**📘 CI/CD Pipeline Tutorial: Creating a BI Dashboard Using AWS EC2, Docker, GitHub, and Jenkins**

**👩‍🏫 Audience**

Beginner to intermediate students exploring DevOps and cloud deployment through hands-on learning.

**📍 Objective**

By the end of this tutorial, audience will:

* Set up a Jenkins-based CI/CD pipeline
* Build and deploy a Dockerized BI dashboard app
* Trigger automated builds via GitHub push events

**🔧 Tech Stack**

* AWS EC2 (Amazon Linux 2)
* Jenkins (CI/CD tool)
* Docker (Containerization)
* GitHub (Version control)
* Dash (Python library for data visualization)

**🪜 Step-by-Step Implementation**

**1. Launch an EC2 Instance**

* Choose Amazon Linux 2 AMI
* Instance type: t2.micro (Free Tier)
* Add 10–15 GB EBS volume (recommended)
* Open ports: **22 (SSH), 80 (HTTP), 8080 (Jenkins), 443 (HTTPS)**
* Download key pair (e.g., jenkins-key.pem)

**2. Connect to EC2 via SSH**

chmod 400 jenkins-key.pem

ssh -i "jenkins-key.pem" ec2-user@<EC2-IP>

**3. Install Dependencies**

sudo yum update -y

sudo yum install git docker -y

sudo systemctl start docker

sudo usermod -aG docker ec2-user

Log out and log in again for Docker group changes to apply.

Clone your GitHub repository (replace with your actual repo URL):

git clone https://github.com/Shefaliarora/aws-bi-dashboard-demo demo

cd demo

Build your Docker image from the cloned app files:

docker build -t bi-dashboard .

Run your BI Dashboard container mapping EC2 port 80 to container port 5000:

docker run -d -p 80:5000 --name bi-dashboard bi-dashboard

Verify the container is running:

docker ps

Your BI Dashboard should now be accessible at: http://<EC2-PUBLIC-IP>

**4. Install Jenkins**

sudo yum install -y java-17-amazon-corretto-devel

That’s the official and easiest way to install Java 17 on Amazon Linux 2.

sudo wget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

sudo yum install jenkins -y

sudo systemctl enable jenkins

sudo systemctl start jenkins

Check Jenkins: http://<EC2-IP>:8080

**5. Unlock Jenkins**

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

Paste password on Jenkins unlock page.

**6. Install Suggested Plugins in Jenkins**

**7. Configure GitHub Repository**

* Create GitHub repo (e.g., aws-bi-dashboard-demo)
* Push your app.py, requirements.txt, Dockerfile

**Sample Files:**

**app.py**

import dash

from dash import html

app = dash.Dash(\_\_name\_\_)

app.layout = html.Div(children=[html.H1('BI Dashboard'), html.P('Hello from Dash!')])

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True, host='0.0.0.0', port=5000)

**requirements.txt**

dash

pandas

plotly

**Dockerfile**

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]

**8. Set Up Jenkins Job**

* New Item → Freestyle Project
* Source Code Management: Git → Add repo URL
* Build Triggers: *GitHub hook trigger for GITScm polling*
* Build Steps: Add Execute Shell:

docker stop bi-dashboard || true

docker rm bi-dashboard || true

docker build -t bi-dashboard .

docker run -d -p 80:5000 --name bi-dashboard bi-dashboard

**9. Create Webhook in GitHub**

* Settings → Webhooks → Add Webhook
* Payload URL: http://<EC2-IP>:8080/github-webhook/
* Content type: application/json
* Events: Just the push event

**10. Test the Pipeline**

* Make a code change in app.py and push to GitHub
* Jenkins should trigger build → Docker image rebuilt → BI dashboard updated on EC2 public IP

**✅ Outcome**

You now have an automated CI/CD pipeline:

* **GitHub** handles source control
* **Jenkins** automates build and deployment
* **Docker** containers run the BI Dashboard on **EC2**

**📈 Future Enhancements**

* Use AWS ECR to store Docker images
* Add monitoring via CloudWatch
* Secure Jenkins with HTTPS and credentials
* Integrate with Slack or email for notifications