## CloudPi Deployment Guide

### 📌 Prerequisites

Ensure the following are installed on your system:

* Docker Desktop
* A stable internet connection
* Minimum 8GB RAM and 20GB free disk space
* Make sure 3000,5001,5005,3306,6379,8088,80 ports are available.

### 🔹 Step 1: Log in to Docker

Before proceeding, log in to Docker using your credentials:

docker login –u <username>

You will need to enter your username and password. If using a token for authentication, enter it as the password.  
For username : contact cloudpi(refer to the email shared)  
when prompted for password : contact cloudpi(refer to the email shared)

### 🔹 Step 2: Clone the Repository

Clone the GitHub repository containing docker-compose.yml, .env, and other required files:

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

### 🔹 Step 3: Change Directory to Docker-Compose File Location

Also, open your command prompt in Administrator mode and Ensure that you are in the correct directory where the docker-compose.yml file is located before running any Docker commands.

Eg:cd "C:\Users\<YourUsername>\<your-repo>"

### 🔹 Step 4: Set Up the .env File

CloudPi requires environment variables to be set at runtime. Make sure you have a .env file in the same directory as docker-compose.yml and add the necessary values.   
**Note:** Replace placeholders with correct values.

### 🔹 Step 5: Pull and Run Docker Containers

Before running the containers, ensure you pull the latest images from Docker Hub:

docker-compose pull

Then, start the application with all environment variables securely injected:

docker-compose up -d

This will pull and run the necessary containers, including MySQL, Flask API, Backend, Frontend, Redis, and Superset.

### 🔹 Step 6: Verify Running Containers

Check if the services are running:

docker ps

To view logs for a specific service:

docker logs app

### 🔹 Step 7: Access the Application

* **CloudPi Application:** <http://localhost:3000>  
    
   For login credentials :contact cloudpi (refer to the email shared with you)
* **MySQL Connection:**

docker exec -it cloudpi-db mysql

### 🔹 Step 8: Stopping Services

To stop and remove all running containers:

docker-compose down -v

To remove all images and containers:

docker system prune -af

### 🔹 Step 9: Security Considerations

* Never hardcode sensitive credentials in docker-compose.yml.
* Always pass credentials at runtime using environment variables.

### 🔹 Troubleshooting

If CloudPi does not start correctly, check logs:

docker-compose logs

If MySQL is not connecting, restart it:

docker restart cloudpi-db

Now your CloudPi system is up and running securely! 🎉