

A PROJECT REPORT ON
DEPLOYING OF DOCKER IMAGE ON EC2 INSTANCE
COURSE
DEVOPS WITH AWS

Submitted By
ASHOK REKHA

Under The Guidance Of
VAMSI BYRAMALA (TRAINER),
A. VENKATESH (MENTOR)



Greatcoder Training institute in Madhapur Hyderabad

#201,2nd floor,Above Ageless Building,Beside Indian Bank Madhapur,Hyderabad-500081
Land Mark:Beside Karachi Bakery, lane, Hyderabad, Telangana 500081.

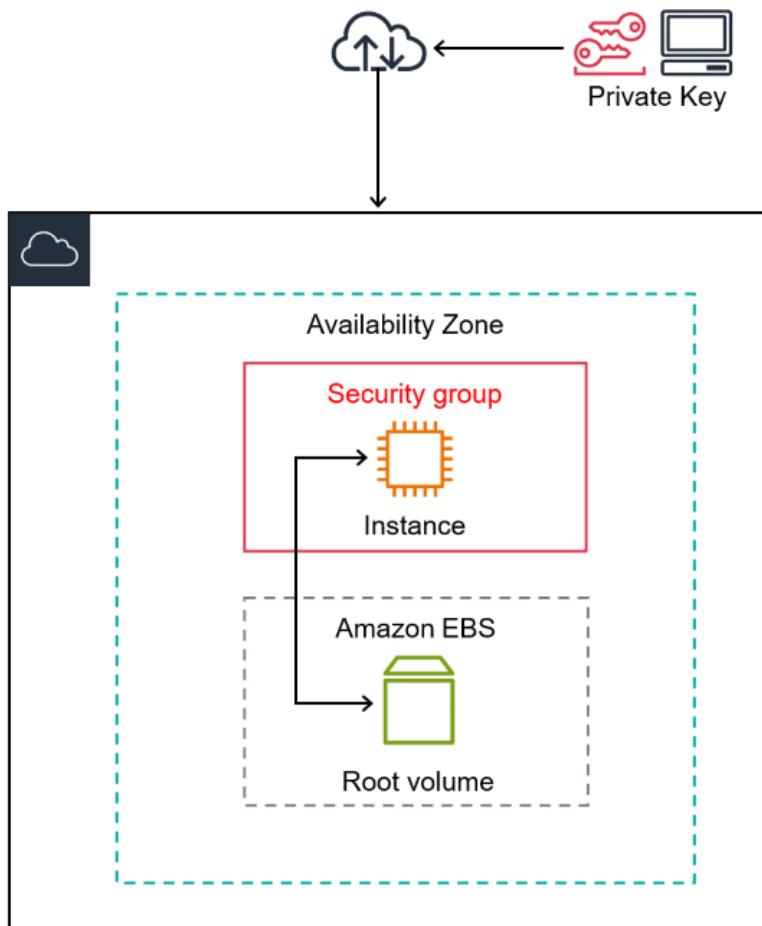
1. Install and configure Jenkins software in your EC2.

Overview:

The instance launched in this tutorial is an Amazon EC2 select an Availability Zone for you.

Availability Zones are multiple, isolated locations within each Region. You can think of an Availability Zone as an isolated data center.

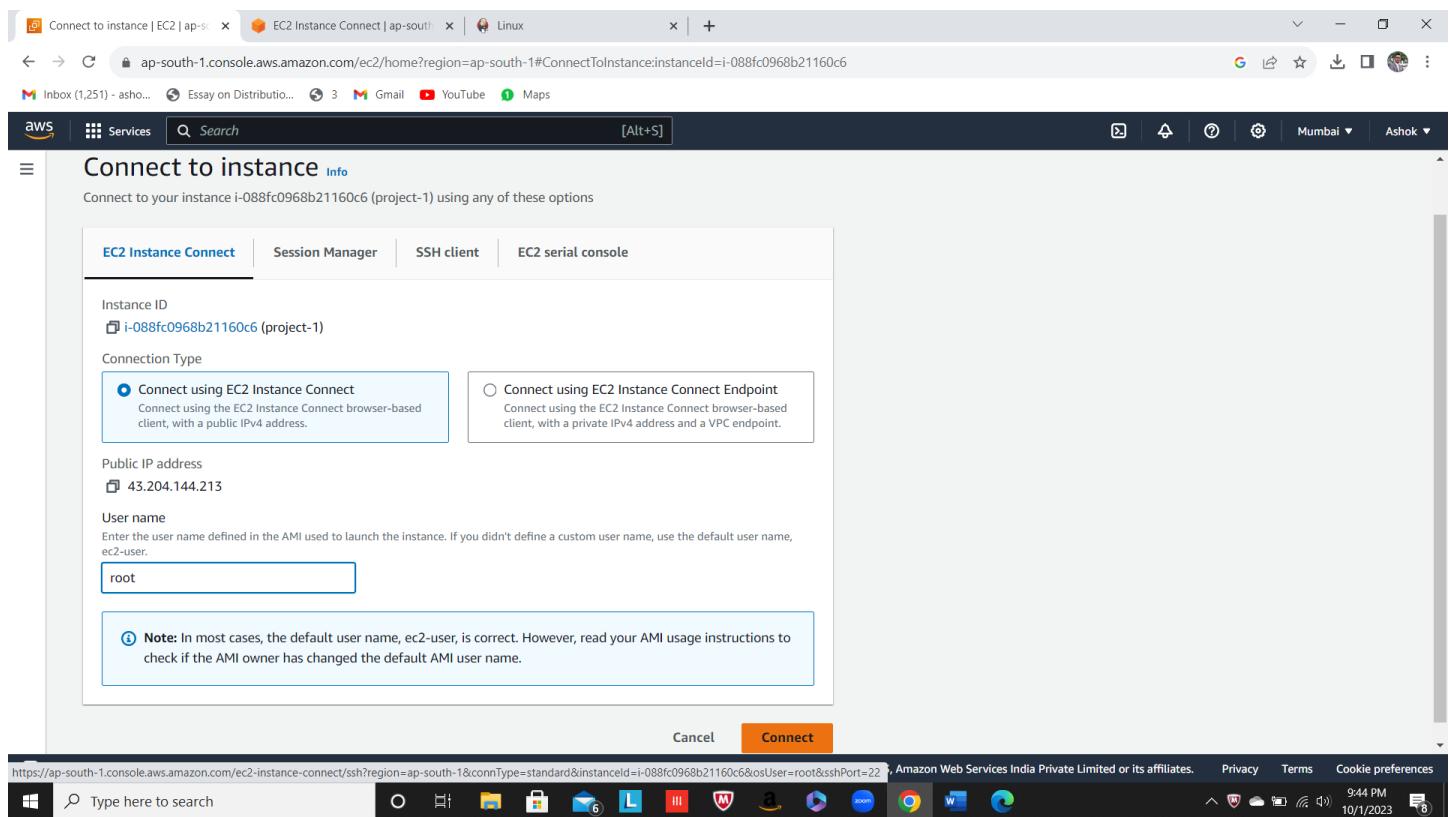
When you launch your instance, you secure it by specifying a key pair (to prove your identity) and a security group (which acts as a virtual firewall to control ingoing and outgoing traffic). When you connect to your instance, you must provide the private key of the key pair that you specified when you launched your instance.



To launch an instance:

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**, and then choose **Launch instance** from the options that appear.
- Under **Name and tags**, for **Name**:Project-1.

- Under **Application and OS Images (Amazon Machine Image)**, do the following:
 - Choose Amazon Linux. This is the operating system (OS) for your instance.
 - From **Amazon Machine Image (AMI)**, select an Amazon Linux 2. Notice that these AMIs are marked **Free tier eligible**.
- Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier.
- Under **Key pair (login)**, for **Project-1**, choose the key pair that you created when getting set up.
 - Warning.
- Keep the default selections for the other configuration settings for your instance.
- A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
- On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running.
- It can take a few minutes for the instance to be ready for you to connect.



Jenkins installation and configure:

- I have installed Jenkins through yum on Red Hat Enterprise Linux, Alma Linux, Rocky Linux, Oracle Linux, and other Red Hat based distributions.
 - Long Term Support release:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```
 - You can enable the Jenkins service to start at boot with the command: `sudo systemctl enable jenkins`
 - You can start the Jenkins service with the command: `sudo systemctl start jenkins`
 - You can check the status of the Jenkins service using the command: `sudo systemctl status jenkins`
- Browse to <http://localhost:8080> and wait until the **Unlock Jenkins** page appears.
- From the Jenkins console log output, copy the automatically-generated alphanumeric password.
- On the **Unlock Jenkins** page, paste this password into the **Administrator password** field and click **Continue**.
- After [unlocking Jenkins](#), the **Customize Jenkins** page appears. Here you can install any number of useful plugins as part of your initial setup.
 - **Install suggested plugins** - to install the recommended set of plugins, which are based on most common use cases.
- When the **Create First Admin User** page appears, specify the details for your administrator user in the respective fields and click **Save and Finish**.
- When the **Jenkins is ready** page appears, click **Start using Jenkins**.

The screenshot shows a web browser window with the Jenkins User Handbook page open. The URL is jenkins.io/doc/book/installing/linux/#red-hat-centos. The page content includes sections on Long Term Support release and Weekly release, along with terminal commands for installation.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
  https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

The screenshot shows a Windows taskbar with a terminal window open on an AWS instance. The terminal window displays the execution of Jenkins repository setup commands. The terminal output is as follows:

```
[root@ip-172-31-10-235 ~]# sudo wget -O /etc/yum.repos.d/jenkins.repo \
> https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2023-10-01 16:16:21-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.154.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=====] 85 --.-K/s    in 0s

2023-10-01 16:16:21 (4.28 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@ip-172-31-10-235 ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-10-235 ~]# sudo yum upgrade
Loaded plugins: extras suggestions, langpacks, priorities, update-motd
jenkins
jenkins/primary_db
No packages marked for update
[root@ip-172-31-10-235 ~]# # Add required dependencies for the jenkins package
[root@ip-172-31-10-235 ~]# sudo yum install fontconfig java-17-openjdk
Loaded plugins: extras suggestions, langpacks, priorities, update-motd
No package java-17-openjdk available.
Resolving Dependencies
--> Running transaction check
--> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
```

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

Inbox (1,251) - ashok... Essay on Distribution... 3 Gmail YouTube Maps

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
fontconfig	x86_64	2.13.0-4.3.amzn2	amzn2-core	253 k
Installing for dependencies:				
dejavu-fОНts-common	noarch	2.33-6.amzn2	amzn2-core	64 k
dejavu-sанс-fОНts	noarch	2.33-6.amzn2	amzn2-core	1.4 M
fontpackages-fОНesystem	noarch	1.44-8.amzn2	amzn2-core	10 k

Transaction Summary

Install 1 Package (+3 Dependent packages)
Total download size: 1.7 M
Installed size: 5.9 M
Is this ok [y/d/N]: y
Is this ok [y/d/N]: y
Downloading packages:
(1/4): dejavu-fОНts-common-2.33-6.amzn2.noarch.rpm 64 kB 00:00:00
(2/4): fontconfig-2.13.0-4.3.amzn2.x86_64.rpm 253 kB 00:00:00
(3/4): fontpackages-fОНesystem-1.44-8.amzn2.noarch.rpm 10 kB 00:00:00
(4/4): dejavu-sанс-fОНts-2.33-6.amzn2.noarch.rpm 1.4 MB 00:00:00

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.144.213 PrivateIPs: 172.31.10.235

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 9:47 PM 10/1/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

Inbox (1,251) - ashok... Essay on Distribution... 3 Gmail YouTube Maps

aws Services Search [Alt+S]

Dependencies Resolved

Package	Arch	Version	Repository	Size
(2/4): fontconfig-2.13.0-4.3.amzn2.x86_64.rpm				253 kB 00:00:00
(3/4): fontpackages-fОНesystem-1.44-8.amzn2.noarch.rpm				10 kB 00:00:00
(4/4): dejavu-sанс-fОНts-2.33-6.amzn2.noarch.rpm				1.4 MB 00:00:00

Total

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing :	fontpackages-fОНesystem-1.44-8.amzn2.noarch	8.8 MB/s 1.7 MB 00:00:00
Installing :	dejavu-fОНts-common-2.33-6.amzn2.noarch	1/4
Installing :	dejavu-sанс-fОНts-2.33-6.amzn2.noarch	2/4
Installing :	fontconfig-2.13.0-4.3.amzn2.x86_64	3/4
Verifying :	fontconfig-2.13.0-4.3.amzn2.x86_64	4/4
Verifying :	dejavu-fОНts-common-2.33-6.amzn2.noarch	1/4
Verifying :	fontpackages-fОНesystem-1.44-8.amzn2.noarch	2/4
Verifying :	dejavu-sанс-fОНts-2.33-6.amzn2.noarch	3/4

Installed:

fontconfig.x86_64 0:2.13.0-4.3.amzn2

Dependency Installed:

dejavu-fОНts-common.noarch 0:2.33-6.amzn2 dejavu-sанс-fОНts.noarch 0:2.33-6.amzn2 fontpackages-fОНesystem.noarch 0:1.44-8.amzn2

Complete!

