



# Deploying web-server using Jenkins-Groovy-code and Kubernetes

**Jenkins** is an open-source automation tool **used to** build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

But there is one task of making a job/pipeline which we do manually till now in all my tasks. So somehow, we say that this thing is not automated.

But Jenkins provides one way to make the job by writing code in Groovy language. For this, Jenkins has one plugin named ***Job\_DSL***. You need to install this only.

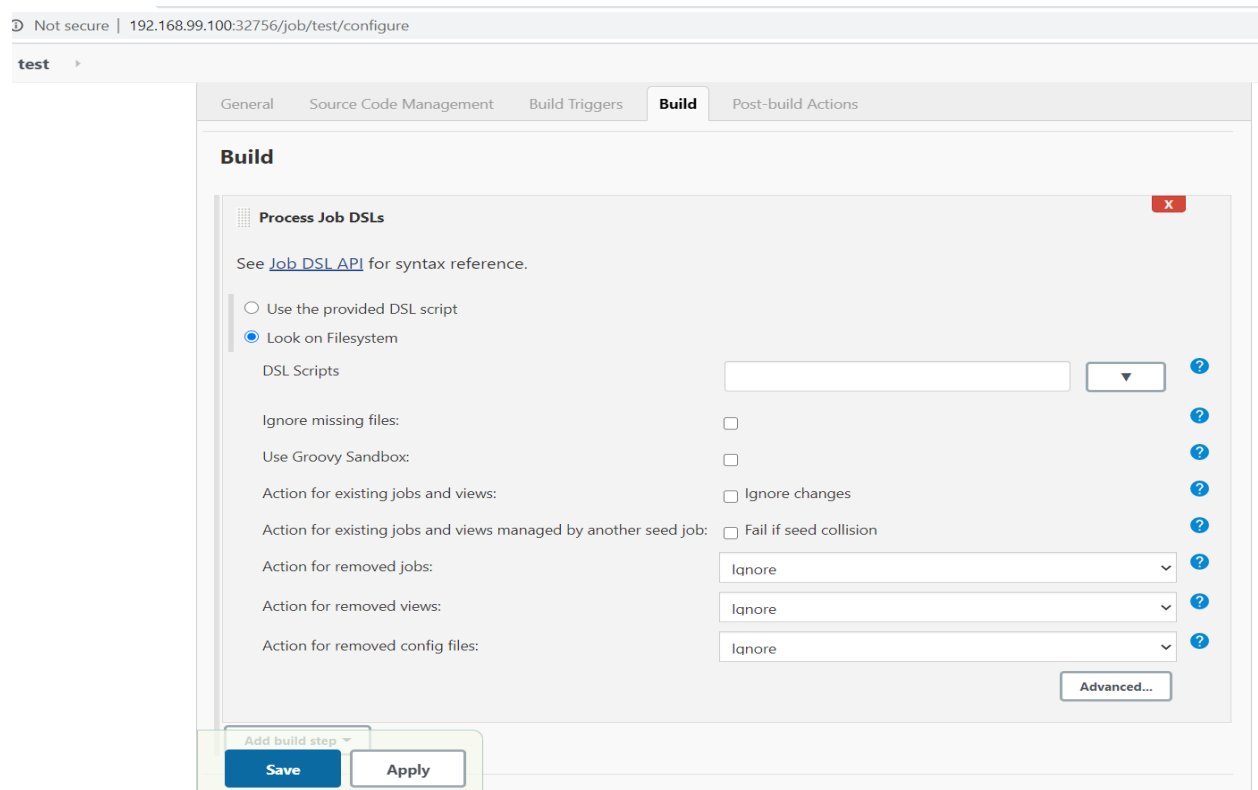
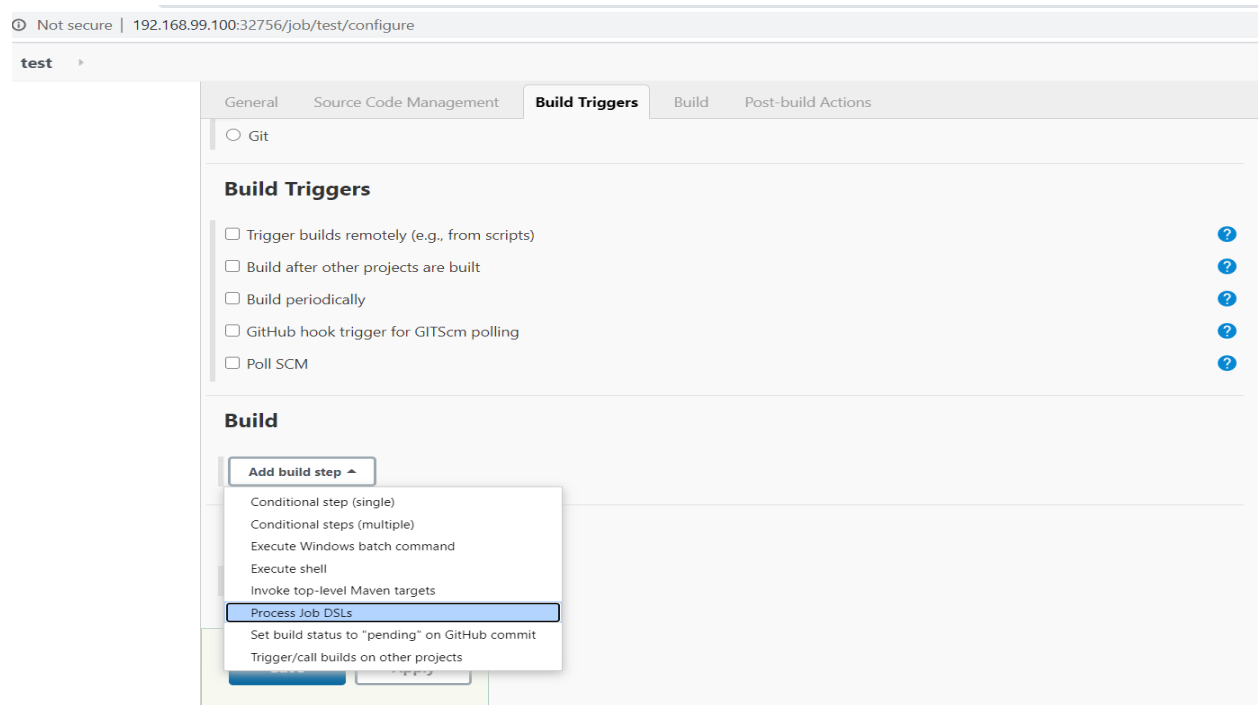
The screenshot shows the Jenkins Update Center interface. The browser address bar indicates the URL is 192.168.99.100:32756/pluginManager/available. The Jenkins logo and search bar are at the top. The left sidebar shows navigation links: Back to Dashboard, Manage Jenkins, and Update Center. The main content area shows a search for 'Job DSL' with tabs for Updates, Available, Installed, and Advanced. A table lists available plugins:

Install	Name	Version	Released
<input checked="" type="checkbox"/>	<a href="#">Job DSL</a> Build Tools This plugin allows Jobs and Views to be defined via DSLs	1.77	4 mo 12 days ago
<input type="checkbox"/>	<a href="#">XML Job to Job DSL</a> Use this plugin to convert your jobs into DSL Groovy scripts	0.1.13	7 mo 12 days ago

At the bottom, there are buttons for 'Install without restart', 'Download now and install after restart', and 'Check now'. A status message indicates 'Update information obtained: 7 min 7 sec ago'.

Now, you are ready to use Groovy code.

After this, Go to create job and in Build, You find one option of Process Job DSL's, click on this.



You have both the option like if you want to give code directly, then choose **Use the provided DSL script** and give the code here. But If you want to run your code as a script, then choose **Look on Filesystem** and give the name of script with **.groovy** extension. Make sure you load your file on Jenkins workspace.

Now developers push the code in GitHub, and we need to create only one job for fetch the code from GitHub, this job is also known as ***Seed Job***.

#### Task Overview:

1. Create container image that's has Jenkins installed using dockerfile.
2. When we launch this image, it should automatically start Jenkins service in the container.
3. Create a job chain of job1, job2, job3 and job4 using build pipeline plugin in Jenkins
4. Seed Job: Pull the Github repo automatically when some developers push the repo to Github.
5. Further on jobs should be pipeline using written code using Groovy language by the developer
6. Job1 :
  1. By looking at the code or program file, Jenkins should automatically start the respective language interpreter installed image container to deploy code on top of Kubernetes ( eg. If code is of PHP, then Jenkins should start the container that has PHP already installed )
  2. Expose your pod so that testing team could perform the testing on the pod
  3. Make the data to remain persistent using PVC ( If server collects some data like logs, other user information )
7. Job2: Test your app if it is working or not.

**8. Job3: if app is not working, then send email to the developer with error messages and redeploy the application after code is being edited by the developer**

**Tools used:**

- Jenkins
- Kubernetes
- Docker
- **Run Jenkins server using Docker image on the top of Kubernetes:**

Dockerfile for creating image...

```
FROM centos
```

```
RUN sudo yum install wget -y
```

```
RUN sudo yum install git -y
```

```
RUN sudo yum install sudo -y
```

```
RUN sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
RUN sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

```
RUN sudo yum install java-11-openjdk.x86_64 -y
```

```
RUN sudo yum install jenkins -y
```

```
RUN sudo yum install net-tools -y
```

```
RUN sudo yum install python36 -y
```

```
RUN sudo yum install /sbin/service -y
```

```
RUN sudo yum install initscripts -y
```

```
RUN sudo yum install httpd -y
```

```
RUN sudo yum install git -y
```

```
RUN sudo yum install vim -y
```

```
RUN sudo yum install openssh-clients -y
```

```
RUN echo -e "jenkins ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

```
RUN sudo curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s  
https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
```

```
RUN sudo chmod +x ./kubectl
```

RUN `sudo mv ./kubectl /usr/bin/`

RUN `sudo mkdir /kube_cert/`

COPY `c* /kube_cert/`

COPY `config /root/.kube/config`

COPY `yml /root/yml`

RUN `sudo yum install ncurses -y`

CMD `java -jar /usr/lib/jenkins/jenkins.war && /bin/bash`

EXPOSE 8080

After creating the image, push the image into Docker Hub. Only thing you need to change is files of .kube folder.

You can use my image also by pull using command ***docker pull anuddeeph/Jenkins\_k8s:v1***

