

Labb: Workflow Dispatch, If-statser och Needs i GitHub Actions

Syfte

Den här labben tränar dig i tre viktiga delar av CI/CD med GitHub Actions:

1. Manuell start av workflows med parametrar (workflow_dispatch)
2. Villkorsstyrd körning av steg (if)
3. Beroenden och flödeskontroll mellan jobb (needs)

Efter labben ska du förstå hur du kan bygga mer flexibla pipelines som reagerar på input och styrs beroende på resultat.

Förutsättningar

- Du har redan ett repository med en fungerande GitHub Actions-konfiguration.
- Du har testat att skapa workflows tidigare.
- Du känner till YAML-grunder.

Uppgift 1 – Workflow Dispatch Input

Mål

Lära dig starta ett workflow manuellt och läsa in användarens input.

Instruktioner

1. Lägg till en workflow_dispatch-trigger som tar in minst en input (till exempel sträng).
2. Lägg till ett jobb som kör ett skript som visar upp värdet från inputen.

Tips:

Du kan komma åt inputen via `${{ github.event.inputs.<namn> }}`.

Reflektion

- Vad händer om du inte anger något värde?
- Hur kan du lägga till fler inputs av olika typer (till exempel choice, boolean)?

Uppgift 2 – If-statser

Mål

Förstå hur du kan använda if:-satser för att styra vilka steg som körs.

Instruktioner

1. Utgå från workflowet i uppgift 1.
2. Lägg till två steg:
 - Ett som bara körs om variabel == ett bestämt värde
 - Ett som bara körs om variabel == något annat
3. Kör workflowet flera gånger med olika val och observera skillnaden i loggarna.
- 4.

Tips:

Villkoret skrivs som if: `${{ github.event.inputs.variabel == 'text' }}`.

Reflektion

- Hur används if: på jobb-nivå jämfört med steg-nivå?
 - Vad händer om inget villkor uppfylls?
-

Uppgift 3 – Needs och flödeskontroll

Mål

Förstå hur man skapar beroenden mellan jobb med needs, och hur if påverkar flödet.

Instruktioner

1. Skapa tre jobb:
 - test
 - build
 - deploy
2. Använd needs: för att styra att:
 - build bara körs efter test
 - deploy bara körs efter build
3. Lägg till en input som styr om tester ska köras eller inte (till exempel run_tests: true/false).
4. Använd if: för att hoppa över test-jobbet vid behov.
5. Kör workflowet både med och utan tester och se hur flödet påverkas.
- 6.

Tips:

- Fundera på vad som händer med needs: när ett jobb hoppar över ett tidigare steg.
- Testa att lägga till if: always() på ett steg som du vill ska köras oavsett fel.
-

Reflektion

- Hur kan du använda needs: för att stoppa ett workflow vid fel?
 - Hur kan du skapa ett jobb som alltid körs sist (till exempel för notifiering)?
-

Tips:

Kika på if: always() och needs: [test, build, deploy].

Inlämning

Lämna in:

- Din workflow-fil i PM
 - Hur du använde workflow_dispatch, if och needs
 - Vilka problem du stötte på