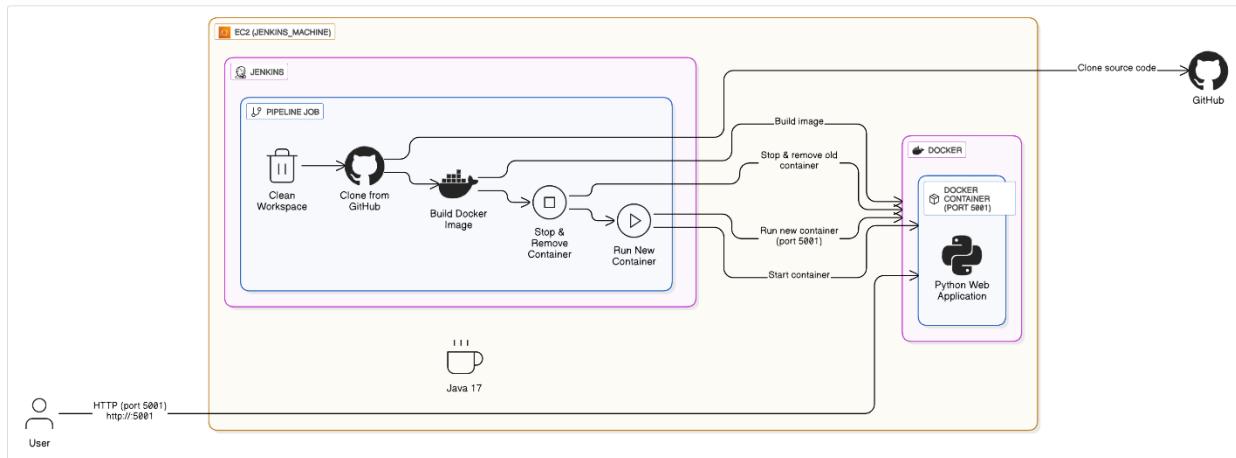


# AUTOMATED DEPLOYMENT OF PYTHON WEB APPLICATION VIA JENKINS PIPELINE ON AWS EC2

**Project Repository:** [https://github.com/dnp176/FinancesTrackProject\\_Live.git](https://github.com/dnp176/FinancesTrackProject_Live.git)



Dhruv Patel (@dnp176)

## EC2 Instance Creation for Jenkins

### Steps:

1. Login to AWS Console
2. Go to EC2 → Launch Instance

### Settings:

- Name: Jenkins\_Machine
- AMI: Amazon Linux 2
- Instance Type: t2.micro
- Storage: 10 GB
- Key Pair: Select/Create

### Security Group:

- **Create new Security Group**
- **Allow following ports:**
  - **22 (SSH)**
  - **8080 (Jenkins)**
  - **5001 (Application)**

Inbound rules (5)									 Manage tags	 Edit inbound rules
	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description		
<input type="checkbox"/>	-	sgr-0830adf623a94d5b	IPv4	SSH	TCP	22	0.0.0.0/0	-		
<input type="checkbox"/>	-	sgr-01ecc6b60870e9a5	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	-		
<input type="checkbox"/>	-	sgr-0cad8725742c6eaf2	IPv4	HTTPS	TCP	443	0.0.0.0/0	-		
<input type="checkbox"/>	-	sgr-0fea7f14836295541	IPv4	HTTP	TCP	80	0.0.0.0/0	-		
<input type="checkbox"/>	-	sgr-004a9423703e1385e	IPv4	Custom TCP	TCP	5001	0.0.0.0/0	-		

