

**TO  
THE  
NEW™**



## **Assessment -21**

### **JENKINS-2**

Trainee Name : Arun Parmar

Mentor Name : Ravi Kumar

College: UPES

**1. Create a jenkins pipeline Job to delete redundant docker images daily at 1 AM UTC. Create a job on Jenkins**

Activities Firefox Web Browser Tue 21:48

Docker #1 Console [Jenkins] - Mozilla Firefox

localhost:9999/job/Docker/1/console

# Jenkins

search monitor Arun Parmar log out

Jenkins Docker #1

- Back to Project
- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#1'
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces

## Console Output

Started by user [Arun Parmar](#)  
Running in Durability Level: MAX\_SURVIVABILITY  
[Pipeline] Start of Pipeline  
[Pipeline] node  
Running on [jenkins](#) in /var/lib/jenkins/workspace/Docker  
[Pipeline] {  
[Pipeline] stage  
[Pipeline] { (Delete docker images) (Delete docker images)  
[Pipeline] sh  
+ docker image prune  
WARNING! This will remove all dangling images.  
Are you sure you want to continue? [y/N] Total reclaimed space: 0B  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS

Page generated: Apr 21, 2020 9:48:06 PM IST [REST API](#) [Jenkins ver. 2.222.1](#)

Activities Firefox Web Browser Tue 21:48

Docker [Jenkins] - Mozilla Firefox

localhost:9999/job/Docker/

# Jenkins

search monitor Arun Parmar log out

Jenkins Docker

disable auto refresh

- Back to Dashboard
- Status
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Rename
- Pipeline Syntax

## Pipeline Docker

[add description](#)  
[Disable Project](#)

[Recent Changes](#)

### Stage View

Average stage times:  
(Average full run time: ~4s)

