

# کنترل‌های CI/CD در زمان Pull Request: تست، کد و شرایط دیپلوی

## مقدمه

یا درخواست ادغام، نقطه حیاتی در چرخه توسعه نرم‌افزار است که در آن تغییرات کد (Pull Request (PR معمولاً) پیشنهادی توسط توسعه‌دهندگان بازبینی، تست و تایید می‌شوند قبل از اینکه به شاخه اصلی کنترل‌های قوی و خودکاری، `aladdin-sandbox` پیاده‌سازی شده برای مونوریپو CI/CD ادغام شوند. سیستم اعمال می‌کند تا از کیفیت، پایداری و امنیت کد اطمینان حاصل شود. این گزارش به تفصیل به PR را در زمان نحوه مدیریت این کنترل‌ها می‌پردازد.

## 1. فعال‌سازی Workflow ها در زمان Pull Request

به گونه‌ای پیکربندی شده‌اند که به طور خودکار در زمان ایجاد یا به‌روزرسانی GitHub Actions Workflows در فایل‌های `on: pull_request` فعال شوند. این فعال‌سازی از طریق تریگر Pull Request یک `.github/workflows/*.yaml` انجام می‌شود:

YAML

```
on:
  push:
    branches:
      - main
    paths:
      - 'apps/backend/**'
      - 'infra/ci-cd/**'
      - 'config/projects.yaml'
  pull_request:
    branches:
      - main
    paths:
      - 'apps/backend/**'
      - 'infra/ci-cd/**'
      - 'config/projects.yaml'
  workflow_dispatch:
    inputs:
      project:
        description: 'Project to run CI/CD for'
        required: true
        default: 'backend'
      deploy_env:
```

```
description: 'Deployment environment (Test, Staging, Production)'
required: false
```

- **تریگر pull\_request**: این تریگر تضمین می‌کند که هر زمان یک PR جدید ایجاد یا به‌روزرسانی شود (مثلاً با `push` کردن تغییرات جدید به برنج PR)، Workflow مربوطه فعال می‌شود.
- **فیلتر branches**: با تعیین Workflow، `branches: - main` تنها برای PRهایی که قصد ادغام در شاخه `main` را دارند، اجرا می‌شود. این امر از اجرای غیرضروری Workflowها برای PRهای داخلی یا موقت جلوگیری می‌کند.
- **فیلتر paths**: همانند تریگر `push`، فیلتر `paths` تضمین می‌کند که Workflow تنها در صورتی اجرا شود که تغییرات در مسیرهای مربوط به پروژه خاصی که Workflow برای آن تعریف شده است (مثلاً `**/apps/backend`) رخ داده باشد. این بهینه‌سازی منابع را حفظ کرده و زمان بازخورد را کاهش می‌دهد.

## 2. کنترل‌های کد و تست در زمان Pull Request

هنگامی که یک Workflow در زمان PR فعال می‌شود، مجموعه‌ای از کنترل‌های خودکار بر روی کد پیشنهادی اعمال می‌شود:

- **بیلد خودکار (Automated Build)**: اولین گام، بیلد کردن کد پروژه است. این مرحله تضمین می‌کند که تغییرات جدید، فرآیند بیلد را خراب نمی‌کنند و پروژه قابل کامپایل/بسته‌بندی است. هرگونه خطای بیلد به سرعت شناسایی و به توسعه‌دهنده گزارش می‌شود.
- **تست‌های خودکار (Automated Tests)**: پس از بیلد موفقیت‌آمیز، تمامی تست‌های خودکار (واحد، یکپارچه‌سازی) مربوط به پروژه اجرا می‌شوند. این تست‌ها صحت عملکرد کد را تایید کرده و از عدم ایجاد رگرسیون (Regression) اطمینان حاصل می‌کنند. نتایج تست‌ها مستقیماً در صفحه PR نمایش داده می‌شوند.
- **تحلیل کیفیت کد (Code Quality Analysis - در آینده)**: در آینده، می‌توان ابزارهای تحلیل کیفیت کد (مانند Linters, Formatters, Static Code Analyzers) را به این مرحله اضافه کرد. این ابزارها می‌توانند استانداردهای کدنویسی را بررسی کرده، پیچیدگی کد را اندازه‌گیری کنند و آسیب‌پذیری‌های احتمالی را قبل از ادغام شناسایی کنند.
- **اسکن امنیتی (Security Scanning - در آینده)**: ادغام ابزارهای اسکن امنیتی (مانند SAST) برای تحلیل کد منبع یا اسکنرهای وابستگی) در این مرحله می‌تواند آسیب‌پذیری‌های امنیتی را در کد پیشنهادی شناسایی کند.