### Launch Instance:

- **Click Launch**
- **Connect via SSH: `ssh -i "key.pem" ec2-user@your-ip`**

## Install Java 17 on EC2

```
sudo apt update  
sudo apt install openjdk-17-jdk -y  
java -version
```

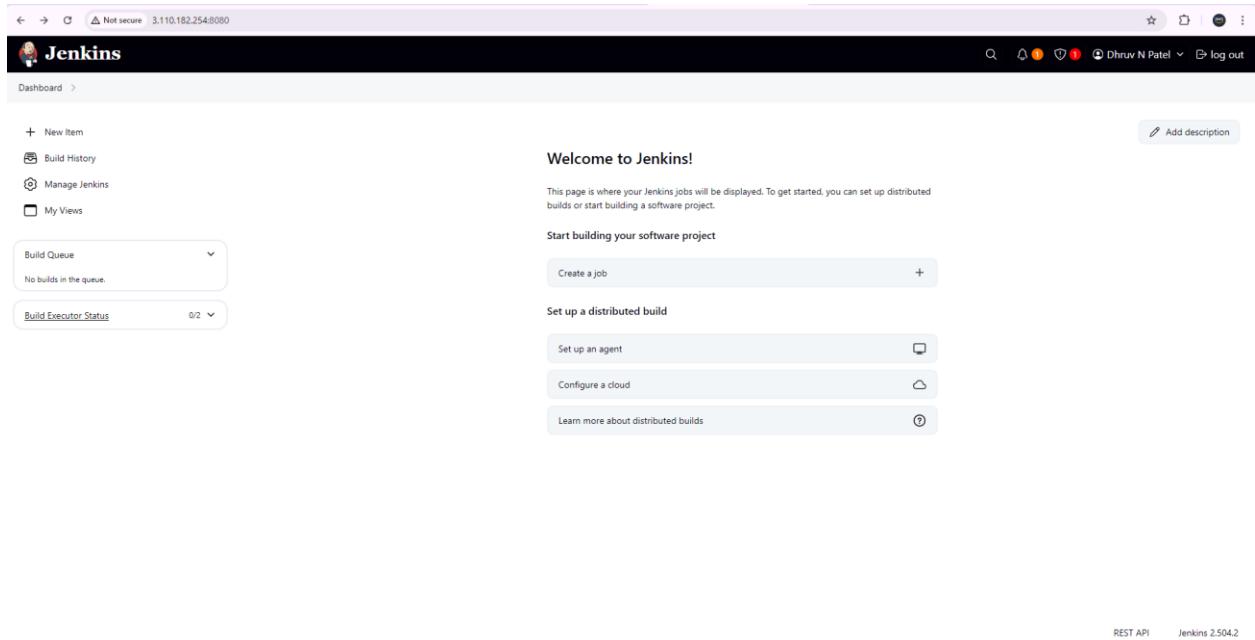
## Install Jenkins on EC2

```
sudo apt update  
  
sudo apt install openjdk-17-jdk -y  
  
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key |  
sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null  
  
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee  
/etc/apt/sources.list.d/jenkins.list > /dev/null  
  
sudo apt update  
  
sudo apt install jenkins -y  
  
sudo systemctl enable jenkins  
  
sudo systemctl start jenkins  
  
sudo systemctl status Jenkins
```

## Access Jenkins:

1. Open browser:  
<http://your-server-ip:8080>
2. Get Jenkins initial password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```



## Install Pipeline: Stage View Plugin in Jenkins

### Steps:

1. Open Jenkins Dashboard
2. Click Manage Jenkins
3. Click Manage Plugins
4. Go to Available tab
5. Search for Pipeline: **Stage View**
6. Check the checkbox
7. Click Install without restart

