

# 1. Create a Jenkins pipeline Job to delete redundant docker images daily at 1 AM UTC.

☒ Poll SCM

Schedule

No schedules so will only run due to SCM changes if triggered by a post-commit hook

## Pipeline

Definition

Pipeline script

Script

```
1 pipeline {
2   agent { label 'master' }
3   stages {
4     stage('build') {
5       steps {
6         echo "removing images"
7         sh "docker image prune -a -f"
8       }
9     }
10  }
11 }
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

```
ayush@ayush:~/Downloads$ docker image ls
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE
nginx               latest            ed21b7a8aee9      13 days ago       127MB
ayushme001/jenkins-docker  2043981ab0aba96c2536c9df6716e31b435f6998  1103327838df      4 weeks ago       84.9MB
ayushme001/jenkins-docker  latest           1103327838df      4 weeks ago       84.9MB
ayushme001/jenkins-docker  v1              1103327838df      4 weeks ago       84.9MB
nginx               v1              80d6f897eed       6 weeks ago       216MB
ubuntu             latest           72300a873c2c      7 weeks ago       64.2MB
openjdk            8-jre-alpine    f7a292bbb70c      11 months ago     84.9MB
ayush@ayush:~/Downloads$
```

## Console Output

```
Started by user ayushtripathi
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/ques-1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] echo
removing images
[Pipeline] sh
+ docker image prune -a -f
Deleted Images:
untagged: ubuntu:latest
untagged: ubuntu@sha256:04d48df82c938587820d7b6006f5071dbbffcceb7ca01d2814f81857c631d44df
untagged: ayushme001/jenkins-docker:2043981ab0aba96c2536c9df6716e31b435f6998
untagged: ayushme001/jenkins-docker:latest
untagged: ayushme001/jenkins-docker:v1
untagged: ayushme001/jenkins-docker@sha256:be8f77552125725e66fe3b5e3dfb69c5d08679882f9781c90ffe
deleted: sha256:1103327838df69f011d5276a1284d8fca3338cd62cab795e4ca14e7ac68dd343
deleted: sha256:b27593dbedc1a07d02aed98a3d06c67bbc282c4b76c8a934a5428852e1d1f72f
deleted: sha256:19561cf79013c4f360691f5e7bfbe9ad32f221c789035bcd803705d4984624b2
untagged: openjdk:8-jre-alpine
untagged: openjdk@sha256:f362b165b870ef129cbe730f29065ff37399c0aa8bcab3e44b51c302938c9193
deleted: sha256:f7a292bbb70c4ce57f7704cc03eb09e299de9da19013b084f138154421918cb4
deleted: sha256:aba8cd4318a8a15f59f305368e49cc3bbd0d6d72171a0b24d08cc00513b14ef4
deleted: sha256:c318fbba92645a83843c387bd78497e7b143ecc7f5ed99dcb7d38fc62442c02a
deleted: sha256:f1b5933fe4b5f49bbe8258745cf396afe07e625bdab3168e364daf7c956b6b81
```

```
ayush@ayush:~/Downloads$ docker images
REPOSITORY          TAG                 IMAGE ID           CREATED            SIZE
nginx                latest             ed21b7a8aee9      13 days ago       127MB
nginx                v1                 80d6f897eed       6 weeks ago       216MB
ayush@ayush:~/Downloads$
```

2. Create a shared library function to convert error and success output into a colourfull output and use it in the upcoming questions(Hint: use ANSI color).
3. Create a function in the same shared library to output git commitID.

# Installing Plugins/Upgrades

## Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Docker API  Success

Docker  Success

Loading plugin extensions  Success

AnsiColor  Success

Loading plugin extensions  Success

➡ [Go back to the top page](#)  
(you can start using the installed plugins right away)

➡ ☐ Restart Jenkins when installation is complete and no jobs are running

## Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL https://github.com/ayushme001/jenkins-ansi.

Credentials - none -

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any') \*/master

Add Branch

Repository browser (Auto)

Additional Behaviours Add

Script Path Jenkinsfile

```

ayush@ayush:~/github/jenkins-ansi$ cat Jenkinsfile
#!/groovy
library identifier: 'jenkins-ansi@master', retriever: modernSCM(
    [$class: 'GitSCMSource',
     remote: 'https://github.com/ayushme001/jenkins-ansi'])
pipeline {
    agent any
    options{
        timestamps()
        ansiColor('xterm')
    }
    stages {
        stage('List All Images') {
            steps {
                script{
                    sh ''' docker image ls
                    '''
                    myscript.info("Successfully Executed")
                    def ret = sh(script: 'dooocker image ls', returnStatus: true)
                    myscript.warning(ret)
                }
            }
        }
        stage('Git Commit Id') {
            steps {
                script{
                    def gitId=sh(script:'git rev-parse HEAD', returnStdout: true)
                    myscript.gitCommitId(gitId)
                }
            }
        }
    }
}

```

