

Automatic Web Application Deployment using Jenkins, Git, Maven, Docker & Kubernetes

1. Project Title

Automatic Deployment of Online Book Store Web Application using Jenkins CI/CD, Docker, and Tomcat

2. Project Description

This project implements a **complete CI/CD automation pipeline** for deploying a Java-based **Online Book Store web application**.

Whenever code is pushed to GitHub:

- Jenkins automatically builds the application using Maven
 - Generates a WAR file
 - Builds a Docker image with Tomcat
 - Pushes the image to Docker Hub
 - Deploys and runs the application inside a Docker container
 - The application becomes accessible through the browser
-

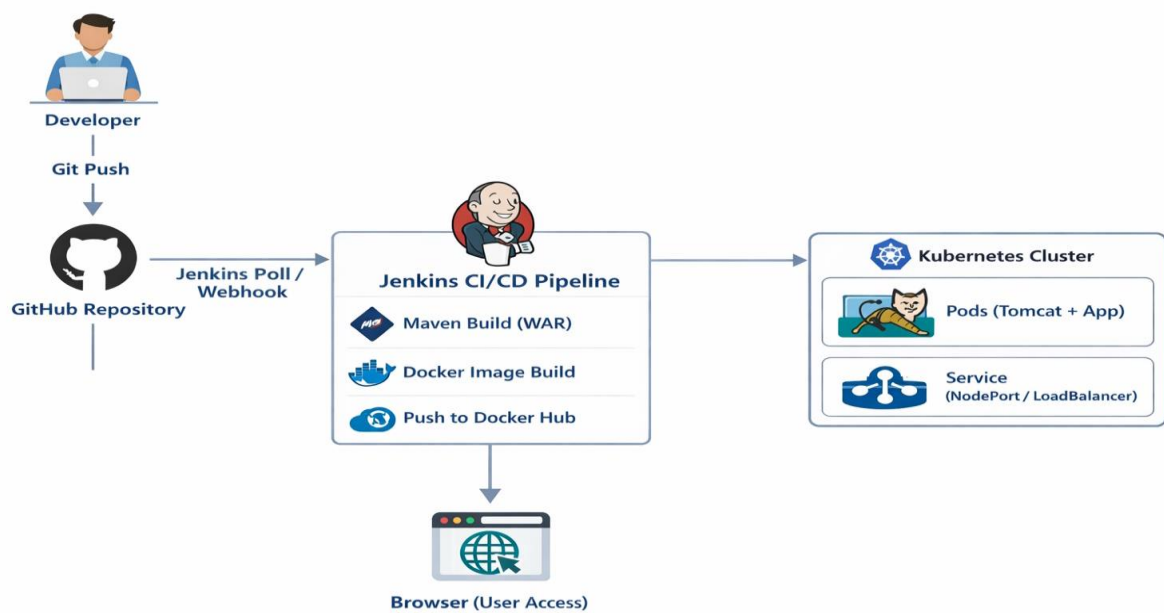
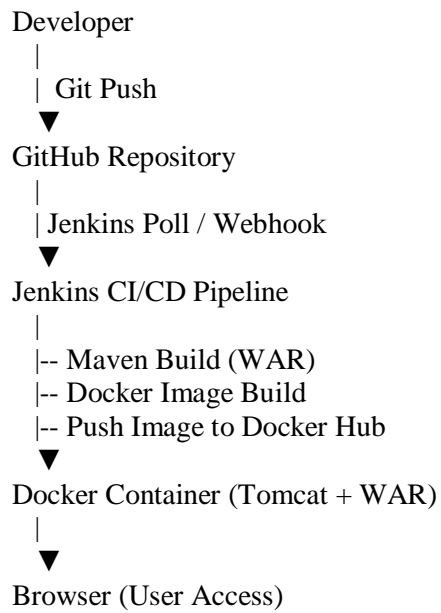
3. Objective

- Automate web application deployment
 - Eliminate manual WAR deployment
 - Learn real-time CI/CD workflow
 - Implement DevOps tools integration
 - Deploy Java application using containers
-

