

# Deploy av java-applikasjon på ubuntu

Truls Thirud

## Spørsmål til DeepSeek

Hva er beste måte å deploye en java-applikasjon på en ubuntu-maskin? Den skal koble seg til en database på samme maskin, og har en embedded web-server. Og hvordan skal den best beskyttes mot f.eks. overbelastnings-angrep?

## Svar fra DeepSeek

Å deploye en Java-applikasjon på en Ubuntu-maskin som kobler seg til en database på samme maskin og har en embedded web-server, krever noen viktige steg for å sikre at applikasjonen kjører sikkert og effektivt. Her er en detaljert guide:

---

### 1. Forberedelser

- **Oppdater systemet:** Sørg for at Ubuntu-maskinen er oppdatert:  
`sudo apt update && sudo apt upgrade -y`
- **Installer Java:** Hvis Java ikke allerede er installert, installer den:  
`sudo apt install openjdk-17-jdk -y`
- **Installer database:** Hvis du bruker en database som MySQL eller PostgreSQL, installer og konfigurer den:  
`sudo apt install mysql-server -y`  
Konfigurer databasen og opprett en bruker og database for applikasjonen din.

---

### 2. Bygg Java-applikasjonen

- Bygg applikasjonen din til en kjørbart JAR-fil. Hvis du bruker Maven, kan du kjøre:  
`mvn clean package`
- Dette vil generere en JAR-fil i target-mappen.

---

### 3. Deploy applikasjonen

- Lag en dedikert bruker for applikasjonen (for bedre sikkerhet):  
`sudo adduser --system --no-create-home --group appuser`
- Lag en mappe for applikasjonen:

```
sudo mkdir /opt/myapp
sudo chown appuser:appuser /opt/myapp
```

- Kopier JAR-filen til mappen:

```
sudo cp target/myapp.jar /opt/myapp/
sudo chown appuser:appuser /opt/myapp/myapp.jar
```

---

#### 4. Konfigurer applikasjonen

- Opprett en konfigurasjonsfil for applikasjonen (f.eks. `application.properties` eller `application.yml`) og legg den i `/opt/myapp/`.
  - Konfigurer applikasjonen til å koble seg til databasen ved å angi riktig JDBC-URL, brukernavn og passord.
- 

#### 5. Opprett en systemd-tjeneste

- Opprett en systemd-tjenestefil for å kjøre applikasjonen som en tjeneste:

```
sudo nano /etc/systemd/system/myapp.service
```

- Legg til følgende innhold:

```
[Unit]
Description=My Java Application
After=syslog.target network.target

[Service]
User=appuser
Group=appuser
ExecStart=/usr/bin/java -jar /opt/myapp/myapp.jar
SuccessExitStatus=143
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

- Start og aktiver tjenesten:

```
sudo systemctl daemon-reload
sudo systemctl start myapp
sudo systemctl enable myapp
```

---

#### 6. Sikkerhetstiltak

- **Bruk en brannmur:** Konfigurer ufw (Uncomplicated Firewall) for å tillate kun nødvendige porter (f.eks. HTTP/HTTPS):

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

- **Begrens tilgang til databasen:** Konfigurer databasen til å kun tillate tilgang fra localhost eller applikasjonens bruker.
- **Bruk HTTPS:** Hvis applikasjonen din eksponerer en web-server, bruk HTTPS med et gyldig sertifikat (f.eks. fra Let's Encrypt).
- **Rate limiting:** Implementer rate limiting i applikasjonen for å forhindre overbelastningsangrep. Dette kan gjøres med biblioteker som **Bucket4j** eller via en reverse proxy som Nginx.
- **Reverse proxy:** Bruk en reverse proxy som Nginx for å håndtere trafikk og legge til ekstra sikkerhetslag:

```
sudo apt install nginx -y
```

Konfigurer Nginx til å videresende trafikk til applikasjonen din.

---

## 7. Overvåkning og vedlikehold

- **Loggføring:** Sørg for at applikasjonen logger til en dedikert loggmappe (f.eks. /var/log/myapp/).
  - **Overvåkning:** Bruk verktøy som **Prometheus** og **Grafana** for å overvåke applikasjonens ytelse og oppdage unormalt trafikkmønster.
  - **Oppdateringer:** Hold applikasjonen, databasen og operativsystemet oppdatert for å sikre mot sårbarheter.
- 

## 8. Beskyttelse mot overbelastningsangrep

- **Rate limiting:** Implementer rate limiting i applikasjonen eller via en reverse proxy som Nginx.
  - **Fail2ban:** Installer og konfigurer Fail2ban for å blokkere IP-adresser som prøver å overbelaste systemet:
- ```
sudo apt install fail2ban -y
```
- **Load balancer:** Hvis applikasjonen din får mye trafikk, vurder å bruke en load balancer for å distribuere belastningen.
- 

Ved å følge disse stegene vil du kunne deploye og sikre en Java-applikasjon på en Ubuntu-maskin, samt beskytte den mot overbelastningsangrep og andre trusler.