

Eksamensdato:	08.12.2022					
Varighet:	6 timer (9:00 – 15:00) + 15 min satt av til levering					
Fagnummer:	DRI113					
Fagnavn:	Infrastructure as Code					
Klasser:	Nettbasert					
Studiepoeng:	6 studiepoeng					
Faglige kontaktpersoner:	Tor Ivar Melling (tlf. 90 66 24 83) E-post: tor.i.melling@tisip.no					
Administrativ kontaktperson:	Ingrid Island (tlf. 92 24 22 01) E-post: ingrid.island@tisip.no					
Hjelpemidler:	Alle hjelpemidler.					
Oppgavesettet består av:	Oppgavesettet består av 5 sider (medregnet forsiden (s.1), informasjonsskriv(s.2)) og 3 oppgaver.					
VIKTIG:	<p>Les hele oppgavesettet og disponer tiden din på best mulig måte. Se på %-inndelingen av oppgavene og beregn forventet arbeidstid på hver oppgave. Dette legger også føring for hvor mye som det forventes at skal produserer under hver oppgave. Denne eksamensbesvarelsen skal løses individuelt og innholdet skal være et produkt av kandidaten som leverer eksamen.</p> <p>Besvarelsen skal leveres på It`s Learning senest kl. 13:15 (15 minutter etter eksamen er ferdig). For sen innlevering medfører stryk på eksamen. Send besvarelse på e-post innen 13:15 ved problemer med It`s Learning.</p>					

Som levering på denne eksamens skal det lastes opp et dokument i It's Learning. Formatet skal være et av formatene: **.doc .docx .pdf .odt**. Ta gjerne med print screen fra developer portalen der det gir mening og merverdi til besvarelsen, men det skal ikke være en veiviser med bilder for hver steg på veien for å fullføre ønsket konfigurasjon.

Eksamenstiden er 4 timer, og det er førende for hvor mye som forventes at skal skrives i en sånn eksamensbesvarelse. Det er lov å benytte seg av innhold i tidligere leverte øvinger i dette faget der hvor det skulle passe, men pass på at en besvarer eksamensoppgaven og det den spør etter.

MERK at ved hjemmeeksamen er alle hjelpemidler tillatt, men det er ikke lov å levere andres arbeid som sitt eget. Det gjelder også for bruk av kunstig intelligens. Dvs. en kan **ikke levere** kunstig intelligens-generert tekst, eksempelvis **tekst generert av ChatGPT**, som sin egen besvarelse. Benytt fagets pensum som utgangspunkt for besvarelsen, og om det blir brukt andre og eksterne kilder, må det oppgis sammen med den delen av besvarelsen. Hvis en blir tatt i juks på eksamen vil en ikke få godkjent eksamen.

Husk å lagre arbeidet fortløpende igjennom eksamen for å unngå å miste store deler av besvarelsen ved uforutsette hendelser. Benytt gjerne løsninger som både lagrer på lokal maskin og eventuelt hos en skytilbyder. Dette kan for eksempel være Word sammen med OneDrive, bruk av Google docs og Google Drive med fler.

1. OPPGAVE 1: TERRAFORM BASICS (20%)

1. Terraform init:
 - Hva er formålet med kommandoen terraform init, og når brukes i arbeidsflyten?
2. Terraform plan:
 - Beskriv hva kommandoen terraform plan gjør. Hvordan bidrar den til å forstå handlingene Terraform vil utføre?
3. Terraform fmt:
 - Forklar funksjonen til kommandoen terraform fmt. Hvorfor er den viktig for kodevedlikehold?
4. Terraform validate:
 - Hva er rollen til kommandoen terraform validate i Terraform-arbeidsflyten? Hvordan bidrar den til påliteligheten til koden?
5. Terraform apply:
 - Diskuter hvordan kommandoen terraform apply brukes. Hvilke forholdsregler bør tas før denne kommandoen utføres?
6. Terraform destroy:
 - Hva gjør kommandoen terraform destroy? I hvilke scenarier er denne kommandoen nødvendig, og hvilke tenke igjennom bør gjøres før den brukes?
7. Terraform output:
 - Beskriv formålet med kommandoen terraform output. Hvordan letter den fasilitering og synligheten av Terraforms output variabler?

2. OPPGAVE 2: FORBEDRE EKSEMPELKODEN(20%)

I eksemplet under opprettes en storage account ved bruk av Terraform i main.tf. Hva bør en gjøre av endringer å gjøre koden mer fleksibel og god? Sett # filnavn.tf i besvarelsen for å illustrere bruk av flere filer i konfigurasjonen din.

```
resource "azurerm_storage_account" "example" {  
  name           = "storageaccount"  
  resource_group_name = "example-resources"  
  location       = "northeurope"  
  account_tier   = "Standard"  
  account_replication_type = "LRS"  
}  
  
resource "azurerm_resource_group" "example" {  
  name     = "example-resources"  
  location = "northeurope"  
}
```

3. OPPGAVE 3: (60%)

Du jobber i et konsulentfirma og er tildelt oppgaven med å implementere en ende-til-ende (end-to-end) arbeidsflyt for Infrastructure as Code for håndtering og utplassering av skyinfrastruktur på Microsoft Azure ved hjelp av Terraform. Målet er å følge kjente praksiser og sikre en smidig, samarbeidsorientert utviklingsprosess. Ende-til-ende referer til en prosess der hele livssyklusen for infrastrukturforvaltning kan både styres og automatiseres gjennom kode. Gi en beskrivelse av hvilke trinn som er involvert i denne arbeidsflyten, og den må minst inneholde følgende:

- Beskriv hvordan du ville strukturere Terraform kodebase for dette prosjektet.
- Forklar konseptet med Terraform workspaces og hvordan du ville benytte dem i dette prosjektet. Gi eksempler på når og hvorfor du ville opprette forskjellige workspaces.
- Greit ut om hvordan du ville bruke Github for versjonskontroll og hvordan du ville sette opp branches for fremtidig utvikling av ulike deler av infrastrukturen.
- Beskriv hvordan du ville konfigurere GitHub Actions for å utføre testing når endringer blir pushet til forskjellige branches. Inkluder eksempler på typer tester du ville kjøre.
- Skisser prosessen med å opprette en pull request (PR) når en funksjon eller feilfix er fullført. Forklar hvordan kodegjennomgang ville bli utført og hvilke kriterier som må være oppfylt før en PR blir sammenslått inn i main branch.
- Hvordan ville du implementere en CI/CD-pipeline for å automatisere utrulling av infrastrukturendringer til Azure når endringer blir sammenslått inn i hovedgrenen. Inkluder eventuelle verktøy eller tjenester du ville bruke for dette formålet.

Lykke til med eksamen.