## Install Docker on EC2

```
sudo apt update

sudo apt install ca-certificates curl gnupg -y

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \

"deb [arch=$(dpkg --print-architecture) 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 > /dev/null

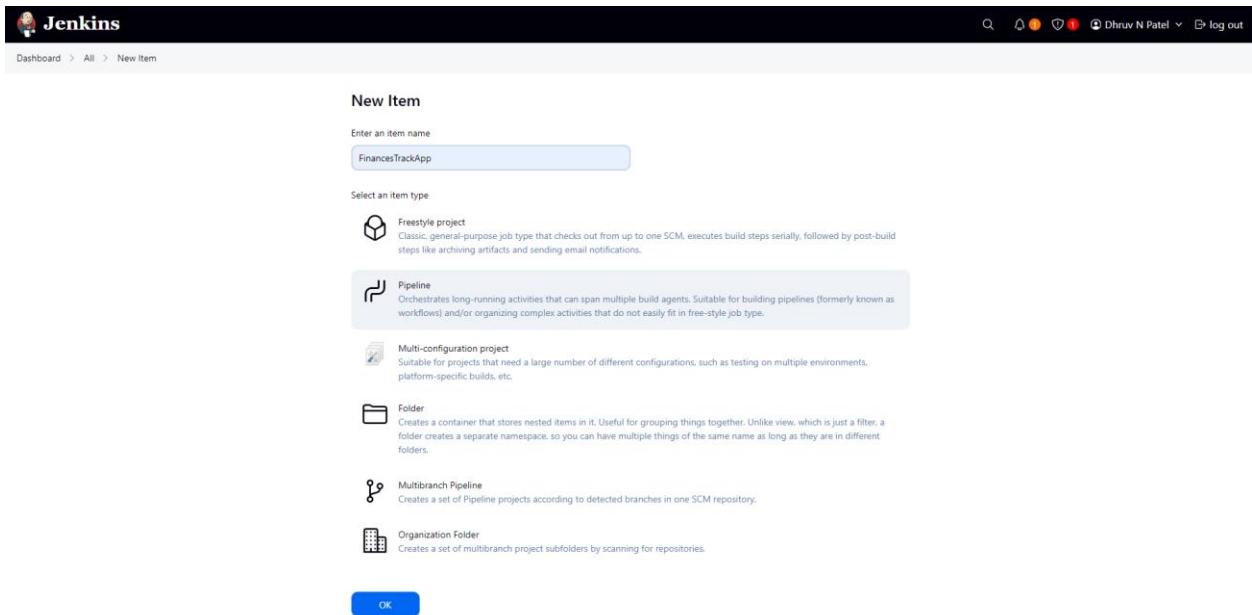
sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

sudo systemctl start docker
sudo systemctl enable docker
sudo docker --version
sudo usermod -aG docker $USER
sudo usermod -aG docker jenkins
sudo systemctl restart docker
sudo systemctl restart Jenkins
```

## Jenkins Pipeline Project Creation

1. Open **Jenkins Dashboard**.
2. Click on **New Item** (top-left corner).
3. Enter a **name** for your pipeline project (e.g., Fintrack\_Pipeline).
4. Select **Pipeline** as the project type.
5. Click **OK**.
6. Scroll down to the **Pipeline** section.
7. In the **Definition** dropdown, select **Pipeline script**.
8. Paste the **Jenkins Pipeline Script** (given above) into the script textbox.
9. Click Save.



The screenshot shows the Jenkins 'New Item' creation interface. At the top, there's a search bar and user information. Below it, the 'New Item' form has a 'Name' field containing 'FinancesTrackApp'. Under 'Select an item type', the 'Pipeline' option is selected, highlighted with a blue border. Other options like 'Freestyle project', 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder' are also listed with their descriptions. At the bottom of the form is an 'OK' button.

## Jenkins Pipeline Script

```
pipeline {
    agent any
    environment {
        APP_NAME      = "fintrack_app"
        CONTAINER_NAME = "${APP_NAME}_container"
        APP_PORT      = "5001"
        IMAGE_TAG     = "${env.BUILD_NUMBER}"
    }
    stages {
        stage('Clean Workspace') {
            steps {
                cleanWs()
            }
        }
        stage('Clone Code') {
            steps {
                git branch: 'main', url:
'https://github.com/dnp176/FinancesTrackProject_Live.git'
            }
        }
        stage('Build Docker Image') {
            steps {
                script {
                    sh "docker build --no-cache -t
${APP_NAME}:${IMAGE_TAG} ."
                }
            }
        }
        stage('Stop & Remove Existing Container') {
            steps {
                script {
                    sh """
                    docker stop ${CONTAINER_NAME} || true
                    docker rm ${CONTAINER_NAME} || true
"""
                }
            }
        }
    }
}
```

```

stage('Run Docker Container') {
    steps {
        script {
            sh "docker run -p ${APP_PORT}:${APP_PORT} --name ${CONTAINER_NAME} -d ${APP_NAME}:${IMAGE_TAG}"
        }
    }
}
post {
    always {
        echo "Pipeline for ${APP_NAME} finished. Container: ${CONTAINER_NAME}, Image Tag: ${IMAGE_TAG}"
    }
}
}

```

**FinancesTrackApp**

This is Sample App For creation CICD Pipeline.