Stage	Time	Changes
Apr 21 21:47	511ms	No Changes

[Delete docker images](#)  
511ms

### Build History

trend

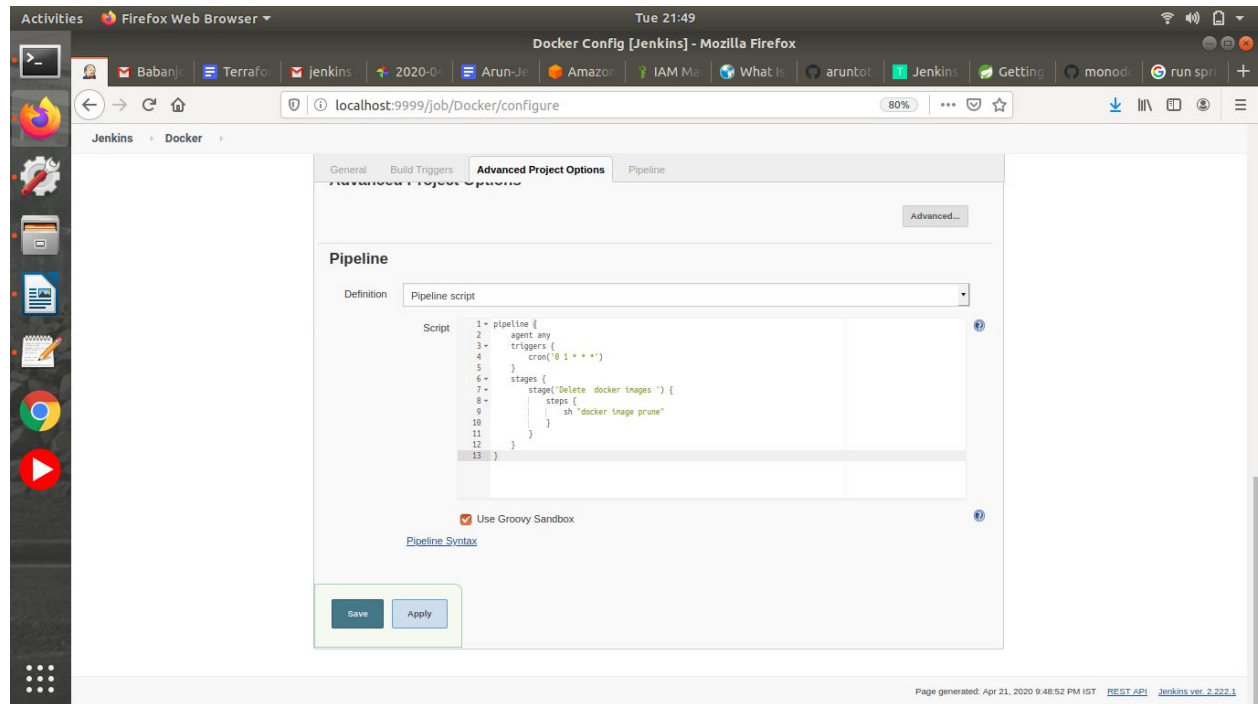
find

