

# Fluid Calendar

[Github](#)

---

## ❏ Prerequisites

Before starting, ensure you have:

- **Docker** installed on your machine
  - **Portainer** set up and running
  - **An available port for the app** (default is 3000, but we will use 3087 in this guide)
  - **An available port for PostgreSQL** (default is 5432, but we will use 5433 for external access)
- 

## ❏ Step 1: Prepare Your Environment

### 1❏ Create a Directory for Fluid Calendar

Since we need to persist the PostgreSQL database, create a directory:

```
mkdir -p /srv/Files/Fluidcalendar/postgres_dev_data
```

This will be used to store PostgreSQL data outside of the container.

---

## ❏ Step 2: Create the `docker-compose.yml` File

### 1❏ Open Portainer and Create a New Stack

1. Go to your **Portainer dashboard**
2. Click on **Stacks** → **Add a new stack**
3. Name it: **fluid-calendar**
4. Copy and paste the following `docker-compose.yml` configuration:

```
services:
  app:
    image: eibrahim/fluid-calendar:latest
  ports:
```

```
- "3087:3000" # External 3087 → Internal 3000
env_file:
  - stack.env
depends_on:
  db:
    condition: service_healthy
restart: unless-stopped

db:
  image: postgres:16-alpine
  environment:
    - POSTGRES_USER=fluid
    - POSTGRES_PASSWORD=fluid
    - POSTGRES_DB=fluid_calendar
  ports:
    - "5433:5432" # External 5433 → Internal 5432
  volumes:
    - /srv/Files/Fluidcalendar/postgres_dev_data:/var/lib/postgresql/data
  healthcheck:
    test: ["CMD-SHELL", "pg_isready -U fluid -d fluid_calendar"]
    interval: 5s
    timeout: 5s
    retries: 5
  restart: unless-stopped
```

## 📁 Step 3: Create the `.env` File

### 1📁 Add an Environment File in Portainer

1. Still in **Portainer**, scroll down to **Environment Variables**
2. Click **Add an Environment File**
3. Name it: `stack.env`
4. Paste the following content:

```
# Database Configuration
DATABASE_URL="postgresql://fluid:fluid@db:5432/fluid_calendar" # Internal Docker communication uses port
5432

# NextAuth Configuration
# Use domain in production, localhost for development
```

```
NEXTAUTH_URL="http://localhost:3087"
NEXT_PUBLIC_APP_URL="http://localhost:3087"
NEXTAUTH_SECRET="32charcomplicatedkey"
NEXT_PUBLIC_SITE_URL="http://localhost:3087"

NEXT_PUBLIC_ENABLE_SAAS_FEATURES=false

RESEND_API_KEY=
RESEND_FROM_EMAIL=
```

## ☐ Step 4: Deploy the Stack

1. Click **"Deploy the stack"** in Portainer
2. Wait for the services to start
3. Open your browser and go to **<http://localhost:3087>**

## ☐ Step 5: Verify Everything is Running

### 1☐ Check Running Containers

In your terminal, run:

```
docker ps
```

You should see **two running containers**:

- fluidcalendar\_app\_1
- fluidcalendar\_db\_1

### 2☐ Check Logs for Errors

If something is wrong, check logs:

```
docker logs -f fluidcalendar-app-1
docker logs -f fluidcalendar-db-1
```

### 3☐ Test the Database Connection

If the app doesn't connect, manually check the database:

```
docker exec -it fluidcalendar-db-1 psql -U fluid -d fluid_calendar
```

If it works, the database is running fine.

---

## ☐ Step 6: Connect Fluid Calendar to Nextcloud

To sync your Nextcloud calendar with Fluid Calendar, use the following details:

- **Username:** *Your Nextcloud username*
  - **Password:** *An app-specific password* (Generate one in Nextcloud under *Settings* → *Security*)
  - **Server URL:** `https://cloud.example.com` (Replace with your Nextcloud instance)
  - **Path:** `/remote.php/dav`
- 

## ☐ Troubleshooting

### ☐ App stuck at "Waiting for database to be ready..."

- **Check that the database container is running:**

```
docker ps | grep fluidcalendar-db
```

- **Ensure you are using port `5432` inside Docker:**

Run this inside the **app container**:

```
docker exec -it fluidcalendar-app-1 sh  
psql "postgresql://fluid:fluid@db:5432/fluid_calendar"
```

If this works, update `.env` to use `5432`, not `5433`.

### ☐ Database Not Persisting

Make sure the volume is mounted correctly:

```
ls -la /srv/Files/Fluidcalendar/postgres_dev_data
```

If the folder is empty, check that Docker has write permissions.

---

## ☐ Conclusion

That's it! You have successfully deployed **Fluid Calendar** on **Portainer** with **Docker**. ☐

If you run into any issues, check the logs and verify the database connection. Hope this helps! ☐

---