



# Full Deployment Guide: EC2 + Docker + Docker Compose

This guide will help you deploy a **frontend + backend** app to an **Ubuntu EC2 instance on AWS**, using Docker and Docker Compose.

---

## 1 Launch EC2 Instance on AWS

**Steps:**

1. Go to **AWS EC2 → Launch Instance**.
  2. **Name:** use a meaningful name (e.g., `resume-app-server`).
  3. **AMI (OS):** choose **Ubuntu (latest LTS)**.
  4. **Key Pair:**
    - Create a new one (type: `.pem`).
  5. Keep the default settings for everything else.
  6. Click **Launch Instance**.
- 

## 2 Connect to EC2 via SSH

**From Mac / Linux:**

```
bash
chmod 600 key-name.pem
ssh -i key-name.pem ubuntu@<ip-address>
```

**From Windows (PowerShell):**

```
bash
ssh -i key-name.pem ubuntu@<ip-address>
```



## 3 Install Required Packages

Run these commands after connecting to your instance:

bash

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install -y docker.io
```

```
sudo apt install -y docker-compose-plugin
```

```
sudo apt install -y docker-compose
```

```
sudo apt install -y curl
```

### Add user to Docker group:

bash

```
sudo usermod -aG docker ubuntu
```

Then log out and back in:

bash

exit

```
ssh -i key-name.pem ubuntu@<ip-address>
```

---

## 4 Clone Your Project from GitHub

bash

```
git clone <your-repo-url>
```



```
cd <project-folder>
```

---

## 5 Backend Setup

### 📦 Backend Dockerfile

text

```
FROM node:20-alpine
COPY . .
RUN npm install
EXPOSE 21000
CMD ["node", "index.js"]
```

### ✳️ Backend docker-compose.yml

text

```
services:
  resume-backend:
    build: .
    container_name: resume-backend
    restart: always
    ports:
      - "8080:21000"    # External : Internal
  deploy:
    resources:
```



limits:

cpus: "1.0"

memory: 1024M

---

## 6 Frontend Setup

### 📦 Frontend Dockerfile

text

```
FROM node:20-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5173
CMD [ "npm", "run", "dev" ]
```

### 🛠️ Frontend docker-compose.yml

text

```
version: "3.8"
```

```
services:
```

```
frontend:
```

```
build:
```



```
context: .

dockerfile: Dockerfile

container_name: resume-frontend

restart: always

ports:

- "5173:5173"

volumes:

- .:/app

- /app/node_modules

command: npm run dev -- --host
```

---

## 7 Configure AWS Security Groups

In AWS EC2 → **Security Groups** → **Inbound Rules**, add:

Type	Protocol	Port	Source	Description
SSH	TCP	22	0.0.0.0/ 0	SSH Access
Custom TCP	TCP	8080	0.0.0.0/ 0	Resume Backend
Custom TCP	TCP	5173	0.0.0.0/ 0	Resume Frontend

---



## 8 Build and Run Containers

Navigate to the respective directories (`backend` or `frontend`):

```
bash
```

```
sudo docker-compose down
```

```
sudo docker-compose up -d --build
```

---

## 9 Verify Running Containers

```
bash
```

```
docker ps
```

You should see:

- `resume-backend`
  - `resume-frontend`
- 

## 10 Enable Firewall (UFW)

```
bash
```

```
sudo ufw allow 8080
```

```
sudo ufw allow 5173
```

```
sudo ufw enable
```

Check status:

```
bash
```

```
sudo ufw status
```



## 11 Test the Backend Locally

Inside the EC2 instance:

bash

```
curl http://localhost:21000
```

---

## 12 Test Backend via Public IP

bash

```
curl http://<ip-address>:8080
```

Example:

bash

```
curl http://51.20.108.111:8080
```

---

## 13 Test in Browser

- **Backend:** <http://<ip-address>:8080>
- **Frontend:** <http://<ip-address>:5173>