Apr 21, 2020 9:47 PM

[Atom feed for all](#) [Atom feed for failures](#)

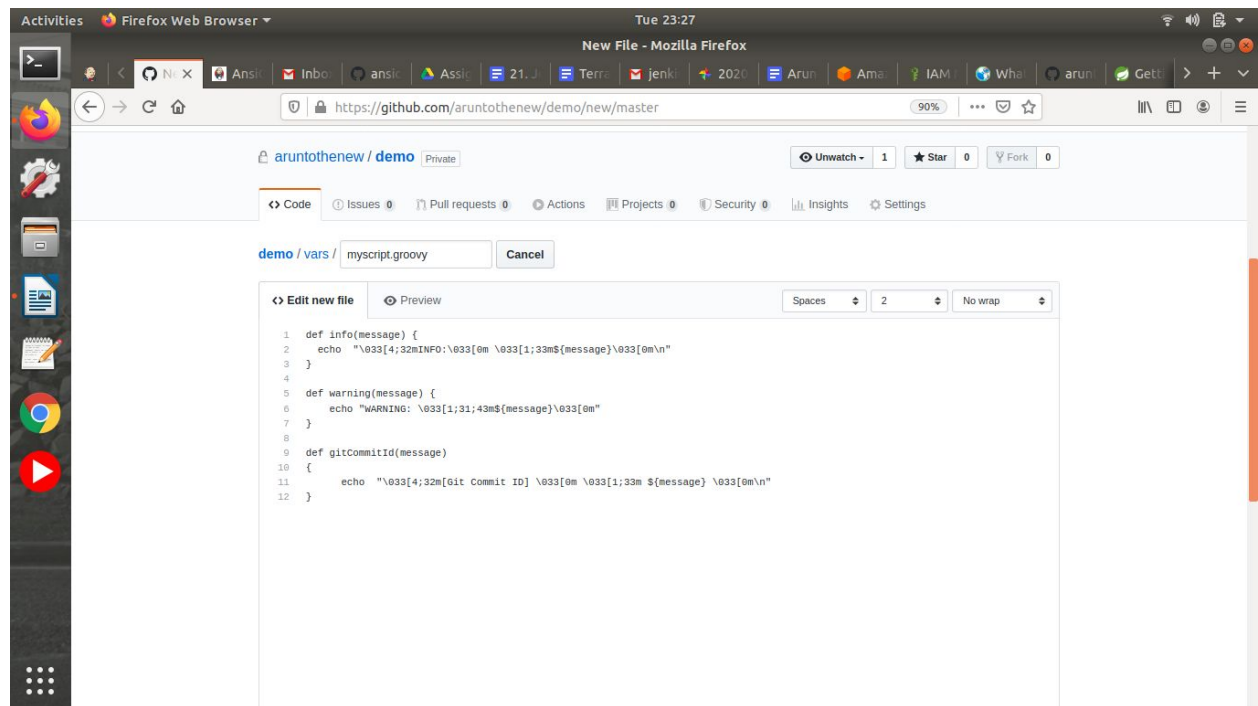
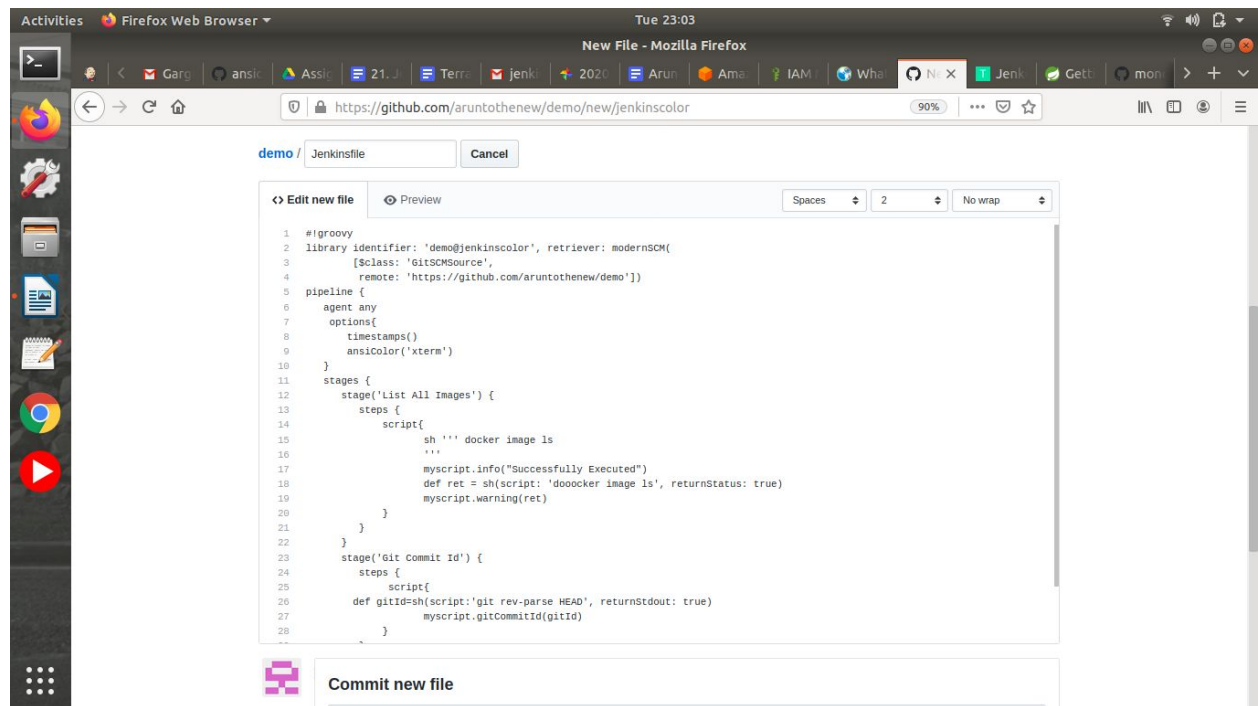
### Permalinks