**Stage View**

Clean Workspace	Clone Code	Build Docker Image	Stop & Remove Existing Container	Run Docker Container	Declarative: Post Actions
333ms	5s	23s	809ms	1s	257ms

Average stage times: (full run time: ~374ms)

**Builds**

No builds

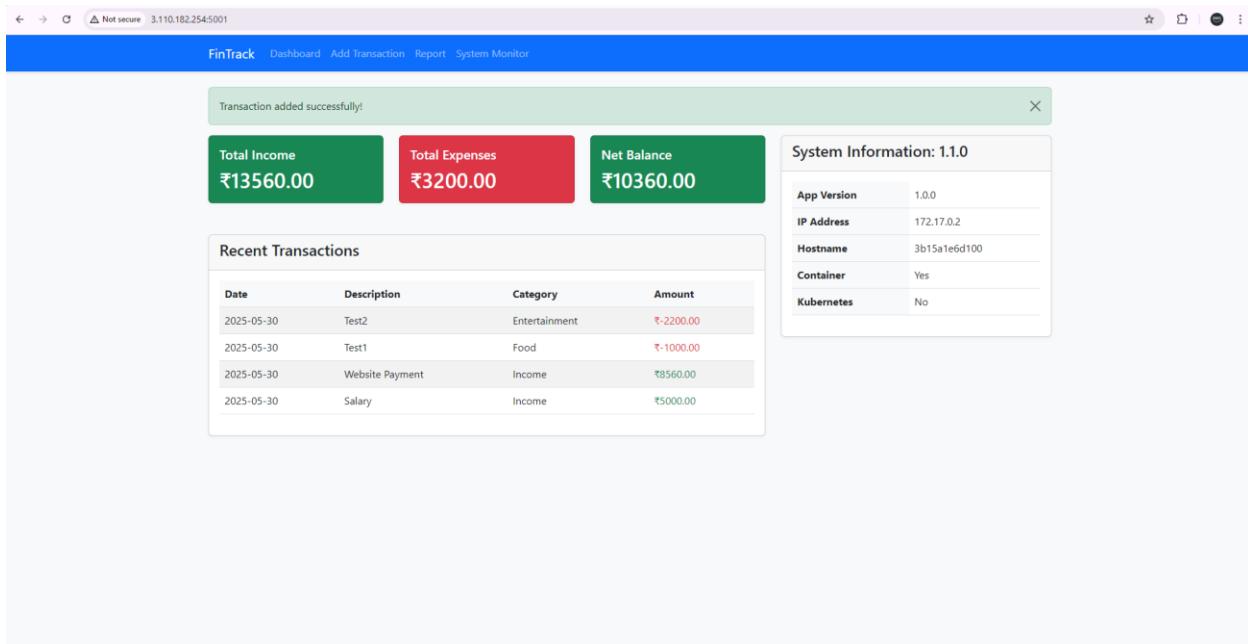
Today: #1 11:39 AM

**Permalinks**

## Application Access

**Once the Docker container is running, open your browser and access your app at:**

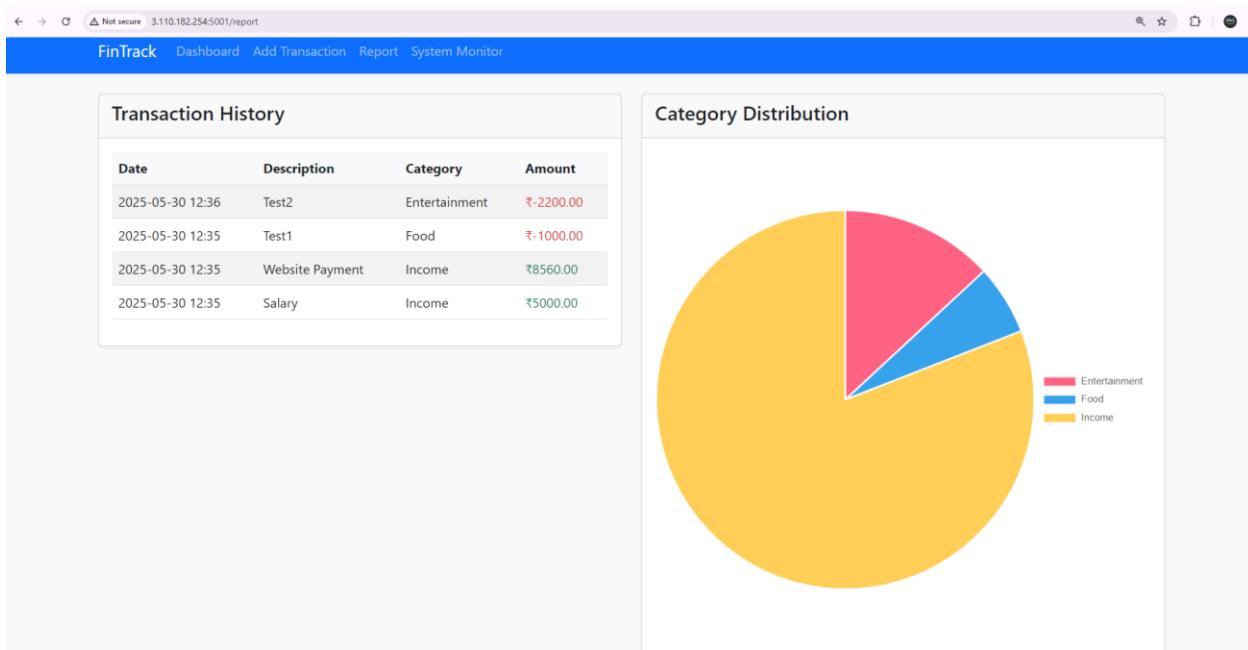
***http://<EC2-Public-IP>:5001***



The screenshot shows the FinTrack application's dashboard. At the top, there are three summary boxes: "Total Income ₹13560.00" (green), "Total Expenses ₹3200.00" (red), and "Net Balance ₹10360.00" (green). Below these is a "Recent Transactions" table with the following data:

Date	Description	Category	Amount
2025-05-30	Test2	Entertainment	₹-2200.00
2025-05-30	Test1	Food	₹-1000.00
