Uke 38 – Prosjekt: Nettverk, Webserver ,Python ,Git, Git-Hub og Raspberry Pi

**Arbeidslogg**

I denne uka fikk vi oppdrag og i oppdrag var det forkjellige ting, jeg var veldig stresse på starten med det gikk greit etter vært.

Oppgaven handler om Nettverk, Webserver, python, Git, Git -hub og mest Raspberry pi. Etter å se på oppgaven jeg var stresse om hvordan jeg skal gjøre det men uansatt jeg begynte. På Mandag hadde vi IM -timen bare fra 10:30 – 12:30 , det var ikke for mye tid men jeg begynte med å lage planen av hele uka slik at jeg vet hva jeg skal gjøre hverdag. Deretter begynte jeg med første del , Første del handler om « Nettverk og tilkoblig». I denne oppgaven skulle vi koble to datamaskiner sammen , første så jeg leste oppgaven veldig nøya for å forstå helt at hva skal gjøres. Deretter det jeg skjønte var å gi en «Statisk IP-Adressen til Raspberry pi, Da snakka jeg med simen om hvordan jeg skal gjøre det så fikk jeg litt veiledning, men uansatt så måtte jeg finne det ut selv at hvordan jeg skal gi «Statisk IP- Adressen til maskinen. Så,en statisk IP er jo en fast IP-adresse som aldri endres, i motsetning til en dynamisk IP som routeren gir automatisk.Da søkte jeg litt på google og youtube så .. Så , brukte jeg først komandoen «sudo nano /etc/dhcpcd.conf» for å og in i GNU 8.3 og der ga jeg til maskinen statisk ip , på sist jeg skrevde :  
interface wlan0  
static ip\_address=10.200.10.18  
static routers=10.200.10.1  
static domain\_name\_servers=8.8.8.8 8.8.4.4

Og sånn ga jeg ip adressen til maskinen der etter skrevde jeg koden «ip a» for å se at Det er nå på statisk IP , og det var. Då, det funka bra. Etter å gi IP til maskines gikk jeg til terminalen (CMD) i PC-en min og skrevde komadoen «ping 10.200.10.18» og fikk jeg reply. Så, jeg pinga mellom maskinene. Etter å ble ferdig med det første del (Nettverk og tilkobling) dokumenterte jeg på word, med skjermbilder og litt forklaringen.

Da på tirsdag begynte jeg med andre del «Server og tjenester». I denne delen måte vi lage en enkel webserver. Da begynte jeg med å lage en mappe som heter «Webserver». Inni denne mappen lagde jeg index fil som heter «index.html». Da brukte jeg komadoen «nano index.html» for å gå inni mappen. Da fikk jeg mulighet til å skrive koden der, og skrevde jeg:  
« <h1>Hei fra min Pi-server!</h1>

<p>Dette er min første nettside på Raspberry Pi.</p> »

Etter å skrive den her koden så lukka jeg index.html men «ctrl+x» og «y» for yes. Og da skrevde jeg komandoen «python3 -m http.server 8000» for å se om porten fungere. Da fikk jeg svar med «Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...» Det betyr at porten fungerer. Da prøvde jeg å åpne nettsiden på nettleseren ved å skrive «http:// <statisk ip>» men det fungerte ikke. Da, så jeg gjennom litt brannmuren, slå av brannmur med komandoen «sudo ufw disable» og prøvde igjen på nettleser men det funka ikke igjen. Så prøvde jeg « sudo ufw enable» for å slå på brannmuren igjen og så på status med «sudo ufw status». Da fikk jeg vite at jeg må «allow» porten. Fordi det var det som blokka nettsiden ved å komme på nettleseren. Fa brukte jeg komandoen «sudo ufw allow 8000» og prøvde igjen med bare å skrive ip addressen så funka nettsiden heldigvis.

Etter å ble ferdig med Del 2 begynte jeg med Del 3 « Python og git-hub».I denne delen skulle vi lage en enkel python script, pushe den på git-hub og klone den i raspberry pi terminalen. Da begynte jeg med å lage en enkel script på python og det var: « print(«Hello world!»)» . Etter å lage denn så pushe jeg denne script i Git-hub. I git.hub så lagde jeg en repository , og pusha koden der. I GitHub lagde jeg et repository for prosjektet mitt og lastet opp Python-scriptet ved å bruke «commit og sync» fra VS Code. Etterpå gikk jeg til Raspberry Pi, åpnet terminalen og brukte «git clone» for å hente scriptet fra GitHub til Pi-en. Første gang funket det ikke fordi jeg måtte lage en «personal access token» på GitHub for å logge inn via terminalen. Etter å ha laget tokenet, oppdaterte jeg Raspberry Pi og prøvde på nytt, og da funket det heldigvis.

Etter at scriptet var klonet, kjørte jeg det på Raspberry Pi for å sjekke at det funket. Scriptet viste “Hello world!” som forventet, og da visste jeg at koden min funket både på PC og Pi.

Etterpå lagde jeg et mer avansert Python-script som viste temperatur og hvor lenge systemet hadde vært på (uptime). Jeg brukte biblioteket «psutil», som hjelper med å hente systeminformasjon. Koden viste temperaturen (eller en tilfeldig temperatur hvis Pi-en ikke hadde sensor) og fortalte om det var en varm eller kald dag. Den viste også hvor lenge systemet hadde vært oppe i timer og minutter. Jeg testet scriptet på Raspberry Pi-terminalen, og det funket som det skulle.

Gjennom hele uka dokumenterte jeg alt jeg gjorde med **skjermbilder og forklaring**, og skrev det i Word for å kunne vise hva jeg hadde gjort og hva jeg lærte. Til slutt hadde jeg gjort alle delene av oppgaven: **nettverk og statisk IP**, **webserver**, og **Python + GitHub**, i tillegg til å lage et ekstra avansert script med systeminformasjon.

Dette prosjektet hjalp meg å lære mye om hvordan man setter opp nettverk, Raspberry Pi, webserver, Python-script og GitHub. Jeg lærte også å feilsøke problemer med brannmur, IP-adresse og GitHub-tilkobling, og fikk en bedre forståelse av hvordan maskiner kan snakke med hverandre på et lokalt nettverk.