[root@ip-172-31-10-235 ~]#

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.144.213 PrivateIPs: 172.31.10.235

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 9:48 PM 10/1/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

aws Services Search [Alt+S]

```
[root@ip-172-31-10-235 ~]# yum install java -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto.x86_64 1:17.0.8+8-1.amzn2.1 will be installed
--> Processing Dependency: java-17-amazon-corretto-headless(x86-64) = 1:17.0.8+8-1.amzn2.1 for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libxtst for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libXrender for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
--> Running transaction check
--> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libSM.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libICE.so.6() (64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Package java-17-amazon-corretto-headless.x86_64 1:17.0.8+8-1.amzn2.1 will be installed
--> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: alsalib for package: 1:java-17-amazon-corretto-headless-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.8+8-1.amzn2.1.x86_64
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.8+8-1.amzn2.1.x86_64
--> Package libX11.x86_64 0:1.6.7-3.amzn2.0.3 will be installed
```

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.144.213 PrivateIPs: 172.31.10.235

CloudShell Feedback

Type here to search

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

9:49 PM 10/1/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

aws Services Search [Alt+S]

```
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.3 for package: libX11-1.6.7-3.amzn2.0.3.x86_64
--> Processing Dependency: libxcb.so.1() (64bit) for package: libX11-1.6.7-3.amzn2.0.3.x86_64
--> Package libxi.x86_64 0:1.7.9-1.amzn2.0.2 will be installed
--> Processing Dependency: libXext.so.6() (64bit) for package: libXi-1.7.9-1.amzn2.0.2.x86_64
--> Package libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2 will be installed
--> Package libXrandr.x86_64 0:1.5.1-2.amzn2.0.3 will be installed
--> Package libXrender.x86_64 0:0.9.10-1.amzn2.0.2 will be installed
--> Package libxt.x86_64 0:1.1.5-3.amzn2.0.2 will be installed
--> Package libxtst.x86_64 0:1.2.3-1.amzn2.0.2 will be installed
--> Running transaction check
--> Package alsalib.x86_64 0:1.1.4.1-2.amzn2 will be installed
--> Package dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package dejavu-serif-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package javapackages-tools.noarch 0:3.4.1-11.amzn2 will be installed
--> Processing Dependency: python-javapackages = 3.4.1-11.amzn2 for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Processing Dependency: libxslt for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Package libICE.x86_64 0:1.0.9-9.amzn2.0.2 will be installed
--> Package libSM.x86_64 0:1.2.2-2.amzn2.0.2 will be installed
--> Package libX11-common.noarch 0:1.6.7-3.amzn2.0.3 will be installed
--> Package libXext.x86_64 0:1.3.3-3.amzn2.0.2 will be installed
--> Package libxcb.x86_64 0:1.12-1.amzn2.0.2 will be installed
--> Processing Dependency: libXau.so.6() (64bit) for package: libxcb-1.12-1.amzn2.0.2.x86_64
--> Package log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2 will be installed
--> Running transaction check
--> Package libXau.x86_64 0:1.0.8-2.1.amzn2.0.2 will be installed
--> Package libxsit.x86_64 0:1.1.28-6.amzn2 will be installed
```

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.144.213 PrivateIPs: 172.31.10.235

CloudShell Feedback

Type here to search

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

9:50 PM 10/1/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-088fc0968b21160c6&osUser=root&sshPort=22#/

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

aws Services Search [Alt+S]

```

Verifying : libXinerama-1.1.3-2.1.amzn2.0.2.x86_64
Verifying : dejavu-sans-mono-fonts-2.33-6.amzn2.noarch
Verifying : python-lxml-3.2.1-4.amzn2.0.4.x86_64
Verifying : 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64
Verifying : libxsigt-1.1.28-6.amzn2.x86_64
Verifying : libXtst-1.2.3-1.amzn2.0.2.x86_64
Verifying : python-javapackages-3.4.1-11.amzn2.noarch
Verifying : libICE-1.0.9-9.amzn2.0.2.x86_64
Verifying : libX11-common-1.6.7-3.amzn2.0.3.noarch
Verifying : javapackages-tools-3.4.1-11.amzn2.noarch

Installed:
java-17-amazon-corretto.x86_64 1:17.0.8+8-1.amzn2.1

Dependency Installed:
alsa-lib.x86_64 0:1.1.4.1-2.amzn2
glib2.x86_64 0:4.1.6-9.amzn2.0.2
libICE.x86_64 0:1.0.9-9.amzn2.0.2
libX11-common.noarch 0:1.6.7-3.amzn2.0.3
libXi.x86_64 0:1.7.9-1.amzn2.0.2
libXrender.x86_64 0:0.9.10-1.amzn2.0.2
libxcb.x86_64 0:1.12-1.amzn2.0.2
python-javapackages.noarch 0:3.4.1-11.amzn2

dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2
java-17-amazon-corretto-headless.x86_64 1:17.0.8+8-1.amzn2.1
libsM.x86_64 0:1.2.2-2.amzn2.0.2
libXau.x86_64 0:1.0.8-2.1.amzn2.0.2
libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2
libXt.x86_64 0:1.1.5-3.amzn2.0.2
libXslt.x86_64 0:1.1.28-6.amzn2
python-lxml.x86_64 0:3.2.1-4.amzn2.0.4

dejavu-serif-fonts.noarch 0:2.33-6.amzn2
javapackages-tools.noarch 0:3.4.1-11.amzn2
libX11.x86_64 0:1.6.7-3.amzn2.0.3
libXext.x86_64 0:1.3.3-3.amzn2.0.2
libXrandr.x86_64 0:1.5.1-2.amzn2.0.3
libXtst.x86_64 0:1.2.3-1.amzn2.0.2
log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2

Complete!
[root@ip-172-31-10-235 ~]#

```

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.144.213 PrivateIPs: 172.31.10.235



CloudShell Feedback

Type here to search

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

9:51 PM 10/1/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Linux

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-088fc0968b21160c6&osUser=root&sshPort=22#/

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

aws Services Search [Alt+S]

```

[root@ip-172-31-10-235 ~]# cd /var/lib/
[root@ip-172-31-10-235 lib]# ls
alternatives authconfig cloud dnclient gssproxy initramfs logrotate misc nfs plymouth rpcbind rpm-state stateless update-motd yum
amazon chrony dbus games hibinit-agent jenkins machines mlocate os-prober postfix rpm rsyslog systemd xfsdump
[root@ip-172-31-10-235 lib]# systemctl start jenkins
[root@ip-172-31-10-235 lib]# systemctl start jenkins
[root@ip-172-31-10-235 lib]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
     Active: active (running) since Sun 2023-10-01 16:49:44 UTC; 40s ago
       Main PID: 5959 (Java)
      CGroup: /system.slice/jenkins.service
             └─5959 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Oct 01 16:49:12 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: 15a7df02d8ff40ac08656db2c39f45f49
Oct 01 16:49:12 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 01 16:49:12 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: ****
Oct 01 16:49:12 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: ****
Oct 01 16:49:12 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: ****
Oct 01 16:49:44 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: 2023-10-01 16:49:44.448+0000 [id=32]           INFO    jenkins.InitReactorRunner$1@on...ization
Oct 01 16:49:44 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: 2023-10-01 16:49:44.484+0000 [id=25]           INFO    hudson.lifecycle.Lifecycle#onR...running
Oct 01 16:49:44 ip-172-31-10-235.ap-south-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Oct 01 16:49:45 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: 2023-10-01 16:49:45.292+0000 [id=47]           INFO    h.m.DownloadService$Downloadab...staller
Oct 01 16:49:45 ip-172-31-10-235.ap-south-1.compute.internal jenkins[5959]: 2023-10-01 16:49:45.293+0000 [id=47]           INFO    hudson.util.Retrier#start: Per...empt #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-10-235 lib]#

```

i-088fc0968b21160c6 (project-1)

PublicIPs: 43.204.148.52 PrivateIPs: 172.31.10.235



The screenshot shows the AWS CloudShell interface with the following details:

- EC2 | ap-south-1**: The active tab.
- EC2 Instance Connect | ap-south-1**: A tab in the background.
- Linux**: A tab in the background.
- Inbound rules**: The current page title.
- Description**: A tooltip for the optional description field.
- Security group rule ID**: `sgr-0a4d2bc74fbeb57f`.
- Type**: `SSH`.
- Protocol**: `TCP`.
- Port range**: `22`.
- Source**: `Custom`.
- Description**: `0.0.0.0/0`.
- Add rule**: A button to add new rules.
- Warning message**: "⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." with a close button.
- Buttons**: `Cancel`, `Preview changes`, and `Save rules`.