2025-05-30	Website Payment	Income	₹8560.00
2025-05-30	Salary	Income	₹5000.00

To the right, there is a "System Information" panel showing details like App Version 1.0.0, IP Address 172.17.0.2, Hostname 3b15a1e6d100, Container Yes, and Kubernetes No.

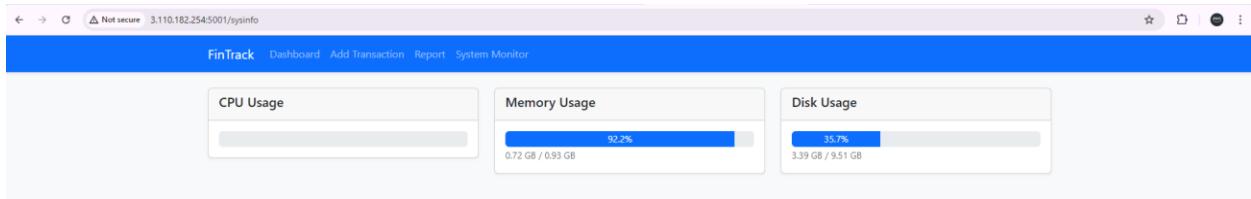


The screenshot shows the FinTrack application's report page. On the left, there is a "Transaction History" table with the same data as the dashboard:

Date	Description	Category	Amount
2025-05-30 12:36	Test2	Entertainment	₹-2200.00
2025-05-30 12:35	Test1	Food	₹-1000.00
2025-05-30 12:35	Website Payment	Income	₹8560.00
2025-05-30 12:35	Salary	Income	₹5000.00

On the right, there is a "Category Distribution" section featuring a pie chart. The chart shows the proportion of transactions by category: Income (yellow), Food (blue), and Entertainment (pink). The legend indicates:

- Entertainment
- Food
- Income



\* \* \*