Continuous Integration & Deployment

Software Engineering - Lab

Marco Robol - marco.robol@unitn.it

Contents of today class

In today's class, we will see how to set-up a continuous integration environment with Github Actions, and how to deploy our application on Render.com.

> **EasyLib repos**
>
> BackEnd - https://github.com/unitn-software-engineering/EasyLib
>
> Vue FrontEnd - https://github.com/unitn-software-engineering/EasyLibVue

> **EasyLib deploys**
>
> Basic Frontend - https://easy-lib.onrender.com/
>
> Vue Frontend - https://easy-lib.onrender.com/EasyLibApp/ or https://unitn-software-engineering.github.io/EasyLibApp/

CI-CD with GitHub Actions

Automate your workflow from idea to production - GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

Workflows

- **Node.js CI** - This workflow will do a clean installation of node dependencies, cache/restore them, build the source code and run tests across different versions of node - https://help.github.com/actions/language-and-framework-guides/using-nodejs-with-github-actions

...if build succeed, we want to deploy to Heroku...

- **Deploy to Render.com** - A very simple GitHub action that allows you to deploy on Render - https://github.com/marketplace/actions/render-github-action

Node.js CI Workflow

`EasyLib/.github/workflows/node.js.yml`

```yaml
name: Node.js CI
on:
  push:
    branches: [ master ]
jobs:
  test:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [14.x]
    steps:
    - uses: actions/checkout@v3
    - name: Use Node.js ${{ matrix.node-version }}
      uses: actions/setup-node@v3
      with:
        node-version: ${{ matrix.node-version }}
        cache: 'npm'
    - run: npm ci #similar to npm install , except it's meant to be used in automated environments
    - run: npm run build --if-present
    - run: npm test
```

Environment variables

`EasyLib/.github/workflows/node.js.yml`

```yaml
name: Node.js CI

    # This is used to load Environment-level secrets, from the specified environment.
    # Instead, repository secrets are loaded by default.
    environment: production

    env:
      SUPER_SECRET: ${{ secrets.SUPER_SECRET }} # Must be set as a GitHub secret
      DB_URL: ${{ secrets.DB_URL }} # Must be set as a GitHub secret
...
```

Deploy on Render.com

`EasyLib/.github/workflows/node.js.yml`

```yaml
...
jobs:
  test:
    ...
  deploy:
    name: Wait for Deploy
    runs-on: ubuntu-latest
    needs: test
    steps:
      - name: Wait for Render Deployment
        uses: bounceapp/render-action@0.6.0
        # https://github.com/marketplace/actions/render-github-action
        with:
          render-token: ${{ secrets.RENDER_TOKEN }}
          github-token: ${{ secrets.GITHUB_TOKEN }}
          service-id: ${{ secrets.RENDER_ID }}
          # srv-xxxxxxxxxxxxxxxxxxxx
          retries: 20
          wait: 16000
          sleep: 30000
```

Questions?

marco.robol@unitn.it

`.gitignore` - Ignoring files from git versioning

- You can start from generic `.gitignore` file generated on www.gitignore.io, such as, https://www.gitignore.io/api/node,windows,linux,visualstudiocode
- **Make sure to always ingore**: `node_modules` `coverage` `.env`
- Put the `.gitignore` file itself under version control `git add .gitignore`