

## Skapa en CI/CD Pipeline med GitHub Actions och AWS

### INLEDNING

#### Bakgrundsbeskrivning

Uppgiften går ut på att ni skall skapa en Maven Spring Boot-applikation och skall implementera en Continuous Integration/Continuous Deployment (CI/CD) pipeline på AWS. Skall också inkludera en Build process på GitHub Actions för test. Målet är att automatisera bygg-, test- och deployprocessen för att effektivisera utvecklingsflödet. Den resulterande webservice-applikationen skall hostas på AWS och innehålla funktionalitet för att svara på olika requests.

#### Varför ska ni utföra detta arbete?

Syftet med uppgiften är att sätta upp en CI/CD-pipeline för att säkerställa en stabil och snabb utvecklingsprocess. Att också lära sig om automatisk testning och distribution av kodändringar, vilket minskar risken för fel och förbättrar kvaliteten på koden som släpps till produktion.

#### Vad ska ni leverera?

Ni förväntas leverera en fullständig Maven Spring Boot-applikation med en fungerande CI/CD-pipeline på AWS. Skall också innehålla en Build process på GitHub Actions. Detta skall ligga i en eller flera .yaml filer i projektet. Syftet är att utföra tester på projektet via JUnit tester. Den färdiga applikationen ska hostas på AWS och kunna hantera olika typer av anrop. Där skall er service självklart kunna hantera anrop från klient eller Postman. För VG och CRUD skall en klient som anropar servicen användas som kan hantera användarens anrop och sedan visa upp responsen. Denna klient kan antingen vara via konsol eller ett GUI. Denna klient behöver INTE ligga online på AWS. Eftersom att både servicen och Databas ligger på en hostad service skall information om dessa, samt endpoints, finnas med i ReadMe-filen.

### ER PROJEKTUPPGIFT

#### Vad ska ni göra?

Ni skall utföra följande:

1. Konfigurera en CI/CD-pipeline på AWS för ett Maven-projektet.
2. Skapa en Build process på GitHub Actions. Använd en Workflow .yaml filer för detta i projektet
3. Implementera web service-funktionalitet i Spring Boot-applikationen.
4. Skriva ett antal JUnit tester i applikationen. Dessa skall köras mot Service klassen eller Controller klassen.
5. Skapa nödvändiga AWS-resurser för att hosta applikationen.
6. Implementera anropshantering för att kunna svara på requests från exempelvis Postman.
7. Skriv dokumentation i ReadMe filen.
8. (För VG) Skapa en klient applikation för att kommunicera med er hostade Service

Hur löser ni uppgiften?

1. Definiera steg för att bygga, testa och distribuera Maven-projektet i AWS.
2. Implementera nödvändig kod i Spring Boot-applikationen för att skapa en fungerande webservice.
3. Skapa en AWS-infrastruktur, inklusive instanser och nödvändiga tjänster för att stödja applikationen.
4. Testa web service-anrop med verktyg som Postman för att säkerställa korrekt funktionalitet.

Använd er av tidigare arbeten som referens och ta hjälp av varandra vid behov. Ta också hjälp via handläggningstider under de utsatta tider då ni har eget arbete under schemalagd lektionstid.

Struktur för arbetet

Vid grupparbete: arbetsfördelning och tidsplan

Uppgiften är en Individuell uppgift

## INLÄMNING OCH REDOVISNING

Inlämning

Projektet skall vara klart och färdigställt för bedömning senast Söndagen den 25 Februari

Redovisning

Uppgiften skall ej redovisas inför klass

## BEDÖMNING OCH ÅTERKOPPLING

Bedömning sker med följande betygsriterier

**För godkänt (G) på projektarbetet skall följande krav uppfyllas:**

- Ett uppbyggt SpringBoot applikation webservice som inkluderar en Workflow fil för GitHub Actions.
- Servicen skall kunna ta emot requests och skicka response tillbaka, med ett enkelt meddelande som response body.
- Deployment kedjan, dvs CI/CD skall vara automatisk. Från det att du pushar upp en ny version på GitHub skall den nya version automatiskt hostas på AWS efter det att alla steg i (AWS) Pipelinen är färdig.
- Dokumentation i ReadMe.
- En beskrivning av AWS Pipeline processen i dokumentationen. Även en beskrivning av den process du har använt dig av, om du har använt en annan process. Kan också inkludera screenshots eller en inspelning av processen och förklarande beskrivning i ReadMe filen.

**För väl godkänt (VG) på projektarbetet skall dessutom följande krav uppfyllas:**

Utöver kraven på Godkänt resultat skall även följande uppfyllas:

- Web Servicen skall inkludera full CRUD funktionalitet mot en online-liggande databas. Ni kan välja fritt vilken DB tjänst ni använder.
- En klient-applikation skall användas för att anropa servicen, med användarens request efter det som användaren

efterfrågar och visa upp servicens respons med användarens efterfrågade data.

- Denna klient behöver inte ligga online
- Klient-applikationen kan antingen vara i Konsol eller valfritt GUI

Återkoppling

2 veckor efter inlämning