# 🚀 Launch CloudPi Using Docker Compose on EC2 (with Optional HTTPS)

This guide will walk you through setting up CloudPi using Docker Compose on an Amazon EC2 instance. Follow each step carefully to ensure a smooth deployment.

## ✅ Step 1: Prepare Your EC2 Instance

### Launch EC2 Instance with the Following Configuration:

* **Instance Name:** CloudPiDockerHost
* **AMI:** Amazon Linux 2023
* **Instance Type:** t2.2xlarge
* **Key Pair:** Use your existing key or create a new one (download the .pem file)
* **Volume Configuration:**
  + Volume Size: 512 GB
  + Volume Type: gp3
* **Networking:**
  + Auto-assign Public IP: Enabled
  + Open these inbound ports in the **Security Group**:
    - TCP: 22 (SSH)
    - TCP: 80, 443 (HTTP/HTTPS)
    - TCP: 3000, 5001, 5005, 6379, 8088, 3306

## ✅ Step 2: Connect to EC2 via SSH

You need to use the PEM file associated with your EC2 instance’s key pair to establish a secure connection.

### 🔹 For ****macOS / Linux Users****

**Where:** Your local terminal

1. **Open Terminal**
2. **Navigate to the directory containing your .pem file**:

cd /path/to/your/pem-file

1. **Connect to EC2**:

ssh -i your-key-name.pem ec2-user@<EC2\_PUBLIC\_IP>

Replace your-key-name.pem with your actual PEM filename, and <EC2\_PUBLIC\_IP> with your EC2 instance’s public IP.

### 🔹 For ****Windows Users using Command Prompt or PowerShell****

**Where:** Command Prompt or PowerShell (Windows 10/11 with OpenSSH)

1. Open **Command Prompt** or **PowerShell**
2. Navigate to the directory where your .pem file is saved, e.g.:

cd "C:\Users\YourUsername\Downloads"

1. Run the SSH command:

ssh -i your-key-name.pem ec2-user@<EC2\_PUBLIC\_IP>

✅ **Note:** Windows must have **OpenSSH client enabled** (enabled by default on Windows 10/11).

If you see a permission denied error, consider moving the PEM file to a safe location and **set restrictive permissions** using PowerShell:

icacls your-key-name.pem /inheritance:r

icacls your-key-name.pem /grant:r "%username%:R"

### 🔹 For ****Windows Users using PuTTY****

If you cannot use .pem directly, follow this alternative:

1. Open **PuTTYgen**
   * Click **Load** and select your .pem file
   * Click **Save private key** to generate a .ppk file
2. Open **PuTTY**
3. In the **Session** tab:
   * Host Name: ec2-user@<EC2\_PUBLIC\_IP>
   * Port: 22
   * Connection type: SSH
4. In the **SSH > Auth** tab:
   * Browse to the .ppk file under **Private key file for authentication**
5. Click **Open** to connect

📝 **Replace**:

* your-key-name.pem with your actual PEM filename
* <EC2\_PUBLIC\_IP> with the public IP of your EC2 instance (available in AWS Console)
* Ensure this is done **on your local system** before proceeding to EC2 setup

## ✅ Step 3: Install Docker, Docker Compose, and Git

***Where:*** *Inside the* ***EC2 terminal*** *(after SSH)*

sudo yum update -y

sudo yum install -y docker

sudo service docker start

sudo usermod -a -G docker ec2-user

🔁 Log out and log back in to apply Docker group permissions.  
exit

**ReConnect to your EC2 instance using SSH**:

ssh -i your-key-name.pem ec2-user@<EC2\_PUBLIC\_IP>

Replace:

* your-key-name.pem with your actual PEM file name
* <EC2\_PUBLIC\_IP> with the **public IP address** of your EC2 instance (visible in AWS Console)

### Install Docker Compose v2:

mkdir -p ~/.docker/cli-plugins

curl -SL https://github.com/docker/compose/releases/download/v2.24.2/docker-compose-linux-x86\_64 -o ~/.docker/cli-plugins/docker-compose

chmod +x ~/.docker/cli-plugins/docker-compose

docker compose version

### Install Git:

sudo yum install git –y

git --version

## ✅ Step 4: Clone CloudPi Docker Repository

### a. Docker Login (Mandatory)

docker login -u <your\_docker\_username>

📧 **Username/Password**: Contact CloudPi (refer to the email shared)

