

# LabAssignment3-Task3

## Continuous Integration/Delivery

Kontinuerlig integration (CI) och kontinuerlig leverans (CD), är en insamling av metoder som gör det möjligt för utvecklare att leverera kodändringar oftare och mer pålitligt för automatisering av olika steg. Denna implementering kallas för CI / CD-pipeline och är en stor del i DevOps.

### Kontinuerlig integration

Det tekniska målet med kontinuerlig integration är att skapa ett konsekvent och automatiserat sätt att bygga, paketera och testa applikationer. Med hjälp av detta så är målet att utvecklare ska "commita" kod oftare, vilket leder till bättre samarbete, mjukvara och möjligheten att eliminera risker tidigt. Fördelen med att integrerar kod på en daglig basis är om t.ex fel uppstår så blir det inte ett stort projekt att gå tillbaka och se vad som gått fel.

### Kontinuerlig leverans

Bygger vidare på kontinuerlig integration och automatiserar byggprocessen/driftsättning samt olika tester beroende på projekt/applikation. Ofta sätter man applikationen på en dev/test eller staging miljö för att testa den. Om alla tester avklaras så skapas en uppgift som manuellt behöver genomföras/bekräftas för att sätta applikationen in produktionsmiljön.

### Kontinuerlig deployment

Här tar vi steget ännu längre och automatiserar även själva release/deploy steget utan någon manuell hantering eller manuell bekräftelse för produktions släpp. Här kan man då också implementera en så kallad "Canary deployment" som gör att bara ett visst antal av besökare/kunder får den nya produkten/applikationen t.ex bara 1-5%, detta för att ytterligare kolla så att allt går som planerat och att inget oväntat uppstår i produktionsmiljön som ej upptäckts i testerna innan och för att förhindra att hela produktionsmiljön går ner om något går fel.

Några verktyg som kan användas inom CI/CD.

### **Jenkins**

Ett av de äldre inom området, gratis och open source. Kan sättas upp som en egen server eller köras i en container. Har många fördelar genom sina plugins.

#### Fördelar

- Gratis
- Full kontroll
- Plugins

#### Nackdelar

- Konfiguration och uppsättning kan ta tid
- En eller fler dedikerade servrar kan behövas

### **TravisCI**

En ci/cd server som är hostad över webben, används t.ex av Github.

#### Fördelar

- Gratis för open source projekt
- Snabb att börja använda
- Ingen dedikerad server behövs

#### Nackdelar

- Begränsningar inom konfiguration

### **TeamCity**

En produkt som ägs av JetBrains, likt Jenkins men har en enklare installations och konfigurations process.

#### Fördelar

- Snabb installation och uppsättning
- Många inbyggda funktioner
- Simpelt och lätt gränssnitt

#### Nackdelar

- Ingen gratis enterprise lösning
- Inte lika stort community som Jenkins har