

Oppskrift system

Dette er en stegvis oppskrift på hvordan man selv kan implementere dette systemet. Vi har delt inn i forskjellige kategorier. Vi forklarer ikke stegene, men vi skriver detaljerte steg.

Klargjøring av system

1. Laste ned Linux Ubuntu og flashe dette over på raspberry pi.
2. Modifisere oppstart protokoll slik at den nå automatisk uten skjerm, kobler seg til ønsket nettverk. Man kan også som vi har gjort, koble direkte til router med hjelp av ethernet kabel.
3. Klargjøring av printere og setup av videokameraer.
4. Koble alle nødvendige kabler til raspberry pi.
5. Laste ned "fing" og skanne lokalt nett etter ip-adresser og finne tilgjengelig raspberry pi.
6. Notere seg ip adresse til raspberry pi, og logge seg inn med brukernavn og passord.

Octoprint:

"Dersom du ikke kjører i root, må du bruke sudo kommando for å override"

1. Lete etter oppdateringer for OS, `sudo apt-get update`.
2. Laste ned relevante oppdateringer, `sudo apt-get upgrade`.
3. Laste ned relevante filer for at docker skal kjøres riktig, lim inn følgende kode;
`apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg2 \
software-properties-common`
4. Reboot av systemet, skriv `reboot` i cli.
5. Kjøre docker installasjons script, `curl -sSL https://get.docker.com | sh`.
6. Sjekk om docker er oppe og kjører, `sudo docker ps`.
7. installere Docker-compose, `sudo apt-get install docker-compose`.
8. lage en directory for octoprint, `mkdir octoprint`.
9. Opprette directory for printeren du anvender, `mkdir "xxx"`.
10. naviger til denne directory, `cd octoprint`, `cd "xxx"`.
11. Finne eksakt adresse til printer og kopiere denne, `pwd`
12. Identifiser dine/din printere/printere, `ls /dev/ttyU* || ls /dev/ttyA*` (her kan det variere etter hva printer man anvender, finn egen dokumentasjon på dette).
13. navigere tilbake til octoprint directory, `cd ..`.
14. Opprette `docker-compose.yml`, `nano docker-compose.yml`.

15. Lim inn docker compose eksemplet som følgende, modifier etter ditt eget printersystem:

```
version: '2.2'
```

```
services:
```

```
  octoprint_reprap:
```

```
    restart: unless-stopped
```

```
    image: octoprint/octoprint
```

```
    ports:
```

```
      - 4000:5000
```

```
    devices:
```

```
      - /dev/ttyUSB0:/dev/ttyACM0 (modifier her etter eget printersystem).
```

```
    volumes:
```

```
      - /root/octoprint/reprap:/home/octoprint (endre her til adresse gitt ved punkt 11).
```

16. Sjekk at docker-compose fungerer og last ned nødvendige filer for at octoprint skal kjøre;, sudo docker-compose up -d.
17. Sjekk at kontainere kjøres, sudo docker ps.
18. For å la vår printer directory mulighet til å skrive og lese uten feil, sudo chmod -R 777 "xxx".
19. Åpne nettleser, skriv inn lokal ip adresse og porten skrevet i docker-compose.yml. (eksempel 10.0.0.201:4000). Om alt er skrevet i henhold til oppskriften, skal du nå bli møtt med Octoprint sitt dashbord.
20. Åpne terminal for å identifisere kameraer, ls /dev vid*.
21. naviger tilbake til docker-compose.yml, og implementer koden gitt under for å koble til kamera;

```
mjpg-streamer:
```

```
  restart: unless-stopped
```

```
  image: openhorizon/mjpg-streamer-pi3
```

```
  # This image just takes the raw mjpg-streamer args -- set to 720p,30 fps
```

```
  command: ./mjpg_streamer -o "output_http.so -w ./www" -i "input_uvc.so -r 1280x720 -d /dev/video0 -f 30"
```

```
  devices:
```

```
    - /dev/video0:/dev/video0
```

```
  ports:
```

```
    - 8080:8080
```

22. Gjør endringer i forhold til dette, her kan det variere i forhold til hvilket system man anvender.
23. Gå til nettleser og sjekk at kameraet fungerer på porten skrevet i docker-compose.yml; (10.0.0.201:8080?action=stream
24. Naviger til octoprint (10.0.0.201:4000) og gå gjennom setup konfigureringen.
25. Implementer stream url til kameraet som kamera i octoprint, nå skal octoprint finne adressen og legge dette ved som overvåkning.