Now, this is the code for running Jenkins on top of Kubernetes using the YAML file.

```
apiVersion: v1
kind: Service
metadata:
  name: jenkins
  labels:
    app: jenkins
spec:
  ports:
    - port: 8080
  selector:
    app: jenkins
  type: NodePort
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: jenkins-claim
  labels:
    app: jenkins
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
```

```

metadata:
  name: jenkins
  labels:
    app: jenkins
spec:
  selector:
    matchLabels:
      app: jenkins
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: jenkins
    spec:
      containers:
      - image: anuddeeph/Jenkins_k8s:v1
        name: jenkins
        ports:
        - containerPort: 8080
          name: jenkins
        volumeMounts:
        - name: jenkins-storage
          mountPath: /root/.jenkins/
      volumes:
      - name: jenkins-storage
        persistentVolumeClaim:
          claimName: jenkins-claim

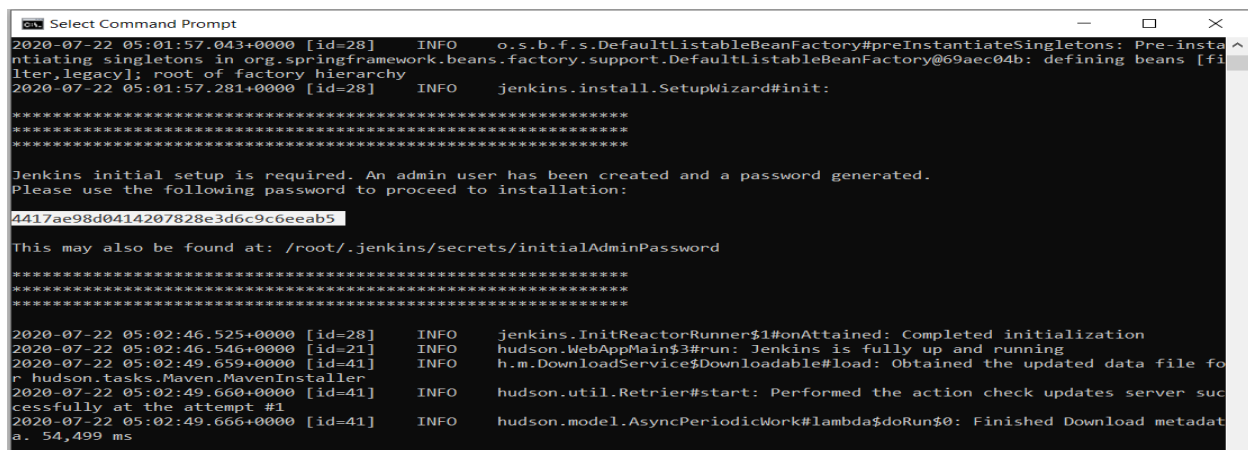
```

To run this, we using the command...

***kubectl apply -f file\_name***

Now Jenkins server is running...

First time when you run, this ask for password, for getting password, RUN ***kubectl logs pod\_name***



```

Select Command Prompt
2020-07-22 05:01:57.043+0000 [id=28] INFO o.s.b.f.s.DefaultListableBeanFactory#preInstantiateSingletons: Pre-insta
ntiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@69aec04b: defining beans [fi
lter,legacy]; root of factory hierarchy
2020-07-22 05:01:57.281+0000 [id=28] INFO jenkins.install.SetupWizard#init:
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
4417ae98d0414207828e3d6c9c6eeab5
This may also be found at: /root/.jenkins/secrets/initialAdminPassword
*****
*****
2020-07-22 05:02:46.525+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2020-07-22 05:02:46.546+0000 [id=21] INFO hudson.WebAppMain$3#run: Jenkins is fully up and running
2020-07-22 05:02:49.659+0000 [id=41] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file fo
r hudson.tasks.Maven.MavenInstaller
2020-07-22 05:02:49.660+0000 [id=41] INFO hudson.util.Retrier#start: Performed the action check updates server suc
cessfully at the attempt #1
2020-07-22 05:02:49.666+0000 [id=41] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$0: Finished Download metadat
a. 54,499 ms

```

Now come to next step which is main part...

## Jobs in Jenkins:

As I say, we need to create only one Seed job only, which create all the job automatically.

- **Seed Job (Pull the GitHub repo when developer pushes some code):**

Whenever the developer pushes any code in GitHub, this job automatically detects and copy in host OS and create other Jobs.

The screenshot displays the Jenkins configuration interface for a job named 'seed job'. The browser address bar shows the URL '192.168.99.100:32756/job/seed%20job/configure'. The 'General' tab is active, showing a description 'This is my seed\_job' and a list of checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Disable this project', and 'Execute concurrent builds if necessary'. The 'Source Code Management' section is expanded, showing 'Git' as the selected provider. The 'Repository URL' is set to 'https://github.com/Anuddeeph/jenkins\_dsl\_code.git', and the 'Credentials' are set to '- none -'. There are 'Save', 'Apply', and 'Add Repository' buttons at the bottom. The 'Add Repository' button is highlighted in green.

ig [Jenkins] x Anuddeeph/jenkins\_dsl\_code x +

▲ Not secure | 192.168.99.100:32756/job/seed%20job/configure

seed job ▶

General Source Code Management **Build Triggers** Build Post-build Actions

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule \*\*\*\*\* ?

⚠ Do you really mean "every minute" when you say "\*\*\*\*\*"? Perhaps you meant "H\*\*\*\*\*" to poll once per hour

Would last have run at Wednesday, July 22, 2020 at 6:23:12 AM Coordinated Universal Time; would next run at Wednesday, July 22, 2020 at 6:23:12 AM Coordinated Universal Time.

☐ Ignore post-commit hooks ?

**Build**

Execute shell x ?

Command `mkdir -p /mycode`  
`cp -rvf * /mycode/`

Save Apply available environment variables

ig [Jenkins] x Anuddeeph/jenkins\_dsl\_code x +

▲ Not secure | 192.168.99.100:32756/job/seed%20job/configure

seed job ▶

General Source Code Management Build Triggers **Build** Post-build Actions

Process Job DSLs x ?

See [Job DSL API](#) for syntax reference.

☐ Use the provided DSL script

☒ Look on Filesystem

DSL Scripts

code.groovy ?

Ignore missing files: ☐ ?

Use Groovy Sandbox: ☐ ?

Action for existing jobs and views: ☐ Ignore changes ?

Action for existing jobs and views managed by another seed job: ☐ Fail if seed collision ?

Action for removed jobs: ignore ?

Action for removed views: ignore ?

Action for removed config files: ignore ?

