

Jenkins CI/CD Pipeline for Java Application on AWS EC2

Freestyle Job, Pipeline with Jenkinsfile, GitHub Webhooks, and Docker Integration

Author: Oluwaseun Osunsola

LinkedIn: <https://www.linkedin.com/in/oluwaseun-osunsola-95539b175/>

Project Repository: <https://github.com/Oluwaseunoa/DevOps-Projects/tree/main/Jenkins-Projects>

Environment & Tools: AWS EC2 (Ubuntu 22.04 LTS), Jenkins 2.462.x (LTS), GitHub, Docker 27.x, OpenJDK 21

Date Completed: December 2025

Project Overview

This hands-on project documents the complete setup of a **secure and automated Jenkins CI/CD pipeline** on an AWS EC2 Ubuntu instance. The pipeline builds, compiles, and runs a simple Java application while demonstrating key DevOps practices.

Key features implemented:

- Jenkins installation and initial security configuration
- Static Elastic IP for persistent access
- **Freestyle project** with Git polling
- **Declarative Pipeline** using **Jenkinsfile** (Pipeline-as-Code)
- **GitHub webhooks** for real-time triggers (replacing polling)
- **Docker integration** for isolated, reproducible builds

The sample application is a basic Java class **Test.java** that prints a customizable greeting.

All steps include terminal commands, screenshots (in **img/** folder), and explanations. Code snippets are provided for reproducibility.

Sample Application: Test.java

```
public class Test {  
  
    public static void main(String[] args) {  
        System.out.println("*****");  
        System.out.println("Hello from GitHub");  
        System.out.println("Hello from Jenkins CI Pipeline");  
        System.out.println("Hello, will the webhook work?");  
        System.out.println("If you see this, the webhook worked and probably the  
pipeline will be triggered!");  
        System.out.println("*****");  
    }  
}
```

Step-by-Step Documentation

1. Server Setup & Jenkins Installation

1. SSH into Ubuntu EC2 server

```

HP@DESKTOP-I9M74R1 MINGW64 ~/Downloads
$ ssh -i "MyKeyPair.pem" ubuntu@ec2-3-235-78-97.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Dec 20 05:46:46 UTC 2025

System load: 0.01 Temperature: -273.1 C
Usage of /: 25.9% of 6.71GB Processes: 115
Memory usage: 23% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.68.167

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Dec 20 05:46:47 2025 from 102.89.83.166
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-68-167:~$
```

2. Update and upgrade system

```

ubuntu@ip-172-31-68-167:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1391 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [311 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1506 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [225 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9504 B]
Get:22 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [916 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [207 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [306 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71.5 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19.4 kB]
Get:27 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2286 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.4 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2413 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [550 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [516 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.3 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [6048 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [488 B]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [64.0 kB]
```

3. Check if Git is installed

```
MINGW64:/c/Users/HP/Downloads
ubuntu@ip-172-31-68-167:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-68-167:~$
```

4. Install OpenJDK 21

```
MINGW64:/c/Users/HP/Downloads
ubuntu@ip-172-31-68-167:~$ sudo apt install openjdk-21-jre -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme alsamixer-conf at-spi2-common at-spi2-core ca-certificates-java
dconf-gsettings-backend dconf-service fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono
gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2-data
libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64
libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcolord2 libcurl2t64 libdatrie1
libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libepoxy0 libfontconfig1 libgbm1 libgdk-pixbuf-2.0-0
libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglvnd0 libglx-mesa0 libgraphite2-3
libgtk-3-0t64 libgtk-3-bin libgtk-3-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4
libllvm20 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpiciaccess0 libpcsselite1 libpixman-1-0 librsvg2-2
librsvg2-common libsharpuyuv0 libsm6 libthai-data libthai0 libtiff6 libvulkan1 libwayland-client0 libwayland-cursor0
libwayland-egl1 libwayland-server0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0
libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libcomposite1 libcursor1 libxdamage1 libxfixes3
libxf2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6t64 libxtst6 libxv1
libxxf86dga1 libxxf86vm1 mesa-libgallium mesa-vulkan-drivers openjdk-21-jre-headless session-migration ubuntu-mono
x11-common x11-utils
Suggested packages:
default-jre alsamixer libasound2-plugins cups-common gvfs liblcms2-utils pscd librsvg2-bin libnss-mdns
fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
luit
The following NEW packages will be installed:
adwaita-icon-theme alsamixer-conf at-spi2-common at-spi2-core ca-certificates-java
dconf-gsettings-backend dconf-service fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono
gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common libasound2-data
libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64
libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcolord2 libcurl2t64 libdatrie1
libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libepoxy0 libfontconfig1 libgbm1 libgdk-pixbuf-2.0-0
libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglvnd0 libglx-mesa0 libgraphite2-3
libgtk-3-0t64 libgtk-3-bin libgtk-3-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4
libllvm20 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpiciaccess0 libpcsselite1 libpixman-1-0 librsvg2-2
librsvg2-common libsharpuyuv0 libsm6 libthai-data libthai0 libtiff6 libvulkan1 libwayland-client0 libwayland-cursor0
libwayland-egl1 libwayland-server0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0
libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libcomposite1 libcursor1 libxdamage1 libxfixes3
libxf2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6t64 libxtst6 libxv1
```

5. Verify Java version

```
MINGW64:/c/Users/HP/Downloads
ubuntu@ip-172-31-68-167:~$ java --version
openjdk 21.0.9 2025-10-21
openJDK Runtime Environment (build 21.0.9+10-Ubuntu-124.04)
openJDK 64-Bit Server VM (build 21.0.9+10-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-172-31-68-167:~$
```

6. Download Jenkins repository key

```
MINGW64:/c/Users/HP/Downloads

ubuntu@ip-172-31-68-167:~$ sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2025-12-20 06:01:48-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/etc/apt/keyrings/jenkins-keyring.asc'

/etc/apt/keyrings/jenkins-keyring.asc 100%[=====] 3.10K --.-KB/s in 0s

2025-12-20 06:01:48 (47.9 MB/s) - '/etc/apt/keyrings/jenkins-keyring.asc' saved [3175/3175]

ubuntu@ip-172-31-68-167:~$
```

7. Add Jenkins repository to APT sources

```
MINGW64:/c/Users/HP/Downloads

ubuntu@ip-172-31-68-167:~$ echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
ubuntu@ip-172-31-68-167:~$
```

8. Update package list

```
MINGW64:/c/Users/HP/Downloads

ubuntu@ip-172-31-68-167:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [30.3 kB]
Fetched 33.2 kB in 0s (86.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-68-167:~$
```

9. Install Jenkins

```
ubuntu@ip-172-31-68-167:~$ sudo apt install jenkins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 95.2 MB of archives.
After this operation, 96.3 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 net-tools amd64 2.10-0.1ubuntu4.4 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.528.3 [95.0 MB]
Fetched 95.2 MB in 1s (64.4 MB/s)
```

10. Enable and start Jenkins

```
ubuntu@ip-172-31-68-167:~$ sudo systemctl enable --now jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-68-167:~$
```

11. Check Jenkins service status

```
ubuntu@ip-172-31-68-167:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-12-20 06:07:18 UTC; 1min 35s ago
     Main PID: 15292 (java)
        Tasks: 37 (limit: 1008)
       Memory: 304.0M (peak: 319.6M)
          CPU: 18.381s
        CGroup: /system.slice/jenkins.service
                └─15292 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war

Dec 20 06:07:14 ip-172-31-68-167 jenkins[15292]: [LF]> This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Dec 20 06:07:14 ip-172-31-68-167 jenkins[15292]: [LF]>
Dec 20 06:07:14 ip-172-31-68-167 jenkins[15292]: [LF]> ****
Dec 20 06:07:14 ip-172-31-68-167 jenkins[15292]: [LF]> ****
Dec 20 06:07:14 ip-172-31-68-167 jenkins[15292]: [LF]> ****
Dec 20 06:07:18 ip-172-31-68-167 jenkins[15292]: 2025-12-20 06:07:18.147+0000 [id=31]           INFO      jenkins.InitReactorRun
Dec 20 06:07:18 ip-172-31-68-167 jenkins[15292]: 2025-12-20 06:07:18.188+0000 [id=24]           INFO      hudson.lifecycle.Life
Dec 20 06:07:18 ip-172-31-68-167 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Dec 20 06:07:18 ip-172-31-68-167 jenkins[15292]: 2025-12-20 06:07:18.490+0000 [id=50]           INFO      h.m.DownloadService$D
Dec 20 06:07:18 ip-172-31-68-167 jenkins[15292]: 2025-12-20 06:07:18.492+0000 [id=50]           INFO      hudson.util.Retrievers$S
ubuntu@ip-172-31-68-167:~$
```

2. Security Group Configuration

12. Navigate to Security Groups

The screenshot shows the AWS EC2 Instances page for an instance named i-0eadd6520581cf0b8. The instance summary section displays basic details like Public IPv4 address (3.235.78.97), Instance state (Running), and VPC ID (vpc-0ba025afc1daa7b72). The Security tab is selected, highlighting the Security groups section which lists sg-09bdfc346f9a1e053 (Launch-wizard-3). A red arrow points to this entry.

