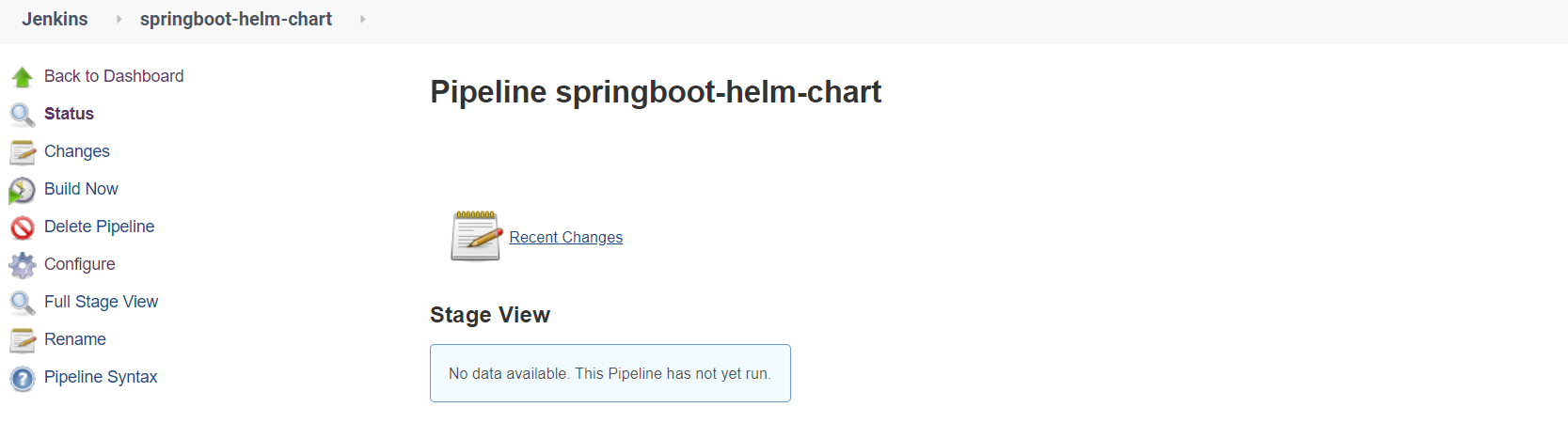
}

}

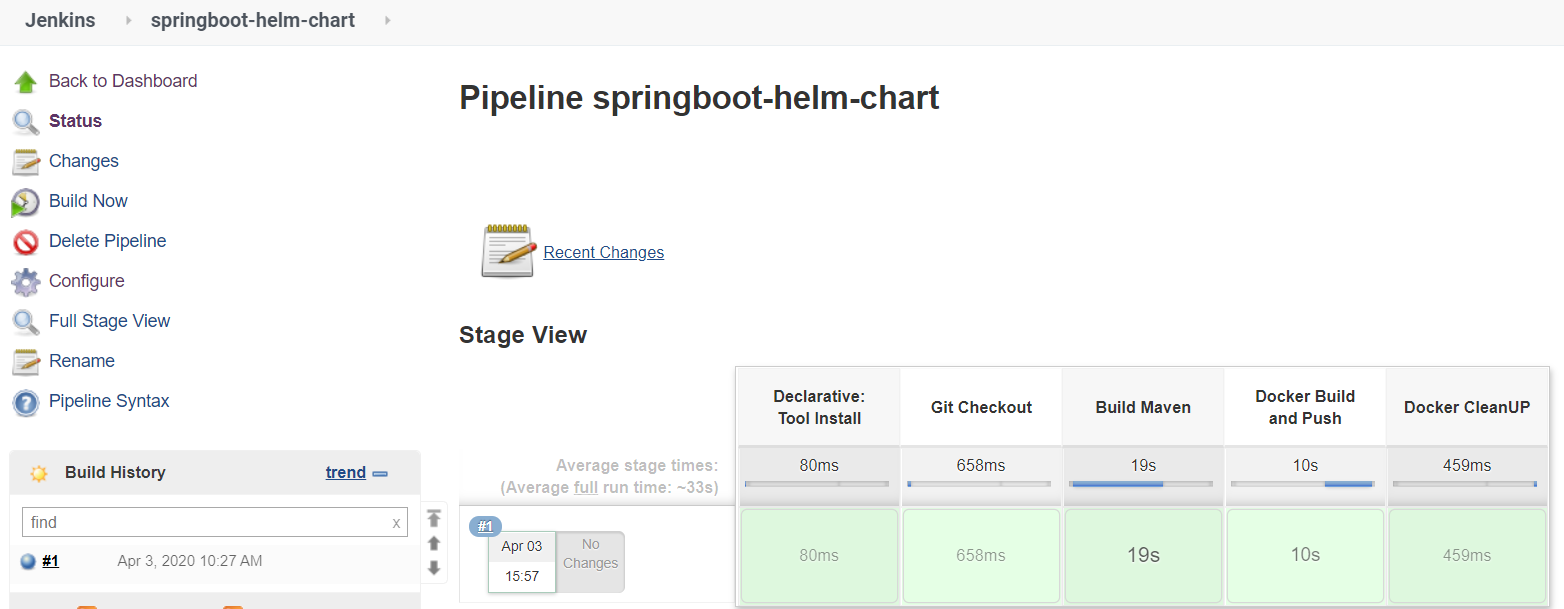
}



Click on **Save**



Click on **Build**



Build success.

Checking in **dockerhub** whether docker image pushed with build number to our docker hub or not