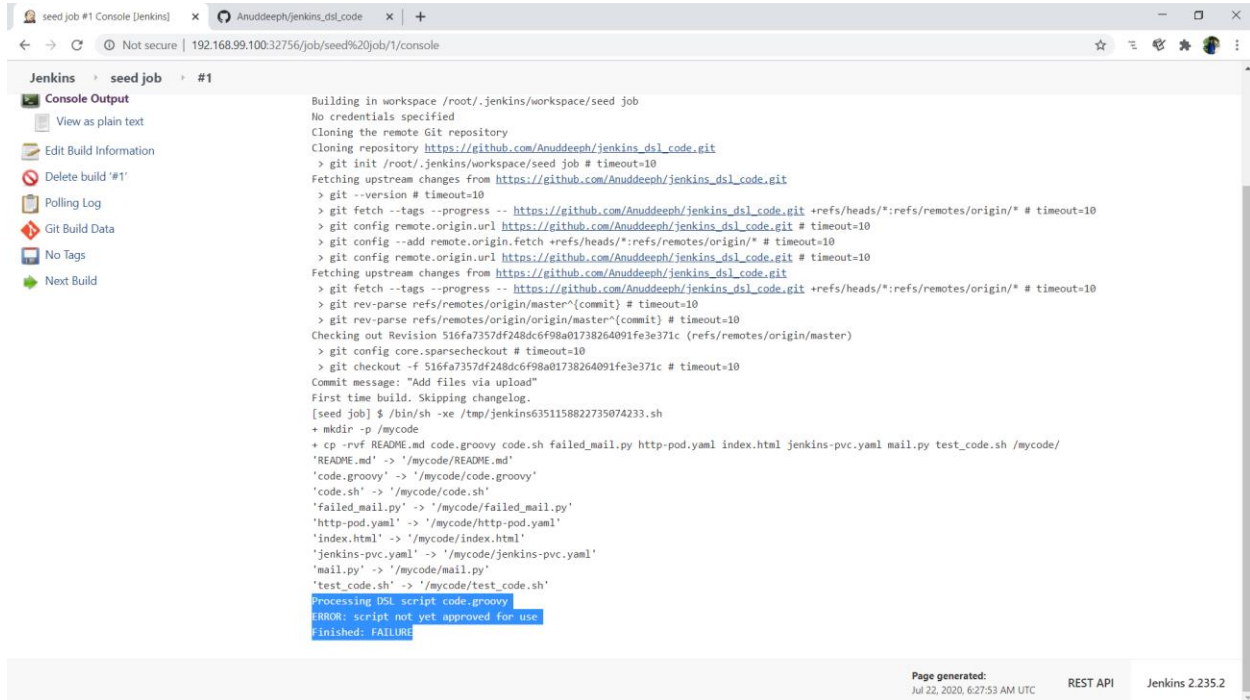
Advanced...

Save Apply



So let's see what happens when we build this job...

First time, when you run it, It gives an error...