Instance summary for i-0eadd6520581cf0b8 (Ubuntu-Server)	
Updated less than a minute ago	
Instance ID	i-0eadd6520581cf0b8
IPv6 address	-
Hostname type	IP name: ip-172-31-68-167.ec2.internal
Answer private resource DNS name	IPv4 (A)
Auto-assigned IP address	3.235.78.97 [Public IP]
IAM Role	-
IMDSv2	Required
Operator	-
Security	
Security details	
IAM Role	-
Security groups	sg-09bdfc346f9a1e053 (Launch-wizard-3)
Owner ID	832959958705
Launch time	Sat Dec 20 2025 06:42:07 GMT+0100 (West Africa Standard Time)

13. Edit inbound rules

Details

Security group name launch-wizard-3	Security group ID sg-09bdfc346f9ab9853	Description launch-wizard-3 created 2025-12-20T05:40:54.383Z	VPC ID vpc-0ba025afc1daa7b72
Owner 832959958705	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Sharing | VPC associations | Tags

Inbound rules (1/1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0d6062439e44bf23d	IPv4	SSH	TCP	22	0.0.0.0/0	-

Manage tags | Edit inbound rules

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

14. Add port 8080 rule

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID sgr-0d6062439e44bf23d	Type Info SSH	Protocol Info TCP	Port range Info 22	Source Info Custom	Description - optional
-	Custom TCP	TCP	8080	Anywhere	0.0.0.0/0

Add rule

⚠ 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.

Cancel | Preview changes | Save rules

15. Security group updated

The screenshot shows the AWS EC2 Security Groups page. A green banner at the top indicates that inbound security group rules were successfully modified on security group sg-09bdfc346f9ab9853 (launched-wizard-3). Below this, the security group details are shown, including its name (sg-09bdfc346f9ab9853 - launch-wizard-3), security group ID (sg-09bdfc346f9ab9853), owner (832959958705), and VPC ID (vpc-0ba025afc1daa7b72). The inbound rules section shows two entries: one for SSH (TCP port 22) and another for a custom TCP rule (TCP port 8080). The page also includes tabs for Outbound rules, Sharing, VPC associations, and Tags.

3. Initial Jenkins Web Setup

16. Access Jenkins via public IP:8080

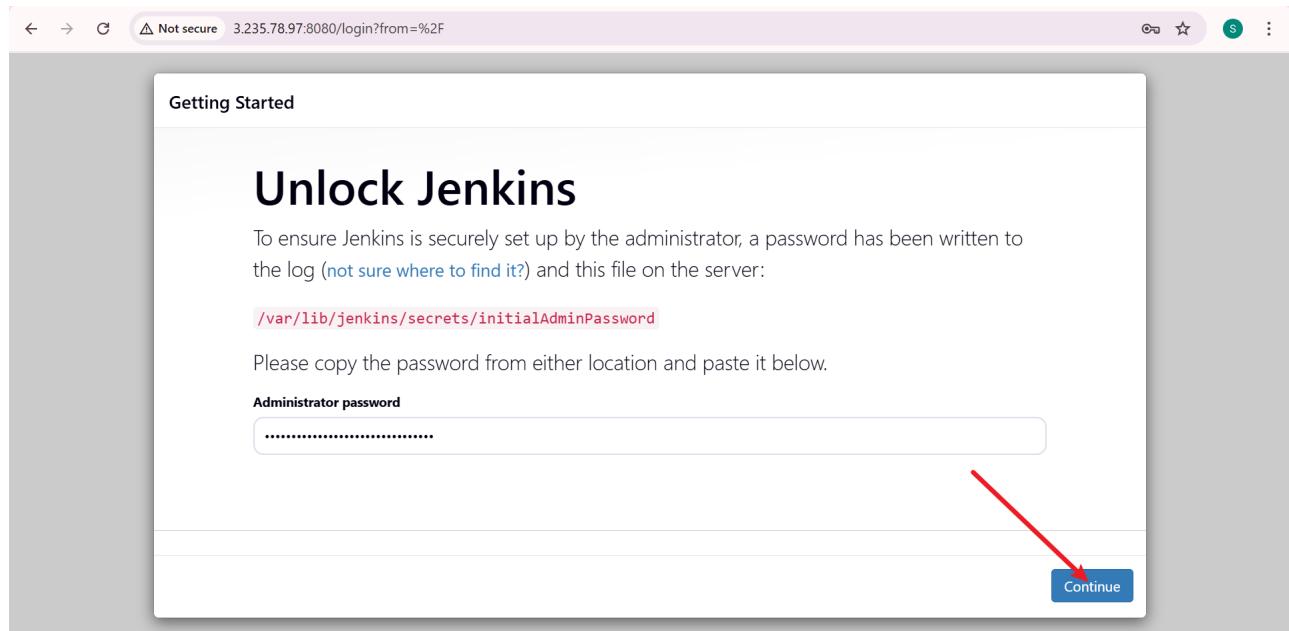
The screenshot shows the Jenkins 'Getting Started' page. It displays the title 'Unlock Jenkins' and instructions for retrieving the initial admin password from the server log. A red arrow points to the 'Administrator password' input field, which is currently empty. A blue 'Continue' button is visible at the bottom right.

17. Retrieve initial admin password

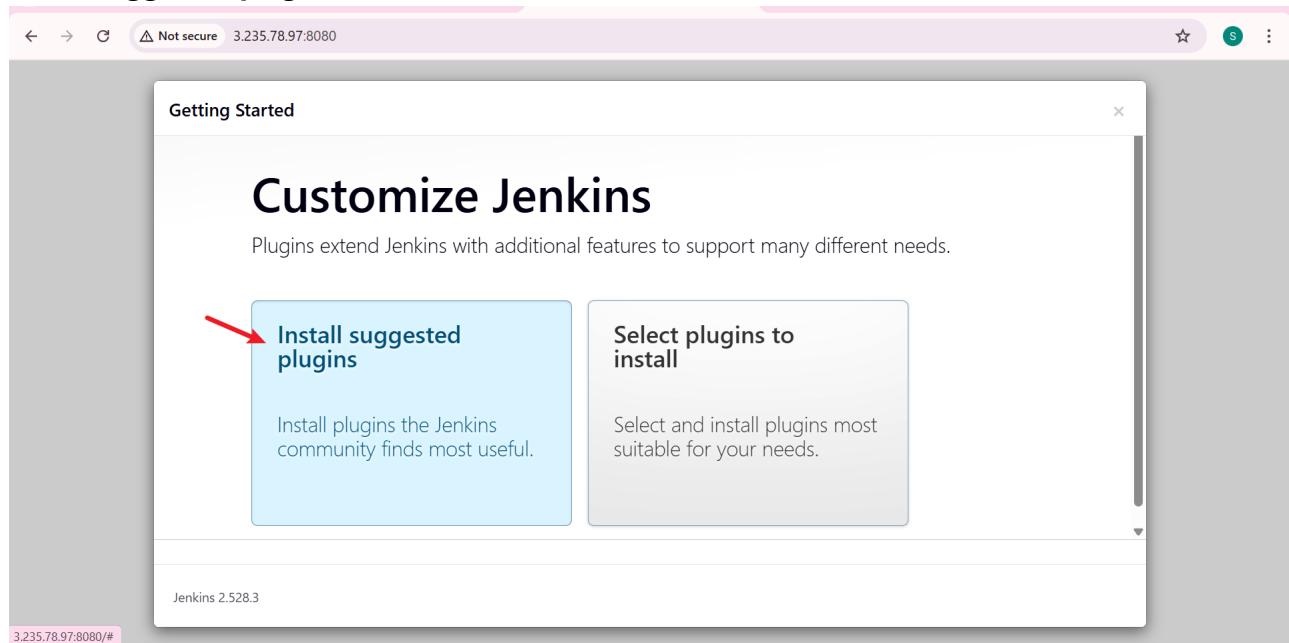
The screenshot shows a terminal window titled 'MINGW64:/c/Users/HP/Downloads'. It displays the command 'sudo cat /var/lib/jenkins/secrets/initialAdminPassword' being run, and the output shows the password 'b82f341341e...dd0b2bd3baf'. A red arrow points to the password output.

```
ubuntu@ip-172-31-68-167:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
b82f341341e...dd0b2bd3baf
```

18. Unlock Jenkins



19. Install suggested plugins



20. Create first admin user

Getting Started

Create First Admin User

Username
Jenkins-Admin-User

Password
.....

Confirm password
.....

Jenkins 2.528.3

Skip and continue as admin

Save and Continue

21. Leave Jenkins URL default

Getting Started

Instance Configuration

Jenkins URL: http://3.235.78.97:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.528.3

Not now

Save and Finish

22. Jenkins dashboard

The screenshot shows the Jenkins dashboard at the URL 3.235.78.97:8080. The page features a header with a Jenkins logo, a search bar, and a 'New Item' button. Below the header, there's a 'Welcome to Jenkins!' message and a section titled 'Start building your software project' with a 'Create a job' button. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). On the right, there are three sections: 'Set up a distributed build' (links to 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'), 'Start building your software project' (with a 'Create a job' button), and 'Welcome to Jenkins!' (with a 'Add description' button). At the bottom right, there are links for 'REST API' and 'Jenkins 2.528.3'.

4. Assign Static Elastic IP

23. Navigate to Elastic IPs

The screenshot shows the AWS EC2 Instances page at the URL us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running. The left sidebar includes sections for Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (with 'Elastic IPs' highlighted by a red arrow), Load Balancing, Auto Scaling, and CloudWatch Metrics. The main content area displays a table of instances, with one row selected for an 'Ubuntu-Server' instance with the ID i-0eadd6520581cf0fb8. The instance is running, has an t3.micro type, and is in the us-east-1f availability zone. It has a Public IPv4 DNS name ec2-3-235-78-97.compute-1.amazonaws.com and a Public IPv4 address 3.235.78.97. The bottom of the page includes links for CloudShell, Feedback, and Console Mobile App, along with copyright information from 2025, Amazon Web Services, Inc. or its affiliates.