4. Tools & Technologies Used

Category	Tools
Version Control	Git, GitHub
CI/CD Tool	Jenkins
Build Tool	Maven
Web Server	Apache Tomcat
Containerization	Docker
Cloud	AWS EC2
OS	Amazon Linux
Scripting	Bash

5. System Architecture



6. Project Workflow

1. Developer pushes code to GitHub
2. Jenkins detects the change
3. Jenkins checks out source code
4. Maven builds the WAR file
5. WAR is copied to Docker build directory
6. Docker image is created with Tomcat
7. Image is pushed to Docker Hub
8. Docker container runs the application
9. Application is accessed through browser

7. Infrastructure Setup

AWS EC2 Instance

- OS: Amazon Linux
 - Security Group:
 - 22 (SSH)
 - 8081 (Application access)
 - 8080 (Tomcat if needed)
-

8. Jenkins Configuration

Installed Tools

- Git
- Maven
- Docker

Jenkins Workspace Path

/var/lib/jenkins/workspace/project-1/

9. Maven Build Output

WAR file generated at:

/var/lib/jenkins/workspace/project-1/target/onlinebookstore.war

10. Dockerfile (Used in Project)

```
FROM tomcat:latest
COPY onlinebookstore.war /usr/local/tomcat/webapps/
EXPOSE 8080
```

11. Bash Deployment Script (push.sh)

```
#!/bin/bash

cp /var/lib/jenkins/workspace/project-1/target/onlinebookstore.war /root

docker build -t project-image .

docker run -d -p 8081:8080 project-image

docker tag project-image kavyakb02/project-image
```

```
docker push kavyakb02/project-image
```

✓ This script:

- Copies WAR from Jenkins workspace
 - Builds Docker image
 - Runs container
 - Tags image
 - Pushes image to Docker Hub
-

12. Docker Image Details

- Image Name: project-image
 - Docker Hub Repo: kavyakb02/project-image
 - Base Image: tomcat:latest
-

13. Jenkins Job Execution Output (Verified)

- ✓ Maven Build – SUCCESS
- ✓ Docker Build – SUCCESS
- ✓ Docker Push – SUCCESS

Docker push output confirms:

Pushed to repository docker.io/kavyakb02/project-image

14. Application Deployment

Running Container

```
docker ps
```

Port Mapping:

8081 (Host) → 8080 (Container)

15. Application Access (Verified)

Application URL:

<http://<EC2-Public-IP>:8081/onlinebookstore/CustomerLogin.html>

- ✓ Online Book Store Login Page successfully loaded in browser.
-

16. Output Screenshot

- Customer Login Page
- Username, Password, Captcha
- Fully functional UI

```
~/m/'
Last login: Wed Dec 31 03:40:22 2025 from 13.233.177.4
[ec2-user@ip-172-31-35-20 ~]$ sudo su
[root@ip-172-31-35-20 ec2-user]# cd ~
[root@ip-172-31-35-20 ~]# ls
Dockerfile  push.sh
[root@ip-172-31-35-20 ~]# cat push.sh
#!/bin/bash

cp /var/lib/jenkins/workspace/project/target/onlinebookstore.war /root

docker build -t project-image .
docker run -d -p 8080:8081 project-image

docker tag project-image kavyakb02/project-image

docker push kavyakb02/project-image

[root@ip-172-31-35-20 ~]# vi push.sh
[root@ip-172-31-35-20 ~]# cat push.sh
#!/bin/bash

cp /var/lib/jenkins/workspace/project-1/target/onlinebookstore.war /root

docker build -t project-image .
docker run -d -p 8080:8081 project-image

docker tag project-image kavyakb02/project-image
```

```
[root@ip-172-31-35-20 ~]# vi push.sh
[root@ip-172-31-35-20 ~]# cat push.sh
#!/bin/bash

cp /var/lib/jenkins/workspace/project-1/target/onlinebookstore.war /root

docker build -t project-image .
docker run -d -p 8081:8080 project-image

docker tag project-image kavyakb02/project-image

docker push kavyakb02/project-image

[root@ip-172-31-35-20 ~]# bash push.sh
[+] Building 8.2s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 170B
=> [internal] load metadata for docker.io/library/tomcat:latest
=> [auth] library/tomcat:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
```

```
=> => sha256:c3f8d69bb599f073d75d39b20fb8acf6f8ea736c4b2b86afc7971bbd8b71fe97 2.72kB / 2.72kB
=> => sha256:627c55a201a9095fd9bb5b05a4c058246c463987d178f778ff22db143ca94635 17.46MB / 17.46MB
=> => sha256:378e3a6f165ea02b44863b980aacd0a6ba57f3b96b08da831cc39bb19eb41760 92.17MB / 92.17MB
=> => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B
=> => extracting sha256:20043066d3d5c78b45520c5707319835ac7d1f3d7f0dded0138ea0897d6a3188
=> => sha256:901b8cfcfda7112bdd66707d3d655b06bf97a440a496fe617c60040989857b40 2.28kB / 2.28kB
=> => sha256:45e2e3388eeb25221c8d602f3ec3ef959da49489322ebe1528c8c855eec199a2 140B / 140B
=> => sha256:841c6b2bc7edd02d64830c140424a05c0d666457bb744795e1358becd348dc84 14.51MB / 14.51MB
=> => extracting sha256:627c55a201a9095fd9bb5b05a4c058246c463987d178f778ff22db143ca94635
=> => extracting sha256:378e3a6f165ea02b44863b980aacd0a6ba57f3b96b08da831cc39bb19eb41760
=> => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1
=> => extracting sha256:901b8cfcfda7112bdd66707d3d655b06bf97a440a496fe617c60040989857b40
=> => extracting sha256:45e2e3388eeb25221c8d602f3ec3ef959da49489322ebe1528c8c855eec199a2
=> => extracting sha256:841c6b2bc7edd02d64830c140424a05c0d666457bb744795e1358becd348dc84
=> [2/2] COPY ./onlinebookstore.war /usr/local/tomcat/webapps/
=> exporting to image
=> exporting layers
=> writing image sha256:251d65def3c2b603201409da2746fd30efa10affac4fc0bde417a92fb890a3eb
=> naming to docker.io/library/project-image
881alc2fc2ec3b31ebb48bb99085de750d288457afc8f1907fd4d0022dac3988
Using default tag: latest
The push refers to repository [docker.io/kavyakb02/project-image]
7bfff707d3a51: Pushed
5f70bf18a086: Mounted from library/tomcat
ea74b0c70fa7: Mounted from library/tomcat
4f31c8da89eb: Mounted from library/tomcat
448e27e229f0: Mounted from library/tomcat
f9e5f9899ee3: Mounted from library/tomcat
818ff17bf41c: Mounted from library/tomcat
e8bce0aab6d8: Mounted from library/tomcat
latest: digest: sha256:717d07018ad63821e9ddc7c461b4a2ee541ac142c37b486df6357fc4c2929a55 size: 2410
[root@ip-172-31-35-20 ~]#
```

← → ↺ ⚠ Not secure 13.201.121.74:8080/job/project-1/1/console

Jenkins

project-1 / #1 / Console Output

[INFO] Configured Artifact: com.github.jsimone:webapp-runner:8.0.30.2:jar

[INFO] Downloading from central: <https://repo.maven.apache.org/maven2/com/github/jsimone/webapp-runner/8.0.30.2/webapp-runner-8.0.30.2.jar>

[INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/com/github/jsimone/webapp-runner/8.0.30.2/webapp-runner-8.0.30.2.jar> (9.1 MB at 65 MB/s)

[INFO] Copying webapp-runner-8.0.30.2.jar to /var/lib/jenkins/workspace/project-1/target/dependency/webapp-runner.jar

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 30.692 s

[INFO] Finished at: 2025-12-31T03:45:39Z

[INFO] -----

Waiting for Jenkins to finish collecting data

[JENKINS] Archiving /var/lib/jenkins/workspace/project-1/pom.xml to onlinebookstore/onlinebookstore-0.0.1-SNAPSHOT/onlinebookstore-0.0.1-SNAPSHOT.pom