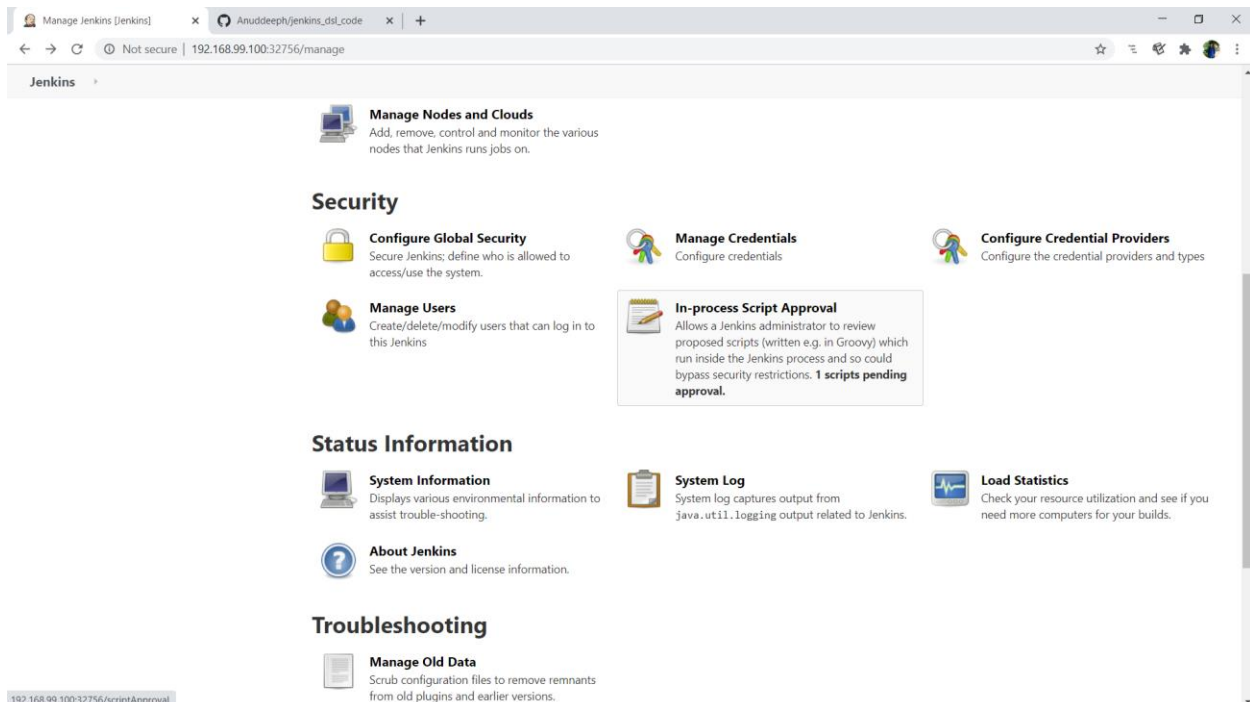
Jenkins > seed job > #1

**Console Output**

Building in workspace /root/.jenkins/workspace/seed job  
No credentials specified  
Cloning the remote Git repository  
Cloning repository [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git)  
> git init /root/.jenkins/workspace/seed job # timeout=10  
Fetching upstream changes from [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git)  
> git --version # timeout=10  
> git fetch --tags --progress -- [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git) +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git config remote.origin.url [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git) # timeout=10  
> git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git config remote.origin.url [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git) # timeout=10  
Fetching upstream changes from [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git)  
> git fetch --tags --progress -- [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git) +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10  
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10  
Checking out Revision 516fa7357df248dc6f98a01738264091fe3e371c (refs/remotes/origin/master)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 516fa7357df248dc6f98a01738264091fe3e371c # timeout=10  
Commit message: "Add files via upload"  
First time build. Skipping changelog.  
[seed job] \$ /bin/sh -xe /tmp/jenkins6351158822735074233.sh  
+ mkdir -p /mycode  
+ cp -rvf README.md code.groovy code.sh failed\_mail.py http-pod.yaml index.html jenkins-pvc.yaml mail.py test\_code.sh /mycode/  
'README.md' -> '/mycode/README.md'  
'code.groovy' -> '/mycode/code.groovy'  
'code.sh' -> '/mycode/code.sh'  
'failed\_mail.py' -> '/mycode/failed\_mail.py'  
'http-pod.yaml' -> '/mycode/http-pod.yaml'  
'index.html' -> '/mycode/index.html'  
'jenkins-pvc.yaml' -> '/mycode/jenkins-pvc.yaml'  
'mail.py' -> '/mycode/mail.py'  
'test\_code.sh' -> '/mycode/test\_code.sh'  
Processing DSL script code.groovy  
**ERROR: script not yet approved for use**  
Finished: FAILURE

Page generated: Jul 22, 2020, 6:27:53 AM UTC REST API Jenkins 2.235.2

To remove this, go to **Jenkins configuration -> In Process Script Approval** and approve the script.



Jenkins > Manage Jenkins [Jenkins] > Anuddeeph/jenkins\_dsl\_code

**Manage Nodes and Clouds**  
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

**Security**

- Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials**  
Configure credentials
- Configure Credential Providers**  
Configure the credential providers and types
- Manage Users**  
Create/delete/modify users that can log in to this Jenkins
- In-process Script Approval**  
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions. **1 scripts pending approval.**

**Status Information**

- System Information**  
Displays various environmental information to assist trouble-shooting.
- System Log**  
System log captures output from java.util.logging output related to Jenkins.
- Load Statistics**  
Check your resource utilization and see if you need more computers for your builds.
- About Jenkins**  
See the version and license information.

**Troubleshooting**

- Manage Old Data**  
Scrub configuration files to remove remnants from old plugins and earlier versions.

192.168.99.100:32756/scriptApproval

In-process Script Approval [Jenkins] x Anuddeeph/jenkins\_dsl\_code x +

Not secure | 192.168.99.100:32756/scriptApproval/

# Jenkins

ScriptApproval

Approve / Deny Groovy script in seed job:

```

job('job1-deploy-web-server') {
    description('This job is for deploy the web-server')
    steps {
        shell('sh /mycode/code.sh')
    }
}
job('job2-test_code') {
    description('This job is for test the code and sent notification to developer')
    triggers {
        scm '@daily'
    }
    upstream {
        upstreamProjects('job1-deploy-web-server')
        threshold('SUCCESS')
    }
    steps {
        shell('sh /mycode/test_code.sh')
    }
}
buildPipelineView('Task 6') {
    filterBuildQueue()
    filterExecutors()
    title('Project for deploy web-server')
    displayedBuilds(5)
    selectedJob('job1-deploy-web-server')
    alwaysAllowManualTrigger()
    showPipelineParameters()
    refreshFrequency(60)
}

```

You can also remove all previous script approvals: [Clear Approvals](#)

Now again build and this time output is...

seed job #3 Console [Jenkins] x Anuddeeph/jenkins\_dsl\_code x +

Not secure | 192.168.99.100:32756/job/seed%20job/3/console

# Jenkins

seed job #3

Console Output

View as plain text

Edit Build Information

Delete build '#3'

Git Build Data

No Tags

Previous Build

```

Running as SYSTEM
Building in workspace /root/.jenkins/workspace/seed job
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/Anuddeeph/jenkins_dsl_code.git # timeout=10
Fetching upstream changes from https://github.com/Anuddeeph/jenkins_dsl_code.git
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/Anuddeeph/jenkins_dsl_code.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 516fa7357df248dc6f98a01738264091fe3e371c (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 516fa7357df248dc6f98a01738264091fe3e371c # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 516fa7357df248dc6f98a01738264091fe3e371c # timeout=10
[seed job] $ /bin/sh -xe /tmp/jenkins8869318816647901183.sh
+ mkdir -p /mycode
+ cp -rvf README.md code.groovy code.sh failed_mail.py http-pod.yaml index.html jenkins-pvc.yaml mail.py test_code.sh /mycode/
'README.md' -> '/mycode/README.md'
'code.groovy' -> '/mycode/code.groovy'
'code.sh' -> '/mycode/code.sh'
'failed_mail.py' -> '/mycode/failed_mail.py'
'http-pod.yaml' -> '/mycode/http-pod.yaml'
'index.html' -> '/mycode/index.html'
'jenkins-pvc.yaml' -> '/mycode/jenkins-pvc.yaml'
'mail.py' -> '/mycode/mail.py'
'test_code.sh' -> '/mycode/test_code.sh'
Processing DSL script code.groovy
Added items:
    GeneratedJob{name='job1-deploy-web-server'}
    GeneratedJob{name='job2-test_code'}
Added views:
    GeneratedView{name='Task 6'}
Finished: SUCCESS

