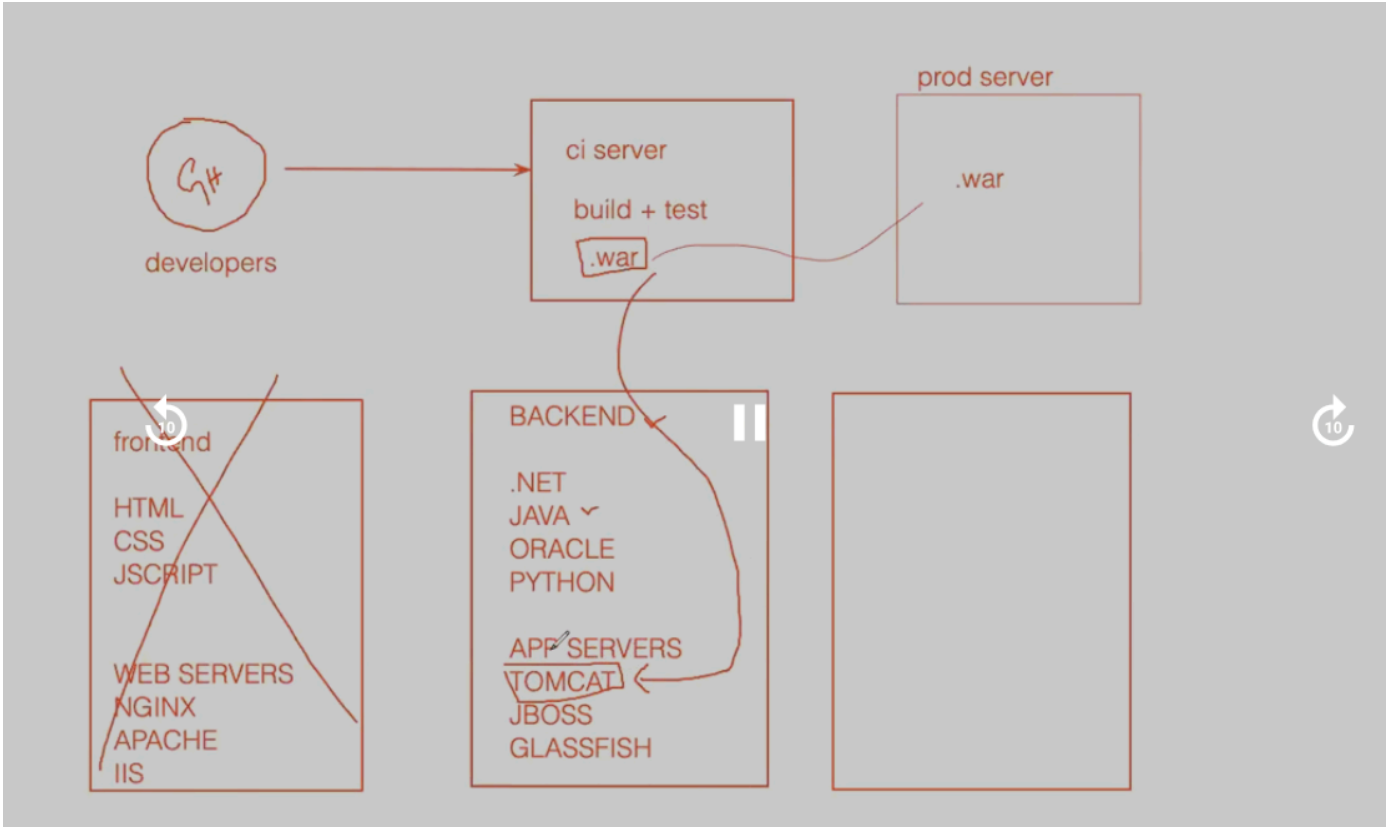
jenkins class 4 (26)

jenkins class 4 (26)

installing the jenkins through the script make excutable

developers put the code into github/bitbucket this code we have to bring ci server
ci server will build + test now it will generate war file when the code java this war file keep it in prod server



Developers push code to GitHub. The CI server builds and tests the code automatically. Frontend code (HTML, CSS, JavaScript) is deployed on web servers like NGINX or Apache, while backend code (Java, .NET, Python) is deployed on application servers like Tomcat or JBoss to handle business logic.

