

DevOps Capstone Project 1:

You have been Hired Sr. DevOps Engineer in Abode Software. They want to implement DevOps Lifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company, their product is available on this GitHub link.

<https://github.com/hshar/website.git>

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration management tool.
2. Git Workflow has to be implemented
3. Code Build should automatically be triggered once commit is made to master branch or develop branch.

If commit is made to master branch, test and push to prod

If commit is made to develop branch, just test the product, do not push to prod

4. The Code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to Git-Hub. Use the following pre-built container for your application:

hshar/webapp

The code should reside in '/var/www/html'

5. The above tasks should be defined in a Jenkins Pipeline, with the following jobs:

Job1: build

Job2: test

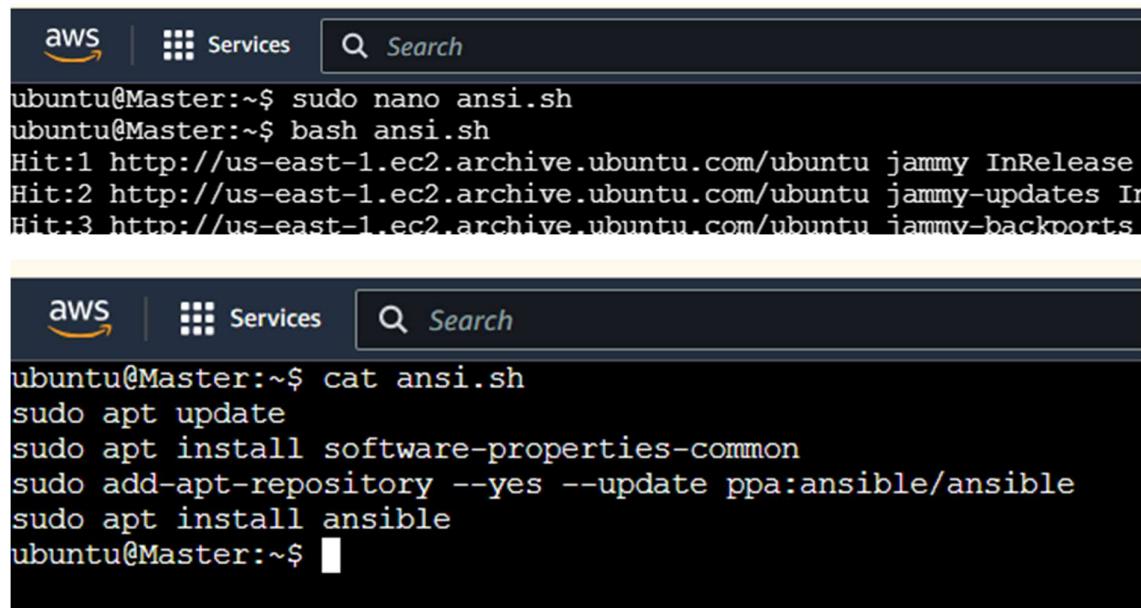
Job3: prod

Solution:

Created 3 instances and updated

Instances (3) Info					
<input type="text"/> Find Instance by attribute or tag (case-sensitive)					
<input type="checkbox"/>	Name ✎	Instance ID	Instance state	Instance type	Status ch
<input type="checkbox"/>	Capstone-Master	i-0ecc5cc2b80fa99a4	Running Q Q	t2.medium	2/2 ch
<input type="checkbox"/>	Prod-Server	i-0479fd8c99b80d35a	Running Q Q	t2.micro	2/2 ch
<input type="checkbox"/>	Test-Server	i-0c90cebe899d1c569	Running Q Q	t2.micro	2/2 ch