```

Page generated: Jul 22, 2020, 6:30:54 AM UTC REST API Jenkins 2.235.2

**Project seed job**

This is my seed job

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

**Workspace**

**Recent Changes**

**Generated Items:**

- [job1-deploy-web-server](#)
- [job2-test\\_code](#)

**Generated Views:**

- [Task 6](#)

**Permalinks**

- [Last build \(#3\), 2 min 22 sec ago](#)
- [Last stable build \(#3\), 2 min 22 sec ago](#)
- [Last successful build \(#3\), 2 min 22 sec ago](#)
- [Last failed build \(#2\), 4 min 47 sec ago](#)
- [Last unsuccessful build \(#2\), 4 min 47 sec ago](#)
- [Last completed build \(#3\), 2 min 22 sec ago](#)

**Build History** [trend](#)

#	Time
#3	Jul 22, 2020, 6:30 AM
#2	Jul 22, 2020, 6:27 AM
#1	Jul 22, 2020, 6:25 AM

Atom feed for all Atom feed for failures

You can see that two jobs and one view are created here.

**Jenkins Dashboard**

**Jobs and Views Table:**

S	W	Name	Last Success	Last Failure	Last Duration
		job1:deploy-web-server	N/A	N/A	N/A
		job2:test_code	N/A	N/A	N/A
		seed job	4 min 23 sec - #3	6 min 48 sec - #2	3.5 sec

Icon: S M L

Legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

**Build Queue**

No builds in the queue.

**Build Executor Status**

1 Idle  
2 Idle

Page generated: Jul 22, 2020, 6:34:34 AM UTC REST API Jenkins 2.235.2

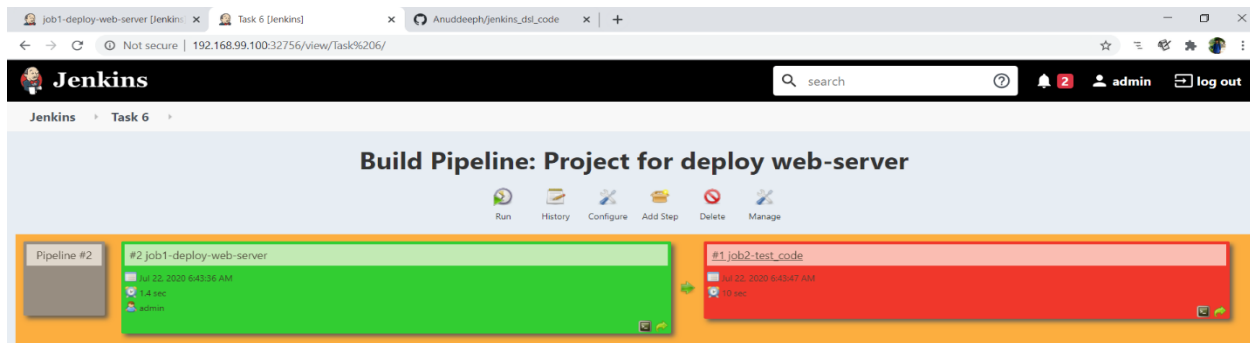
**Build Pipeline: Project for deploy web-server**

[Run](#) [History](#) [Configure](#) [Add Step](#) [Delete](#) [Manage](#)

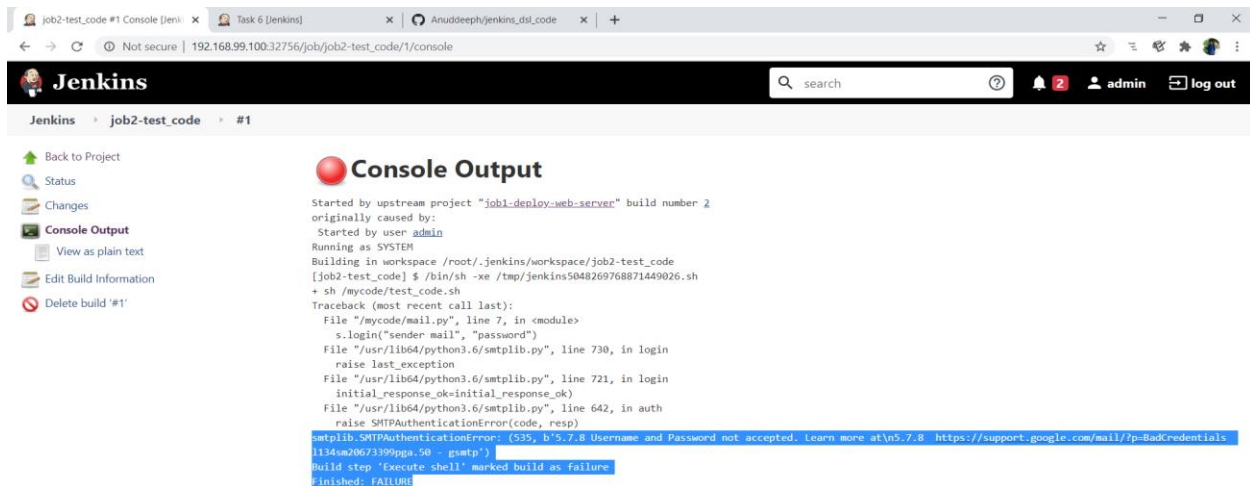
This view has no jobs associated with it. You can either add some existing jobs to this view or create a new job in this view.

And this is the pipeline view...

