**📁 .github/workflows/**

This folder contains **GitHub Actions workflows** — YAML files that define how your project should automatically respond to events like code pushes, pull requests, or scheduled tasks.

Each .yml file inside this folder defines **a specific job or pipeline**, such as:

* 🧪 Run tests
* 🧹 Lint code
* 🚀 Deploy to staging/production
* 🐳 Build and push Docker images
* 🔁 Schedule cron jobs
* 📦 Publish npm or Python packages

**🧱 Basic Folder Structure**

bash

CopyEdit

.github/

└── workflows/

├── ci.yml # Build, Test, Lint

├── deploy.yml # Deploy to staging/production

├── docker-build.yml # Build and push Docker images

├── cron-maintenance.yml # Scheduled cleanup jobs

Now, let’s break down **one by one** with **detailed configuration + explanation**.

Great! Let's generate **real-world ci.yml and deploy.yml** files specifically tailored for a **NestJS project**.

1. **✅ ci.yml – Continuous Integration for NestJS**

This workflow will:

* Run on push or pull\_request to main or develop
* Install dependencies
* Lint the codebase
* Run tests (Jest)
* Optionally build the project

**📄 File: .github/workflows/ci.yml**

name: 🧪 NestJS CI

on:

push:

branches: [main, develop]

pull\_request:

branches: [main, develop]

jobs:

build-test:

name: 🧰 Lint, Test, Build

runs-on: ubuntu-latest

steps:

# Checkout repo code

- name: ⬇️ Checkout repository

uses: actions/checkout@v4

# Setup Node.js

- name: ⚙️ Setup Node.js

uses: actions/setup-node@v4

with:

node-version: 18

cache: 'npm'

# Install dependencies

- name: 📦 Install dependencies

run: npm ci

# Run ESLint

- name: 🧹 Lint code

run: npm run lint

# Run unit tests with coverage

- name: 🧪 Run tests

run: npm run test:cov

# Build project (optional but recommended)

- name: 🏗️ Build project

run: npm run build

1. **✅ deploy.yml – Deploy to Staging/Production (via SSH)**

This workflow deploys to a remote server using SSH and rsync.

**📄 File: .github/workflows/deploy.yml**

name: 🚀 NestJS Deploy to Production

on:

push:

branches:

- main

jobs:

deploy:

name: 📤 Deploy to Server

runs-on: ubuntu-latest

steps:

- name: ⬇️ Checkout code

uses: actions/checkout@v4

- name: ⚙️ Setup Node.js

uses: actions/setup-node@v4

with:

node-version: 18

# Install production dependencies

- name: 📦 Install dependencies

run: npm ci

# Build the NestJS app

- name: 🛠️ Build NestJS app

run: npm run build

# Setup SSH key (saved in GitHub Secrets)

- name: 🔐 Set up SSH connection

uses: webfactory/ssh-agent@v0.8.0

with:

ssh-private-key: ${{ secrets.SSH\_PRIVATE\_KEY }}

# Deploy using rsync to remote server

- name: 📂 Deploy to Remote Server

run: |

rsync -avz --delete ./dist/ user@your-server:/var/www/your-nestjs-app

# Optional: Restart server process (e.g., with PM2)

- name: 🔄 Restart App with PM2

run: |

ssh user@your-server "cd /var/www/your-nestjs-app && pm2 restart all"

**🔐 Required GitHub Secrets**

You must define the following in your GitHub repo under:

**Settings > Secrets and Variables > Actions**

| **Name** | **Description** |
| --- | --- |
| SSH\_PRIVATE\_KEY | Your server's SSH private key (for rsync/ssh) |
| *(optional)* DOCKER\_USERNAME, DOCKER\_PASSWORD | If using DockerHub |

**✅ 3. docker-build.yml – Docker Image Build & Push**

**Purpose:** Build a Docker image and push it to DockerHub or GitHub Container Registry.

**📘 DockerHub Example:**

yaml

CopyEdit

name: 🐳 Docker Build & Push

on:

push:

branches: [main]

jobs:

docker:

runs-on: ubuntu-latest

steps:

- name: ⬇️ Checkout code

uses: actions/checkout@v4

- name: 🔐 Login to DockerHub

uses: docker/login-action@v3

with:

username: ${{ secrets.DOCKER\_USERNAME }}

password: ${{ secrets.DOCKER\_PASSWORD }}

- name: 🛠️ Build Docker Image

run: docker build -t myapp:latest .

- name: 📤 Push to DockerHub

run: docker tag myapp:latest mydockerhubuser/myapp:latest && docker push mydockerhubuser/myapp:latest

**✅ 4. cron-maintenance.yml – Scheduled Tasks**

**Purpose:** Run scripts like DB cleanup, archiving, notifications, or health checks.

**📘 Example:**

yaml

CopyEdit

name: 🔁 Weekly Maintenance

# 👇 Trigger every Monday at 2:00 AM UTC

on:

schedule:

- cron: '0 2 \* \* 1'

jobs:

maintenance:

runs-on: ubuntu-latest

steps:

- name: ⬇️ Checkout code

uses: actions/checkout@v4

- name: 🧹 Run cleanup script

run: |

npm run cleanup:db

**🛠 Reusable Best Practices**

| **Tip** | **Description** |
| --- | --- |
| ✅ Use secrets | Never hard-code sensitive data |
| ✅ Use cache | Use actions/cache to speed up installs |
| ✅ Separate CI & Deploy | Clear separation makes debugging easier |
| ✅ Use matrix builds | Test across multiple versions (Node.js, Python, OS, etc.) |
| ✅ Use status badges | Add them in your README.md |

**🧪 Status Badge Example for README**

md

CopyEdit

![CI Status](https://github.com/your-org/your-repo/actions/workflows/ci.yml/badge.svg)