24. Allocate Elastic IP

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' page open. On the right, there is a large orange button labeled 'Allocate Elastic IP address'. A red arrow points directly at this button, indicating the next step.

25. Confirm allocation

The screenshot shows the 'Allocate Elastic IP address' confirmation dialog. At the bottom right, there is a large orange button labeled 'Allocate'. A red arrow points directly at this button, indicating the final step to complete the allocation.

26. Name the IP

The screenshot shows the AWS Management Console interface for managing Elastic IP addresses. In the left sidebar, under the 'EC2' section, 'Elastic IP addresses' is selected. A green success message at the top states 'Elastic IP address allocated successfully. Elastic IP address 100.49.165.144'. Below this, the 'Elastic IP addresses (1/1)' table shows one entry: Jenkins-IP, which is a Public IP with Allocation ID eipalloc-07eae73b8c4a2bdf3. A modal window titled 'Edit Name' is open over the table, showing the current name 'Jenkins-IP' in the input field and a red arrow pointing to the 'Save' button.

27. Select the IP

This screenshot shows the same AWS interface as the previous one, but with a red arrow pointing to the IP address '100.49.165.144' in the 'Allocated IPv4 address' column of the table. The table row for Jenkins-IP is highlighted in blue. The rest of the interface, including the summary details below, remains identical to the previous screenshot.

28. Associate Elastic IP

The screenshot shows the AWS EC2 console with the URL `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ElasticIpDetails:AllocationId=eipalloc-07eae73b8c4a2bdf3`. The page displays the details of an Elastic IP address, including its allocation ID, type (Public IP), scope (VPC), and associated instance ID (None). The 'Actions' dropdown menu at the top right contains an orange-highlighted 'Associate Elastic IP address' option.

Summary	
Allocated IPv4 address 100.49.165.144	Type Public IP
Association ID -	Scope VPC
Network interface ID -	Network interface owner account ID -
Address pool Amazon	Network border group us-east-1
	Allocation ID eipalloc-07eae73b8c4a2bdf3
	Associated instance ID -
	Public DNS -
	Service managed -
	Reverse DNS record -
	Private IP address -
	NAT Gateway ID -

Tags(1)

Key	Value
Name	Jenkins-IP

29. Select instance and associate

The screenshot shows the 'Associate Elastic IP address' wizard. It starts with the URL `> Elastic IP addresses > 100.49.165.144 > Associate Elastic IP address`. The first step, 'Associate Elastic IP address', asks to choose an instance or network interface to associate with the Elastic IP address. It shows the selected instance 'i-0eadd6520581cf0b8' and the private IP '172.31.68.167'. The second step, 'Reassociation', asks if the Elastic IP address can be reassigned to another resource. The 'Allow this Elastic IP address to be reassociated' checkbox is unchecked. At the bottom right are 'Cancel' and 'Associate' buttons, with 'Associate' highlighted by a red arrow.

30. Association successful

The screenshot shows the AWS Elastic IP details page for IP address 100.49.165.144. A green banner at the top indicates "Elastic IP address associated successfully." Below this, the IP address is listed as "100.49.165.144". The "Summary" section contains various details such as Allocation ID, Associated instance ID, Public DNS, and Reverse DNS record. A "Tags(1)" section shows a single tag named "Name" with value "Jenkins-IP".

31. Verify EIP on instance

The screenshot shows the AWS Instances page. It lists one instance, "Ubuntu-Server" (ID i-0eadd6520581cf0b8), which is running. The instance has a Public IPv4 address of 100.49.165.144, highlighted with a red box. Below the instances table, the instance details for "Ubuntu-Server" are shown, including its public IP address.

5. Update Jenkins URL to Elastic IP

32. Manage Jenkins

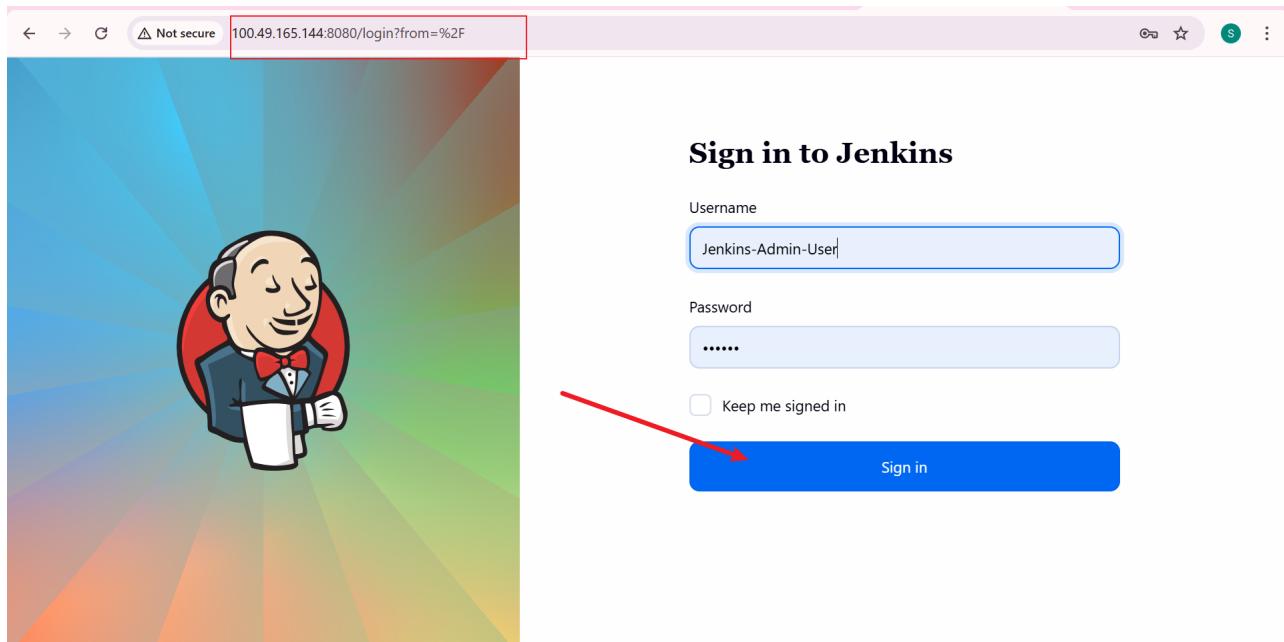
The screenshot shows the Jenkins dashboard at the URL 3.235.78.97:8080. The top navigation bar includes icons for back, forward, search, and more. The main header says "Jenkins" with a logo. Below it, there's a "New Item" button and a "Build History" link. On the left, there are two dropdown menus: "Build Queue" (No builds in the queue) and "Build Executor Status" (0/2). The central area features a "Welcome to Jenkins!" message, a "Start building your software project" section with a "Create a job" button, and a "Set up a distributed build" section with links for "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". In the top right, there's a "Manage Jenkins" button with a notification bubble saying "One or more notifications". At the bottom, there are links for "REST API" and "Jenkins 2.528.3".

33. Temporary site access check

The screenshot shows a browser error page with the title "This site can't be reached". The URL in the address bar is 3.235.78.97:8080/manage. The page includes a network icon, the text "3.235.78.97 took too long to respond.", a "Try:" section with three items, and an "ERR_CONNECTION_TIMED_OUT" error code. There are "Reload" and "Details" buttons at the bottom.

- Checking the connection
- [Checking the proxy and the firewall](#)
- [Running Windows Network Diagnostics](#)

34. Jenkins on new IP



35. Continue to Manage Jenkins

The screenshot shows the Jenkins dashboard at the URL '100.49.165.144:8080/manage'. The dashboard has a header with the Jenkins logo and a 'Welcome to Jenkins!' message. It includes sections for 'Build Queue' (empty), 'Build Executor Status' (0/2), and 'Start building your software project' (with a 'Create a job' button). There is also a 'Set up a distributed build' section with links for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. In the top right corner, there is a 'Manage Jenkins' link with a notification badge. A red arrow points to this 'Manage Jenkins' link.

36. Configure System

The screenshot shows the Jenkins 'Manage Jenkins' interface at the URL <http://100.49.165.144:8080/manage/>. The 'System Configuration' section is highlighted with a red arrow pointing to the 'System' item. Other items in this section include 'Plugins' and 'Clouds'. To the right, there are sections for 'Tools', 'Nodes', and 'Appearance'. A search bar at the top right contains the placeholder 'Search settings'.

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

System Configuration

- System
- Plugins
- Clouds

Tools
Configure tools, their locations and automatic installers.

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

Appearance
Configure the look and feel of Jenkins

Security

37. Jenkins URL section

The screenshot shows the Jenkins 'System' configuration page at the URL <http://100.49.165.144:8080/manage/configure>. The 'Jenkins Location' section is highlighted with a red box around the 'Jenkins URL' input field, which contains the value 'http://3.235.78.97:8080/'. Below it, the 'System Admin e-mail address' field shows 'address not configured yet <nobody@nowhere>'. At the bottom, there are 'Save' and 'Apply' buttons.

0

Restrict project naming

Jenkins Location

Jenkins URL ?
http://3.235.78.97:8080/

System Admin e-mail address ?
address not configured yet <nobody@nowhere>

Serve resource files from another domain

Resources Base URL ?