[JENKINS] Archiving /var/lib/jenkins/workspace/project-1/target/onlinebookstore.war to onlinebookstore/onlinebookstore-0.0.1-SNAPSHOT/onlinebookstore-0.0.1-SNAPSHOT.war

channel stopped

Finished: SUCCESS

REST API Jenkins 2.528.3

← → ↺ ⚠ Not secure 13.201.121.74:8080/job/project-1/1/

Jenkins

project-1 / #1

Status

✓ #1 (31-Dec-2025, 3:45:02 am)

Add description Keep this build forever

<> Changes

📄 Console Output

📄 Edit Build Information

🗑 Delete build '#1'

🕒 Timings

🔗 Git Build Data

🔄 Redeploy Artifacts

👤 See Fingerprints

🕒 Started by user admin

🕒 This run spent:

- 4 ms waiting;
- 36 sec build duration;
- 36 sec total from scheduled to completion.

git

Revision: 58c7929bfff69672bb1c4f24abc53b887dbc91ec1

Repository: <https://github.com/shashirajraja/onlinebookstore.git>

- refs/remotes/origin/master

Module Builds


🟢 onlinebookstore

30 sec

Started 44 min ago

Took 36 sec

← → ↻ ⚠ Not secure 13.201.121.74:8080/job/project-1/ ☆ 📄 ⬇ 👤 ⋮

 **Jenkins** / project-1

Status

<> Changes

Workspace

Build Now

Configure

Delete Maven project

Modules

Rename

project-1

This build is for devops project

Permalinks

- Last build (#1), 42 min ago
- Last stable build (#1), 42 min ago
- Last successful build (#1), 42 min ago
- Last completed build (#1), 42 min ago

Builds

Filter

/

Today


✓ #1 3:45 am

▼

Edit description

REST API Jenkins 2.528.3

← → ↻ 🌐 hub.docker.com/repository/docker/kavyakb02/project-image/tags ☆ 📄 ⬇ 👤 ⋮

 **hub** Explore My Hub 🔍 Search Docker Hub CtrlK 🔔 🔄 ⋮ K

kavyakb02
Docker Personal

Repositories

Hardened Images

Collaborations

Settings

Default privacy

Notifications

Billing

Usage

Pulls

Storage

Repositories / project-image / Tags

kavyakb02/project-image

Last pushed 41 minutes ago · ☆0 · ⬇15

Add a description

Add a category

General Tags Image Management BETA Collaborators Webhooks Settings

Sort by Newest

Filter tags

Delete

TAG

latest

Last pushed 41 minutes by kavyakb02

Digest

OS/ARCH

Last pull

Compressed size

717d07018ad6

linux/amd64

less than 1 day

152.01 MB

Docker commands


To push a new tag to this repository:

docker push kavyakb02/project-image:tagname

Public view

docker pull kavyakb02/project-image:latest

← → ↻ ⚠ Not secure 13.201.121.74:8081/onlinebookstore/CustomerLogin.html ☆ 📄 ⬇ 👤 ⋮

 Home Login Register

Welcome to Online Book Store

Customer LOGIN

Are you Seller?, Click Here to Login as Admin

Username : Enter your Username

Password : Enter Password

78566 Enter Captcha

Login as an User

17. Advantages of This Project

- Fully automated deployment
 - Faster delivery
 - Reduced manual errors
 - Containerized application
 - Real-world DevOps workflow
-

18. Explanation

“In this project, I automated the deployment of a Java web application using Jenkins CI/CD. Jenkins pulls code from GitHub, builds a WAR using Maven, creates a Docker image with Tomcat, pushes it to Docker Hub, and runs the container automatically. The application is accessible through a browser using Docker port mapping.”

19. Future Enhancements

- Kubernetes deployment (EKS / Minikube)
 - Jenkinsfile-based pipeline
 - Webhook integration
 - Load Balancer service
 - Monitoring with Prometheus
-

20. Conclusion

This project successfully demonstrates **end-to-end CI/CD automation** using **industry-standard DevOps tools**. It reflects a **real production-style deployment pipeline** suitable for DevOps Engineer and Cloud roles.