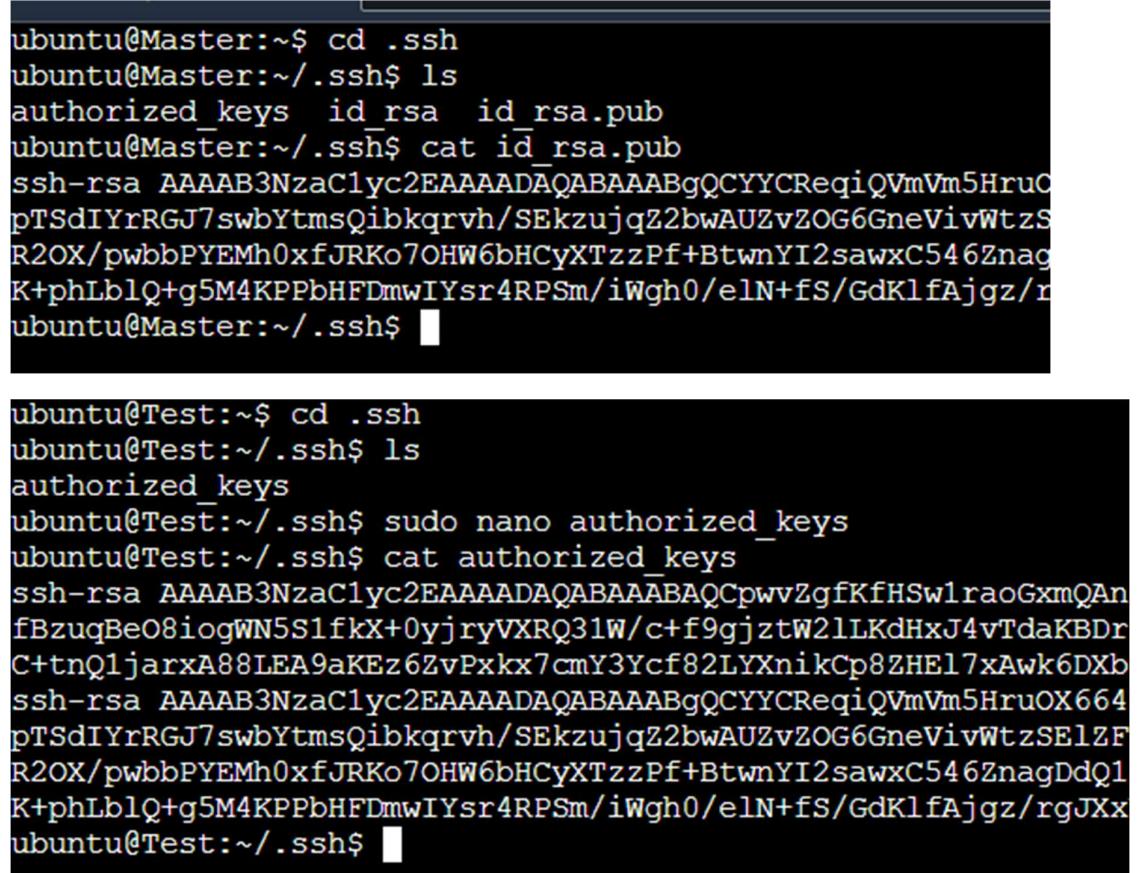
Installing ansible



```
aws | Services | Search

ubuntu@Master:~$ sudo nano ansi.sh
ubuntu@Master:~$ bash ansi.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
ubuntu@Master:~$ cat ansi.sh
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
ubuntu@Master:~$
```

