

WEEK 4 REPORT – DEPLOYMENT AND PRODUCTION SETUP

Objectives

- Deploy the **Whitefly Detection System** into a production-ready environment.
- Implement **continuous integration and delivery (CI/CD)** using GitHub Actions.
- Set up **monitoring, logging, and rollback** support for reliability.

Summary of Activities

1. GitHub Actions (CI/CD Pipeline)

- **Purpose:** Automated the entire build, test, and deployment process for both frontend and backend.
- **Configuration:** .github/workflows/ci-cd.yml
- **Functions:**
 - Code checkout, dependency installation, and project build.
 - Docker image creation for the Django backend.
 - Deployment triggers for both **Render (backend)** and **Vercel (frontend)**.
- **Enhancements:**
 - Rollback hooks configured by *Aine Levi* to restore previous builds in case of deployment failure.

2. Vercel (Frontend Deployment)

- **Purpose:** Hosted and deployed the React-based frontend.
- **Action Used:** vercel/action-deploy@v1
- **Features:**
 - Automatic builds from GitHub.
 - Global CDN distribution for faster access.
 - Managed environment variables for API connectivity.

3. Render (Backend Deployment)

- **Purpose:** Hosted the Django backend using Docker containers.
- **Action Used:** johnbeynon/render-deploy-action@v0.0.8
- **Highlights:**
 - Database integration and automatic scaling.
 - Rollback hooks for stable recovery during deployment errors.

5. Monitoring and Logging

- We configured **Django logging handlers** for build and backend error tracking.
- Integrated **Prometheus** and **Grafana** dashboards for monitoring uptime, performance, and system health.

6. ClickUp Updates

- Tasks for **production deployment**, **rollback setup**, and **monitoring configuration** were marked as complete, indicating the successful completion of all Week 4 milestones.

Deliverables

- Fully automated CI/CD workflow
- Frontend deployed on Vercel
- Backend deployed on Render
- Monitoring and logging dashboards active
- Rollback mechanism for deployment safety

Challenges & Future Plans

- Encountered **memory limitations** on Render, causing occasional response delays.
- The team plans to **optimize model handling and build performance** in Week 5.