## 3. بازبینی کد انسانی (Human Code Review)

علاوه بر کنترل‌های خودکار، بازبینی کد انسانی یک جزء حیاتی از فرآیند PR است:

- **تایید توسط همکاران**: توسعه‌دهندگان دیگر کد پیشنهادی را بازبینی می‌کنند تا از کیفیت، خوانایی، رعایت بهترین شیوه‌ها و مطابقت با الزامات کسب و کار اطمینان حاصل کنند. این بازبینی‌ها می‌توانند به شناسایی اشکالاتی که تست‌های خودکار ممکن است از دست بدهند، کمک کنند.

- بررسی نتایج CI/CD: بازیکنندگان کد می‌توانند نتایج اجرای Workflow های CI/CD را مستقیماً در صفحه PR مشاهده کنند. این شامل وضعیت بیلد، نتایج تست و هرگونه هشدار یا خطای تولید شده توسط ابزارهای تحلیل است.

## 4. شرایط دیپلوی و پیش‌نمایش (Deployment Conditions and Previews)

- در زمان PR، معمولاً دیپلوی به محیط پروداکشن انجام نمی‌شود. با این حال، می‌توان شرایطی را برای دیپلوی‌های موقت یا پیش‌نمایش تنظیم کرد:
- **دیپلوی به محیط پیش‌نمایش (Preview Environment):** برای برخی پروژه‌ها (به ویژه فرانت‌اند)، می‌توان یک مرحله دیپلوی خودکار به یک محیط پیش‌نمایش موقت (Ephemeral Preview Environment) را در Workflow PR پیکربندی کرد. این محیط به بازیکنندگان کد و ذینفعان اجازه می‌دهد تا تغییرات را در یک محیط زنده و ایزوله قبل از ادغام مشاهده کنند.
- **عدم دیپلوی به Staging/Production:** Workflow های مربوط به دیپلوی به محیط‌های Staging و Production معمولاً تنها با تریگر push به شاخه main و پس از تایید دستی فعال می‌شوند، نه در زمان PR. این امر از استقرارهای ناخواسته یا بدون بازبینی به محیط‌های حساس جلوگیری می‌کند.

## 5. وضعیت‌های PR و محافظت از شاخه (PR Statuses and Branch Protection)

Pull Request به (Status Checks) ها را به عنوان وضعیت Workflow نتایج اجرای GitHub Actions استفاده شوند main گزارش می‌دهد. این وضعیت‌ها می‌توانند برای محافظت از شاخه:

- **قوانین محافظت از شاخه (Branch Protection Rules):** در تنظیمات مخزن GitHub، می‌توان قوانینی را برای شاخه main تعریف کرد که ادغام PR را مشروط به موارد زیر کند:
  - **تایید موفقیت‌آمیز وضعیت‌ها:** تمامی وضعیت‌های CI/CD (بیلد، تست، تحلیل کد) باید با موفقیت انجام شده باشند.
  - **تایید بازبینی کد:** حداقل تعداد مشخصی از بازیکنندگان کد باید PR را تایید کرده باشند.
  - **عدم وجود تغییرات در حال بررسی:** PR نباید دارای تغییراتی باشد که هنوز توسط CI/CD بررسی نشده‌اند.
- این قوانین تضمین می‌کنند که تنها کدهای با کیفیت و تست شده می‌توانند به شاخه اصلی ادغام شوند، که به حفظ پایداری و امنیت کلی سیستم کمک می‌کند.

## نتیجه‌گیری

سیستم CI/CD پیاده‌سازی شده، با فعال‌سازی خودکار Workflow ها در زمان Pull Request، اجرای کنترل‌های جامع کد و تست، تسهیل بازبینی کد انسانی، و اعمال قوانین محافظت از شاخه، یک فرآیند قوی و ایمن برای ادغام تغییرات کد فراهم می‌کند. این رویکرد به طور قابل توجهی ریسک‌های مرتبط با معرفی

اشکالات یا آسیب‌پذیری‌ها به شاخه اصلی را کاهش داده و به تیم توسعه اطمینان می‌دهد که تنها کدهای با کیفیت بالا به پروداکشن راه پیدا می‌کنند.