One Diagram in Words (Easy to Draw)

Developer
↓
GitHub
↓
CI Server (Build & Test)
↓
Frontend → Web Server (NGINX)

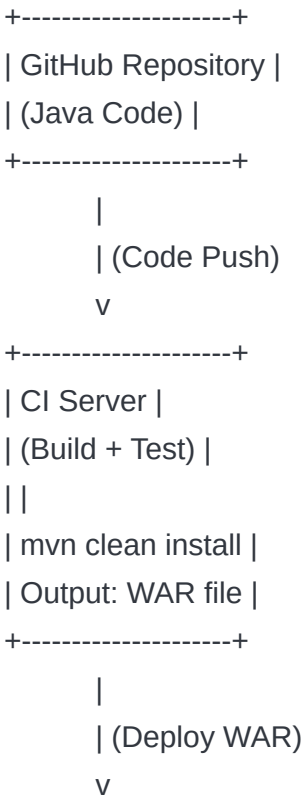
Backend → App Server (Tomcat)

🔥 Web Server vs Application Server – Difference Table

Feature	Web Server	Application Server
Purpose	Serves static content	Executes business logic
Handles	HTML, CSS, JavaScript, images	Java, .NET, Python, backend code
Type of Work	Display content	Process requests
Processing Power	Low	High
Business Logic	❌ No	✅ Yes
Database Access	❌ No	✅ Yes
Protocols Used	HTTP / HTTPS	HTTP, RMI, JMS, etc.
Runs On	OS level	Runtime (JVM, .NET CLR)
Examples	NGINX, Apache, IIS	Tomcat, JBoss, GlassFish
Performance	Very fast	Slower than web server
Use Case	Frontend hosting	Backend APIs, services

NestJS and Python frameworks act as application servers because they execute backend business logic, whereas web servers like NGINX are used to serve static content and route requests.

📌 Java CI/CD Deployment – Diagram Representation



+--------------------+
| Production Server |
| Tomcat Setup |
| |
| Deploy WAR file |
| Application Runs |
+--------------------+

In a CI/CD pipeline, Java code is pushed to GitHub, built and tested by a CI server to generate a WAR file, which is then deployed on a Tomcat application server in the production environment.

Install Jenkins and
create a job and here clone repo i.e one inside job because source code we have to bring to ci server.install git in jenkins server.
select master branch source code we have in master branch save and build build is successfull .
we can see in servers in this we have pom.xml and src

second stage2: for build we use maven
install maven depency it have java jdk 1.8.0

firstinstall java jdk then maven we donot get any issues
mvn -v

for maven we have alternatives that is ant and gradle

here go to build step
here we use maven select invoke top level maven targets
in goals--> give here clean package and build here no need to give mvn because it is alredy mvn

mvn clean package //here build will success then it will generate war file
if we get warfile means build happen and if we get surefire test happen

here both build and test done

now its time install tomcat to deploy war file

take one server
goto root user
tomcat have one depency i.e java so install java 11 or any in prod server

sudo apt update
sudo apt install openjdk-11-jdk -y
java -version

tomcat invited by apache2
so wehave website i.e dlcdn.apache.org
goto tomcat and goto tomcat9 then goto binary last it end with tar.gz file with 11 mb this file we haev to install in server

cd /opt
sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.85/bin/apache-tomcat-9.0.85.tar.gz
sudo tar -xvzf apache-tomcat-9.0.85.tar.gz
sudo mv apache-tomcat-9.0.85 tomcat9
sudo chmod -R 755 /opt/tomcat9
readlink -f $(which java)
cd /opt/tomcat9/bin
./startup.sh
ps -ef | grep tomcat