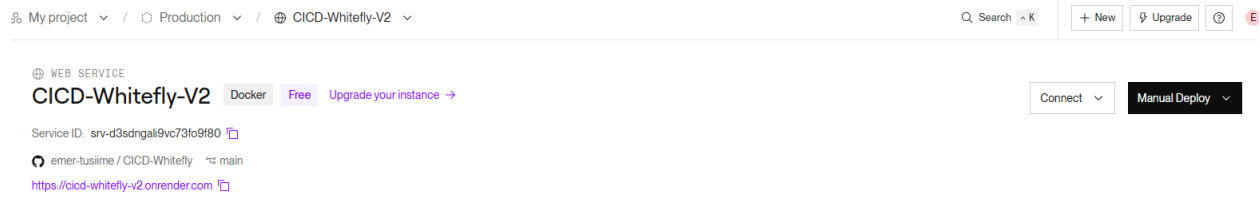


Figure 1 Setting up Deployment Environment

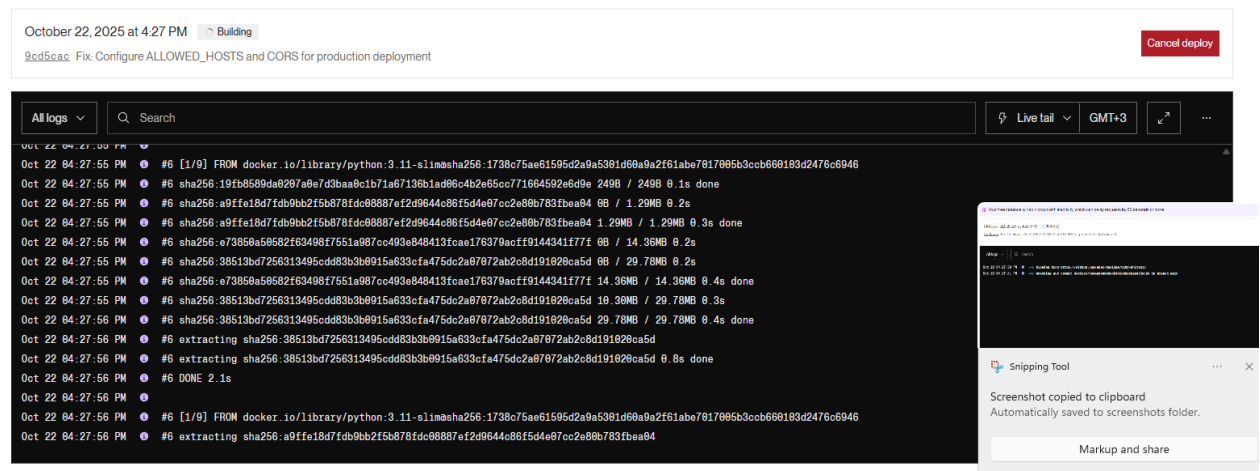


Figure 2 During the Proyes of auto build and testing

```
All logs Search Live tail GMT+3
Oct 22 04:50:43 PM #16 exporting config sha256:7148cc26e83139e3dd83d98c8219ec7e5efc2dbb228598934c123d567e7521 0.0s done
Oct 22 04:50:44 PM #16 DONE 1.3s
Oct 22 04:50:44 PM #17 exporting cache to client directory
Oct 22 04:50:44 PM #17 preparing build cache for export
Oct 22 04:50:44 PM #17 writing cache image manifest sha256:d4e452074682c1ce53ec55de9ae54131eb8ad5017eea2cc6d7d04883a924ca2f done
Oct 22 04:50:44 PM #17 DONE 0.1s
Oct 22 04:50:51 PM Pushing image to registry...
Oct 22 04:50:52 PM Upload succeeded
Oct 22 04:50:57 PM --> Deploying...
Oct 22 04:51:10 PM [2025-10-22 13:51:10 +0000] [1] [INFO] Starting gunicorn 21.2.0
Oct 22 04:51:10 PM [2025-10-22 13:51:10 +0000] [1] [INFO] Listening at: http://0.0.0.0:8000 (1)
Oct 22 04:51:10 PM [2025-10-22 13:51:10 +0000] [1] [INFO] Using worker: sync
Oct 22 04:51:10 PM [2025-10-22 13:51:10 +0000] [25] [INFO] Booting worker with pid: 25
Oct 22 04:51:14 PM Not Found: /
Oct 22 04:51:17 PM --> Your service is live 🎉
```

Figure 3 Successful deployment and the system is running

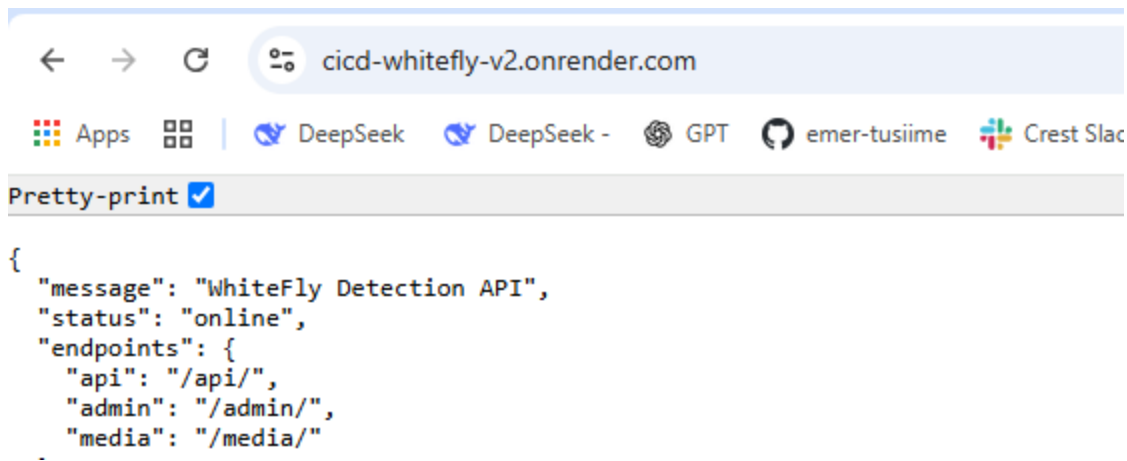


Figure 4 Ssystem is running and online

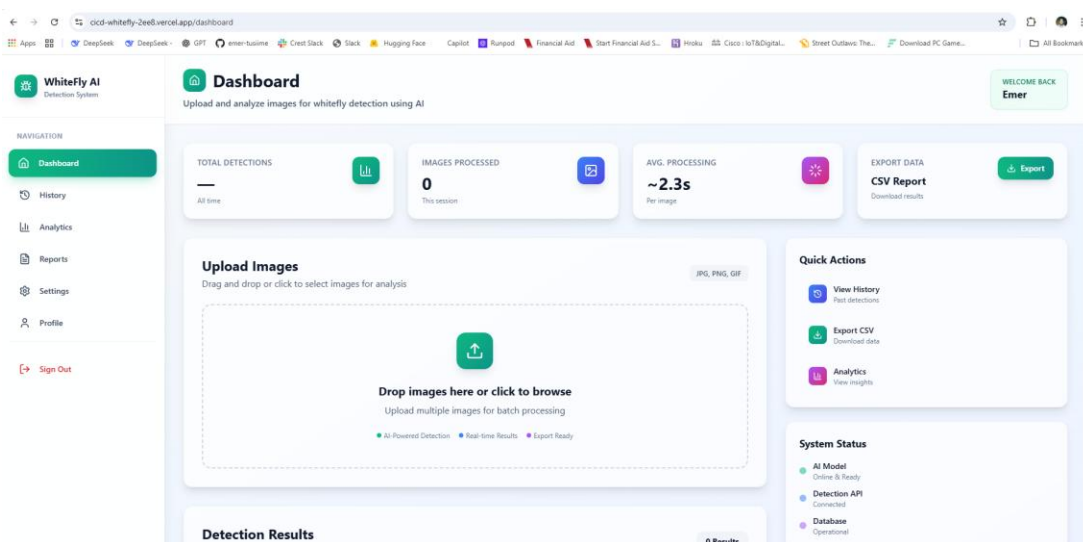


Figure 5 Application running after successful deployment