Creating keys in Project1 and installing in Test and Prod server



```
ubuntu@Master:~$ cd .ssh
ubuntu@Master:~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Master:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCYYCReqiQVmVm5HruO
pTSdIYrRGJ7swbYtmsQibkqrvh/SEkzujqZ2bwAUZvZOG6GneVivWtzS
R2OX/pwbbPYEMh0xfJRKO7OHW6bHCyXTzzPf+BtwnYI2sawxC546znag
K+phLb1Q+g5M4KPPbHFdmwIYsr4RPSm/iWgh0/e1N+fS/GdK1fAjgz/r
ubuntu@Master:~/ssh$
```



```
ubuntu@Test:~$ cd .ssh
ubuntu@Test:~/ssh$ ls
authorized_keys
ubuntu@Test:~/ssh$ sudo nano authorized_keys
ubuntu@Test:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCyyCReqiQVmVm5HruO
fBzuqBeO8iogWN5S1fkX+0yjryVXRQ31W/c+f9gjztW21LKdHxJ4vTdaKBDr
C+tnQ1jarxA88LEA9aKEz6ZvPxkx7cmY3Ycf82LYXnikCp8ZHE17xAwk6DXb
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCYYCReqiQVmVm5HruO
pTSdIYrRGJ7swbYtmsQibkqrvh/SEkzujqZ2bwAUZvZOG6GneVivWtzSE1ZF
R2OX/pwbbPYEMh0xfJRKO7OHW6bHCyXTzzPf+BtwnYI2sawxC546znagDdQ1
K+phLb1Q+g5M4KPPbHFdmwIYsr4RPSm/iWgh0/e1N+fS/GdK1fAjgz/rgJxx
ubuntu@Test:~/ssh$
```

```

ubuntu@Prod:~$ cd .ssh
ubuntu@Prod:~/._ssh$ ls
authorized_keys
ubuntu@Prod:~/._ssh$ sudo nano authorized_keys
ubuntu@Prod:~/._ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCPwvZgfKfHSw1raoGxmQAn
zuqBe08iogWN5S1fkX+0yjryVXRQ31W/c+f9gjztW21LKdHxJ4vTdaKBDr1d
Q1jarxA88LEA9aKEz6ZvPxkx7cmY3Ycf82LYXnikCp8ZHE17xAwk6DXb/H8e
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCYYCReqiQVmVm5HruOX664
SdIYrRGJ7swbYtmsQibkqrvh/SEkzujqZ2bwAUZvZOG6GneVivWtzSELZFPm
/pwbbPYEMh0xfJRKo7OHW6bHCyXTzzPf+BtwnYI2sawxC546ZnagDdQ1Fci0
lQ+g5M4KPPbHFdmwIYsr4RPSm/iWgh0/e1N+fS/GdK1fAjgz/rgJxxYkEx2U
ubuntu@Prod:~/._ssh$ 

```

Adding hosts in Project1

```

aws | Services | Q Search [Alt+S]
GNU nano 6.2
[host]
Test ansible_host=172.31.86.228
Prod ansible_host=172.31.95.83

```

```

aws | Services | Q Search [Alt+S]
ubuntu@Master:/etc/ansible$ sudo nano hosts
ubuntu@Master:/etc/ansible$ ansible -m ping all
The authenticity of host '172.31.85.236 (172.31.85.236)' can't be established.
ED25519 key fingerprint is SHA256:101vJmxSj6teQpJtEDS7aT+nMmj2KGUJ1Fx28Clvrro.
This key is not known by any other names
The authenticity of host '172.31.87.194 (172.31.87.194)' can't be established.
ED25519 key fingerprint is SHA256:3GYN4EULYqRgU5f71UgqOtZSzcxVA4pjVMfBKEFVMz8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
172.31.85.236 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
172.31.87.194 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@Master:/etc/ansible$ 

```

Installing b.sh and c.sh to run in play.yaml

```
ubuntu@Master:/etc/ansible$ cat b.sh
sudo apt-get update
sudo apt install openjdk-17-jdk -y
sudo apt install docker.io -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
ubuntu@Master:/etc/ansible$ cat c.sh
sudo apt update
sudo apt install openjdk-17-jdk -y
sudo apt install docker.io -y
ubuntu@Master:/etc/ansible$
```

i-0ecc5cc2b80fa99a4 (Capstone-Master)

