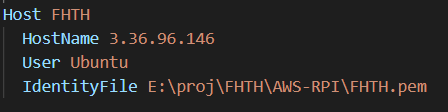
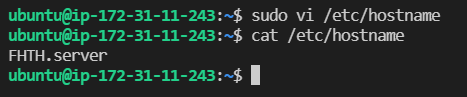
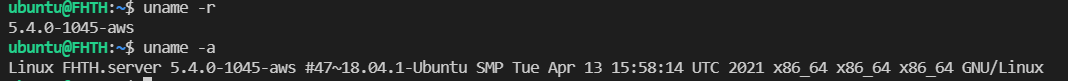
Raspberry pi 2B 와 AWS EC2 서버간 통신 구현

1. AWS
   1. backend(nodejs)
      1. url 핸들링 구현
         1. ./connected
         2. ./disconnected
   2. Frontend(VueX)
      1. 서버의 status 변수에 종속된 화면 제공
2. RPI rpi\_test1.py 구현
   1. axios로 접속 구현
3. AWS 서버 생성
   1. Info
      1. Ubuntu 18.04 LTS
      2. Bionic-18.04
   2. 서버 ssh 접속
      1. Vscode의 Remote Explorer 플러그인 사용
         1. Config 설정
            1. 
      2. 시스템 설정
         1. Hostname 설정
            1. 
         2. Apt 설정
            1. Remote Server 변경





* + - * 1. Apt update && apt install vim npm -y
      1. 시간 설정 - /etc/localtime 생성
         1. 
      2. 설정 결과 확인
         1. 
      3. 다중 유저 환경 구성
         1. 모두가 접근할 수 있게 ubuntu user 권한 변경 (777)
         2. ubuntu 홈에 git clone [ssh]

git branch -r 로 원격 저장소의 브랜치 확인

git checkout -t [원격저장소]

* + - * 1. 유저 생성

sudo adduser User >> 1234

계정확인 grep bash /etc/passwd

* + - * 1. 그룹생성

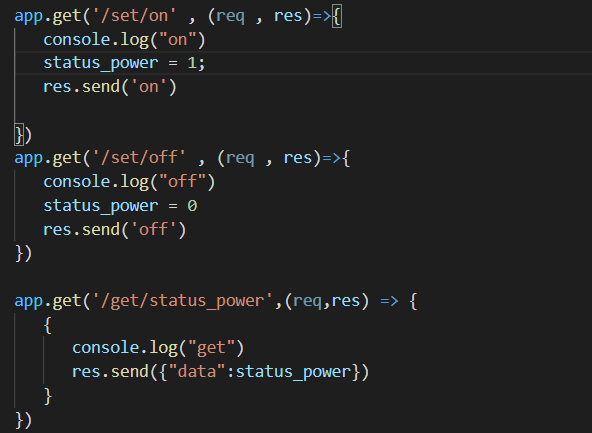
sudo gpasswd -a [user] [group]

sudo vi /etc/group 으로 확인

또는 sudo groups [user]로 확인

* + - * 1. 권한 설정

sudo vi /etc/sudoer

* + 1. Backend
       1. Npm 생성 & 의존성 설정
          1. cors - [크로스 사이트 스크립팅](https://evan-moon.github.io/2020/05/21/about-cors/) 예방
          2. express – REST API
          3. sequelize
          4. 
    2. Frontend
       1. Npm 초기화 & 의존성
          1. Npm init
          2. Axios - 웹 request 라이브러리
          3. Element-ui
          4. @vue/cli
       2. Vue.js

1. REST API

|  |  |
| --- | --- |
| POST /set/status/:id/ | :id의 status 갱신 |

* 1. RPI
  2. AWS
     1. Backend

|  |  |  |
| --- | --- | --- |
| WebSite |  |  |
| /login/ | Post | userId세션을 얻음 |
| /register/ | Post | 계정 DB에 등록   1. UserID 2. UserPW 3. [DeviceType : deviceID]  * 1 / 0 |
| /register/:id | Get | Select …   * 1 / 0 |
| /userInfo/:session.Id | Get | {   1. UserID 2. UserPW 3. [DeviceType : deviceID]   } |
| /userInfo/:session.Id | Post | Update Id == session.user.Id   1. UserPW 2. [DeviceType : devcieID] |
| /:session.id/device | Get | Session.user.DeviceId 으로 조회  If Session.user.DeviceId가 없음   * Send(“Error”)   Else   * Send(status) |
| RPI |  |  |
| /set/status/:id | Post | :id 기기 정보 갱신 |
|  |  |  |

* + 1. DB

|  |  |  |
| --- | --- | --- |
| WebSite |  |  |
| Post | /auth/login | accessToken 휙득 |
|  |  |  |