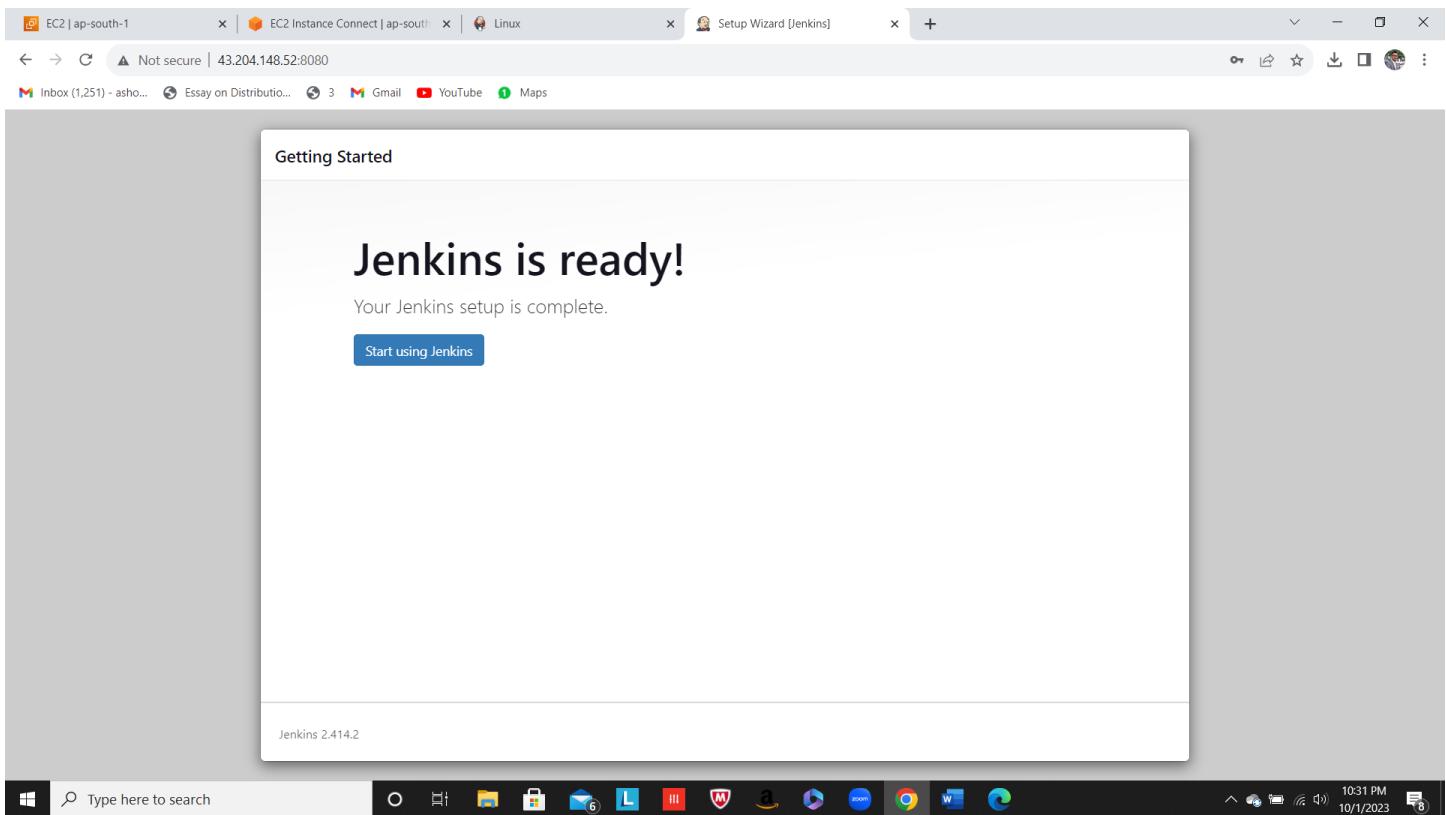
The screenshot shows the Jenkins 'Getting Started' page with the following details:

- Sign in [Jenkins]**: The active tab.
- Not secure | 43.204.148.52:8080/login?from=%2F**: The URL in the address bar.
- Inbox (1,251) - ashok...**: A tab in the background.
- Getting Started**: The main heading.
- Unlock Jenkins**: The main title of the guide.
- 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:** A note about the initial password location.
- /var/lib/jenkins/secrets/initialAdminPassword**: The path to the initial password file.
- Please copy the password from either location and paste it below.** Instructions for copying the password.
- Administrator password**: A placeholder for the copied password.
- Continue**: A blue button at the bottom right.

The screenshot shows two windows from the AWS CloudShell interface.

Top Window: A terminal session titled "Setup Wizard [Jenkins]" showing the command `cat /var/lib/jenkins/secrets/initialAdminPassword` and its output, a long hex string: `15a7df02d8ff40ac8656db2c39f45f9`.

Bottom Window: A "Getting Started" dialog titled "Instance Configuration". It displays the Jenkins URL input field containing `http://43.204.148.52:8080/`. Below the input field is a descriptive text about the Jenkins URL's purpose. At the bottom, it shows Jenkins version 2.414.2 and provides "Not now" and "Save and Finish" buttons.



Dashboard >

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

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

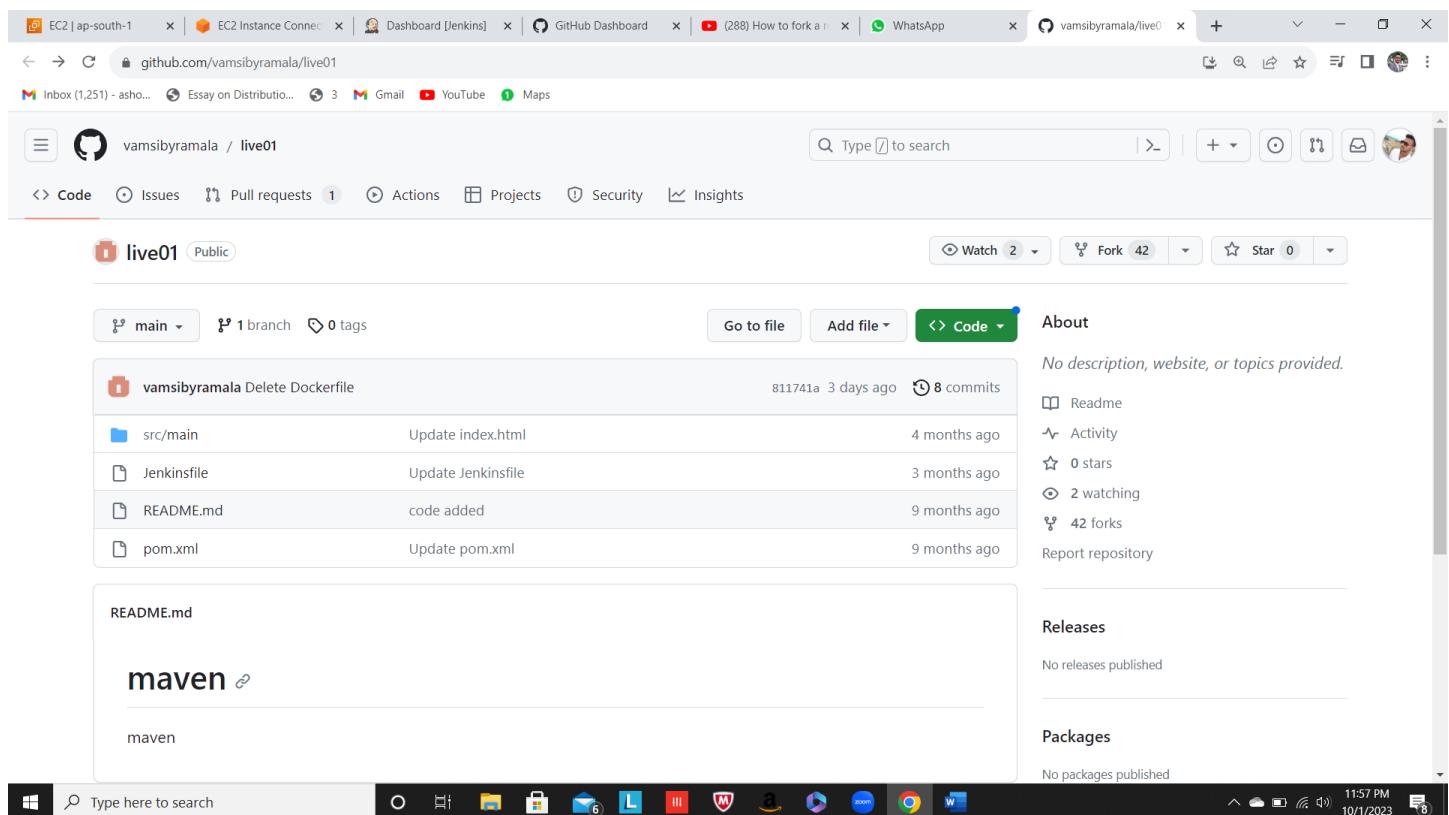
2. Fork the sample repository to your GitHub (<https://github.com/byramala/gctech.git>).

Overview:

A fork is a new repository that shares code and visibility settings with the original “upstream” repository. Forks are often used to iterate on ideas or changes before they are proposed back to the upstream repository, such as in open source projects or when a user does not have write access to the upstream repository.

Fork Procedure:

- On GitHub.com, navigate to the **vamsibyramala/live01** repository.
- In the top-right corner of the page, click **Fork**.
- Under "Owner," select the dropdown menu and click an owner for the forked repository.
- By default, forks are named the same as their upstream repositories. Optionally, to further distinguish your fork, in the "**Ashokrekha/live01**" field, type a name.
- Optionally, in the "Description" field, type a description of your fork.
- Optionally, select **Copy the DEFAULT branch only**.
- Click **Create fork**.



EC2 | ap-south-1 | EC2 Instance Connect | Dashboard [Jenkins] | GitHub Dashboard | (288) How to fork a | WhatsApp | Fork vamsibyramala/live01/fork

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks](#).

Required fields are marked with an asterisk (*).

Owner * Repository name *

Ashokrekha / live01
live01 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Copy the main branch only
Contribute back to vamsibyramala/live01 by adding your own branch. [Learn more](#).

ⓘ You are creating a fork in your personal account.

Create fork

Type here to search

11:59 PM 10/1/2023

EC2 | ap-south-1 | EC2 Instance Connect | Dashboard [Jenkins] | Ashokrekha/live01 | (288) How to fork a | WhatsApp | Ashokrekha/live01

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

Ashokrekha / live01

Type to search

Code Pull requests Actions Projects Wiki Security Insights Settings

live01 Public
forked from vamsibyramala/live01

main 1 branch 0 tags

Go to file Add file Code

This branch is up to date with vamsibyramala/live01:main.

Contribute Sync fork

vamsibyramala Delete Dockerfile 8 commits 3 days ago

src/main Update index.html 4 months ago

Jenkinsfile Update Jenkinsfile 3 months ago

README.md code added 9 months ago

pom.xml Update pom.xml 9 months ago

README.md

maven

About

No description, website, or topics provided.

Readme Activity 0 stars 0 watching 43 forks

Releases

No releases published Create a new release

Packages

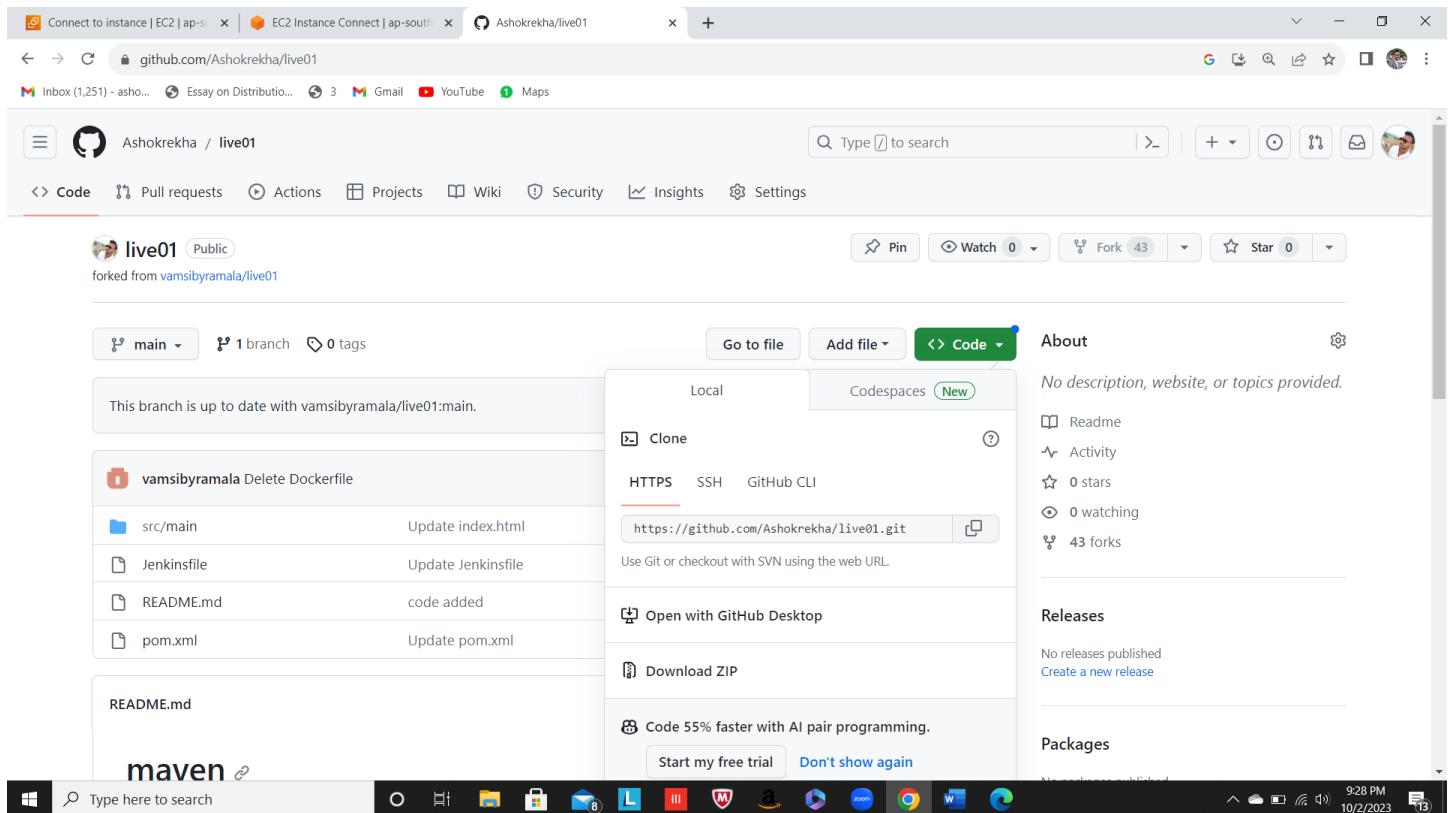
Type here to search

12:00 AM 10/2/2023

3. Configure Jenkins job to pull the latest code from git sample repository (step 2) and build the java projects.

Create a Job:

- Click on New item.
- Enter a Job Name:live01, select “Freestyle project” and hit the “OK” button.
- You will be redirected to the job configuration page where you can see the following settings
- **General Settings:** The section contains the general setting of the job like discarding old builds, support parameters, Disabling the project, etc.
- **Source Code Management:** The section contains source code options such as GIT, SVN, etc.
- **Build Triggers:** The section contents trigger settings that trigger the build based on the specific condition match
- **Build:** The section contains the build steps that can be performed by adding a Batch or shell command
- **Post-build Actions:** The section contains the build steps that can be performed after the build action.
- Build a job by clicking on the “*Build Now*”.
- Check console output.



Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Ashokrekha/live01 | +

← → C 🔍 ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-088fc0968b21160c6&osUser=root®ion=ap-south-1&sshPort=22#/

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

AWS Services Search [Alt+S]

```
[root@ip-172-31-10-235 ~]# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.40.1-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

Package	Arch	Version	Repository	Size
---------	------	---------	------------	------

i-088fc0968b21160c6 (project-1)

PublicIPs: 3.110.158.56 PrivateIPs: 172.31.10.235

CloudShell Feedback

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Type here to search

9:32 PM 10/2/2023

Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | Ashokrekha/live01 | +

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

AWS Services Search [Alt+S]

```
[root@ip-172-31-10-235 ~]# git clone https://github.com/Ashokrekha/live01.git
Cloning into 'live01'...
remote: Enumerating objects: 182, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 182 (delta 8), reused 16 (delta 4), pack-reused 159
Receiving objects: 100% (182/182), 744.16 KiB | 13.05 MiB/s, done.
Resolving deltas: 100% (15/15), done.
[root@ip-172-31-10-235 ~]# ls
live01
[root@ip-172-31-10-235 ~]# cd live01/
[root@ip-172-31-10-235 live01]# ls -a
. .. .git Jenkinsfile pom.xml README.md src
[root@ip-172-31-10-235 live01]# cd .git/
[root@ip-172-31-10-235 .git]# ls
branches config description HEAD hooks index info logs objects packed-refs refs
[root@ip-172-31-10-235 .git]#
```

i-088fc0968b21160c6 (project-1)

PublicIPs: 3.110.158.56 PrivateIPs: 172.31.10.235

CloudShell Feedback

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Type here to search

9:35 PM 10/2/2023

- git clone repository code.
- ls (live01).
- cd live01(. git, jenkinsfile, pom.xml, README.md, src).

```
[root@ip-172-31-10-235 ~]# yum install java -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package 1:java-17-amazon-corretto-17.0.8+8-1.amzn2.1.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-10-235 ~]# java --version
openjdk 17.0.8.1 2023-08-22 LTS
OpenJDK Runtime Environment Corretto-17.0.8.8.1 (build 17.0.8.1+8-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.8.8.1 (build 17.0.8.1+8-LTS, mixed mode, sharing)
[root@ip-172-31-10-235 ~]#
```

i-088fc0968b21160c6 (project-1)
PublicIPs: 3.110.158.56 PrivateIPs: 172.31.10.235

```
[root@ip-172-31-10-235 ~]# mvn -v
Apache Maven 3.0.5 (Red Hat 3.0.5-17)
Maven home: /usr/share/maven
Java version: 17.0.8.1, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.192-183.736.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-10-235 ~]#
```

i-088fc0968b21160c6 (project-1)
PublicIPs: 3.110.158.56 PrivateIPs: 172.31.10.235

- install maven (yum install maven -y).
- mvn -v (to check maven version).
- Now go to Jenkins dashboard.
- Create new job > new item: live01.

- Freestyle project > ok.
- Do configure settings as shown in below images.
- Build now and check console output.
- Go to server check cd /var/lib/jenkins/workspace/live01/
- ls (jenkinsfile, pom.xml, README.md, src).
- Build steps > invoke top-level maven target.
- Maven version: maven3
- Goals: install
- Apply and save
- Build now and check console output.
- Go to server check cd /var/lib/jenkins/workspace/live01/
- ls (jenkinsfile, pom.xml, README.md, src, target).

The screenshot shows the Jenkins configuration page for the 'live01' job. The left sidebar lists configuration sections: General (selected), Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main content area is titled 'Configure' and shows the 'Discard old builds' section. Under 'Strategy', 'Log Rotation' is selected. The 'Days to keep builds' field is empty, and the 'Max # of builds to keep' field contains the value '2'. Below these fields are three unchecked checkboxes: 'GitHub project', 'This project is parameterised', and 'Throttle builds'. At the bottom are 'Save' and 'Apply' buttons. The browser's address bar shows the URL as 3.110.92.60:8080/job/live01/configure. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar View, Mail, Photos, OneDrive, Edge, Google Chrome, and File Explorer.

The screenshot shows the Jenkins configuration page for a project named 'live01'. The 'General' tab is selected. Under 'Source Code Management', the 'Git' provider is chosen, and the repository URL is set to `https://github.com/Ashokrekha/live01.git`. The 'Credentials' dropdown is set to '- none -'. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > live01 > Configuration

Configure

JDK
JDK to be used for this project
openJDK8

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Source Code Management

None

Git

Repositories

Repository URL: `https://github.com/Ashokrekha/live01.git`

Credentials: - none -

Save Apply

The screenshot shows the Jenkins configuration page for a project named 'live01'. The 'Build Triggers' tab is selected. Under 'Branches to build', the 'Branch Specifier' is set to `*/main`. The 'Repository browser' is set to '(Auto)'. In the 'Additional Behaviours' section, there is an 'Add' button. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > live01 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Branches to build

Branch Specifier (blank for 'any')
*/main

Add Branch

Repository browser
(Auto)

Additional Behaviours

Add

Build Triggers

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

Build after other projects are built

Save Apply

The screenshot shows a Jenkins console output page. The title bar includes tabs for various Jenkins instances and a search bar. The main content area has a header "Console Output" with a green checkmark icon. On the left, there's a sidebar with links for Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build, Git Build Data, Docker Fingerprints, and Previous Build. The main pane displays the build log:

```
Started by user Ashok
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/live01
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/live01/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Ashokrekha/live01.git # timeout=10
Fetching upstream changes from https://github.com/Ashokrekha/live01.git
> git --version # timeout=10
> git --version # 'git' version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Ashokrekha/live01.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 811741af5ee028be8c2cb756d87e536c417a8628 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 811741af5ee028be8c2cb756d87e536c417a8628 # timeout=10
Commit message: "Delete Dockerfile"
> git rev-list --no-walk 811741af5ee028be8c2cb756d87e536c417a8628 # timeout=10
[live01] $ /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/maven3/bin/mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Project Webapp archetype 0.0.1-SNAPSHOT
[INFO] -----
```

The screenshot shows an AWS CloudShell terminal window. The title bar includes tabs for various AWS services and a search bar. The main pane displays a terminal session with the following command history:

```
aws Services Search [Alt+S]
[root@ip-172-31-10-235 ~]# cd /var/lib/jenkins/workspace/live01/
[root@ip-172-31-10-235 live01]# ls
Dockerfile Jenkinsfile pom.xml README.md src target
[root@ip-172-31-10-235 live01]#
```

Below the terminal, a message box displays:

i-088fc0968b21160c6 (project-1)
PublicIPs: 3.110.92.60 PrivateIPs: 172.31.10.235

The screenshot shows the Jenkins configuration interface for a job named 'live01'. The left sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Configure' and contains two build step sections:

- Build Steps**:
 - Invoke top-level Maven targets**: Maven Version is set to 'maven3' and Goals are set to 'install'. An 'Advanced' button is present.
 - Docker Build and Publish**: Repository Name is set to 'ashok223/live01'.

At the bottom are 'Save' and 'Apply' buttons. The taskbar at the bottom shows various open windows including 'Inbox - ashokrekha5...', 'Folder shared with y...', '4.DevOps - Google...', 'Connect to instance | x', 'EC2 Instance Connect | x', 'live01 Config [Jenkins] | x', 'ashok223/live01 gen | x', and system icons.

The screenshot shows the Jenkins console output for a build step. The log message is as follows:

```
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/live01/src/main/resources
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ vamsi ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ vamsi ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/live01/src/test/resources
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ vamsi ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ vamsi ---
[INFO] No tests to run.
[INFO]
[INFO] --- war:3.3.1:war (default-war) @ vamsi ---
[INFO] Packaging webapp
[INFO] Assembling webapp [vamsi] in [/var/lib/jenkins/workspace/live01/target/live]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/live01/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/live01/target/live.war
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ vamsi ---
[INFO] Installing /var/lib/jenkins/workspace/live01/pom.xml to /var/lib/jenkins/.m2/repository/vamsi/maven/com/vamsi/0.0.1-SNAPSHOT/vamsi-0.0.1-SNAPSHOT.pom
[INFO] Installing /var/lib/jenkins/workspace/live01/target/live.war to /var/lib/jenkins/.m2/repository/vamsi/maven/com/vamsi/0.0.1-SNAPSHOT/vamsi-0.0.1-SNAPSHOT.war
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
```

The taskbar at the bottom shows various open windows including 'Inbox - ashokrekha5...', 'Folder shared with y...', '4.DevOps - Google...', 'Connect to instance | x', 'EC2 Instance Connect | x', 'live01 #4 ashok223/live01 | x', 'ashok223/live01 gen | x', and system icons.

A screenshot of a web browser window displaying an EC2 instance terminal session. The URL is `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-088fc0968b21160c6&osUser=root®ion=ap-south-1&sshPort=22#/`. The terminal window shows the following command history:

```
[root@ip-172-31-10-235 ~]# cd /var/lib/jenkins/workspace/live01/
[root@ip-172-31-10-235 live01]# ls
Dockerfile Jenkinsfile pom.xml README.md src target
[root@ip-172-31-10-235 live01]# cd target/
[root@ip-172-31-10-235 target]# ls
classes generated-sources live live.war maven-archiver maven-status
[root@ip-172-31-10-235 target]#
```

i-088fc0968b21160c6 (project-1)
Public IPs: 3.110.92.60 Private IPs: 172.31.10.235



4. Continue to configure your Jenkins job to build a docker image for the project and push the working docker image on the Docker Hub.

Docker Build and Publish with Jenkins:

Jenkins is a continuous integration platform which can automate almost all the software development platforms in terms of build the code, publish the package to the repository and deploying it. Before getting into automating the Docker Image Build and Publish process, let's discuss shortly on how this Docker build and publish is working. When the developer is working with the Docker application, a docker image will be created by docker build . command provisioned from Dockerfile. Just like .jar or .war file java applications are stored in the antifactory repository, docker image can be stored in a repository called docker registry and the same can be pulled when we need to deploy it on application runtime servers.

A plugin from Cloudbees called [CloudBees Docker Build and Publish plugin](#). To install this plugin, follow the following steps.

1. Go to **Manage Jenkins -> Manage Plugins**.
 2. Click the **Available** tab.
 3. Type **Docker Build and Publish** in the **Filter** box.
 4. Check the **CloudBees Docker Build and Publish plugin** and click **install** button
 5. This will install the plugin
- Add Dockerfile in github.
 - Add plugins shown in below image.
 - Install docker in project-1 insatnce.
 - Now start docker.
 - Change user mode to jenkins.
 - Restart jenkins.
 - Go to jenkins dashboard.
 - In build steps add docker build and publish.
 - Do changes as shown in below images.
 - Apply and save.
 - Build now and check console output.
 - Go to project-1 instance and check created image ashok223/live01.
 - Go to Docker hub and check pushed docker image ashok223/live01.

A screenshot of a web browser displaying a GitHub repository named 'live01'. The user is editing the 'Dockerfile' in the 'main' branch. The code currently contains two lines:

```
1 FROM tomcat:9
2 COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
```

The GitHub interface includes a sidebar with files like 'src', 'Dockerfile', 'Jenkinsfile', 'README.md', and 'pom.xml'. A status bar at the bottom shows keyboard shortcuts for navigating tabs.

A screenshot of a web browser displaying the Jenkins plugin manager. The search bar is set to 'docker'. The results show several Docker-related plugins:

- Docker API Plugin**: Version 3.3.1-79.v20b_53427e041. Status: Enabled (green checkmark).
- Docker Commons Plugin**: Version 439.va_3cb_0a_6a_fb_29. Status: Enabled (green checkmark).
- Docker Pipeline**: Version 572.v950f58993843. Status: Enabled (green checkmark).
- docker-build-step**: Version 2.9. Status: Enabled (green checkmark).

The Jenkins header bar shows the URL as 'Not secure | 3.110.158.56:8080/manage/pluginManager/installed'.

Screenshot of a web browser showing the Jenkins 'Installed plugins' page. A search bar at the top contains the text 'cloudbees'. The results list three plugins:

- CloudBees Docker Build and Publish plugin** 1.4.0: Described as enabling building Dockerfile based projects and publishing built images/repos to the docker registry. It is enabled.
- Folders Plugin** 6.848.v3b_fd7839a_81: Described as allowing users to create "folders" to organize jobs. It is enabled.
- Github Branch Source Plugin** 1741.va_3028eb_9fd21: Described as supporting Multibranch projects and organization folders from GitHub. It is enabled.

The Jenkins interface includes a sidebar with links for Updates, Available plugins, Installed plugins (which is selected), Advanced settings, and Download progress. The bottom right corner shows 'REST API' and 'Jenkins 2.414.2'.

Screenshot of a web browser showing the Jenkins 'Download progress' page. The sidebar indicates the 'Download progress' tab is selected. The main content area shows the progress of various plugin downloads:

Task	Status
Authentication Tokens API	Success
Docker Commons	Success
SSH server	Success
CloudBees Docker Build and Publish	Success
Loading plugin extensions	Success
Docker Pipeline	Success
Loading plugin extensions	Success
Javadoc	Success
JSch dependency	Success
Maven Integration	Success
Apache HttpComponents Client 5.x API	Success
Docker API	Success
docker-build-step	Success
Loading plugin extensions	Success

The Jenkins interface includes a sidebar with links for Updates, Available plugins, Installed plugins (which is selected), Advanced settings, and Download progress. The bottom right corner shows '10:45 PM' and '10/2/2023'.

[root@ip-172-31-10-235 ~]# yum install docker -y

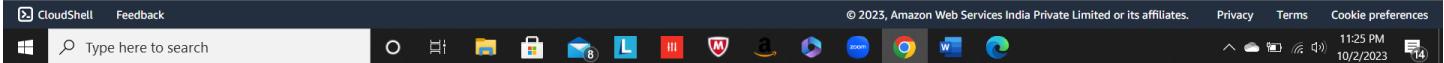
```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.23-1.amzn2.0.1 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: pigz for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.6.19-1.amzn2.0.3 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.7-3.amzn2 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
docker	x86_64	20.10.23-1.amzn2.0.1	amzn2extra-docker	41 M
Installing for dependencies:				
containerd	x86_64	1.6.19-1.amzn2.0.3	amzn2extra-docker	28 M

i-088fc0968b21160c6 (project-1)

Public IPs: 3.110.158.56 Private IPs: 172.31.10.235



[root@ip-172-31-10-235 ~]# systemctl start docker

[root@ip-172-31-10-235 ~]# systemctl status docker

```
* docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-10-02 17:55:22 UTC; 20s ago
     Docs: https://docs.docker.com
   Process: 10195 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Process: 10193 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 10198 (dockerd)
    Tasks: 7
   Memory: 79.9M
      CPU: 0.000 CPU(s) (0.000%)
      CGroup: /system.slice/docker.service
              └─10198 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

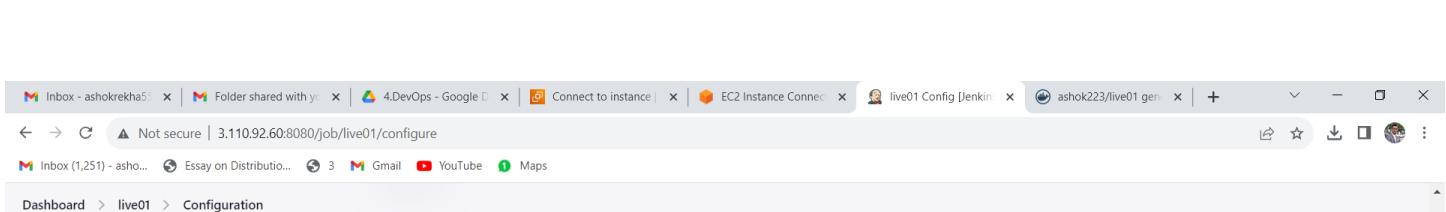
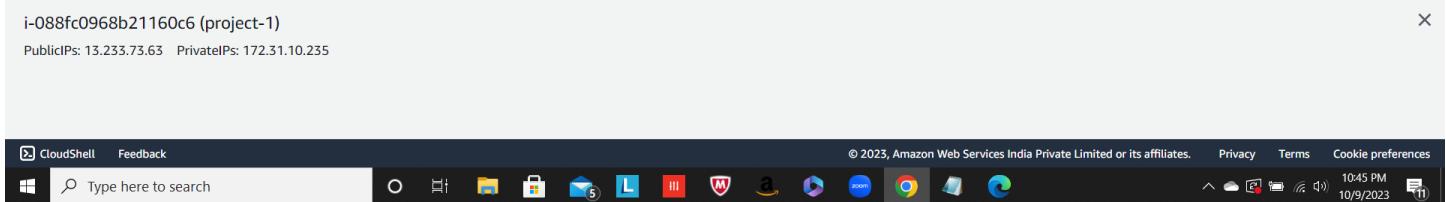
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.320686973Z" level=info msg="ClientConn switching balancer t...le=grpc
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.358570017Z" level=warning msg="Your kernel does not support...weight"
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.359011239Z" level=warning msg="Your kernel does not support...device"
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.359975582Z" level=info msg="Loading containers: start."
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.566389351Z" level=info msg="Default bridge (docker0) is ass...dress"
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.623122483Z" level=info msg="Loading containers: done."
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.721048146Z" level=info msg="Docker daemon" commit=6051f14 g...0.10.23
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.7215956177Z" level=info msg="Daemon has completed initialization"
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal systemd[1]: Started Docker Application Container Engine.
Oct 02 17:55:22 ip-172-31-10-235.ap-south-1.compute.internal dockerd[10198]: time="2023-10-02T17:55:22.758728543Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
```

i-088fc0968b21160c6 (project-1)

Public IPs: 3.110.158.56 Private IPs: 172.31.10.235



```
[root@ip-172-31-10-235 ~]# systemctl start jenkins
[root@ip-172-31-10-235 ~]# usermod -a -G docker jenkins
[root@ip-172-31-10-235 ~]# systemctl restart jenkins
[root@ip-172-31-10-235 ~]#
```



Configure

Docker Build and Publish

Repository Name: ashok223/live01

Tag: \${BUILD_NUMBER}

Docker Host URI:

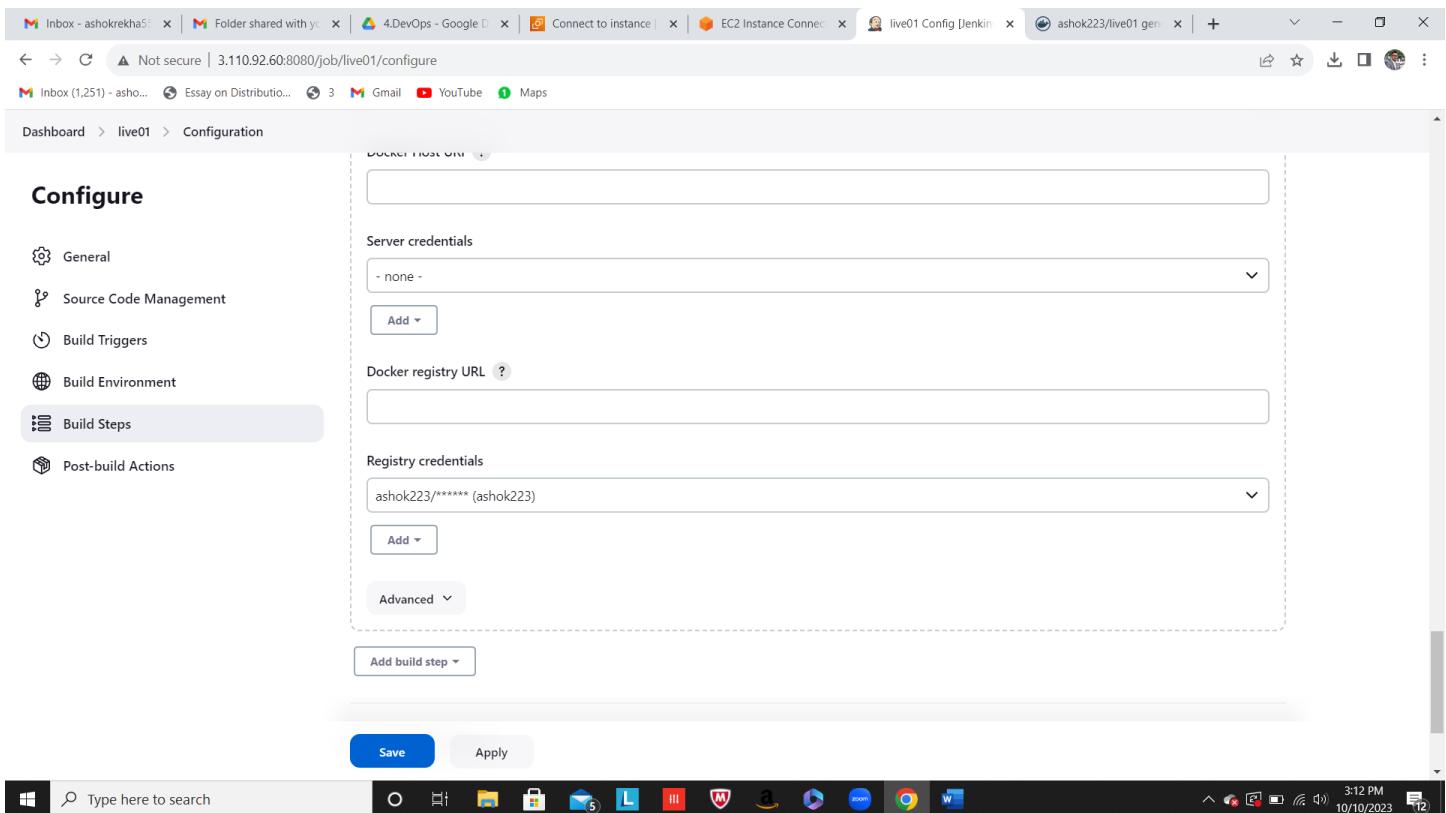
Server credentials: - none -

Add

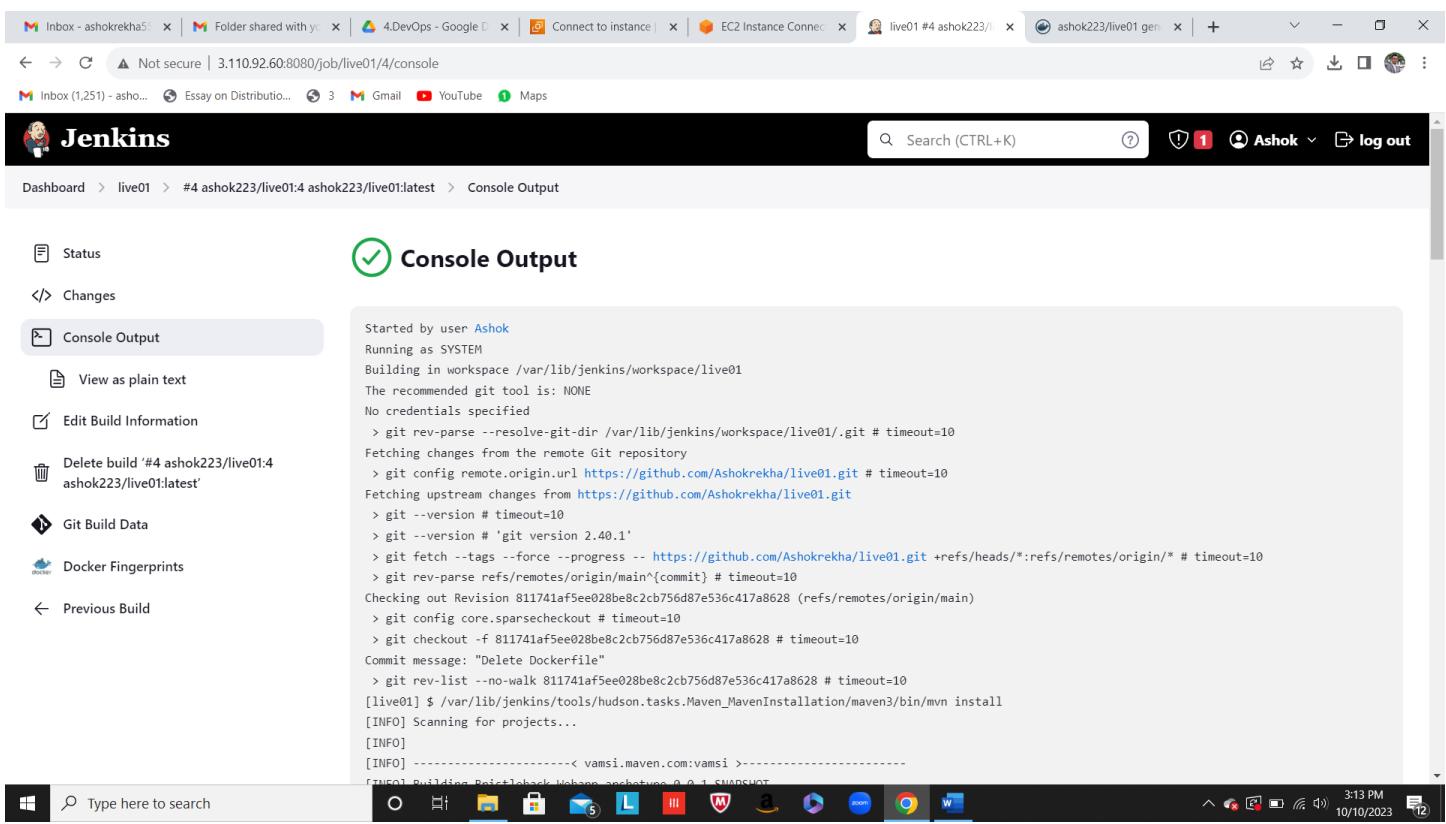
Docker registry URL:

Registry credentials

Save Apply



The screenshot shows the Jenkins configuration interface for a job named 'live01'. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment, **Build Steps**, and Post-build Actions. The main panel is titled 'Configure' and contains fields for 'Server credentials' (set to 'none'), 'Docker registry URL' (empty), 'Registry credentials' (set to 'ashok223/***** (ashok223)'), and an 'Advanced' section. At the bottom are 'Save' and 'Apply' buttons.



The screenshot shows the Jenkins console output for build #4 of the 'live01' job. The output details the build process, starting with cloning from a GitHub repository and then executing Maven commands to build the project. The output ends with a successful build message.

```
Started by user Ashok
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/live01
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/live01/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Ashokrekha/live01.git # timeout=10
Fetching upstream changes from https://github.com/Ashokrekha/live01.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Ashokrekha/live01.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 811741af5ee028be8c2cb756d87e536c417a8628 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 811741af5ee028be8c2cb756d87e536c417a8628 # timeout=10
Commit message: "Delete Dockerfile"
> git rev-list --no-walk 811741af5ee028be8c2cb756d87e536c417a8628 # timeout=10
[live01] $ /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/maven3/bin/mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Project-Maven 0.0.1-SNAPSHOT
[INFO] -----------------------------------------------------------------
```

The screenshot shows a browser window with multiple tabs open. The active tab displays the Docker build output for a container named 'live01'. The logs show the preparation of various Docker layers, with many layers being marked as 'already exists'. The final line of the log indicates a successful build with a digest and size of 2414.

```
WARNING: Support for the legacy /v2/docker/ configuation file and v1 API format is deprecated and will be removed in an upcoming release
The push refers to repository [docker.io/ashok223/live01]
322d1f7084fc: Preparing
75119d43939: Preparing
f684a60d6850: Preparing
a0fe8de9470f: Preparing
9be8a63d2be6: Preparing
cd59091cblbf: Preparing
151ff94a03ea: Preparing
a7b7cb7db02e: Preparing
793368f2be0c: Preparing
01d4e4b4f381: Preparing
cd59091cblbf: Waiting
151ff94a03ea: Waiting
a7b7cb7db02e: Waiting
793368f2be0c: Waiting
01d4e4b4f381: Waiting
9be8a63d2be6: Layer already exists
f684a60d6850: Layer already exists
a0fe8de9470f: Layer already exists
75119d43939: Layer already exists
a7b7cb7db02e: Layer already exists
cd59091cblbf: Layer already exists
151ff94a03ea: Layer already exists
322d1f7084fc: Layer already exists
793368f2be0c: Layer already exists
01d4e4b4f381: Layer already exists
latest: digest: sha256:88abcb02c620b8c27c51600b93e88458341bf8ade65b9185404ec95adf2ee2ce size: 2414
Finished: SUCCESS
```



The screenshot shows an AWS CloudShell interface. A terminal window is open, displaying the output of the 'docker images' command. The output lists several Docker images, including 'ashok223/live01' (tag 4) and 'test' (tag latest), both created 16 hours ago with a size of 439MB. Other images listed are 'tomcat' (tag 9) and 'i-088fc0968b21160c6' (tag latest).

```
[root@ip-172-31-10-235 ~]# docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
ashok223/live01    4        440768d9f37a  16 hours ago  439MB
ashok223/live01    latest   440768d9f37a  16 hours ago  439MB
test                latest   440768d9f37a  16 hours ago  439MB
tomcat              9        ce95ab0490a9  7 days ago   427MB
[root@ip-172-31-10-235 ~]#
```



Inbox - ashokrekha5... | Folder shared with you... | 4.DevOps - Google... | Connect to instance | EC2 Instance Connect | live01 #4 ashok223... | ashok223/live01 Tag... +

hub.docker.com/r/ashok223/live01/tags

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

Hackathon time. Join us for the Docker AI/ML Hackathon now through November 7th. [Join us](#)

docker hub Search Docker Hub Explore Repositories Organizations Help Upgrade ashok223

Explore ashok223/live01

ashok223/live01 ☆ Manage Repository

Image

Pulls 1

Overview Tags

Sort by Newest Filter Tags

TAG

latest Last pushed 27 minutes ago by ashok223

DIGEST 88abcb02c620 OS/ARCH linux/amd64 LAST PULL 22 minutes ago COMPRESSED SIZE 207.07 MB

https://hub.docker.com/r/ashok223/live01/tags

Type here to search

docker pull ashok223/live01:lat...

3:30 PM 10/10/2023

5. Pull the docker image from the registry and deploy it on a local machine or EC2 instance.

Introduction to Docker pull:

[Docker](#) is a popular open-source platform that allows developers to automate the deployment and scaling of applications using containerization. Containerization provides a lightweight and isolated environment to run applications consistently across different systems. In the Docker ecosystem, the process of acquiring container images from a remote repository is known as “pulling.” The Docker pull command plays a crucial role in obtaining container images from a specified Docker registry. It allows developers to fetch pre-built images that can be used as the base for building and running containerized applications. By pulling images from a registry, developers can save time and effort by leveraging existing solutions and dependencies instead of starting from scratch.

- docker pull ashok223/live01 (to pull docker image from docker hub).
- docker images (to check created image).
- Now deploy created image it will create container.
(docker run -d -p 8081:8080 --name project-1 ashok223/live01).
- docker ps -a (to check all created containers).
- Now check application with public ip:8081.

```
[root@ip-172-31-10-235 ~]# docker pull ashok223/live01
Using default tag: latest
latest: Pulling from ashok223/live01
Digest: sha256:88abccb02c620b8c27c51600b93e088458341bf8ade65b9185404ec95adf2ee2ce
Status: Downloaded newer image for ashok223/live01:latest
docker.io/ashok223/live01:latest
[root@ip-172-31-10-235 ~]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
ashok223/live01    4        440768d9f37a   17 hours ago  439MB
ashok223/live01    latest   440768d9f37a   17 hours ago  439MB
tomcat             9        ce95ab0490a9   7 days ago   427MB
[root@ip-172-31-10-235 ~]# docker run -d -p 8081:8080 --name project-1 ashok223/live01
f6723d08c2b60735ff4640b294cf340f33305832a9764f29b3dc59f5fe51aaab
[root@ip-172-31-10-235 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
f6723d08c2b6        ashok223/live01   "catalina.sh run"   16 seconds ago   Up 15 seconds     0.0.0.0:8081->8080/tcp, :::8081->8080/tcp   project-1
1b22b189f47a        440768d9f37a   "bash"             17 hours ago     Exited (0) 17 hours ago   peaceful_kalam
[root@ip-172-31-10-235 ~]# 
```

i-088fc0968b21160c6 (project-1)
PublicIPs: 3.110.92.60 PrivateIPs: 172.31.10.235

Folder shared with you | 4.DevOps - Google | Connect to instance | EC2 Instance Connect | Dashboard [Jenkins] | ashok223/live01 Tag | 3.110.92.60 | + | - | X

