

# Continuous Integration / Continuous Deployment/Delivery

This document outlines the CI/CD process designed to streamline development, testing, and deployment across the project repositories. GitHub Actions are leveraged to automate and simplify these workflows.

CI/CD pipeline ensures a robust, efficient, and consistent process for code integration, testing, and deployment, facilitating seamless collaboration and product reliability.

## Tests

- **PR based** - These actions are triggered when a Pull Request is opened or updated with new commits. They ensure code quality and readiness for integration.
- **End to End** - These actions are triggered after a successful deployment to validate the product's overall functionality.

### Python syntax validation (PR based)

Ensure consistent formatting across Python files using the Black formatter. Files are checked before running unit tests, maintaining uniform style across services.

### Unit Tests (PR based)

Validate the functionality of individual components of a microservice. Tests are executed to ensure that singular parts of the service function as intended.

### Docker build Test (PR based)

Verify the creation of a valid Docker image from the repository's Dockerfile. Automatically builds the image to confirm all dependencies are included and the Dockerfile is correctly configured.

## Synthetic tests (End to End)

Simulate real-user interactions with the deployed application to ensure microservice compatibility and system reliability. Tests mimic browser operations on the product, which is deployed within docker containers. The source code for these tests is located in the SyntheticTest repository.

## Releases

Each repository includes a GitHub Action to automate release creation upon any push or merge to the `main` branch.

**Version Tag Format:** `<short_year>.<week_number>.<release_count>`

- `short_year` - Two-digit year (e.g., 2024 becomes 24).
- `<week_number>` - ISO week number.
- `<release_count>` - Sequential release number for the week.

For a release created in the 12th week of 2024, and being the 3rd release of the week, the version tag will be `24.12.3`.

## Deployment

Each release triggers the deployment process. As the application is microservices-based, only the service related to the updated code is rebuilt and redeployed. Other services remain operational without interruptions.