PublicIPs: 52.91.102.254 PrivateIPs: 172.31.30.73

Yaml file created

```
ubuntu@Master:/etc/ansible$ sudo nano play.yaml
ubuntu@Master:/etc/ansible$ sudo nano b.sh
ubuntu@Master:/etc/ansible$ sudo nano c.sh
ubuntu@Master:/etc/ansible$ sudo cat play.yaml
---
- name: task for master
  hosts: localhost
  become: true
  tasks:
    - name: executing b.sh script in master
      script: b.sh
- name: task for slaves
  hosts: all
  become: true
  tasks:
    - name: executing c.sh script in slaves
      script: c.sh
ubuntu@Master:/etc/ansible$
```

Checking syntax and dry run it

```
script: c.sh
ubuntu@Master:/etc/ansible$ ansible-playbook play.yaml --syntax-check

playbook: play.yaml
ubuntu@Master:/etc/ansible$ ansible-playbook play.yaml --check

PLAY [task for master] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [executing b.sh script in master] ****
skipping: [localhost]

PLAY [task for slaves] ****
TASK [Gathering Facts] ****
```

i-0ecc5cc2b80fa99a4 (Capstone-Master)

Running playbook

```
TASK [Gathering Facts] ****
ok: [172.31.85.236]
ok: [172.31.87.194]

TASK [executing c.sh script in slaves] ****
changed: [172.31.87.194]
changed: [172.31.85.236]

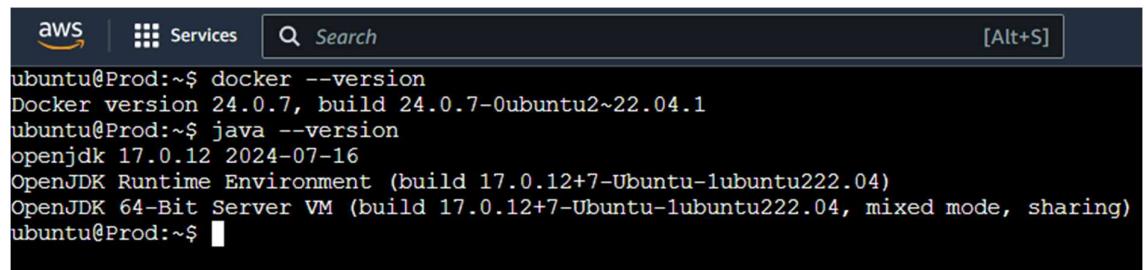
PLAY RECAP ****
172.31.85.236 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.87.194 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@Master:/etc/ansible$ 
```

i-0ecc5cc2b80fa99a4 (Capstone-Master)

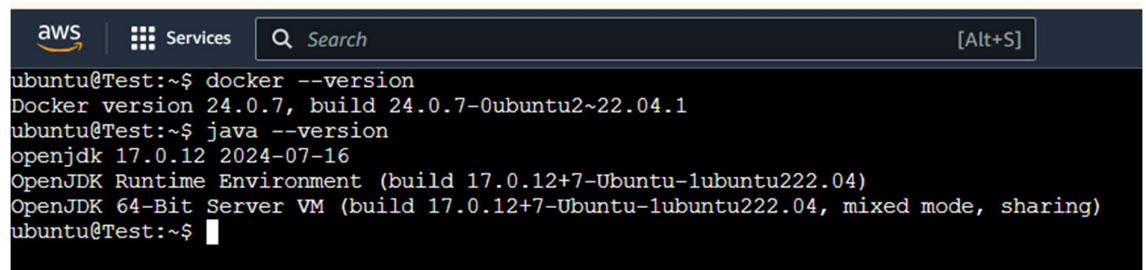
PublicIPs: 52.91.102.254 PrivateIPs: 172.31.30.73

Docker and java got installed in prod and test



The screenshot shows a terminal window with the AWS logo and 'Services' button in the top bar. The search bar contains 'Search' and the keyboard shortcut '[Alt+S]' is shown on the right. The terminal output shows the following commands and their results:

```
ubuntu@Prod:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
ubuntu@Prod:~$ java --version
openjdk 17.0.12 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu222.04, mixed mode, sharing)
ubuntu@Prod:~$ 
```