### b. Clone the Repository

git clone https://github.com/PurpleDataInc-TX/cloudpi.git

cd cloudpi

Ensure these files are present:

ls

# You should see:

# docker-compose.yml

# .env

# Cloudrestartnew.sh

⚠️ Ensure .env contains correct values. Contact CloudPi for any missing environment values or credentials.

🎯 Update cloudpi-app and cloudpi-db versions inside docker-compose.yml if needed. Confirm the version from CloudPi support.

## ✅ Step 5: Upload SSL Certificates (If Using HTTPS)

### On Your Local Machine where you are connected to instance using ssh:

scp -i your-key-name.pem -r ./certs ec2-user@<EC2\_PUBLIC\_IP>:/home/ec2-user/certs

Replace:

* your-key-name.pem with your actual PEM file name
* <EC2\_PUBLIC\_IP> with the **public IP address** of your EC2 instance (visible in AWS Console)

Ensure the following files exist in /home/ec2-user/certs/:

├── ca\_bundle.crt

├── cloudpi\_certificate.crt

└── cloudpi\_private.key

Set appropriate permissions:

chmod 600 /home/ec2-user/certs/cloudpi\_private.key

## ✅ Step 6: Run Docker Compose

cd ~/cloudpi

### docker compose up –d

## ✅ Step 7: Configure and Run Cloud Restart Script

### 1. Copy Script into the App Container

docker ps # to get the container name (usually cloudpi-app)

docker cp Cloudrestartnew.sh cloudpi-app:/app/

docker exec -it cloudpi-app /bin/bash

### 2. Inside the Container:

cd /app

chmod +x Cloudrestartnew.sh

### 3. (Optional) Edit HTTPS Settings

apt-get update && apt-get install -y nano

nano Cloudrestartnew.sh

To **enable HTTPS**:

use\_https=true

SUBDOMAIN=<your\_domain\_name>

To **disable HTTPS**:

use\_https=false

Save with Ctrl + O, then press Enter, and exit with Ctrl + X.

### 4. Run the Script

./Cloudrestartnew.sh

## ⚠️ Troubleshooting: Script Execution Error

If you see:

bash: ./Cloudrestartnew.sh: cannot execute: required file not found

Try the following fixes **inside the container**:

apt-get update && apt-get install -y dos2unix

dos2unix /app/Cloudrestartnew.sh

chmod +x /app/Cloudrestartnew.sh

### /app/Cloudrestartnew.sh

**✅ Step 8: Verify the Setup and Useful Docker Commands**

**Where:** On your **local browser** and **EC2 terminal**

| **Description** | **URL** |
| --- | --- |
| Without HTTPS | http://<EC2\_PUBLIC\_IP>:3000 |
| With HTTPS (DNS setup) | https://<YOUR\_DOMAIN\_NAME> |

📝 **Note:**

* Replace <EC2\_PUBLIC\_IP> with your EC2 instance's **public IP address** (from AWS Console).
* Replace <YOUR\_DOMAIN\_NAME> with your **configured domain name** (e.g., cloudpie.ai) if using HTTPS.

**🔹 Common Docker Commands (Run in EC2 Terminal)**

| **Task** | **Command** |
| --- | --- |
| **List running containers** | docker ps |
| **Access the app container** | docker exec -it cloudpi-app /bin/bash |
| **Exit from the container** | exit |
| **View all service logs** | docker compose logs |
| **View logs for cloudpi-app only** | docker compose logs cloudpi-app |
| **Stream real-time logs** | docker compose logs -f  *(Use Ctrl + C to stop)* |
| **Stop all containers** | docker compose down |

## 🧹 Full Clean-Up Commands (Optional)

cd ~/cloudpi

docker compose down

docker system prune -a --volumes -f

Let the CloudPi team know once deployment is complete or if assistance is needed.