Save **Apply**

38. Update to Elastic IP

Jenkins / Manage Jenkins / System

0

Restrict project naming

Jenkins Location

Jenkins URL ?
http://100.49.165.144:8080/

System Admin e-mail address ?
address not configured yet <nobody@nowhere>

Notification e-mails from Jenkins to project owners will be sent with this address in the from header. This can be just "foo@acme.org" or it could be something like "Jenkins Daemon <foo@acme.org>"

Save Apply

39. Reload to confirm

← → ⌛ Not secure 100.49.165.144:8080

Jenkins

+ New Item

Build Queue

No builds in the queue.

Build Executor Status

0/2

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 ?

6. Freestyle Job with Git Polling

40. GitHub repo overview

The screenshot shows a GitHub repository page for 'java-code'. The main file displayed is 'Test.java' with the following code:

```
1 < public class Test {  
2 <     public static void main(String[] args) {  
3 <         System.out.println("*****");  
4 <         System.out.println("Hello from GitHub");  
5 <         System.out.println("*****");  
6 <     }  
7 < }  
8 < }
```

The code prints a greeting message 'Hello from GitHub' to the console. The GitHub interface includes a sidebar with 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings' tabs. A search bar at the top right allows users to search for files.

41. Create new job

The screenshot shows the Jenkins dashboard at the URL 100.49.165.144:8080. The 'Jenkins' logo is visible at the top left. In the center, there is a table showing build status for a single item:

S	W	Name	Last Success	Last Failure	Last Duration
		FreestyleMaven	N/A	N/A	

On the left side, there are two dropdown menus: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). At the top right, there is a 'New Item' button with a red arrow pointing to it. Other buttons include 'Build History', 'All', '+', 'Add description', and filter icons. At the bottom right, there are links for 'REST API' and 'Jenkins 2.528.3'.

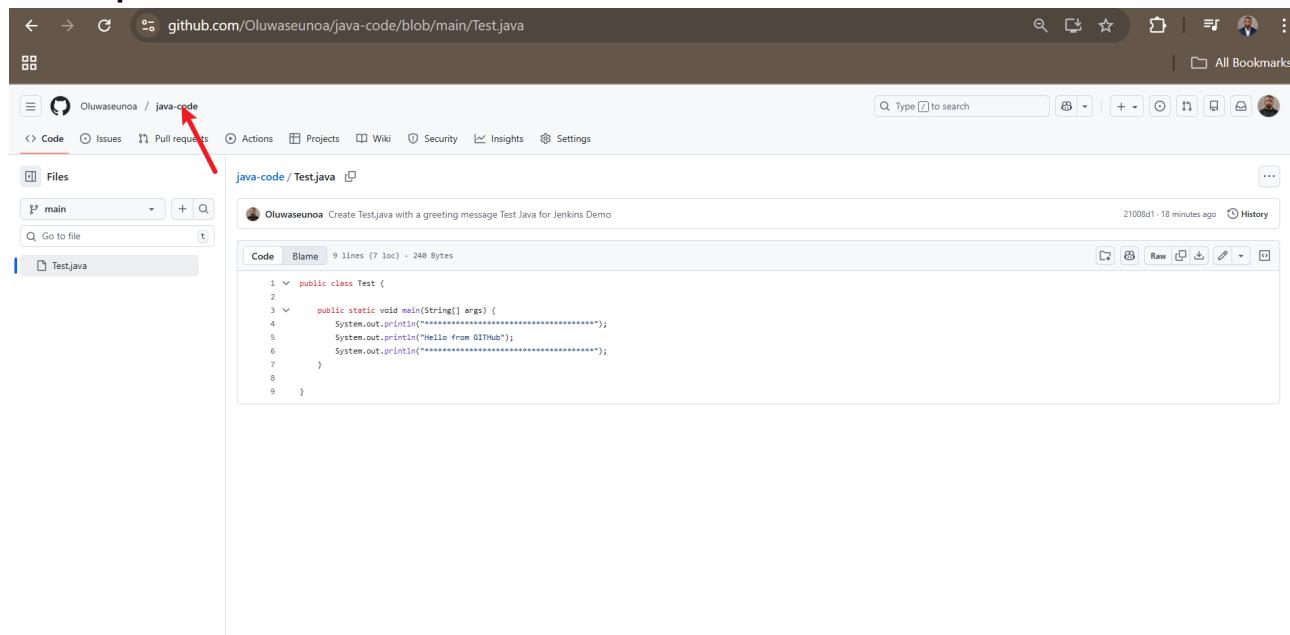
42. Name and select Freestyle

The screenshot shows the Jenkins 'New Item' configuration page. In the 'Enter an item name' field, 'Java-Project-Git' is typed. Under 'Select an item type', the 'Freestyle project' option is selected and highlighted with a red box. A red arrow points from the bottom left towards the 'OK' button at the bottom of the list. The 'OK' button is blue with white text.

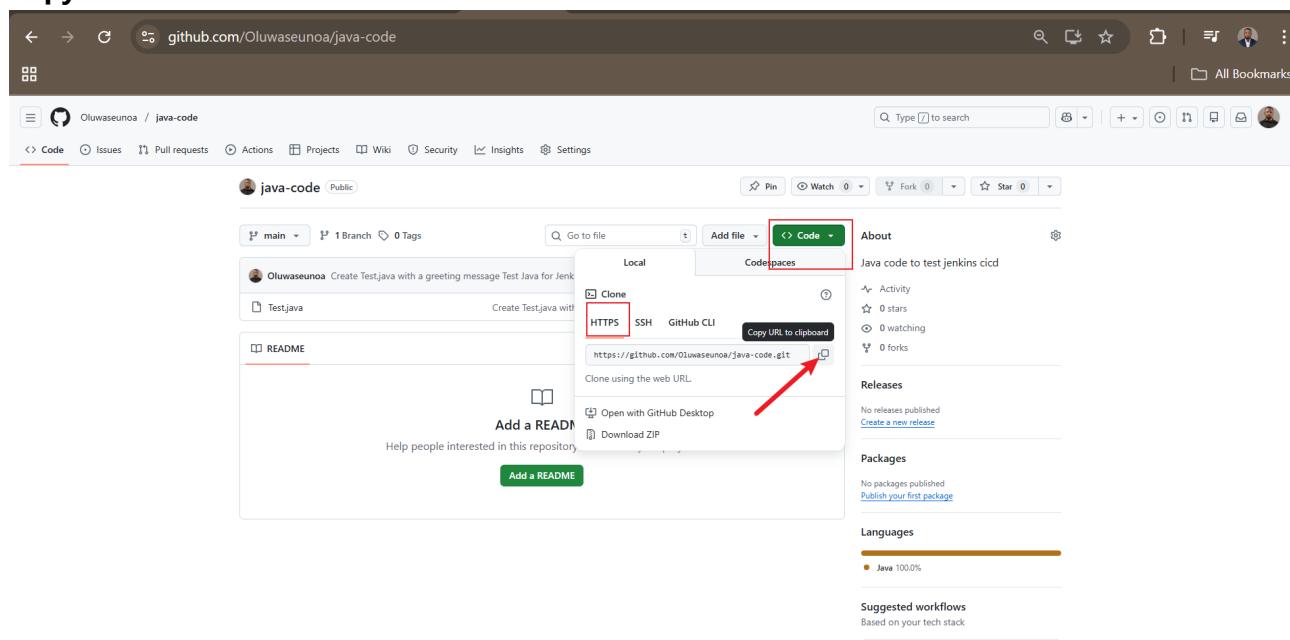
43. Source Code Management → Git

The screenshot shows the Jenkins 'Configure' page for the 'Java-Project-Git' job. The 'General' tab is selected in the sidebar. In the 'Source Code Management' section, the 'Git' tab is highlighted with a red circle. Below it, there is a 'Repositories' section with a 'Repository URL' field containing a placeholder message: 'Please enter Git repository.' A red arrow points from the bottom left towards the 'Git' tab. At the bottom of the page are 'Save' and 'Apply' buttons.

44. Go to repo



45. Copy HTTPS URL



46. Paste URL in Jenkins

The screenshot shows the Jenkins configuration interface for a job named "Java-Project-Git". The "Source Code Management" section is active. Under "Git", the "Repository URL" field contains the value "https://github.com/Oluwaseunoa/java-code.git", which is highlighted with a red box. Below it, the "Branch Specifier (blank for 'any')" field contains "*main", also highlighted with a red box. At the bottom, there are "Save" and "Apply" buttons.

47. Specify branch: main

The screenshot shows the Jenkins configuration interface for the same job. The "Branches to build" section is active. It includes a "Branch Specifier" field containing "*main", which is highlighted with a red box. There are "Advanced" and "+ Add Repository" buttons above the branches section. At the bottom, there are "Save" and "Apply" buttons.

48. Add Execute shell build step

The screenshot shows the Jenkins configuration interface for a job named "Java-Project-Git". The "Build Steps" section is highlighted with a red box. A modal window titled "Add build step" is open, listing various build steps. The "Execute shell" option is selected and highlighted with a red box. A red arrow points from the "Execute shell" option to the "Save" button at the bottom of the configuration page.

49. Enter command: `javac Test.java`

The screenshot shows the Jenkins configuration interface for the same job. The "Execute shell" step has been configured with the command "javac Test.java" in the "Command" field. A red arrow points from the "Save" button at the bottom of the configuration page to the "Save" button inside the "Execute shell" step configuration panel.