- [Last build \(#1\) 37 sec ago](#)
- [Last stable build \(#1\) 37 sec ago](#)
- [Last successful build \(#1\) 37 sec ago](#)
- [Last completed build \(#1\) 37 sec ago](#)

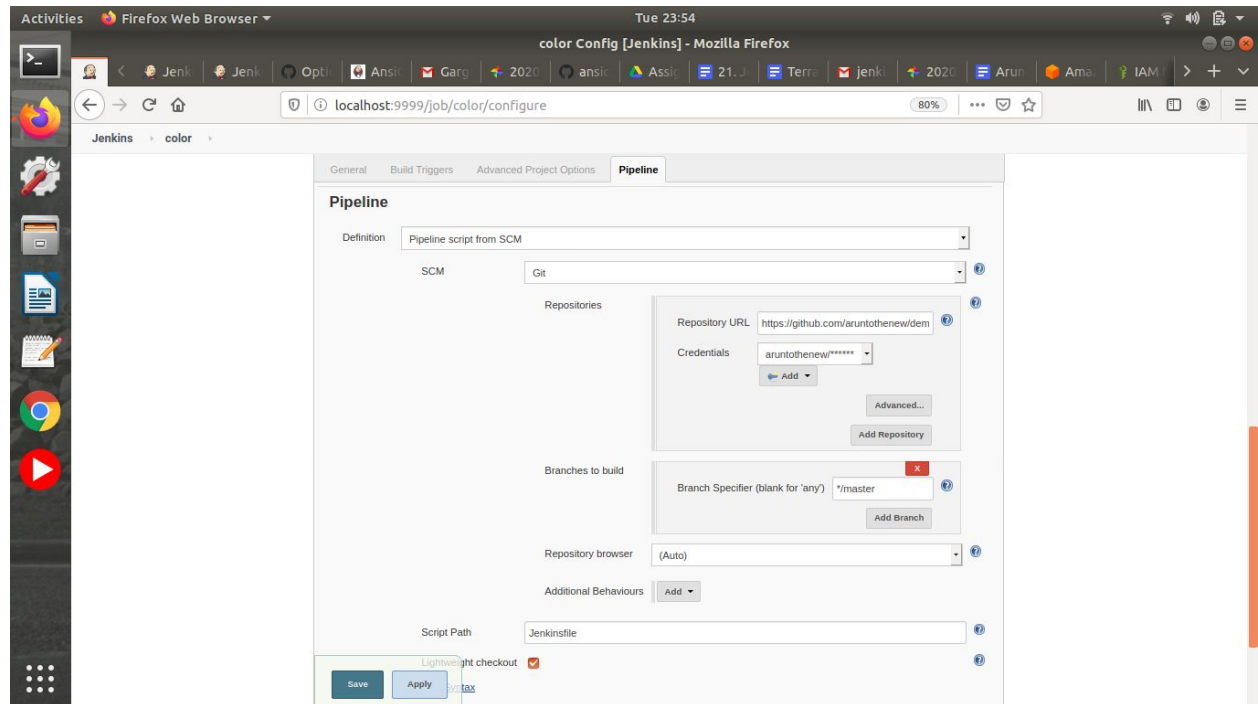


**2. Create a shared library function to convert error and success output into a colorful output and use it in the upcoming questions(Hint: use ANSI color).**

Create repo and create jenkins file and vars folder



Configure pipeline



## Build the pipeline

```

23:12:23 tomcat latest 6ab907c973d2 11 days ago 528MB
23:12:23 nginx latest ed21b7a8aee9 3 weeks ago 127MB
[Pipeline] echo
23:12:23 INFO: Successfully Executed
23:12:23
[Pipeline] sh
23:12:24 + docker image ls
23:12:24 /var/lib/jenkins/workspace/sharedlib@tmp/durable-8346f4d6/script.sh: 1:
/var/lib/jenkins/workspace/sharedlib@tmp/durable-8346f4d6/script.sh: docker: not found
[Pipeline] echo
23:12:24 WARNING: 127
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Git Commit Id)
[Pipeline] script
[Pipeline] {
[Pipeline] sh

```

## 3. Create a function in the same shared library to output git commitID.

```

stage('Git Commit Id') {
  steps {
    script{
      def gitId=sh(script:'git rev-parse HEAD', returnStdout: true)
      myscrip[0].gitCommitId(gitId)
    }
  }
}

```

```

23:12:24 + git rev-parse HEAD
[Pipeline] echo
23:12:24 [Git Commit ID]_ 0d139c4465e6f78d7caf7af5eccf122710868e
23:12:24
23:12:24
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // ansiColor
[Pipeline] }
[Pipeline] // timestamps
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

#### 4. Take a sample react application and deploy it on EKS

- a. You can use this repo or any other sample (<https://github.com/gothinkster/react-redux-realworld-example-app>).
- b. Create a Dockerfile for react application
- c. Build and publish the image to ECR (create ECR repo of your name) and the image must have the git commit id in its name.
- d. Deploy this image on EKS.

**Resources werent available so Prashant told us to leave this question**