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

[AWS Services Search [Alt+S]]

```
[root@ip-172-31-10-235 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f6723d08c2b6 ashok223/live01 "catalina.sh run" 4 minutes ago Up 4 minutes 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp project-1
lb22b189f47a 440768d9f37a "bash" 17 hours ago Exited (0) 17 hours ago
[root@ip-172-31-10-235 ~]# docker exec -it f6723d08c2b6 bash
root@f6723d08c2b6:/usr/local/tomcat# ls
apache-tomcat-9.0.80.tar.gz BUILDING.txt CONTRIBUTING.md LICENSE native-jni-lib README.md RUNNING.txt test webapps.dist
bin conf lib log NOTICE RELEASE-NOTES webapps webxml
root@f6723d08c2b6:/usr/local/tomcat# cd webapps
root@f6723d08c2b6:/usr/local/tomcat/webapps# ls
root@f6723d08c2b6:/usr/local/tomcat/webapps# cd ..
root@f6723d08c2b6:/usr/local/tomcat# cp -r webapps.dist/* webapps/
root@f6723d08c2b6:/usr/local/tomcat# cd webapps
root@f6723d08c2b6:/usr/local/tomcat/webapps# ls
docs examples host-manager manager ROOT
root@f6723d08c2b6:/usr/local/tomcat/webapps# cd ..
root@f6723d08c2b6:/usr/local/tomcat# cd webapps.dist/
root@f6723d08c2b6:/usr/local/tomcat/webapps.dist# ls
docs examples host-manager manager ROOT
root@f6723d08c2b6:/usr/local/tomcat/webapps.dist#
```

i-088fc0968b21160c6 (project-1)

Public IPs: 3.110.92.60 Private IPs: 172.31.10.235



Connect to instance | EC2 | ap-south-1 | EC2 Instance Connect | ap-south-1 | vamsi cake zone

Inbox (1,251) - ashok... Essay on Distributio... 3 Gmail YouTube Maps

CloudShell Feedback

Type here to search

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Windows Start button Taskbar icons System tray

CAKEZONE

EMAIL US info@example.com CALL US +012 345 6789

HOME ABOUT US MENU & PRICING MASTER CHEFS PAGES CONTACT US

Super Crispy CAKEZONE THE BEST CAKE IN LONDON

Read More Play Video

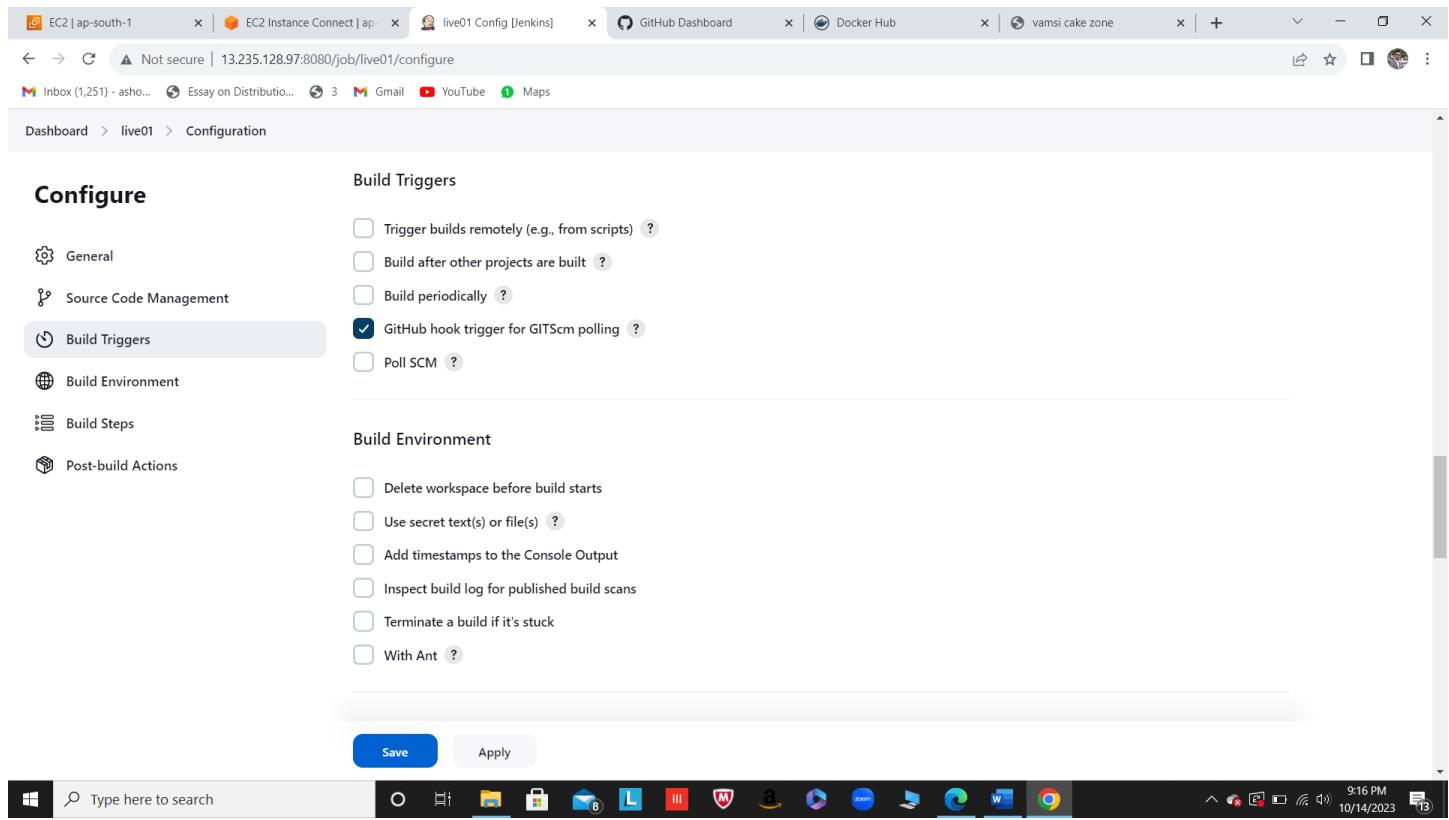
Large image of a chocolate cake with cherries and whipped cream.

Windows Start button Taskbar icons System tray

6. Configure your Jenkins to run automatically on any commits happening to your sample (forked) repository.

Steps for Jenkins GitHub Webhook integration:

- Create a Jenkins build job that uses a GitHub URL.
- Click the *GitHub hook trigger for GITScm polling* checkbox on the build job.
- Create and copy a Jenkins API token for the Jenkins user who will run the build job.
- Create a trigger in your GitHub repository's settings page.
- Set the GitHub payload URL to be your Jenkins' IP address with /github-webhook/ appended to it.
- Set the Jenkins API token as the GitHub webhook secret token.
- Save the GitHub Webhook and then Jenkins builds will occur when a commit is pushed to the repo.



EC2 | ap-south-1 | EC2 Instance Connect | ap- | live01 [Jenkins] | Docker Hub | vamsi cake zone

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

Ashokrekha / live01

Type to search

Code Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Add webhook

Type here to search

9:18 PM 10/14/2023

EC2 | ap-south-1 | EC2 Instance Connect | ap- | live01 [Jenkins] | Add webhook | Docker Hub | vamsi cake zone

Inbox (1,251) - ash... Essay on Distributio... 3 Gmail YouTube Maps

Ashokrekha / live01

Type to search

Code Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *

http://13.235.128.97:8080/job/live01/github_webhook

Content type

application/x-www-form-urlencoded

Secret

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

Let me select individual events

Type here to search

9:21 PM 10/14/2023

Content type
application/x-www-form-urlencoded

Secret

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

Type here to search

Code Pull requests Actions Projects Wiki Security Insights Settings

Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/#ping-event>.

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

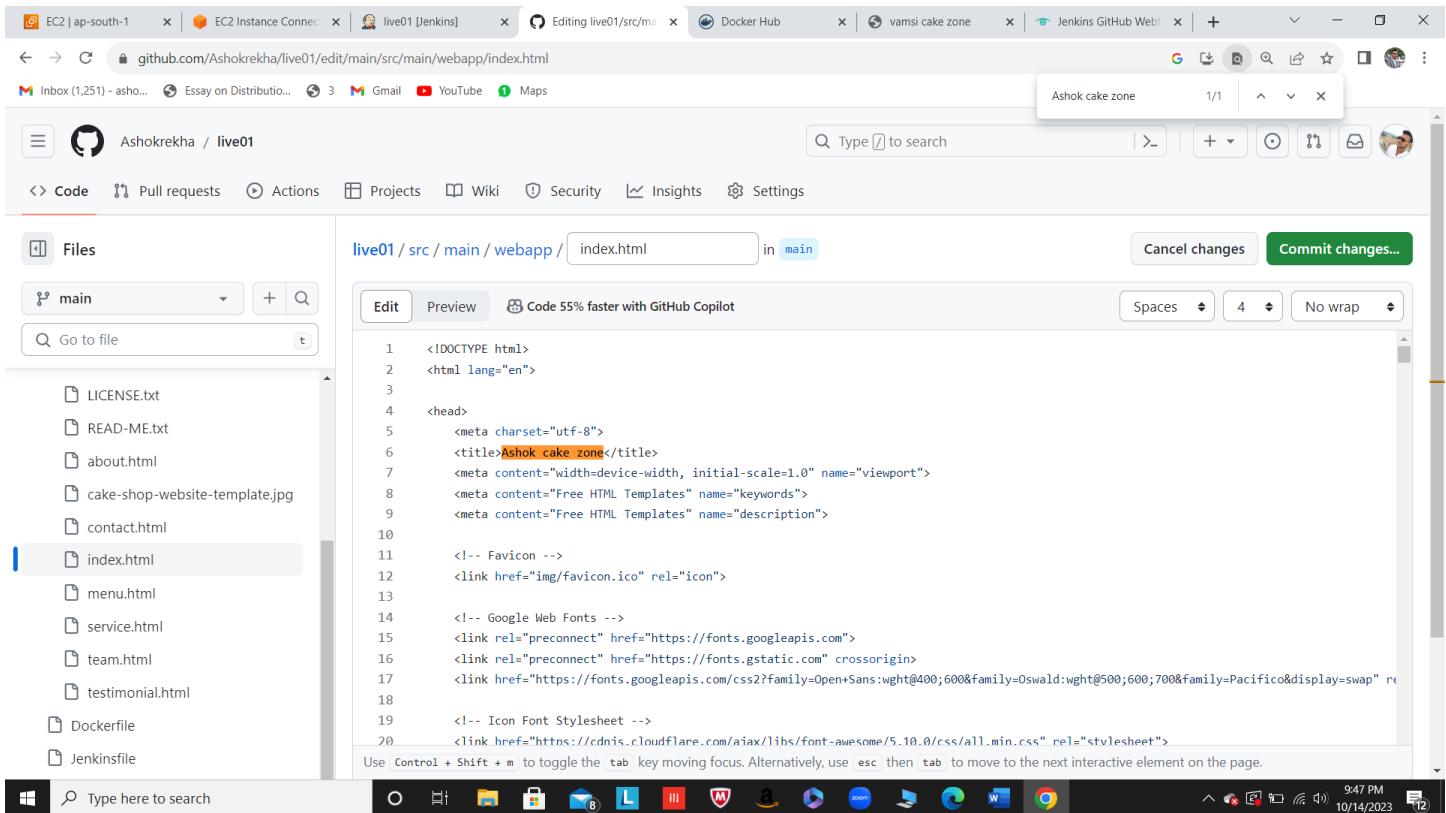
Webhooks

Add webhook

http://13.235.128.97:8080/job/live0... (push)