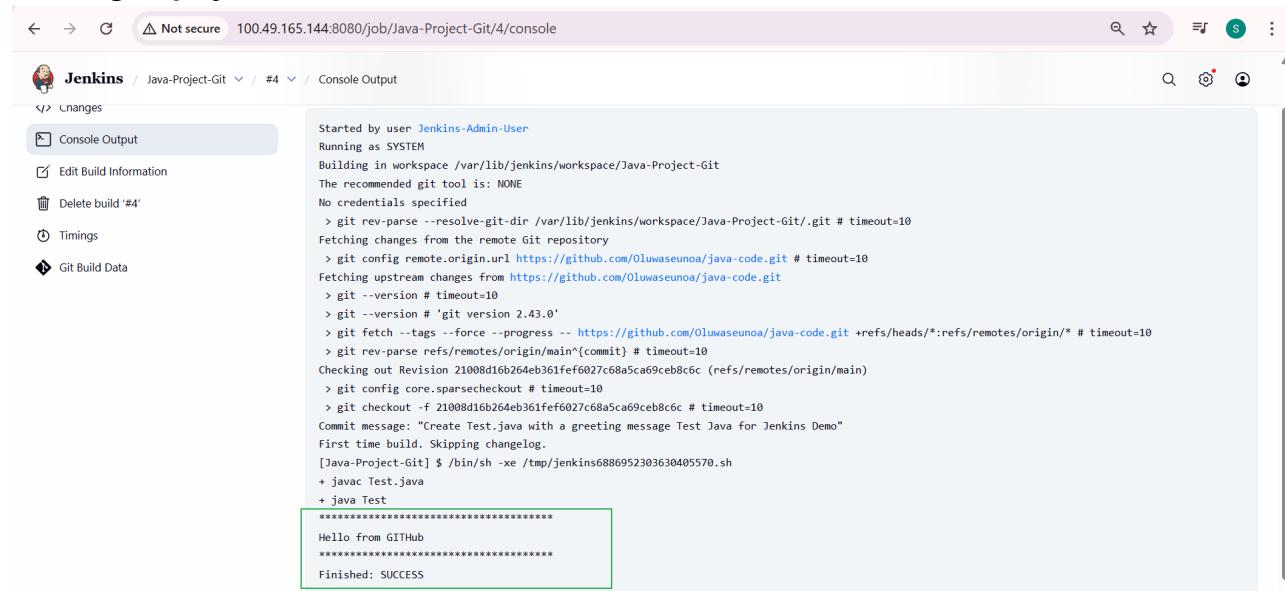
50. Build Now

The screenshot shows the Jenkins Java-Project-Git project page. On the left, there's a sidebar with options: Status, Changes, Workspace, Build Now (which has a red arrow pointing to it), Configure, Delete Project, and Rename. The main area is titled "Java-Project-Git" and "Permalinks". Below that is a "Builds" section with a message "No builds". At the bottom right, it says "REST API Jenkins 2.528.3".

51. Console output

This screenshot is similar to the previous one, showing the Jenkins Java-Project-Git project page. The sidebar includes "Build Now" (with a red arrow pointing to it) and "Console Output" (also highlighted with a red arrow in a context menu). The "Builds" section shows a single build entry: "Today #3 3:42 PM". The bottom right corner indicates "REST API Jenkins 2.528.3".

52. Greeting displayed



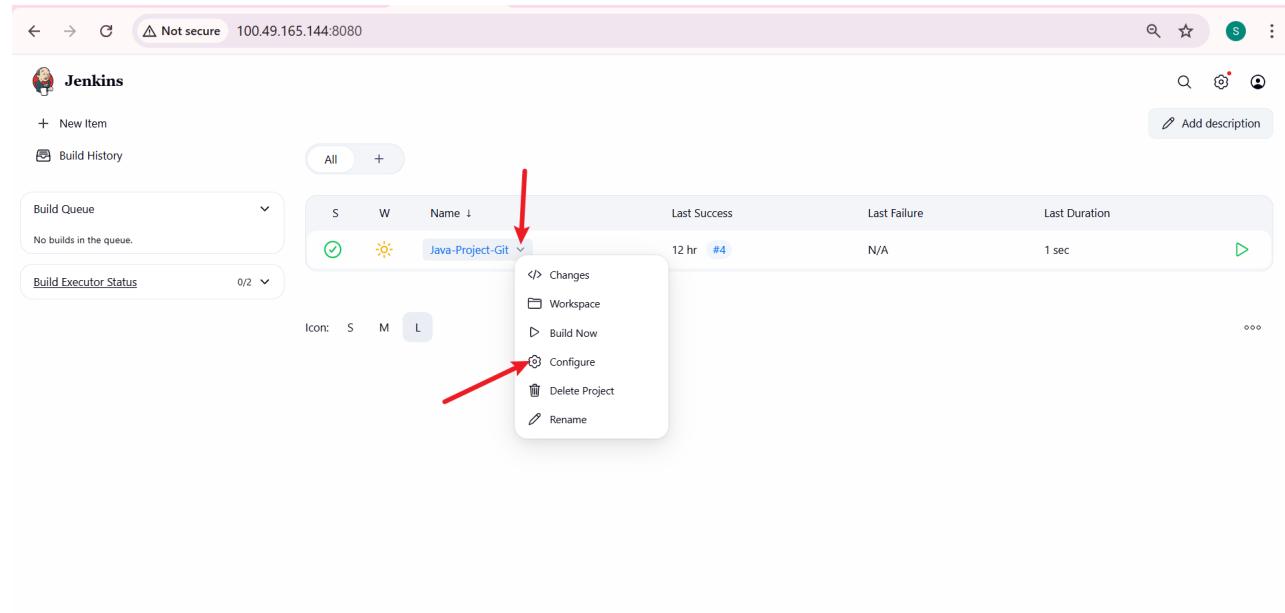
The screenshot shows the Jenkins console output for a build named "Java-Project-Git". The output shows the build process, including cloning from GitHub, compiling with javac, and running the resulting Java application. A red box highlights the final output:

```

Started by user Jenkins-Admin-User
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Java-Project-Git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Java-Project-Git/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Oluwaseunoa/java-code.git # timeout=10
Fetching upstream changes from https://github.com/Oluwaseunoa/java-code.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch -t --force --progress -- https://github.com/Oluwaseunoa/java-code.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 21008d16b264eb361fef6027c68a5ca69ceb8c6c (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 21008d16b264eb361fef6027c68a5ca69ceb8c6c # timeout=10
Commit message: "Create Test.java with a greeting message Test Java for Jenkins Demo"
First time build. Skipping changelog.
[Java-Project-Git] $ /bin/sh -xe /tmp/jenkins6886952303630405570.sh
+ javac Test.java
+ java Test
*****
Hello from GitHub
*****
Finished: SUCCESS

```

53. Configure job again



The screenshot shows the Jenkins job list. A context menu is open over the "Java-Project-Git" job entry. Red arrows point to the "Configure" option in the menu, which is highlighted.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Java-Project-Git	12 hr #4	N/A	1 sec

Icon: S M L

- <> Changes
- Workspace
- Build Now
- Configure**
- Delete Project
- Rename

54. *Enable Poll SCM (H/5 * * *)

The screenshot shows the Jenkins 'Configuration' page for a project named 'Java-Project-Git'. The left sidebar has 'Triggers' selected. In the main area, under 'Triggers', 'Poll SCM' is checked and set to run 'H/5 * * *'. A red box highlights the 'Poll SCM' checkbox and the schedule input field. A red arrow points from the 'Save' button at the bottom to the 'Poll SCM' section.

55. Push change to GitHub

The screenshot shows a VS Code interface with multiple tabs open. The 'Test.java' tab is active, displaying Java code. A red box highlights the line 'System.out.println("Hello from Jenkins CI Pipeline");'. The terminal tab shows the command 'git push' being run, with output indicating a new line was added to 'Test.java'. The Explorer sidebar shows a file structure with 'J Testjava' selected.

```

public class Test {
    public static void main(String[] args) {
        System.out.println("*****");
        System.out.println("Hello from GitHub");
        System.out.println("Hello from Jenkins CI Pipeline");
        System.out.println("*****");
    }
}

[main e68afec] Added a new line to Test.java
1 file changed, 1 insertion(+)

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main) ...
To https://github.com/Oluwaseunosa/java-code.git
  21008d1..e68afec  main -> main

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main)
$ 

```

56. Change visible on GitHub

The screenshot shows a GitHub repository named 'java-code'. A commit has been made to the file 'Test.java'. The commit message is 'Added a new line to Test.java' and was made 2 minutes ago. The code editor displays the Java code for the 'Test' class. A specific line, 'System.out.println("Hello from Jenkins CI Pipeline");', is highlighted with a red box.

```

1 public class Test {
2
3     public static void main(String[] args) {
4         System.out.println("*****");
5         System.out.println("Hello from GitHub");
6         System.out.println("Hello from Jenkins CI Pipeline");
7         System.out.println("*****");
8     }
9
10 }

```

57. Trigger build

The screenshot shows the Jenkins interface for the 'Java-Project-Git' project. In the left sidebar, there is a 'Build Now' button, which is highlighted with a red arrow. The main area shows the build history, with the last four builds listed: #5 (5:15 AM), #4 (3:52 PM), #3 (12:45 PM), and #2 (12:45 PM). The Jenkins version is 2.528.3.