install_tomcat9.sh

```
#!/bin/bash

# ----------------------------
# Variables
# ----------------------------
TOMCAT_VERSION=9.0.85
TOMCAT_USER=tomcat
INSTALL_DIR=/opt
JAVA_HOME_PATH=/usr/lib/jvm/java-11-openjdk-amd64

# ----------------------------
# Step 1: Update system
# ----------------------------
sudo apt update -y

# ----------------------------
# Step 2: Install Java 11
# ----------------------------
sudo apt install openjdk-11-jdk -y

# ----------------------------
# Step 3: Set JAVA_HOME
# ----------------------------
```

```
export JAVA_HOME=$JAVA_HOME_PATH
echo "export JAVA_HOME=$JAVA_HOME_PATH" >> ~/.bashrc
source ~/.bashrc

# ----------------------------
# Step 4: Download Tomcat
# ----------------------------
cd $INSTALL_DIR
sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v\${TOMCAT_VERSION}/bin/apache-tomcat-\${TOMCAT_VERSION}.tar.gz

# ----------------------------
# Step 5: Extract Tomcat
# ----------------------------
sudo tar -xvzf apache-tomcat-${TOMCAT_VERSION}.tar.gz
sudo mv apache-tomcat-${TOMCAT_VERSION} tomcat9

# ----------------------------
# Step 6: Set permissions
# ----------------------------
sudo chmod -R 755 tomcat9

# ----------------------------
# Step 7: Start Tomcat
# ----------------------------
cd tomcat9/bin
./startup.sh

# ----------------------------
# Step 8: Verify
# ----------------------------
echo "Tomcat started successfully!"
echo "Access Tomcat at: http://<server-ip>:8080"
```

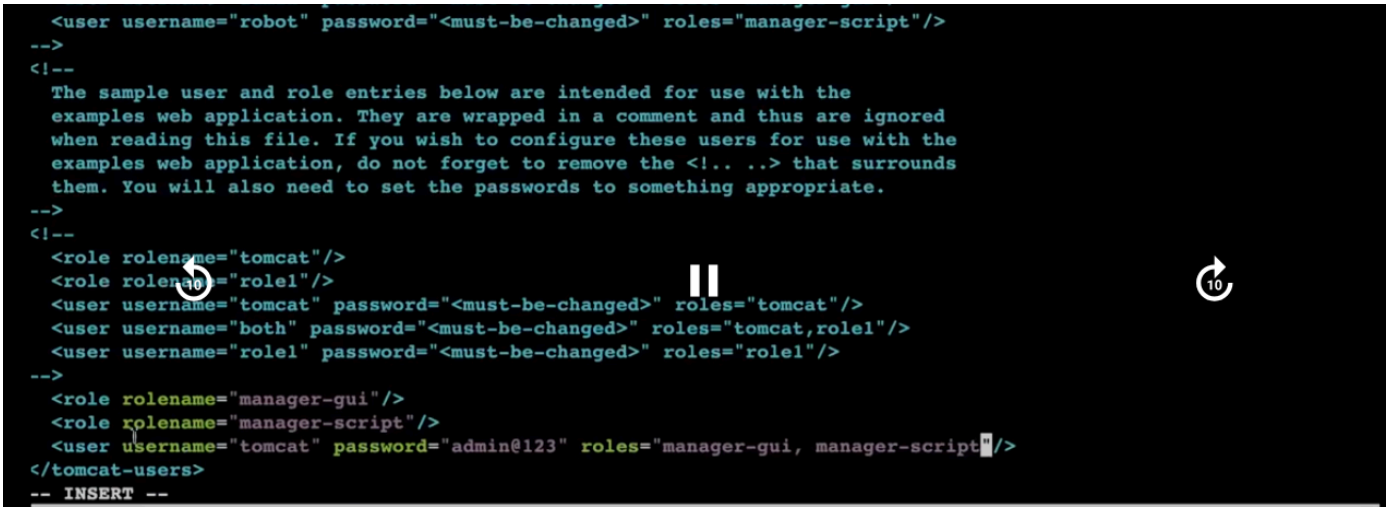
```
chmod +x install_tomcat9.sh

./install_tomcat9.sh
```

in the manager app we can see the all deployed apps
but if we go manager app it will ask permissions i.e access denied
so we ahve to edit context .xml file
the path is
/opt/tomcat9/webapps/manager/META-INF/context.xml

here goto 21 line delete2 lines now again access
now again it will ask username and password
for that we haev to go conf/
vim tomcat-users.xml
here

goto lastline



<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="admin@123" roles="manager-gui, manager-script"/>

then stop and start tomcat
then goto jenkins
install the plugin i.e deploy to container
then goto job configuration i.e sourcec code is one
goto configuration last we haev post build action here you can select deploy war to container
here select war file path i.e jenkins server
/target/*.war
context path any i.e swiggy
containers select tomcat9 --> add credentials here username tomcat ,password admin@123 description tomcat login details
tomcat url : ip:8080
save and deploy

it will deploy i.e build + test + deploy

and we can see the build wise console logs i.e
goto jenkins default workspace path i.e there build folder will create