```

ayush@ayush:~/github/jenkins-ansi/vars$ cat myscript.groovy
def info(message) {
    echo "\033[4;32mINFO:\033[0m \033[1;33m${message}\033[0m\n"
}

def warning(message) {
    echo "WARNING: \033[1;31;43m${message}\033[0m"
}

def gitCommitId(message)
{
    echo "\033[4;32m[Git Commit ID] \033[0m \033[1;33m ${message} \033[0m\n"
}
ayush@ayush:~/github/jenkins-ansi/vars$

```

```

912MB
17:10:17 nginx latest
127MB
17:10:17 nginx v1
216MB
[Pipeline] echo
17:10:17 INFO: Successfully Executed
17:10:17
[Pipeline] sh
17:10:17 + doooocker image ls
17:10:17 /var/lib/jenkins/workspace/ques-2-3-pipeline@tmp/durable-54c7932a/script.sh: 1: /var/lib/jenk
pipeline@tmp/durable-54c7932a/script.sh: doooocker: not found
[Pipeline] echo
17:10:17 WARNING: 127
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Git Commit Id)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
17:10:18 + git rev-parse HEAD
[Pipeline] echo
17:10:18 [Git Commit ID] 848331d3ffffbae3e5c1a4f1ec6d5f2f87b0d3668
17:10:18
17:10:18

```

#### 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 image to ECR (create ECR repo of your name) and image must have the git commit id in its name.
- d. Deploy this image on EKS.
- e. Send Slack notification/Mail/google chat notification for build pass, abort and fail.



```
jenkins@ayush:~$ aws ecr get-login-password --region us-east-1 | docker login --
username AWS --password-stdin 187632318301.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/co
nfig.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
jenkins@ayush:~$
```

## Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL https://github.com/ayushme001/ttnbootcamp

Credentials - none -

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any') \*/jenkins

Add Branch

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat Jenkinsfile
pipeline {
    agent any
    stages {
        stage('Docker Build') {
            steps {
                echo 'building docker'
                sh 'docker build -t 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:${git rev-parse HEAD} .'
            }
        }
        stage('pushing image to ECR') {
            steps {
                echo 'pushing image'
                sh 'docker push 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:${git rev-parse HEAD}'
            }
        }
        stage('Deploying on eks') {
            steps {
                echo 'deploying on eks'
                sh 'kubectl create -f deployment.yml'
                sh 'sleep 10'
                sh 'kubectl get pods'
                sh 'kubectl create -f loadbalancer.yml'
                sh 'sleep 5'
                sh 'kubectl get svc'
                sh 'kubectl describe pods'
            }
        }
    }
    post {
        success {
            emailx body: 'Its a success', recipientProviders: [[class: 'DevelopersRecipientProvider'], [class: 'RequesterRecipientProvider']], subject: 'success build'
        }
        failure {
            emailx body: 'Its a failed', recipientProviders: [[class: 'DevelopersRecipientProvider'], [class: 'RequesterRecipientProvider']], subject: 'failed build'
        }
    }
}
```

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat deployment.yml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: react-app
spec:
  selector:
    matchLabels:
      app: react-app
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: react-app
    spec:
      containers:
      - name: react-app
        image: 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:latest
        ports:
        - containerPort: 4100
```

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat loadbalancer.yml
apiVersion: v1
kind: Service
metadata:
  name: react-lb
spec:
  type: LoadBalancer
  selector:
    app: react-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 4100

ayush@ayush:~/github/ttnbootcamp-tothenew$
```

