

## Projektuppgift: DevOps Pipeline

**Examinationsbeskrivning:** Ni ska implementera en DevOps-pipeline som hanterar en API-baserad applikation. Du ska sedan individuellt skapa en pipeline från push till er personliga branch till deploy med kvalitetssäkring längs vägen. Slutligen lämnar du in en länk till din branch, länk till deployad app och rapport på genomfört arbete och teoretiska bitarna inom DevOps.

### Förutsätter:

Examinationsuppgiften förutsätter att du har varit delaktig i utvecklingen av en gemensam (gruppvis) webapp enligt de kriterier som återfinns i *Projektuppgift\_Gemensam*.

### Individuellt arbete: DevOps-pipeline:

Du ska skapa din egen pipeline för att automatisera testning, byggande och distribution av applikationen.

1. Pipelinen ska bestå av din **egna** workflow på din egen branch/fork av din grups webapp. Denna pipeline startar vid push till din branch hela vägen till deploy. Det ska vara tydliga namngivningar för bland annat workflows, jobs och steps.
2. Användning av environment-variabler, GitHub-kontextvariabler och GitHub Secrets (till exempel, API-nycklar och Docker-credentials).
3. Logik och beroenden (användning av needs och if-satser) ska användas i pipelinen.
4. Pipelinen ska kunna triggas manuellt med workflow\_dispatch och erbjuda en parameter för att logga fel:
  - a. Om parametern är satt till true och testerna misslyckas, ska en logg-error artefakt genereras.
  - b. Om parametern är satt till false, skapas ingen loggfil, oavsett testresultat.
5. När alla tester är godkända ska pipelinen:
  - a. Skapa en Docker-image av applikationen.
  - b. Publicera Docker-Imagen på DockerHub som är kopplad till en Azure-app service (med användning av Azure-credentials).

*Detta projekt kräver ingen specifik teknikstack utan kan implementeras med verktyg och språk du själv väljer. Dock ska arbetet följa DevOps-principerna som introducerats i kursen.*

### Rapport och Självreflektion:

Efter att ha genomfört projektet ska varje student skriva en individuell rapport som beskriver sitt arbete. Rapporten ska vara strukturerad enligt följande:

1. **Beskrivning av din egen pipeline:**
  - a. Redogör för de jobb och steps som ingår i din pipeline.
  - b. Motivera dina designval.

- c. Dokumentera vilka fel du upptäckte i under arbetet kopplat till pipelinen och beskriv din felsökningsprocess.
2. **Självreflektion:**
- a. Reflektera över din egen insats: utmaningar du mötte, kunskaper du fått och vad du anser är viktigt för en effektiv DevOps-pipeline.
  - b. Redogör för dina lärdomar kring DevOps och pipelines från kursens gång. Punkter som kan vara bra att ha med är DevOps bakgrund, principer, roller, termer med mera.

*Rapporten får vara så lång som du bedömer är nödvändigt för att redovisa kunskaperna men max 2 sidor.*

## Bedömningskriterier

1. **Tekniska krav (G):**
  - Alla krav på projektets applikation och pipeline är uppfyllda.
  - Pipelinen kommer att testas och ska fungera med korrekt konfiguration av secrets och beroenden. Detta innebär att ni behöver bjuda in mig som kollaboratör till repot.
2. **Rapport och reflektion (G/VG):**
  - **G:** Rapporten är strukturerad, visar förståelse för kursmålen och redovisar en genomförd pipeline och felsökningsprocess.
  - **VG:** Rapporten innehåller en djupare reflektion över lärdomar, med insiktsfulla resonemang om hur DevOps-principer appliceras.

**För VG krävs G på punkt 1 och VG på punkt 2.**

**Notera: Sen inlämning begränsar möjligheten att få ett högre betyg.**

---

## Inlämning och deadline

- Deadline för inlämning: **16/11, kl 23:55** på LearnPoint under Examination: Projektuppgift & Individuell rapport.
- Det ska tydligt framgå i GitHub-repot vems branch samt workflow tillhör vem. Ett förslag till nomenklatur är initialer och gruppnummer. Till exempel MG10.
- Ni lämnar in: En länk till er branch med fungerande workflow, en länk till er deployade webapp och en rapport (PDF).