58. New line in console

The screenshot shows the Jenkins Console Output page for a build named "Java-Project-Git #6". The log output shows a series of git commands being run, followed by a block of text from a Jenkins pipeline script:

```

Started by user Jenkins-Admin-User
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Java-Project-Git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Java-Project-Git/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Oluwaseunoa/java-code.git # timeout=10
Fetching upstream changes from https://github.com/Oluwaseunoa/java-code.git
> git --version # timeout=10
> git fetch -t --tags --force --progress -- https://github.com/Oluwaseunoa/java-code.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision e68afec3ee867747ff58f29e9768ae380fbe99 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f e68afec3ee867747ff58f29e9768ae380fbe99 # timeout=10
Commit message: "Added a new line to Test.java"
> git rev-list --no-walk e68afec3ee867747ff58f29e9768ae380fbe99 # timeout=10
[Java-Project-Git] $ /bin/sh -xe /tmp/jenkins11238369311256570208.sh
+ javac Test.java
+ java Test
*****
Hello from GitHub
Hello from Jenkins CI Pipeline
*****
Finished: SUCCESS

```

59. Back to dashboard

The screenshot shows the Jenkins Console Output page for the same build. A red arrow points to the Jenkins logo in the top left corner of the header.

7. Pipeline Job with Jenkinsfile & GitHub Webhook

60. Create Pipeline job

New Item

Enter an item name
Java-Pipeline

Select an item type

Pipeline Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job 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.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

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

OK

60b. Enable GitHub hook trigger

Configure

General

Triggers

Pipeline

Advanced

Do not allow the pipeline to resume if the controller restarts

GitHub project

Pipeline speed/durability override

Preserve stashes from completed builds

This project is parameterized

Throttle builds

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

Build after other projects are built

Build periodically

GitHub hook trigger for GITScm polling (checked)

Poll SCM

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

Save Apply

61. Pipeline from SCM

The screenshot shows the Jenkins configuration interface for a job named "Java-Pipeline". The left sidebar has sections: General, Triggers, Pipeline (which is highlighted with a red box), and Advanced. The main area is titled "Configure" and "Definition". A dropdown menu under "Definition" shows "Pipeline script from SCM" selected (also highlighted with a red box). Below it, "Pipeline script" is also listed. A sub-section titled "Repositories" contains a "Repository URL" field with the placeholder "https://github.com/OluwaseunOja/java-code.git" and a note "Please enter Git repository.". A "Credentials" dropdown shows "- none -". Buttons at the bottom are "Save" and "Apply".

62. Git SCM config

The screenshot shows the Jenkins configuration interface for the same job "Java-Pipeline". The left sidebar has sections: General, Triggers, Pipeline (highlighted with a red box), and Advanced. The main area is titled "Configure" and "SCM". A dropdown menu under "SCM" shows "Git" selected (highlighted with a red box). Below it, "Repositories" is listed. A "Repository URL" field contains "https://github.com/OluwaseunOja/java-code.git". A "Branches to build" section shows a "Branch Specifier (blank for 'any') ?" field containing "/main" (highlighted with a red box). Buttons at the bottom are "Save" and "Apply".

63. Script path: Jenkinsfile

The screenshot shows the Jenkins Pipeline configuration page for a job named "Java-Pipeline". The "Advanced" tab is selected. A red box highlights the "Script Path" input field, which contains the value "Jenkinsfile". Below it is a checked checkbox for "Lightweight checkout". At the bottom right are "Save" and "Apply" buttons, with a red arrow pointing to the "Save" button.

63b. Configuration saved

The screenshot shows the Jenkins pipeline status page for the "Java-Pipeline" job. The left sidebar includes options like Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main area displays the "Permalinks" section with links to the last five builds. The "Builds" section shows three successful builds from today: #4 at 6:36 AM, #3 at 6:27 AM, and #2 at 5:54 AM. A red arrow points to the successful build link for #4.

64. Create Jenkinsfile

```
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main)
$ touch Jenkinsfile
```

65. Add pipeline script

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Clone Repository') {
6       steps {
7         git branch: 'main',
8         url: 'https://github.com/Oluwaseunoa/java-code.git'
9       }
10    }
11
12    stage('Compile Java Code') {
13      steps {
14        sh 'javac Test.java'
15      }
16    }
17
18    stage('Run Application') {
19      steps {
20        sh 'java Test'
21      }
22    }
23  }
24}
```

GitHub Repo link

66. Push Jenkinsfile

The screenshot shows the Jenkinsfile in the code editor and its execution in the terminal. The Jenkinsfile defines a pipeline with two stages: 'Clone Repository' and 'Compile Java Code'. The terminal output shows the commit and push process to GitHub.

```

1 pipeline {
2     agent any
3
4     stages {
5         stage('Clone Repository') {
6             steps {
7                 git branch: 'main',
8                 url: 'https://github.com/Oluwaseunoa/java-code.git'
9             }
10        }
11
12        stage('Compile Java Code') {
13            steps {
14                sh 'javac Test.java'
15            }
16        }
17    }
18 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

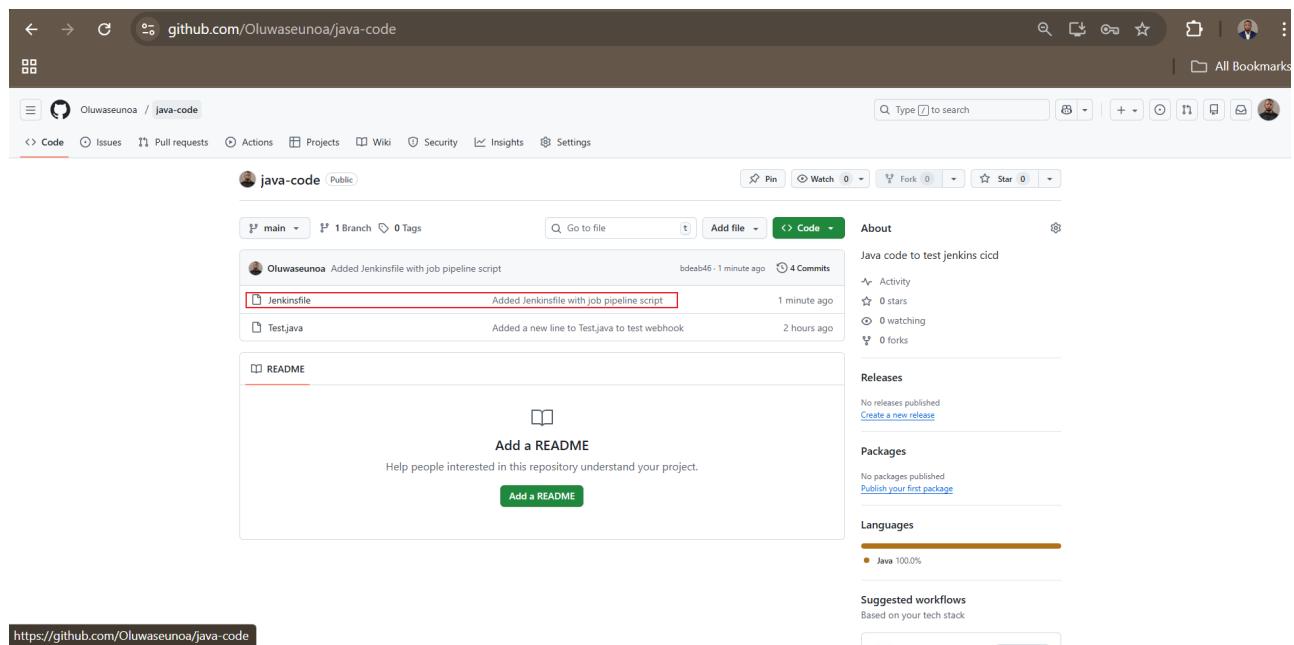
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main)
$ git commit -m "Added Jenkinsfile with job pipeline script"
[main bdeab46] Added Jenkinsfile with job pipeline script
1 file changed, 24 insertions(+)
create mode 100644 Jenkinsfile

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.

```

Ln 18, Col 35 (91 selected) Spaces: 4 UTF-8 CRLF {} Groovy Prettier

67. Jenkinsfile on GitHub



68. GitHub Settings

A screenshot of a GitHub repository page for 'java-code'. The top navigation bar shows the repository name 'Oluwaseunoa / java-code'. The 'Settings' tab is highlighted with a red arrow. The main content area displays basic repository statistics: 1 branch, 0 tags, 3 commits, and 1 hour ago. There is a section to 'Add a README' with a button to 'Add a README'. To the right, there are sections for 'About', 'Releases', 'Packages', and 'Languages' (Java 100.0%).

69. Webhooks section

A screenshot of the GitHub repository settings page for 'java-code'. The 'General' tab is selected. On the left, a sidebar menu has 'Webhooks' highlighted with a red arrow. The main content area includes sections for 'General' (repository name 'java-code', 'Template repository', 'Require contributors to sign off on web-based commits'), 'Default branch' (set to 'main'), and 'Releases' (with an 'Enable release immutability' checkbox). A footer link points to 'docs.github.com/repositories/.../creating-a-repository-from-a-template'.

70. Add webhook

The screenshot shows the GitHub settings interface for a repository named 'Oluwaseunoa/java-code'. The left sidebar has 'Webhooks' selected under the 'Code and automation' section. The main area is titled 'Webhooks' with a sub-instruction: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#)'. A red arrow points to the 'Add webhook' button at the top right of this section.

71. Copy Jenkins webhook URL

The screenshot shows the Jenkins Pipeline configuration page for a job named 'Java-Pipeline'. The 'Advanced' tab is selected in the left sidebar. In the main area, there is a 'Repository browser' dropdown set to '(Auto)', an 'Additional Behaviours' section with a '+ Add' button, and a 'Script Path' field containing 'Jenkinsfile'. A checkbox for 'Lightweight checkout' is checked. At the bottom, there are 'Save' and 'Apply' buttons. The page footer indicates 'REST API' and 'Jenkins 2.528.3'.

72. Configure webhook

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://100.49.165.144:8080/github-webhook/

Content type
application/json

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

© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

73. Webhook added

github.com/Oluwaseunoa/java-code/settings/hooks

Oluwaseunoa / java-code

Code Issues 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 **Webhooks** **Add webhook**

Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions Models

Webhooks Copilot Environments Codespaces Pages

Copilot Environments Codespaces Pages

Advanced Security Deploy keys Secrets and variables

Integrations GitHub Apps Email notifications

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a `POST` request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ http://100.49.165.144:8080/github-webhook... (push)
Last delivery was successful.

Edit **Delete**

74. Push new change

