AUtomation & DevOPS

## BY Kristofer Fangrat

### DATA ENGINEER STOCKHOLMS TEKNISKA INSTITUT

### 2024-11-16

1. Teoretisk bakgrund
2. Beskriv vad DevOps innebär och varför DevOps-principer är viktiga.

DevOps är en uppsättning metoder och verktyg som syftar till att integrera och automatisera arbetet mellan mjukvaruutveckling och IT-drift.  
DevOps-principer är viktiga eftersom de främjar snabbare leverans av mjukvara, högre kvalitet och bättre samarbete mellan team.

1. Förklara pipelines och dess roll i automatiserad systemutveckling.

En pipeline är en serie automatiserade steg som kod går igenom från utveckling till produktion. Pipelines spelar en central roll i automatiserad systemutveckling genom att säkerställa kontinuerlig integration (CI) och kontinuerlig leverans (CD), vilket minskar risken för fel och påskyndar leveransprocessen.

1. Beskriv vanliga termer inom DevOps, samt olika roller och ansvarsområden.

* CI/CD: Kontinuerlig integration och kontinuerlig leverans.
* Infrastructure as Code (IaC): Hantering av infrastruktur genom kod.
* Monitoring: Övervakning av systemets prestanda och tillgänglighet.
* DevOps Engineer: Ansvarar för att implementera och underhålla DevOps-processer.
* SRE (Site Reliability Engingeer): Fokuserar på att upprätthålla systemets tillförlitlighet och prestanda.

1. Beskrivning av min pipeline
2. Redogör för dem jobb och steps som ingår i din pipeline.

Min pipeline består av följande jobb och steg:

* Build-and-test: Klonar repository, installerar beroenden, kör tester och loggar eventuella fel.
* Build-and-push: Bygger Docker-image, taggar och pushar den till DockerHub.
* Deploy: Distribuerar Docker-Image till Google Cloud Run.

1. Motivera dina designval.

Jag valde att dela upp pieplinen i separata jobb för att säkerställa att varje steg är isolerat och kan felsökas individuellt. Att använda Docker för att paketera applikationen gör det enkelt att distribuera till olika miljöer.

1. Beskrivning av min pipeline.
2. Dokumentera vilka fel du upptäckte i den existerande pipelinen och beskriv felsökningsprocessen.

Jag upptäckte att applikationen inte lyssnade på rätt port, vilket orsakade att den inte startade korrekt på Google Cloud Run. Jag felsökte genom att granska loggarna och verifiera konfigurationen i Dockerfile och Streamlit-applikationen.

1. Ge åtgärdsförslag och motivera dem.
2. Jag föreslår att lägga till kontroller för None-värden i applikationen för att förhindra TypeError. Dessutom bör Dockerfile och pipeline konfigureras för att säkerställa att applikationen lyssnar på rätt port.
3. Självreflektion
4. Reflektera över din egen instats i projektet: Utmaningar jag mötte, kunskaper jag fått och vad jag anser är viktigt för en effektiv DevOps-pipeline.

Jag mötte utmaningar med att konfigurera pipelinen korrekt och hantera None-värden i applikationene. Jag lörde mig vikten av att ha robusta tester och att noggrant granska loggar för felsökning. En effektiv DevOps-pipeline bör vara automatiserad, pålitlig och enkel att underhålla.

1. Redgör för dina lärdomar kring DevOps och pipelines.

Jag har lärt mig att DevOps-principer och pipelines är avgörande för att säkerställa snabb och pålitlig leverans av mjukvara. Att ha en väl-konfigurerad pipeline minskar risken för fel och gör det enklare att distribuera och underhålla applikationer.