The screenshot shows a terminal window with the AWS logo and 'Services' button in the top bar. The search bar contains 'Search' and the keyboard shortcut '[Alt+S]' is shown on the right. The terminal output shows the following commands and their results:

```
ubuntu@Test:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
ubuntu@Test:~$ java --version
openjdk 17.0.12 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu222.04, mixed mode, sharing)
ubuntu@Test:~$ 
```

```

aws | Services | Search [Alt+S]
ubuntu@Master:/etc/ansible$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
ubuntu@Master:/etc/ansible$ jenkins --version
2.462.1
ubuntu@Master:/etc/ansible$ java --version
openjdk 17.0.12 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu22.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu22.04, mixed mode, sharing)
ubuntu@Master:/etc/ansible$ 

```

Forked the repository mentioned in the project

The screenshot shows a GitHub repository named 'Project1'. It is a public fork from 'hshar/website'. The repository has one branch ('master') and no tags. The README file is present. The commit history shows two commits by 'Ubuntu' from 5 years ago.

File	Commit Message	Time
modified	Ubuntu modified	883b439 · 5 years ago
final	Ubuntu images	5 years ago
modified	Ubuntu index.html	5 years ago

Connecting Master to Jenkins

```

ubuntu@Master:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
a287f6386364433b855585095fbafeb8
ubuntu@Master:~$ 

```

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	folders OWASP Markup Formatter ** ASM API ** JSON Path API ** Structs ** Pipeline: Step API ** Token Macro Build Timeout ** bouncycastle API ** Credentials ** Plain Credentials ** Variant ** SSH Credentials Credentials Binding ** SCM API ** Pipeline: API ** commons-lang3 v3.x Jenkins API Timestamper ** Caffeine API ** Script Security
✓ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	
⌚ LDAP	⌚ Email Extension	⌚ Mailer	⌚ Dark Theme	

Installing plug-ins

ssh agent

Install Name Released

SSH Agent 376.v8933585c69d3 13 days ago

This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.

Plugins

-  Updates
-  Available plugins
-  Installed plugins
-  Advanced settings
-  Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Ionicons API

 Success

Folders

 Success

OWASP Markup Formatter

 Success

ASM API

 Success

JSON Path API

 Success

Setup an agent

New node

Node name

Test

Type

 Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide high integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

Nodes

+ New Node

Configure Monitors

36 min

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.33 GiB	 0 B	4.33 GiB	0ms 
	Test		N/A	N/A	N/A	N/A	N/A 

Icon: S M L

Leg

New node

Node name

Type

Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Copy Existing Node

Create

Nodes							
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.06 GiB	! 0 B	4.06 GiB	0ms ⚙️
2	Prod	Linux (amd64)	In sync	4.98 GiB	! 0 B	4.98 GiB	26ms ⚙️
3	Test	Linux (amd64)	In sync	4.98 GiB	! 0 B	4.98 GiB	20ms ⚙️
Data obtained		0.33 sec	0.33 sec	0.32 sec	0.33 sec	0.33 sec	0.32 sec

Creating job

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

+

New Item

Enter an item name

sample-job

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

<https://github.com/Miinal10/Project1.git>



Creating webhook URL

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages

- Security
- Code security and analysis
- Deploy keys
- Secrets and variables

- Integrations
- GitHub Apps
- Email notifications

payload URL

Content type *

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.

Enable SSL verification Disable (not recommended)

Which events would you like to trigger this webhook?

Just the push event.
 Send me everything.
 Let me select individual events.

Active
We will deliver event details when this hook is triggered.