Edit Delete

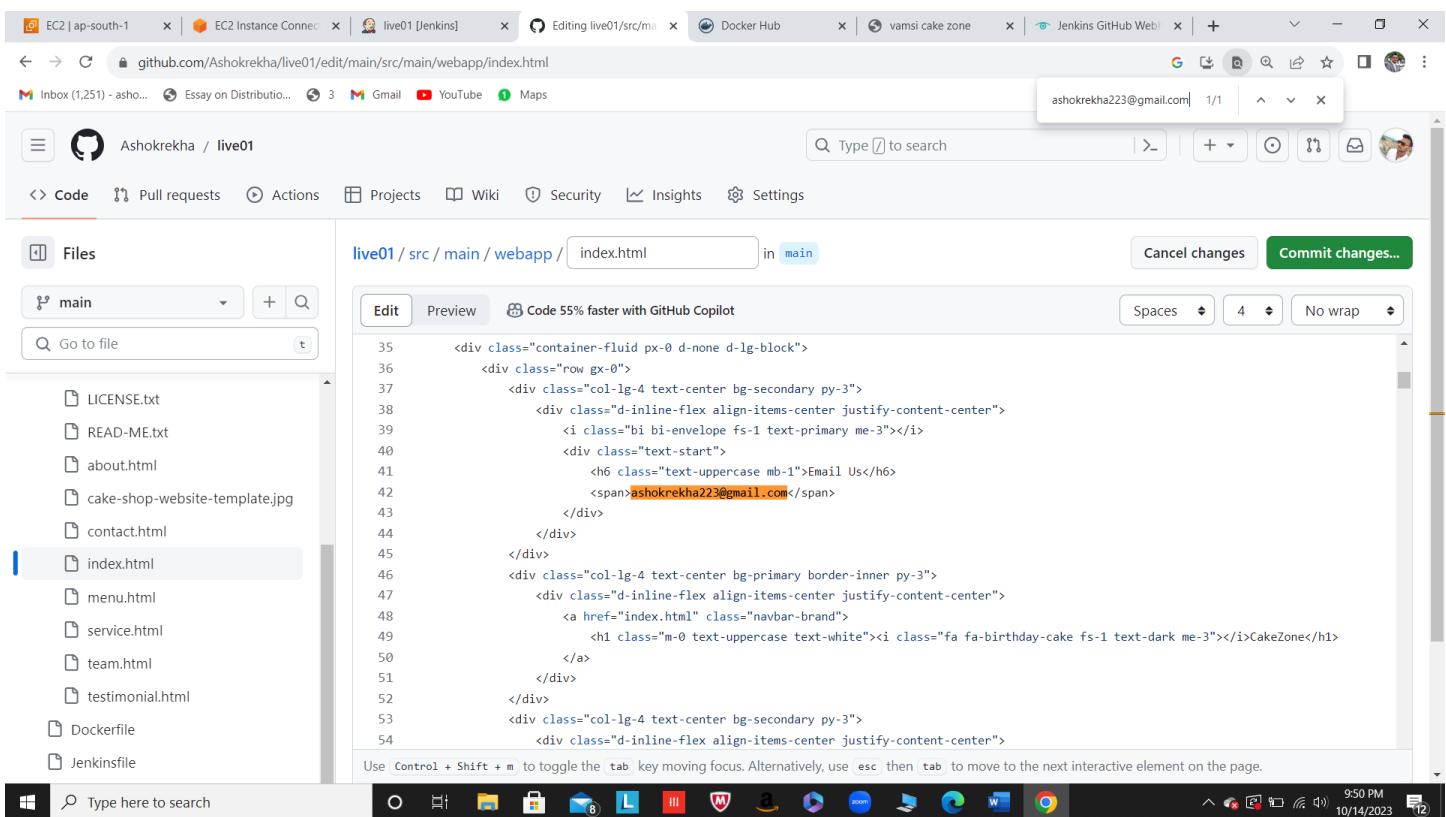
7. Assuming your auto-trigger configurations are correct, now the Jenkins Job should run automatically, and update (push the new) docker image docker registry (as in step 4).



The screenshot shows a web browser window with multiple tabs. The active tab is 'github.com/Ashokrekha/live01/edit/main/src/main/webapp/index.html'. The page displays the GitHub interface for the 'live01' repository, specifically the 'index.html' file. The code editor shows the following snippet:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Ashok cake zone</title>
7   <meta content="width=device-width, initial-scale=1.0" name="viewport">
8   <meta content="Free HTML Templates" name="keywords">
9   <meta content="Free HTML Templates" name="description">
10
11  <!-- Favicon -->
12  <link href="img/favicon.ico" rel="icon">
13
14  <!-- Google Web Fonts -->
15  <link rel="preconnect" href="https://fonts.googleapis.com">
16  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
17  <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&family=Oswald:wght@500;600;700&family=Pacifico&display=swap" rel="stylesheet">
18
19  <!-- Icon Font Stylesheet -->
20  <link href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.10.0/css/all.min.css" rel="stylesheet">
```

The status bar at the bottom right indicates the time is 9:47 PM on 10/14/2023.



The screenshot shows a web browser window with multiple tabs. The active tab is 'github.com/Ashokrekha/live01/edit/main/src/main/webapp/index.html'. The page displays the GitHub interface for the 'live01' repository, specifically the 'index.html' file. The code editor shows the following snippet, with the addition of a new email address:

```
35 <div class="container-fluid px-0 d-none d-lg-block">
36   <div class="row gx-0">
37     <div class="col-lg-4 text-center bg-secondary py-3">
38       <div class="d-inline-flex align-items-center justify-content-center">
39         <i class="bi bi-envelope fs-1 text-primary me-3"></i>
40         <div class="text-start">
41           <h6 class="text-uppercase mb-1">Email Us</h6>
42           <span>ashokrekha223@gmail.com</span>
43         </div>
44       </div>
45     </div>
46     <div class="col-lg-4 text-center bg-primary border-inner py-3">
47       <div class="d-inline-flex align-items-center justify-content-center">
48         <a href="index.html" class="navbar-brand">
49           <h1 class="m-0 text-uppercase text-white"><i class="fa fa-birthday-cake fs-1 text-dark me-3"></i>CakeZone</h1>
50         </a>
51       </div>
52     </div>
53     <div class="col-lg-4 text-center bg-secondary py-3">
54       <div class="d-inline-flex align-items-center justify-content-center">
```

The status bar at the bottom right indicates the time is 9:50 PM on 10/14/2023.

Started by user Ashok
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/live01
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/live01/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Ashokrekha/live01.git # timeout=10
Fetching upstream changes from https://github.com/Ashokrekha/live01.git
> git --version # timeout=10
> git --version # 'git' version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Ashokrekha/live01.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision e05ff9bab7546ac07db4bb698bfd8bfff66c3cd47 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f e05ff9bab7546ac07db4bb698bfd8bfff66c3cd47 # timeout=10
Commit message: "Update index.html"
> git rev-list --no-walk f4589ecaf7d40bb4e833fafbaf24b78888a5d856 # timeout=10
[live01] \$ /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/maven3/bin/mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< vamsi.maven.com:vamsi >-----
[INFO] Building Project Webapp archetype 0.0.1-SNAPSHOT

f7548163a693: Preparing
b346713d7197: Preparing
dc91ca13bf7: Preparing
36f61647eced: Preparing
5bc7f77dbbd9: Preparing
4a597c75ae9e: Preparing
591f657a08e3: Preparing
fb91a4b6eff7: Preparing
256d88da4185: Preparing
4a597c75ae9e: Waiting
591f657a08e3: Waiting
fb91a4b6eff7: Waiting
256d88da4185: Waiting
5bc7f77dbbd9: Mounted from library/tomcat
b346713d7197: Mounted from library/tomcat
dc91ca13bf7: Mounted from library/tomcat
36f61647eced: Mounted from library/tomcat
4a597c75ae9e: Mounted from library/tomcat
591f657a08e3: Mounted from library/tomcat
fb91a4b6eff7: Mounted from library/tomcat
256d88da4185: Mounted from library/tomcat
f7548163a693: Pushed
latest: digest: sha256:578113fd3f1ca662c4638fa4baa87fb4b009adc3aeaf2ad17306261028ef2687 size: 2207
Finished: SUCCESS

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

ashok223 / live01

Description

This repository does not have a description

Last pushed: 4 minutes ago

Docker commands

To push a new tag to this repository:

```
docker push ashok223/live01:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	--	4 minutes ago

[See all](#) [Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

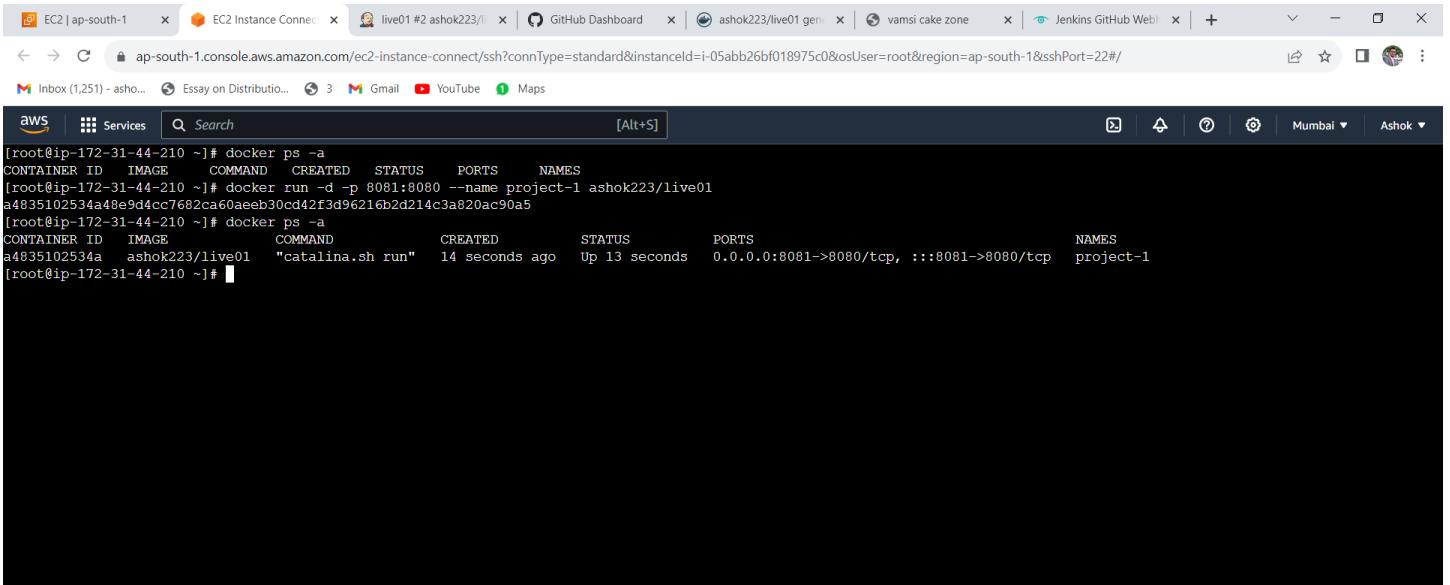
```
[root@ip-172-31-44-210 ~]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED       SIZE
ashok223/live01  latest   438b7dd0332e  3 minutes ago  473MB
tomcat          9        cb2ad02d3907  14 hours ago   455MB
ashok223/live01  <none>   227c6cbd0132  3 days ago    445MB
tomcat          <none>   ce95ab0490a9   11 days ago   427MB
[root@ip-172-31-44-210 ~]#
```

i-05abb26bf018975c0 (project-1)

PublicIPs: 13.235.128.97 PrivateIPs: 172.31.44.210

```
CloudShell Feedback
Type here to search
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
958 PM 10/14/2023
```

8. Deploy the newly pushed docker image on your local or EC2 instance and now it must be showing you the updated web page (site) with the new commit when refreshed in the browser.



```
[root@ip-172-31-44-210 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@ip-172-31-44-210 ~]# docker run -d -p 8081:8080 --name project-1 ashok223/live01
a4835102534a48e9d4cc7692ca60aeeb30cd42f3d96216b2d214c3a820ac90a5
[root@ip-172-31-44-210 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a4835102534a ashok223/live01 "catalina.sh run" 14 seconds ago Up 13 seconds 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp project-1
[root@ip-172-31-44-210 ~]#
```

i-05abb26bf018975c0 (project-1)
Public IPs: 13.235.128.97 Private IPs: 172.31.44.210

