**Shared Code**

For Shared code library create Job on Jenkins (or other CI tools) that will be triggered after changes, it will get package from Nugget and run unit tests. If Unit tests are OK – All passed.

Current API version should be built with updated package and API tests should be performed.

**NOTE:** Depends of your delivery process. If the package can be delivered to production separately, API from release tag (for example), should be built with new package and API tests should be performed on DEV environment, that after deploying on stage UI Tests (End to END) and API tests should be performed, That after deploy to production E2E and API tests should be permed too. This scenario depends of your process and delivery methods. If release includes some other components to these tests should be performed for actual version for API and /or Frontend that are going to be released

Get API Nuget package

Run Unit tests

Build API with new package

Deploy API and Run API tests (Dev environment)

Deploy API and Run API tests (pre-prod environment)

Deploy API and Run API tests (prod environment) and E2E tests

**Push**

After each commit service sources should be cloned built and Unit tests should be permed. If Unit tests are passed, push should be deployed to environment (dev).

Build **Push service**

Run Unit Tests

Deploy to Dev Environment

Run E2E tests

Deploy to Pre-production Environment

Run E2E tests

Deploy to Pre-Production Environment

Run E2E tests

**API:**

On each commit to API project should be built, unit tests should be performed if all tests are passed, API tests should be performed.

Build API

Run Unit tests

Deploy API to Dev Environment

Run API Tests

Run E2E tests (UI tests Frontend)

Deploy API to Pre-production Environment

Run API Tests

Run E2E tests

RUN E2E tests

Deploy Print service to Production and Run API tests

Run API Tests

Run E2E tests

**Printing**: After each commit service sources should be cloned built and Unit tests should be permed.

Then integration with API should be checked.

Build Printing service

Run Unit tests

Deploy API to Dev Environment

RUN API Tests

(API calls related to print)

Deploy API to Pre-production Environment

Run API Tests

RUN E2E tests

Deploy API and Run API tests

Run API Tests

RUN E2E tests

**Frontend**

Build **Frontend project**

**Run Frontend Unit tests**

Deploy **Frontend** to Dev Environment

RUN UI Tests

Deploy **Frontend** Pre-production Environment

RUN UI Tests

RUN E2E tests

Deploy **Frontend** Production Environment

RUN UI Tests (E2E)

**NOTE:** Depends of your delivery process. Each component should be checked in integration with related components on each environment (usually dev, pre-production and production), unit tests should be performed for arch component after each commit

**If some step is failed changes should be roll backed. If we need some manual test. This should be done after testing on DEV.**

**By E2E Tests I mean UI Tests**