```

1  public class Test {
2
3      public static void main(String[] args) {
4          System.out.println("*****");
5          System.out.println("Hello from GitHub");
6          System.out.println("Hello from Jenkins CI Pipeline");
7          System.out.println("Hello, will the webhook work?");
8          System.out.println("If you see this, the webhook worked and probably the pipeline will be triggered!");
9          System.out.println("*****");
10     }
11
12
13

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[main 27fef02] Added this line to see if pipeline will be triggered!
1 file changed, 2 insertions(+), 1 deletion(-)

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main) ...
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Oluwaseunoa/java-code.git
 bdeab46..27fef02 main -> main

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Jenkins-Projects/Jenkins CI Pipeline for Java Application with GitHub Integration/java-code-v2/java-code (main)

75. Automatic build triggered

← → ⚡ Not secure 100.49.165.144:8080/job/Java-Pipeline/

Jenkins / Java-Pipeline

Status: **Java-Pipeline** (green checkmark)

Changes: Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, GitHub Hook Log.

Permalinks:

- Last build (#5), 1 min 54 sec ago
- Last stable build (#5), 1 min 54 sec ago
- Last successful build (#5), 1 min 54 sec ago
- Last completed build (#5), 1 min 54 sec ago

Builds:

Build #	Timestamp
#5	9:33AM
#4	6:36AM
#3	6:27AM
#2	5:54AM

Today: #5 9:33AM, #4 6:36AM, #3 6:27AM, #2 5:54AM

REST API Jenkins 2.528.3

76. View console

The screenshot shows the Jenkins Java-Pipeline job page. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. Below that is a list of builds with filters. A green arrow points to the build #5 entry in the list. Another green arrow points to the 'Console Output' option in the context menu that appears when hovering over the build number. The main content area shows the pipeline stages: stage, Compile Java Code, sh, javac Test.java, stage, Run Application, sh, java Test. A message box highlights the output from the java Test stage, which includes a webhook trigger message.

77. New line in output

This screenshot shows the Jenkins Java-Pipeline job page with build #5 selected. A green arrow points to the 'Console Output' link in the build history. The expanded console output window shows the pipeline stages and the expanded java Test stage output. The output from the java Test stage is highlighted with a green box, showing a message about a successful webhook trigger.

8. Docker Integration

78. SSH & update server

```
ubuntu@ip-172-31-68-167:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-inRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
*****Hello from GitHub
Hello from Jenkins CI Pipeline
Hello, will the webhook work?
If you see this, the webhook worked and probably the pipeline will be triggered!
*****
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

79. Install required packages

```
ubuntu@ip-172-31-68-167:~$ sudo apt install ca-certificates curl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-68-167:~$
```

80. Create GPG key directory

```
ubuntu@ip-172-31-68-167:~$ sudo install -m 0755 -d /etc/apt/keyrings
ubuntu@ip-172-31-68-167:~$
```

81. Download Docker GPG key

```
ubuntu@ip-172-31-68-167:~$ curl -fsS https://download.docker.com/linux/ubuntu/gpg | \
sudo tee /etc/apt/keyrings/docker.gpg > /dev/null
ubuntu@ip-172-31-68-167:~$
```

82. Add Docker repository

```
ubuntu@ip-172-31-68-167:~$ echo "deb [signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$ (lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list
deb [signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu noble stable
ubuntu@ip-172-31-68-167:~$
```

83. Update package list

```
ubuntu@ip-172-31-68-167:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [49.5 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 http://download.docker.com/linux/ubuntu/noble/stable amd64 Packages [41.1 kB]
Fetched 89.6 kB in 1s (126 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-68-167:~$
```

84. Install Docker

```
ubuntu@ip-172-31-68-167:~$ sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Suggested packages:
  cgroups-mount | cgroup-lite docker-model-plugin
The following NEWER packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 91.3 MB of archives.
After this operation, 364 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 slirp4netns amd64 1.2.1-1+ndk12 [34.4 kB]
Get:4 http://download.docker.com/linux/ubuntu/noble/stable amd64 containerd.io amd64 2.2.1-1ubuntu2.24.04-noble [23.4 MB]
Get:5 http://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce amd64 5:29.1.3-1ubuntu24.04-noble [16.3 MB]
Get:6 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-buildx-plugin amd64 0.30.1-1ubuntu24.04-noble [21.0 MB]
Get:7 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-rootless-extras amd64 5:29.1.3-1ubuntu24.04-noble [16.4 MB]
Get:8 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-compose-plugin amd64 5:29.1.3-1ubuntu24.04-noble [6383 kB]
Get:9 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-compose-plugin amd64 5.0.0-1ubuntu24.04-noble [7709 kB]
Reading 1 file in archive...done
Selecting previously unselected package containerd.io.
(Reading database... 120289 files and directories currently installed.)
Preparing to unpack .../0-containerd.io_2.2.1-1ubuntu24.04-noble_amd64.deb ...
Unpacking containerd.io (2.2.1-1ubuntu24.04-noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../1-docker-ce_2.1.1-1ubuntu24.04-noble_amd64.deb ...
Unpacking docker-ce (2.1.1-1ubuntu24.04-noble) ...
Selecting previously unselected package docker-ce-cll.
Preparing to unpack .../2-docker-ce_5%2a29.1.3-1ubuntu24.04-noble_amd64.deb ...
Unpacking docker-ce (5:29.1.3-1ubuntu24.04-noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1ubuntu24.04-noble_amd64.deb ...
```

85. Add jenkins user to docker group

```
ubuntu@ip-172-31-68-167:~$ sudo usermod -aG docker jenkins
ubuntu@ip-172-31-68-167:~$
```

86. Add ubuntu user to docker group

```
ubuntu@ip-172-31-68-167:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-68-167:~$
```

87. Restart Jenkins

```
ubuntu@ip-172-31-68-167:~$ sudo systemctl restart Jenkins
[
```

88. Test Docker

```
ubuntu@ip-172-31-68-167:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17e9e7b93d91: Pull complete
Digest: sha256:d44aab6242e0cace87e2ec17a2ed3d779d18fbfd03042ea58f2995626396a274
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

ubuntu@ip-172-31-68-167:~$
```

89. Create Dockerfile

The screenshot shows a code editor interface with several tabs at the top: 'Docker Containers', 'Test.java', 'Jenkinsfile', 'README.md ...\\Getting Started with Docker with NodeJS', and 'README.md ...\\Getting Started'. The 'Jenkinsfile' tab is active, displaying the following Jenkins Pipeline script:

```
pipeline {
    agent any

    stages {
        stage('Clone Repository') {
            steps {
                git branch: 'main',
                url: 'https://github.com/Oluwaseunoa/java-code.git'
            }
        }

        stage('Compile Java Code') {
            steps {
                sh 'javac Test.java'
            }
        }
    }
}
```

Below the code editor is a terminal window titled 'bash - java-code'. It shows the following command being run:

```
$ touch Dockerfile
```

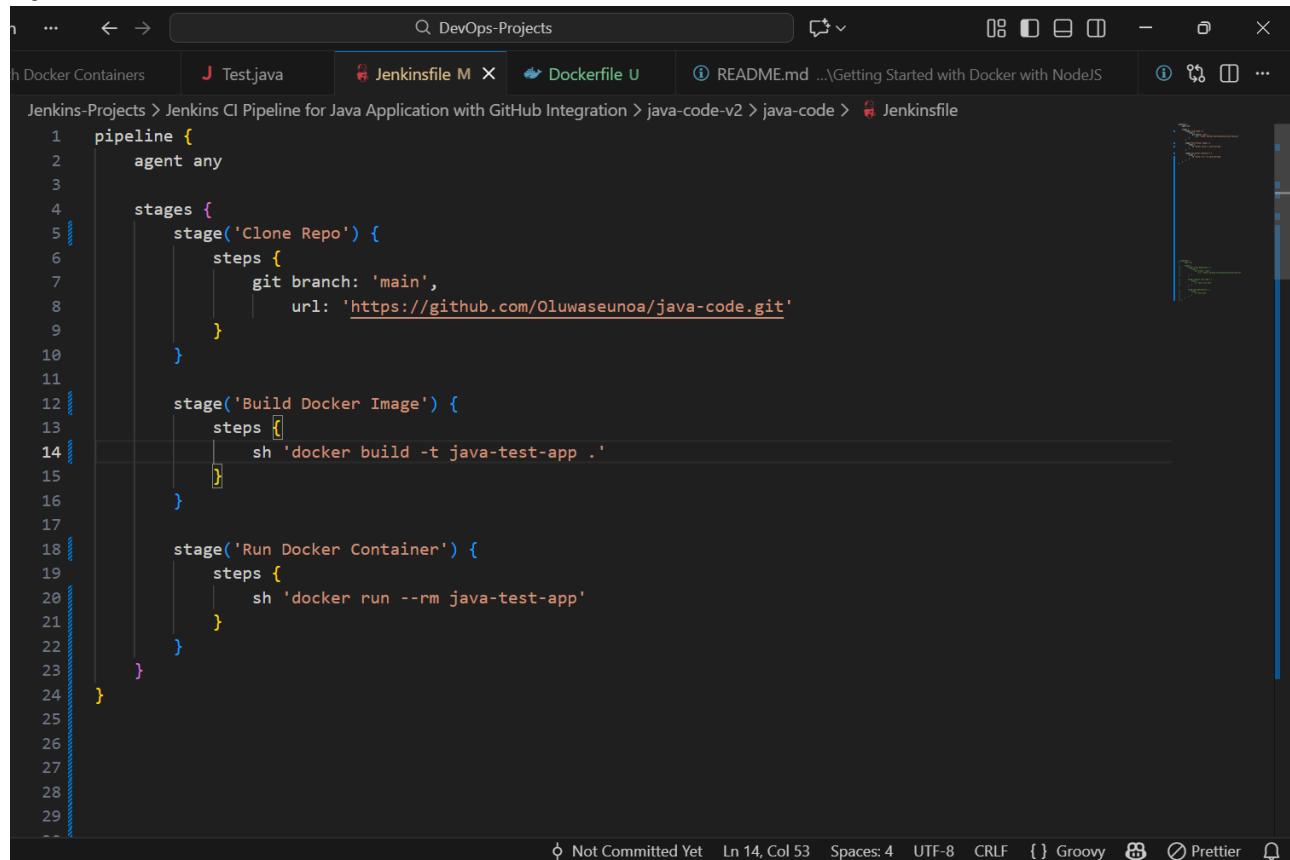
The terminal window has a red box around the command. At the bottom of the terminal window, there is some Jenkins pipeline log output.

90. Add Dockerfile content

The screenshot shows a code editor interface with several tabs at the top: 'Docker Containers', 'Test.java', 'Jenkinsfile', 'Dockerfile', 'README.md ...\\Getting Started with Docker with NodeJS', and 'README ...'. The 'Dockerfile' tab is active, displaying the following Dockerfile content:

```
FROM eclipse-temurin:21-jdk
WORKDIR /app
COPY Test.java .
RUN javac Test.java
CMD ["java", "Test"]
```

91. Update Jenkinsfile for Docker



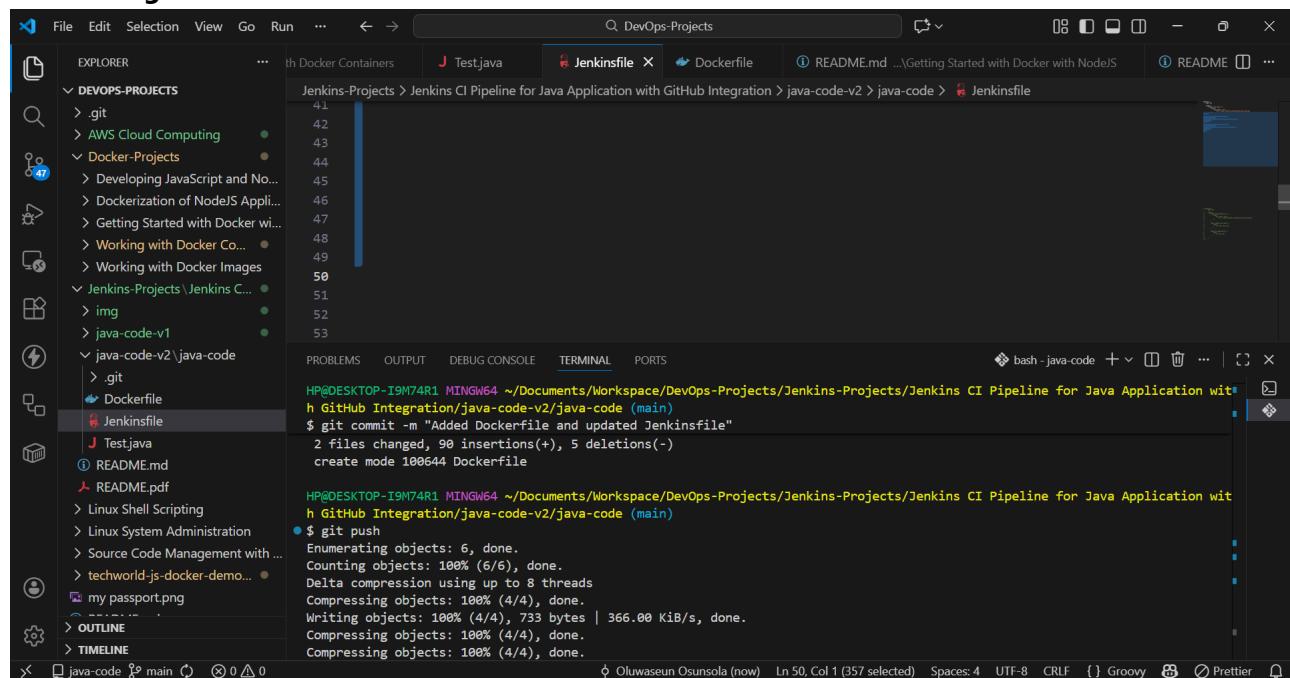
```

1 pipeline {
2     agent any
3
4     stages {
5         stage('Clone Repo') {
6             steps {
7                 git branch: 'main',
8                 url: 'https://github.com/Oluwaseunoa/java-code.git'
9             }
10        }
11
12        stage('Build Docker Image') {
13            steps [
14                sh 'docker build -t java-test-app .'
15            ]
16        }
17
18        stage('Run Docker Container') {
19            steps [
20                sh 'docker run --rm java-test-app'
21            ]
22        }
23    }
24
25
26
27
28
29

```

The screenshot shows a code editor with a dark theme. The tab bar at the top includes 'DevOps-Projects', 'Jenkinsfile M X', 'Dockerfile U', 'README.md ...', and 'Jenkinsfile'. The main area displays a Jenkins pipeline script. The script defines a pipeline with three stages: 'Clone Repo', 'Build Docker Image', and 'Run Docker Container'. Each stage contains a single step involving Docker commands. The code editor interface includes a status bar at the bottom with file information like 'Not Committed Yet' and line numbers.

92. Push changes



The screenshot shows a code editor with a dark theme, similar to the previous one. The tab bar at the top includes 'DevOps-Projects', 'Jenkinsfile X', 'Dockerfile', 'README.md ...', and 'Jenkinsfile'. The left sidebar shows a file tree under 'EXPLORER' with categories like 'DEVOPS-PROJECTS', 'Docker-Projects', and 'Jenkins-Projects'. The 'Jenkinsfile' is selected in the tree. The main area shows the Jenkinsfile content with line numbers 41 through 53. Below the code is a terminal window showing the command-line output of a 'git commit' and 'git push' operation. The terminal shows the user is on a Windows machine ('MINGW64') and is committing changes to a repository named 'java-code-v2'. The commit message is 'Added Dockerfile and updated Jenkinsfile'. The push command is successful, showing object enumeration, counting, compression, and writing statistics. The status bar at the bottom indicates the file is 'main' and has 0 changes.

93. Build triggered

The screenshot shows the Jenkins Java-Pipeline job page. On the left, there is a sidebar with various options: Status, Changes, Build Now (highlighted with a red arrow), Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main area is titled "Java-Pipeline" and shows "Permalinks". Below that is a "Builds" section with a message "No builds". At the top, the URL is 100.49.165.144:8080/job/Java-Pipeline/ and there are standard browser navigation and search tools.

94. Build successful

The screenshot shows the Jenkins Java-Pipeline job page after a build has been triggered. The sidebar remains the same. A context menu is open over the build entry "#3 10:43 AM" in the "Builds" list. The menu items include: Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, GitHub Hook Log, Console Output (highlighted with a green arrow), Edit Build Information, Delete build #3, Timings, Git Build Data, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The "Builds" list shows the successful build entry "#3 10:43 AM". At the bottom right, it says REST API Jenkins 2.528.3.

95. Greeting works in Docker

The screenshot shows a Jenkins console output for a Java Pipeline job. The output includes Docker commands like `docker run --rm java-test-app` and logs from the application inside the container. A green box highlights a specific message: "Hello from GitHub", "Hello from Jenkins CI Pipeline", "Hello, will the webhook work?", and "If you see this, the webhook worked and probably the pipeline will be triggered!". Below this, the pipeline ends with "Finished: SUCCESS".

```
#9 exporting manifest list sha256:fe5e6f309949aa4fa7aea9b77ae603ab485a286ae3b4ea74ca5434f7150fc4ae 0.0s done
#9 naming to docker.io/library/java-test-app:latest
#9 naming to docker.io/library/java-test-app:latest 0.0s done
#9 unpacking to docker.io/library/java-test-app:latest 0.0s done
#9 DONE 0.2s
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Docker Container)
[Pipeline] sh
+ docker run --rm java-test-app
*****
Hello from GitHub
Hello from Jenkins CI Pipeline
Hello, will the webhook work?
If you see this, the webhook worked and probably the pipeline will be triggered!
*****
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.528.3

Lessons Learned & Best Practices

- **Webhooks > Polling** — Real-time, efficient, less resource usage.
- **Pipeline-as-Code** — Version-controlled, auditable, reproducible.
- **Docker agents** — Ensure consistent environments, isolate dependencies.
- **Security** — Use Elastic IP with HTTPS (e.g., via Nginx reverse proxy), enable RBAC, use Jenkins credentials plugin.
- **Future Improvements** — Add Maven/Gradle for real projects, unit tests (JUnit), artifact archiving, deployment to AWS ECR/ECS, notifications (Slack/Email).

This project provides a strong foundation for scaling to production-grade CI/CD.