* + 1. d
    2. FrontEnd

1. /RPI 카메라 구현할 라이브러리
2. /AWS에서 설정값 받아올 URL

**210618**

1. **데이터베이스**
   * + - 1. **DB : MongoDB**
         2. **이유**

**스키마가 정리되지않았음**

**Agile 방식으로 설계와 구현을 동시에 진행하는데에 NoSQL이 적합하다고 판단.**

**벤처기업 THE VC에서도 사용하고 분산형 데이터베이스 프레임워크 ELK에서도 사용되므로 한번 접해보고 싶음**

1. **Data structure**
   1. **BackEnd**

|  |  |  |
| --- | --- | --- |
| **user** |  |  |
| **id** | **Int** |  |
| **email** | **varchar(20)** |  |
| **pw** | **varchar(20)** |  |
| **device** | **[type, serial]** |  |
| **Device** |  |  |
| **Id** | **Int** |  |
| **type** | **varchar(20)** |  |
| **registedDevice** |  |  |
| **id** |  |  |
| **serial** |  |  |
| **type** |  |  |
| **status** | **Object** |  |

* 1. **FrontEnd**

|  |  |  |
| --- | --- | --- |
| **/auth\_info** |  |  |
|  | **accessToken** |  |
| **/usdr\_info** |  |  |
| **email** | **Int** |  |
| **devciceList** |  |  |
|  | **serial** |  |
|  | **type** |  |
|  | **status** |  |
|  |  |  |
|  |  |  |

1. **토큰 휙득**

|  |  |  |  |
| --- | --- | --- | --- |
| **cookies** | **client** | **server** | **DB** |
|  | **/auth/v1/getToken {id, pw}** |  |  |
|  |  |  | **(auth) ; accessToken**  **email** |
|  |  | **response {accessToken}** |  |
| **accessToken** |  |  |  |

1. **데이터 요청**

|  |  |  |  |
| --- | --- | --- | --- |
| **cookies** | **client** | **server** | **DB** |
|  | **(header)**  **accessToken** |  |  |
|  |  |  | **accessToken 유효성 검사** |
|  |  |  | **id에 맞는 정보 확인{accessToken}** |
|  |  | **response data** |  |

**210619**

mongoose 미들웨어 추가

mongoDB ubuntu 설치

- https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/

Virtual Box 구성 (Ubuntu 18.04

- https://theniceguy.tistory.com/13

npm(dotenv, bcrypt)

* dotenv
  + https://devhyun.com/blog/post/23

**210621**

- mongodb with nodejs : <http://mongodb.github.io/node-mongodb-native/3.4/quick-start/quick-start/>

- https://poiemaweb.com/mongoose

- mongodb in ubuntu : https://devst.tistory.com/68

* login, logout
* 세션구현
  + <https://berkbach.com/node-js%EC%99%80-cookie-session%EC%9C%BC%EB%A1%9C-%EC%82%AC%EC%9A%A9%EC%9E%90-%EC%A0%95%EB%B3%B4-%EC%A0%80%EC%9E%A5-part-3-4868ad575fd1>

210622

* 회원가입 front페이지 제작
* device url 변경(/api/divice/…
* emit
  + <https://velog.io/@unani92/Vue-JS-props-emit-%EC%97%B0%EC%8A%B5%ED%95%98%EA%B8%B0-%EC%9C%A0%ED%88%AC%EB%B8%8C-%ED%81%B4%EB%A1%A0%EC%BD%94%EB%94%A9>

210623

* 로그인 화면, 로그인시 client 로컬에 jwt를 얻고 /main으로 리다이렉트
* login에서 Jwt 인증이 되면 /main으로 바로 넘어가도록 구현
* vue에서 jwt로 인증 구현하기[https://minu0807.tistory.com/64\*](https://minu0807.tistory.com/64*)
* https://kdinner.tistory.com/60
  + vue-cookies 설치
  + <https://vuex.vuejs.org/kr/guide/actions.html>
  + <https://github.com/vuejs/vuex/blob/dev/examples/shopping-cart/components/ShoppingCart.vue>

210625

* 세션 얻기 전
  + Home.vue
  + Signup.vue
* 세션 얻은 후

|  |  |
| --- | --- |
|  |  |
| /Home |  |
| /SignUp |  |
|  |  |

0627

* 텍스트이(가) 표시된 사진

  자동 생성된 설명

0629

[아마존 서버로 옮기기]

1. git add
2. git commit -m “msg”
3. git pull origin
4. git push origin AWS

[몽고DB 복구]

윈도우용 덤프 프로그램 : <https://www.mongodb.com/try/download/database-tools?tck=docs_databasetools> Database Tool

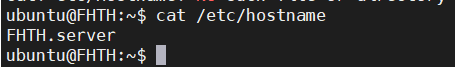
명령어: https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=kkforgg&logNo=220661830576

텍스트이(가) 표시된 사진

자동 생성된 설명`

ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

질문1



왜 .뒤에는 안나오지???

질문2

Npm i @vue/cli

ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

공부