

Blockchain or Bust- Kontinuasjonsoppgave PGR301 20204

Krav til leveransen

- Eksamensoppgaven, kode og nødvendige filer er tilgjengelig i GitHub-repo: <https://github.com/glennbechdevops/PGR301-konte-2024>
- Du skal ikke opprette en fork av dette repositoryet, Lag et nytt GitHub repository for din besvarelse
- Når du leverer inn oppgaven via WiseFlow, vennligst opprett et tekstdokument som kun inneholder en kobling til ditt repository.
- Vennligst bruk et tekstdokumentformat, ikke PDF, Word eller PowerPoint.
- Dersom du er bekymret for plagiat fra medstudenter, kan du arbeide i et privat repository og deretter gjøre det offentlig tilgjengelig like før innleveringsfristen.

Når oppgaven evalueres, vil han/hun:

- Se gjennom repositoryet ditt og gå til fanen "Actions" på GitHub for å bekrefte at Workflows faktisk fungerer som de skal.
- Vurdere drøftelsesoppgavene. Du må opprette en fil, README.md for besvarelsen i ditt repository.
- Sensoren vil lage en "fork" (en kopi) av ditt repository og deretter kjøre GitHub Actions Workflows med sin egen GitHub-bruker for å bekrefte at alt fungerer som forventet.

Om bruk av KI

- Det er viktig at du forstår all kode og alle deler av det du leverer. Dersom svar på en oppgave inneholder elementer som ikke er nødvendig for å løse oppgaven, vil dette medføre trekk av poeng.
- Det samme gjelder forøvrig copy/paste fra

Evaluering

- Oppgave 1. GitHub Actions workflow - 30 Poeng
- Oppgave 2. Terraform - 30 Poeng
- Oppgave 3. Docker hub og GitHub Actions - 40 Poeng

Oppgavebeskrivelse

Blockchain Busters (BCB) er et nystartet selskap innen kvantitativ trading av kryptovaluta. BCB vil analysere handler og markedssdata fra ulike kryptovalutabørser for å utnytte ulike signaler i markedet. De vil bruke programvare og AI for å gjennomføre høyfrekvent, automatisk trading.

Selskapet er nylig etablert, og din oppgave er å sørge for en god start på programvareutviklingen. Først og fremst ønsker selskapet å utforske de åpne APIene til den norske kryptovalutabørsen NBX.

Utviklingen er i en tidlig fase, og all koden BCB har, finner du i dette repositoryet:

For å sikre minimal forsinkelse og optimal ytelse, har CTO valgt å bruke programmeringsspråket Go for all utvikling. Som student ved Kristiania, har du kanskje ikke mye erfaring med dette språket - men som en dyktig DevOps-Ingeniør, tenker du at du kan "DevOpse" hva som helst så lenge løsningen er basert på containere!

- [NBXbook.go](#)
- [Dockerfile](#)

Oppgave 1 - GitHub Actions Workflow

Test Koden

Du kan teste koden med, ved hjelp av Docker, ved å kjøre kommandoene

```
docker build . -t nbx
docker run nbx
```

A. Det første steget er å implementere kontinuerlig integrasjon. Lag en GitHub Actions workflow som, ved hver push til en hvilken som helst branch, kompilerer koden og lager et containerimage ved hjelp av [Dockerfile](#) i prosjektet.

B. I PGR301 diskuterte vi en arbeidsflyt med beskyttede branches (f.eks., main), og hvordan man kan forhindre at en pull-request blir gjort mot hovedbranchen hvis koden ikke kompilerer, tester feiler, osv. Beskriv hvordan du konfigurerer dette i GitHub med ord eller skjermbilder.

C. Reflekter over fordeler og ulemper med en arbeidsflyt der minst to personer i et team må godkjenne endringer mot main branch.

Oppgave 2 - Terraform - Infrastruktur som Kode

- Opprett en konto på Docker Hub hvis du ikke allerede har det.

BCB ønsker å konfigurere alle repositoriene sine på Docker Hub ved hjelp av Terraform. De har funnet en Terraform provider som kan opprette Docker Hub repositories.

- <https://registry.terraform.io/providers/BarnabyShearer/dockerhub/latest/docs> Ifølge dokumentasjonen er dette ressursen man må bruke:

```
resource "dockerhub_repository" "project" {
  name          = "project-name"
  namespace     = "your-dockerhub-username"
  description   = "Project description"
}
```

- Hint: brukernavnet ditt på Docker Hub brukes for verdien [namespace](#)

A. Skriv Terraformkode som lager et Docker Hub repository som heter "nbx".

- Lag en egen mappe med navn "infra" i ditt repository
- Terraformkoden skal kreve Terraform versjon 1.6.4 eller høyere.
- Du trenger ikke tenke på Terraform backend-konfigurasjon, Terraformkoden skal ikke kjøre i en GitHub Actions workflow
- Docker-Hub brukernavn, og navn på repository skal ikke hardkodes. Du må skal bruke terraform variabler

B. Kjør Terraform apply minst en gang, slik at et repository som heter "nbx" blir opprettet. Slett filen "Terraform.tfstate" kjør Terraform apply en gang til. Forklar hvorfor du får en feilmelding, og hvordan du kan løse dette

Oppgave - Docker Hub push

- Oppdater din GitHub Actions workflow fra oppgave 1 slik at hver push til *kun main-branchen* bygger et containerimage og pusher container image til repository du lagde i oppgave 2.
- Ved hver push til en annen branch enn *main*, skal det fortsatt bygges et containerimage, men det skal altså ikke gjøres en push til Docker Hub.

Hvilke docker kommando kan sensor bruke for å kjøre ditt container image på sin maskin?