Add webhook

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(Auto)

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

```
ubuntu@Master:~$ cd /var
ubuntu@Master:/var$ ls
backups cache crash lib local lock log mail opt run snap spool tmp
ubuntu@Master:/var$ cd lib/
ubuntu@Master:/var/lib$ ls
PackageKit chrony           dhcp   hibinit-agent misc   private   sudo
amazon   cloud              docker jenkins os-prober python  systemd  u
apport   command-not-found dpkg   landscape pam    shells.state tpm    u
apt     containerd          git    logrotate plymouth shim-signed ubuntu-advantage u
boltd   dbus               grub   man-db   polkit-1 snapd   ubuntu-fan   u
ubuntu@Master:/var/lib$ cd jenkins/
ubuntu@Master:/var/lib/jenkins$ cd workspace/
ubuntu@Master:/var/lib/jenkins/workspace$ ls
sample-job
ubuntu@Master:/var/lib/jenkins/workspace$ cd sample-job/
ubuntu@Master:/var/lib/jenkins/workspace/sample-job$ ls
images index.html
ubuntu@Master:/var/lib/jenkins/workspace/sample-job$
```

Files are available in given path below

✓ Console Output

```
Started by user Minal
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/sample-job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Miinal10/Project1.git
> git init /var/lib/jenkins/workspace/sample-job # timeout=10
Fetching upstream changes from https://github.com/Miinal10/Project1.git
```

✓ Console Output

```
Started by user Minal
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Test
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Miinal10/Project1.git
```

All +

S	W	Name	Last Success	Last Failure	Last Duration	
✓	☀️	sample-job	7 min 54 sec #2	N/A	3.3 sec	▶
✓	☀️	Test	42 sec #1	N/A	6.7 sec	▶

```

ubuntu@Prod:~$ ls
jenkins
ubuntu@Prod:~$ cd jenkins/
ubuntu@Prod:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@Prod:~/jenkins$ cd workspace/
ubuntu@Prod:~/jenkins/workspace$ ls
Test
ubuntu@Prod:~/jenkins/workspace$ cd t
-bash: cd: t: No such file or directory
ubuntu@Prod:~/jenkins/workspace$ cd Test
ubuntu@Prod:~/jenkins/workspace/Test$ ls
images  index.html
ubuntu@Prod:~/jenkins/workspace/Test$ 

```

New branch is created

Default				
Branch	Updated	Check status	Behind Ahead	Pull request
master	now	Default		⋮

Your branches				
Branch	Updated	Check status	Behind Ahead	Pull request
develop	now	0 0	0 0	⋮

Active branches				
Branch	Updated	Check status	Behind Ahead	Pull request
develop	now	0 0	0 0	⋮

Copy Test job to create Prod job and change branch

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as test platform-specific builds, etc.

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. folder creates a separate namespace, so you can have multiple things of the same name in different folders.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Copy from

Test

OK

Throttle builds ?

Execute concurrent builds if necessary ?

Restrict where this project can be run ?

Label Expression ?

Test

Label Test matches 1 node. Permissions or other restrictions provided by plugins may further limit what nodes are available.

Git ?

Repositories ?

Repository URL ?

https://github.com/Miinal10/Project1.git

Credentials ?

- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/develop

Add Branch

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Started by user [Minal](#)

Running as SYSTEM

Building remotely on [Test](#) in workspace /home/ubuntu/jenkins/workspace/Prod

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository <https://github.com/Minal10/Project1.git>

> git init /home/ubuntu/jenkins/workspace/Prod # timeout=10

Fetching upstream changes from <https://github.com/Minal10/Project1.git>

All +

S	W	Name	Last Success	Last Failure	Last Duration	
✓	☀	Prod	1 min 21 sec #1	N/A	0.35 sec	▷
✓	☀	sample-job	15 min #2	N/A	3.3 sec	▷
✓	☀	Test	7 min 55 sec #1	N/A	6.7 sec	▷