Push on RUN and your all jobs are running now... 🤖

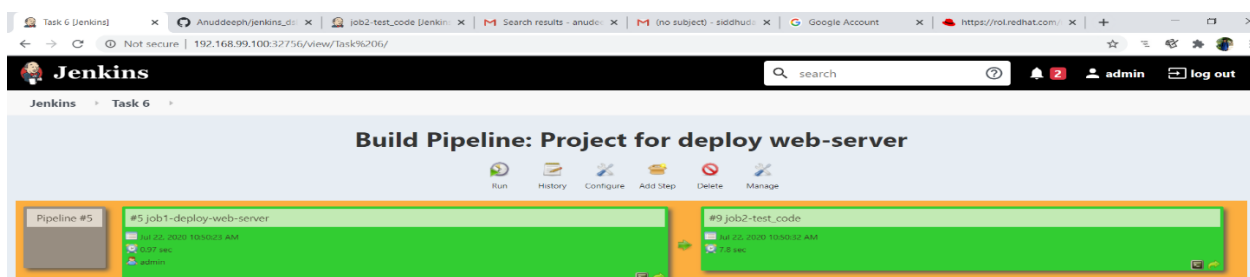


Job 2 is not running now; this conflict comes because I am not uploaded my mail password in GitHub. If you provided the right id/password, it works... 👍



Page generated:  
Jul 22, 2020, 6:45:53 AM UTC   REST API   Jenkins 2.235.2

After giving the required credentials,



Task 6 [Jenkins] | Anuddeeph/jenkins\_dsl\_code | job2-test\_code #9 Console [Jenkins]

Not secure | 192.168.99.100:32756/view/Task%206/job/job2-test\_code/9/console

# Jenkins

Jenkins > Task 6 > job2-test\_code > #9

- Back to Project
- Status
- Changes
- Console Output**
  - View as plain text
- Edit Build Information
- Delete build '#9'
- Previous Build

## Console Output

Started by upstream project "[job1-deploy-web-server](#)" build number [5](#) originally caused by:  
 Started by user [admin](#)  
 Running as SYSTEM  
 Building in workspace /root/.jenkins/workspace/job2-test\_code  
 [job2-test\_code] \$ /bin/sh -xe /tmp/jenkins11027070168506843248.sh  
 + sh /mycode/test\_code.sh  
 Finished: SUCCESS

Task 6 [Jenkins] | Anuddeeph/jenkins | job2-test\_code [Jenkins] | Search results - an | (no subject) - sidd | Inbox (4,973) - an | Google Account | https://rolredhat |

mail.google.com/mail/u/1/#inbox/FMfcgww/WjVcnPspcZgwcTQqrwgfnfm

Compose

Inbox 332

Starred

Snoozed

Important

Sent

Drafts 5

(no subject) > Inbox

siddhudarling1@gmail.com  
to bcc: me

Hey Developer, Our website is up.

4:19 PM (3 minutes ago)

Reply Forward

Yo...This completed and now this is much automated than previous tasks...

Now let's see what I write in my groovy file(code.groovy)...

- Job1(Deploy deployment for website):**

It launches the deployment with PVC and services. Here I have taken the example of webpages, so I have used the apache webserver.

```
job('job1-deploy-web-server') {
    description('This job is for deploy the web-server')
    steps {
        shell('sh /mycode/code.sh')
    }
}
```

And code.sh is...

```
if kubectl get pods | grep httpd
then
echo "service is running"
else
kubectl create -f /mycode/http-pod.yaml
sleep 20
fi
POD=$(kubectl get pod -l app=httpd -o
jsonpath="{.items[0].metadata.name}")
kubectl cp /mycode/*.html $POD:/var/www/html/
```

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is 192.168.99.100:32756/job/job1-deploy-web-server/configure. The Jenkins logo and name are in the top left. A search bar and user information (admin) are in the top right. The main content area shows the configuration for the job 'job1-deploy-web-server'. The 'General' tab is selected, showing a description 'This job is for deploy the web-server'. Below the description are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Disable this project', and 'Execute concurrent builds if necessary'. An 'Advanced...' button is also present. The 'Source Code Management' section shows 'None' selected. The 'Build Triggers' section is partially visible at the bottom, showing 'Save' and 'Apply' buttons.

job1-deploy-web-server Config | Task 6 [Jenkins] | Anuddeeph/jenkins\_dsl\_code

Not secure | 192.168.99.100:32756/job/job1-deploy-web-server/configure

Jenkins

Search

admin log out

Jenkins > job1-deploy-web-server

General Source Code Management Build Triggers Build Post-build Actions

Description This job is for deploy the web-server

[Plain text] Preview

☐ Discard old builds

☐ GitHub project

☐ This project is parameterized

☐ Disable this project

☐ Execute concurrent builds if necessary

Advanced...