## Console Output

```
Started by user ayushtripathi
Obtained Jenkinsfile from git https://github.com/ayushme001/ttnbootcamp-tothenew
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/ques-4-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ayushme001/ttnbootcamp-tothenew # timeout=10
Fetching upstream changes from https://github.com/ayushme001/ttnbootcamp-tothenew
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/ayushme001/ttnbootcamp-tothenew +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/jenkins^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/jenkins^{commit} # timeout=10
Checking out Revision 454f9660c1cf84ee5413eb097e3c399bfb0c2c4f (refs/remotes/origin/jenkins)
> git config core.sparsecheckout # timeout=10
> git checkout -f 454f9660c1cf84ee5413eb097e3c399bfb0c2c4f # timeout=10
Commit message: "added"
> git rev-list --no-walk b70db3e0995c2eef7fbc061bc06cae9dea6e0425 # timeout=10

[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Docker Build)
[Pipeline] echo
building docker
[Pipeline] sh
+ git rev-parse HEAD
+ docker build -t 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:454f9660c1cf84ee5413eb097e3c399bfb0c2c4f .
Sending build context to Docker daemon 16.22MB

Step 1/7 : FROM node:10
---> bd83fcef19d
Step 2/7 : WORKDIR /usr/src/app
---> Using cache
---> 1887595987e4
Step 3/7 : RUN apt install git -y
---> Using cache
---> a8707dbb9735
Step 4/7 : RUN git clone https://github.com/gothinkster/react-redux-realworld-example-app.git /usr/src/app
---> Using cache
---> ed89ad1a02a0
Step 5/7 : RUN npm install
---> Using cache
---> b2aad42f8d03
Step 6/7 : CMD [ "npm", "start" ]
---> Using cache
---> 1111b9aa1302
```

---



```
[Pipeline] stage
[Pipeline] { (pushing image to ECR)
[Pipeline] echo
pushing image
[Pipeline] sh
+ git rev-parse HEAD
+ docker push 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:454f9660c1cf84ee5413eb097e3c399bfb0c2c4f
The push refers to repository [187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr]
0de1639f1af5: Preparing
614c9e0e36c3: Preparing
bdabb219d1dc: Preparing
adacd70e0959: Preparing
7a139c38a90f: Preparing
b58f9ccd46af: Preparing
8247c5671f2d: Preparing
45ac74adb5b4: Preparing
d485cbb6a5e: Preparing
391c89959588: Preparing
588545a7a2a3: Preparing
8452468a5e50: Preparing
55b19a5e648f: Preparing
8247c5671f2d: Waiting
45ac74adb5b4: Waiting
d485cbb6a5e: Waiting
391c89959588: Waiting
588545a7a2a3: Waiting
8452468a5e50: Waiting
```

---

```
[Pipeline] { (Deploying on eks)
[Pipeline] echo
deploying on eks
[Pipeline] sh
+ kubectl create -f deployment.yml
deployment.apps/react-app created
[Pipeline] sh
+ sleep 10
[Pipeline] sh
+ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
react-app-54b48d68b-9zvff          1/1     Running   0           12s
react-app-54b48d68b-dlzhz          1/1     Running   0           12s
[Pipeline] sh
+ kubectl create -f loadbalancer.yml
service/react-lb created
[Pipeline] sh
+ sleep 5
[Pipeline] sh
+ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP
PORT(S)                            AGE
kubernetes                          ClusterIP           10.100.0.1      <none>
443/TCP                             55m
react-lb                            LoadBalancer       10.100.214.69   a372daaab5e484babbb114407a15e2f8-231714168.us-east-1.elb.amazonaws.com
80:32579/TCP                        8s
```

```

Normal Scheduled 24s default-scheduler Successfully assigned default/react-app-54b48d68b-
tkhzz to ip-192-168-193-232.ec2.internal
Normal Pulling 23s kubelet, ip-192-168-193-232.ec2.internal Pulling image "187632318301.dkr.ecr.us-east-
1.amazonaws.com/ayush-ecr:latest"
Normal Pulled 23s kubelet, ip-192-168-193-232.ec2.internal Successfully pulled image "187632318301.dkr.ecr.us-
east-1.amazonaws.com/ayush-ecr:latest"
Normal Created 23s kubelet, ip-192-168-193-232.ec2.internal Created container react-app
Normal Started 22s kubelet, ip-192-168-193-232.ec2.internal Started container react-app
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] emailxnt
Sending email to: ayush.tripathi@tothenew.com
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## success build Inbox x



**address not configured yet**

to me ▾

Its a success

↩ Reply

➡ Forward

← → ↻ ⓘ Not secure | a372daab5e484babbb114407a15e2f8-231714168.us-east-1.elb.amazonaws.com  
 🍏 Apple Final Cu... 🍏 Apple Final Cu... 📁 On Demand Tr... 📁 On Demand Tr... 🗨 GitHub Stude... 🗨 GitHub Stude...

conduit

# conduit

A place to share your knowledge.

Global Feed



juuzek

Tue Apr 14 2020




**Test**

abolt jur mader

Final version

Images (1)

 Find Images

<input type="checkbox"/>	Image tag	Image URI	Pushed at ▼
<input type="checkbox"/>	latest, b70db3e0995c2eef7fbc061bc06cae9dea6e0425, 454f9660c1cf84ee5413eb097e3c399bfb0c2c4f	 187632318301.dkr.ecr.us-east-1.amazonaws.com/ayush-ecr:latest	04/14/20, 11:48:34 AM