Creating Dockerfile

A screenshot of a GitHub code editor interface. At the top, it shows 'Project1 / Dockerfile' and 'in master'. There are 'Cancel changes' and 'Commit changes...' buttons. Below the header, there are 'Edit' and 'Preview' buttons, and settings for 'Spaces', '2', and 'No wrap'. The code editor contains the following Dockerfile:

```
1 FROM ubuntu
2 RUN apt-get update
3 RUN apt-get install apache2 -y
4 COPY ./var/www/html/
5 ENTRYPOINT apachectl -D FOREGROUND
```

Dockerfile is created in develop branch

A screenshot of a GitHub repository page for 'Project1'. The top navigation bar shows 'develop' selected. A commit by 'Miinal10' titled 'Create Dockerfile' is shown, with the commit hash '6690e47 · now'. Below the commit, a message says 'This branch is 1 commit ahead of hshar/website:master.' There are 'Contribute' and 'Sync' buttons. A table below lists files with their last commit details:

Name	Last commit message	Last comm
images	final	5 years ago
Dockerfile	Create Dockerfile	
index.html	modified	5 years ago

```
images index.html
ubuntu@Prod:~/jenkins/workspace/Test$ ls
Dockerfile images index.html
ubuntu@Prod:~/jenkins/workspace/Test$
```

```
ubuntu@Test:~/jenkins/workspace/Prod$ ls
Dockerfile images index.html
ubuntu@Test:~/jenkins/workspace/Prod$
```

Adding built-in steps in Prod

Build Steps

≡ Execute shell ?

Command

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

```
sudo docker build . -t image1
sudo docker run -itd -p 80:80 image1
```

```
done.
Removing intermediate container 2fdaa5413cf1
--> 71a852d5aa16
Step 4/5 : COPY . /var/www/html/
--> a81a209a79f6
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in a3180d5e62cd
Removing intermediate container a3180d5e62cd
--> 1f9fb53aeb53
Successfully built 1f9fb53aeb53
Successfully tagged testimage:latest
+ sudo docker run -it -p 80:80 -d testimage
620e952178b63952a0069409cd04938edc77c0d17af63561ae55dc255f5f8689
Finished: SUCCESS
```

```
--> 3cfb2fcadd44
Step 4/5 : COPY . /var/www/html/
--> 7bd564056c8c
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in cff667c41d32
Removing intermediate container cff667c41d32
--> 7e39852d875e
Successfully built 7e39852d875e
Successfully tagged image1:latest
+ sudo docker run -itd -p 80:80 image1
dcff293a1a59259f839be84e281350cb933fd1dbfe7a68142e92dd00078e2b30
Finished: SUCCESS
```

Prod server



The screenshot shows a web browser window with the URL `54.145.163.172`. The page displays the text "Hello world!" above the GitHub logo. The GitHub logo features a black cat head with large white eyes and a small smile, standing on a blue circular base.

GitHub Project Overview:

- Files:** develop (selected), images, Dockerfile, index.html
- Project1 / index.html:** Miinal10 Update index.html
- Code:** Blame, 9 lines (9 loc) · 250 Bytes, Code 55% faster with GitHub Copilot

```
1 <html>
2 <head>
3 <title> Intellipaat </title>
4 </head>
5 <body style = "background-image:url('images/github3.jpg'); background-size: 100%">
6 <h2 ALIGN=CENTER>Hello world!</h2>
7 <h3>This is edited in develop branch for testing</h3>
8 </body>
9 </html>
```

Docker Build Log:

```
--> 0099a5f065a0
Step 3/5 : RUN apt-get install apache2 -y
--> Using cache
--> 71a852d5aa16
Step 4/5 : COPY . /var/www/html/
--> Using cache
--> bece0734a30d
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Using cache
--> 48a9c8bc407f
Successfully built 48a9c8bc407f
Successfully tagged testimage:latest
+ sudo docker run -it -p 81:80 -d testimage
ebcebcb5c1cd8c6ae100492ae8ac6c1ee590bb48561cd9c450e58a7d9452f1df
Finished: SUCCESS
```

Hello world!

This is edited in develop branch for testing



Github