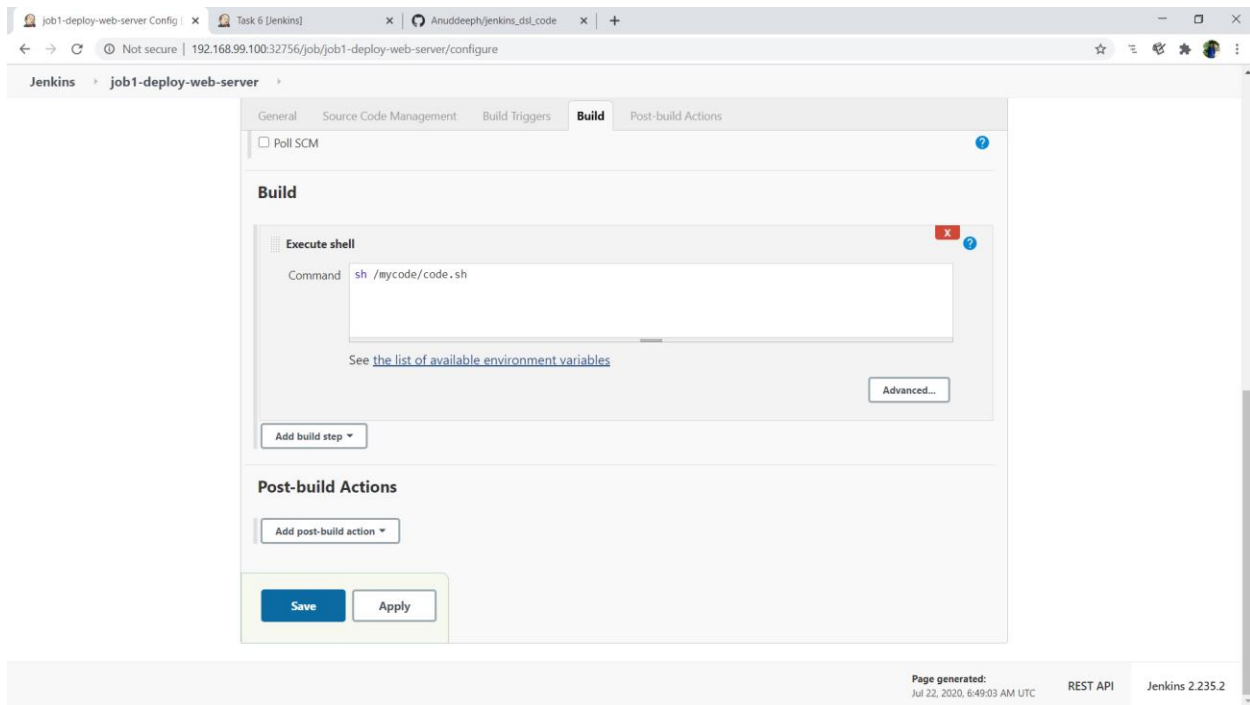
Source Code Management

☒ None

☐ Git

Build Triggers

Save Apply



The YAML code is launching for the website is below...

```
apiVersion: v1
kind: Service
metadata:
  name: httpd
  labels:
    app: httpd
spec:
  ports:
    - nodePort: 80
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: httpd
  type: NodePort
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: httpd-claim
  labels:
    app: httpd
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
---
```

```

apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: httpd
  labels:
    app: httpd
spec:
  selector:
    matchLabels:
      app: httpd
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: httpd
    spec:
      containers:
        - image: anuddeeph/apache-webserver-php
          name: httpd
          ports:
            - containerPort: 8080
              name: httpd
          volumeMounts:
            - name: httpd-storage
              mountPath: /var/www/html/
      volumes:
        - name: httpd-storage
          persistentVolumeClaim:
            claimName: httpd-claim

```

- **Job2(Testing that website is running or not) and Job3(Sent a mail to the developer):**

After successfully build, Job1 will trigger Job2 and it checks the website that is working or not. If not working, it sent a notification to the developer.

Groovy code for this...

```

job('job2-test_code') {
    description('This job is for test the code and sent notification
to developer')
    triggers {
        scm('@daily')
        upstream {
            upstreamProjects('job1-deploy-web-server')
            threshold('SUCCESS')
        }
    }
    steps {
        shell('sh /mycode/test_code.sh')
    }
}

```



I write one script for do test here(test\_code.sh)...

```
#!/bin/bash
    status=$(curl -o /dev/null -s -w "%http_code%"
http://192.168.99.101:32033)
    if [[ $status==200 ]]
    then
    python3 /mycode/mail.py
    else
    python3 /mycode/failed_mail.py
    fi
```

Config [Jenkins] x Task 6 [Jenkins] x Anuddeeph/jenkins\_dsl\_code x +

Not secure | 192.168.99.100:32756/job/job2-test\_code/configure

# Jenkins

job2-test\_code

General

Source Code Management

Build Triggers

Build

Post-build Actions

Description

This job is for test the code and sent notification to developer

[Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This project is parameterized

☐ Disable this project

☐ Execute concurrent builds if necessary

Advanced...

Source Code Management

☒ None

☐ Git

Build Triggers

Save

Apply

scripts)

job2-test\_code

General Source Code Management **Build Triggers** Build Post-build Actions

## Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Projects to watch job1-deploy-web-server

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule @daily ?

Would last have run at Tuesday, July 21, 2020 at 9:29:20 PM Coordinated Universal Time; would next run at Wednesday, July 22, 2020 at 9:29:20 PM Coordinated Universal Time.

☐ Ignore post-commit hooks ?

## Build

Save

Apply



Not secure | 192.168.99.100:32756/job/job2-test\_code/configure

job2-test\_code

General Source Code Management Build Triggers **Build** Post-build Actions

☐ Ignore post-commit hooks

### Build

**Execute shell**

Command `sh /mycode/test_code.sh`

See [the list of available environment variables](#)

Advanced...

Add build step

### Post-build Actions

Add post-build action

Save Apply

Page generated:  
Jul 22, 2020, 6:52:20 AM

**For creating a pipeline view, we use the below code...**

```
buildPipelineView('Task 6') {  
    filterBuildQueue()  
    filterExecutors()  
    title('Project for deploy web-server')  
    displayedBuilds(5)  
    selectedJob('job1-deploy-web-server')  
    alwaysAllowManualTrigger()  
    showPipelineParameters()  
    refreshFrequency(60)  
}
```

Pull all the code in one file and your groovy script is ready 🍷

**GitHub Link...** [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git)

Thanks for reading... 🍷

Give a thumbsup 👍, if you like it...

If have any query, please DM me...

Let's connect with each other... 😊

This is the submission of the **#6th** task given by Mr. [Vimal Daga](#) Sir under the DevOps assembly line training.

Normally we create Job in Jenkins manually, but there is one more way to create job using code. I try to deploy web-server using groovy code on the top of Kubernetes.

For more detail, please check the article once..

. GitHub Link: [https://github.com/Anuddeeph/jenkins\\_dsl\\_code.git](https://github.com/Anuddeeph/jenkins_dsl_code.git)

**#vimaldaga #automation #jenkins #dsl #kubernetes #docker #github #righteducation #devops**