Implementering av caddy, sekundær raspberry pi.

“Ved tilkobling av sekundær Raspberry PI, følg klargjøring av system før kommende punkter. Når alt er koblet til og du finner IP adressen til den sekundære RaspBerry PI, er du klar til å følge stegene under”

1. Lete etter oppdateringer for OS, `sudo apt-get update`.
2. Last ned relevante oppdateringer, `sudo apt-get upgrade`.
3. Finn en stabil utgave, og installer det som passer ditt system, lim inn følgende kode:

```
sudo apt update
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo apt-key
add -
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo
tee /etc/apt/sources.list.d/caddy-stable.list
```

4. `sudo apt update`
5. `sudo apt install caddy`
6. For å validere installasjonen din; `$ caddy version`
7. For å sjekke caddy sin lokalisering; `$ which caddy`
8. For å opprette nødvendige brukere, grupperinger og biblioteker lim inn følgende kode;

```
sudo groupadd -g 33 www-data
sudo useradd \
  -g www-data --no-user-group \
  --home-dir /var/www --no-create-home \
  --shell /usr/sbin/nologin \
  --system --uid 33 www-data
```

```
sudo mkdir /etc/caddy
sudo chown -R root:root /etc/caddy
sudo mkdir /etc/ssl/caddy
sudo chown -R root:www-data /etc/ssl/caddy
sudo chmod 0770 /etc/ssl/caddy
```

9. Om du har fulgt disse punktene kan du nå sjekke status på caddy; `$ sudo systemctl status caddy.service`
10. Opprett en `caddyfile.yml` og manipuler routene og redirectingen.
11. Med et opprettet subdomene er du nå klar til å redirekte data fra hoved Raspberry PI, til sekundær Raspberry PI og til dedikert Subdomene.

12. Slik ser vår Caddy fil ut, her er det koblet til Basic Auth og vi har kjørt reverse proxy for å sende denne dataen til Subdomenet.

Route: Her definerer vi hvor ip adresse og port skal sende data på subdomenet.

Denne filen vil variere i forhold til hvor mange enheter du bruker, men kan bli brukt som en mal til ditt oppsett.

```
3D.alignracing.no {
```

```
    tls internal
    basicauth * {
        align JDJhJDE0JG9NZFRZL0N4Znl0WU55YXpPTDZ3YWWnTzIIUFJRSGRrZ21jd2QwZGUwa0JLRXpOMzdiZExT
    }
    redir /webcam/1 /webcam/1/
    redir /webcam/1 /webcam/2/
    redir /printer/1 /printer/1/
    redir /printer/2 /printer/2/
```

TLS internal (basicauth) *
definer brukernavn, kryptert passord.

```
route /webcam/1/* {
    uri strip_prefix /webcam/1
    reverse_proxy 10.0.0.201:8080 # Cam 1
}
route /webcam/2/* {
    uri strip_prefix /webcam/2
    reverse_proxy 10.0.0.201:8081 # Cam 2
}
handle_path /printer/1/* {
    #uri strip_prefix /printer/1
    reverse_proxy 10.0.0.201:4080 # PRINTER02
}
#reverse_proxy /printer/2/ 10.0.0.201:4081
reverse_proxy 10.0.0.90:4000 # OctoFarm
}
```

Route: Hvor skal data ende på domenet.
Reverse proxy: Hent data fra IP adresse og Port.

```
PRINTER02.alignracing.no, http://PRINTER02.alignracing.no {
    reverse_proxy 10.0.0.201:4080
```

```
    tls internal
    @authIP {
        not remote_ip 10.0.0.0/24
    }
    basicauth @authIP {
        align JDJhJDE0JG9NZFRZL0N4Znl0WU55YXpPTDZ3YWWnTzIIUFJRSGRrZ21jd2QwZGUwa0JLRXpOMzdiZExT
    }
}
```

```
PRINTER03.alignracing.no, http://PRINTER03.alignracing.no {
    reverse_proxy 10.0.0.201:4081
```

```
    tls internal
    @authIP {
        not remote_ip 10.0.0.0/24
    }
    basicauth @authIP {
        align JDJhJDE0JG9NZFRZL0N4Znl0WU55YXpPTDZ3YWWnTzIIUFJRSGRrZ21jd2QwZGUwa0JLRXpOMzdiZExT
    }
}
```