

# IDATT2104 - Arbeidskrav 2

Gruppe 18

April 16, 2022

# Contents

<b>1</b>	<b>Websocket</b>	<b>3</b>
1.1	Teori . . . . .	3
1.2	Praktisk . . . . .	3
<b>2</b>	<b>Bruk av SSH med CI/CD</b>	<b>3</b>
2.1	Enkel klient/tjener . . . . .	3

# 1 Websocket

## 1.1 Teori

Det er en del forskjeller fra HTTP og Websockets. Den største forskjellen går på hvordan en får tak i ny data. Dersom en ønsker å skaffe ny data etter hvert som ny data dukker opp, og man bruker HTTP, så er klienten nødt til å sende en ny forespørsel til tjeneren for hver gang den ønsker ny data. A.K.A Polling med intervall. Tjeneren vil sende en respons tilbake kun dersom en tjener spør om det. Websockets derimot, setter opp en to-veis tilkobling mellom tjeneren og klienten, som gjør at tjeneren og klienten kan få data etterhvert som ny data dukker opp. Det er event-basert, slik at når tjeneren mottar data fra en klient, så kan tjeneren sende et event til alle tilkoblede klienter som sier at ett eller annet har skjedd, med noe tilhørende data.

Websockets er veldig attraktive for sanntids-oppdateringer. Grunnen til dette, er at det tar en del mer ressurser dersom vi skal sette opp en helt ny TCP-tilkobling og handshakes for hver eneste gang vi vil ha data. Dersom vi ønsker å gjøre dette veldig ofte, så blir det mye overhead som vi ikke ønsker. Websockets fikser dette ved å ha litt overhead når vi setter opp en tilkobling og gjøre et Websocket-handshake, men når vi overfører data, så er meldingene mindre, og gjør at vi kan sende data langt oftere i praktiske tilfeller.

## 1.2 Praktisk

# 2 Bruk av SSH med CI/CD

## 2.1 Enkel klient/tjener