

Examensuppgift: CI/CD med Docker, Kubernetes och Ansible

Uppgiftsbeskrivning

1. Bygg en Docker-image

- Skapa en Docker-image som serverar Nginx på port 80 med en enkel egengjord `index.html` (det är denna filen du kan göra uppdateringar i för att enkelt se att din CI/CD fungerar).

2. Skapa en GitHub Actions-pipeline

- Bygg imagen och pusha den till GitHub Container Registry (GHCR).
- Se till att pipelinen bygger om imagen varje gång utan att använda cachelagrade lager.

3. Automatisera CI/CD med semantisk versionering

- Förbättra pipelinen så att den endast triggas när en ny annoterad git-tag pushas.
- Tagga imagen med samma versionsnummer som den pushade taggen.
- Notera att commits och taggar pushas separat (push commit först, sedan taggen).

4. Deploya till Kubernetes

- Skapa en Kubernetes-deployment som använder imagen från steg 1 och Inkludera en NodePort-service för att exponera applikationen.

(Testa deploymenten lokalt med `minikube service service_name`.)

5. Skapa en Ansible-playbook för rolling updates

- Skriv en ansible-playbook som:
 - i. uppdaterar Kubernetes-deploymenten till den senaste versionen av bilden.
 - ii. Validerar att deploymenten lyckades.
 - iii. Om deploymenten misslyckas, rullar tillbaka till den tidigare fungerande versionen.

6. Dokumentation och framtida utveckling

- Skapa en tydlig och välskriven **README.md** med instruktioner för hur man använder repositoryt.
- Lägg till GitHub Issues där du försöker motivera vad nästa steg hade varit i resan att förbättra CI/CD flödet genom att illustrera bottlenecks, risker och så vidare. Tänk utifrån ITIL 4-principer.

Slutresultatets förväntade filstruktur

```
|— .github
|   |— workflows
|       |— docker-image-build-pipeline.yml
|— Dockerfile
|— README.md
|— deploy-playbook.yml
|— index.html
|— nginx-deployment.yml
```

Bedömningskriterier

Godkänt (G):

- En fungerande CI/CD-lösning där examinator kan kлона repositoryt och:
 - Deploya Nginx med en NodePort-service till Minikube.
 - Uppdatera **index.html**, commita och pusha en annoterad semantisk versionstag.

- Pipeline bygger imagen, taggar den korrekt och pushar till GHCR.
- Man manuellt uppdaterar en `image version` variabel i Ansible-playbooken och kör playbooken.
- Playbooken uppdaterar deploymenten i klustret, validerar att uppdateringen lyckades eller rullar tillbaka vid fel.
- Examinatorn kan sedan testa att köra `'minikube service service-name'` och validera att ändringarna till `index.html` nu är automatiskt deployade till klustret.

Väl Godkänt (VG):

- Githistoriken är välbyggd med commits som följer best practice.
- Studenten har implementerat enkel automatisk testning av imagen innan den pushas till container-registret.
- Studenten visar kompetent användning av miljövariabler för att öka flexibiliteten och minska hårdkodning.

Instruktioner för att skicka in

Skapa ett repository på GitHub och se till att samtliga filer är inkluderade. Skicka in länken till repot tillsammans med en kort beskrivning av eventuella utmaningar du stött på.