시퀄라이즈 getSomething, removeSomething

  const User = sequelize.define('User', {

    id: {

      type: DataTypes.STRING(30),

      allowNull: false

      unique: true

      primaryKey: true,

    },

.. 중략

  User.associate = (db) => {

    db.User.belongsToMany(db.Item, { through: "Carts", as: 'Item'}); // 장바구니

  };

const Item = sequelize.define('Item', {

id

.. 중략

  Item.associate = (db) => {

    db.Item.belongsToMany(db.User, { through: "Carts", as: 'user'});

  };

수정한 내용

User.associate = (db) => { // 릴레이션(관계) 정의

... 중략

db.User.belongsToMany(db.Item, { through: "Carts", as: 'CartItem'}); // 장바구니

};

시퀄라이즈 다대다 테이블은 { through: "Carts", as: 'CartItem'} 이런식으로 써주면

UserID, ItemID가 있는 Carts라는 테이블을 자동으로 생성해준다.

또 자동으로 as의 값 기준으로 get, remove 등의 함수도 쓸수있는데

item.getUser, user.removeItem 이런식으로 쓰면 된다.

user.getItem을 하면 Carts테이블에서 User가 가진 Item을 불러오고

item.getUser를 하면 Carts테이블에서 Item이 가진 User를 불러온다.

const user = await User.findOne({

where: { id: req.params.userId },

});

user를 불러온 후

await user.removeItem(삭제할 아이템 아이디);

를 통해 카트 테이블 특정 유저의 특정 아이템을 삭제하려고 시도했다.

그런데 시퀄라이즈 쿼리 로그를 보면 DELETE 쿼리문을 실행하지 않고 엉뚱하게 UPDATE를 실행시키는 것이다.

반대로

await item.removeUser(삭제할 유저 아이디);

를 하니 정상적으로 DELETE문이 실행되었다.

Executing (default): DELETE FROM `Carts` WHERE `ItemId` = 25 AND `UserId` IN ('t2')

Executing (default): DELETE FROM `Carts` WHERE `UserId` = 't2' AND `ItemId` IN (25)

배포 전 작업

back/app.js

if (process.env.NODE\_ENV === 'production') { //배포모드일 떄

  app.use(morgan('combiend'));

  app.use(hpp());

  app.use(helmet());

  // npm i pm2 cross-env helmet hpp

} else {

  app.use(morgan('dev')); // 개발모드

}

back/package.json

  "scripts": {

    "dev": "nodemon app",

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "cross-env NODE\_ENV=production pm2 start app"

  },

깃허브

100MB이상 파일 에러

<https://medium.com/@stargt/github%EC%97%90-100mb-%EC%9D%B4%EC%83%81%EC%9D%98-%ED%8C%8C%EC%9D%BC%EC%9D%84-%EC%98%AC%EB%A6%AC%EB%8A%94-%EB%B0%A9%EB%B2%95-9d9e6e3b94ef>

git rm -r --cached .

git add .

git commit -m "fixed untracked files"

강제 pull

git fetch --all

git reset --hard origin/main

git pull origin main

리눅스 명령어

폴더 삭제

rm -r dir1

모바일 dev툴

chrome://inspect/#devices

https://ap-northeast-2.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-2#Instances:  
**아마존 AWS 이용**

1. git에 업로드

<https://github.com/moamoa-code/excuse-moa>

2. 아마존 프라이빗 키 생성 ~.pem 파일

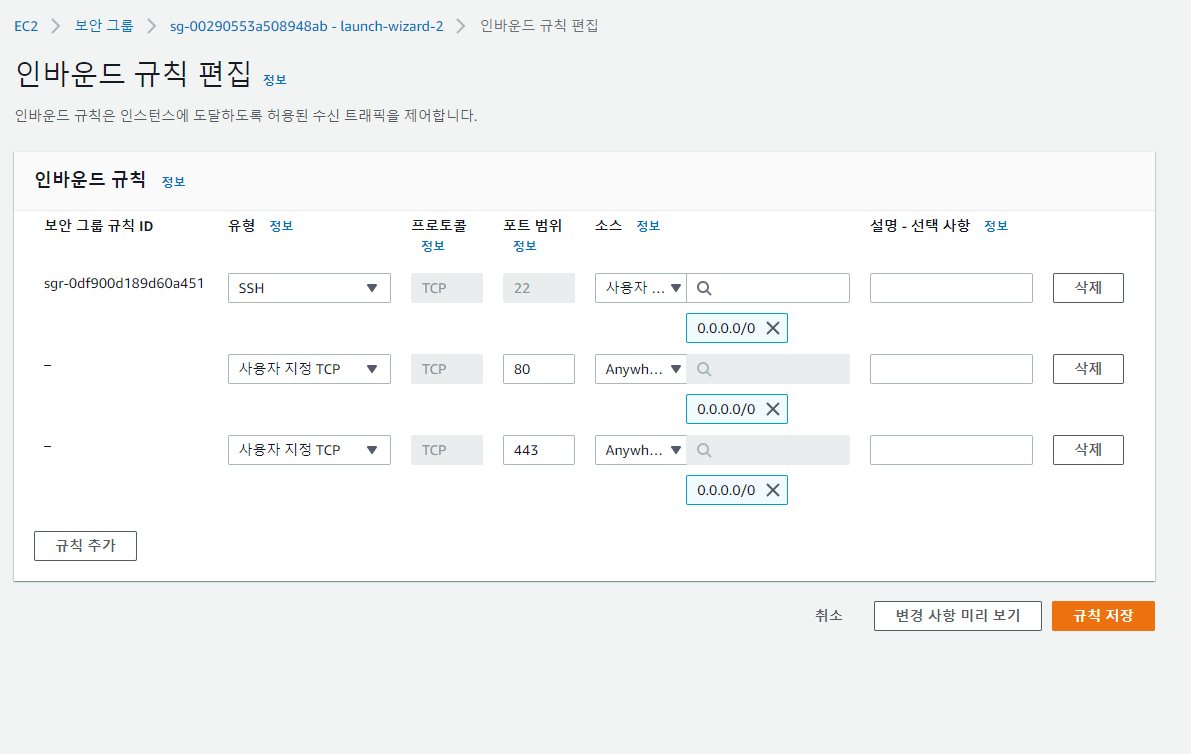
아마존 인스턴스에 연결 SSH 클라이언트

~.pem 파일 있는곳에서 명령어

ssh -i "excuse-moa.pem" [ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com](mailto:ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com)

깃 아이디, 개인토큰으로 로그인 <https://github.com/settings/tokens>

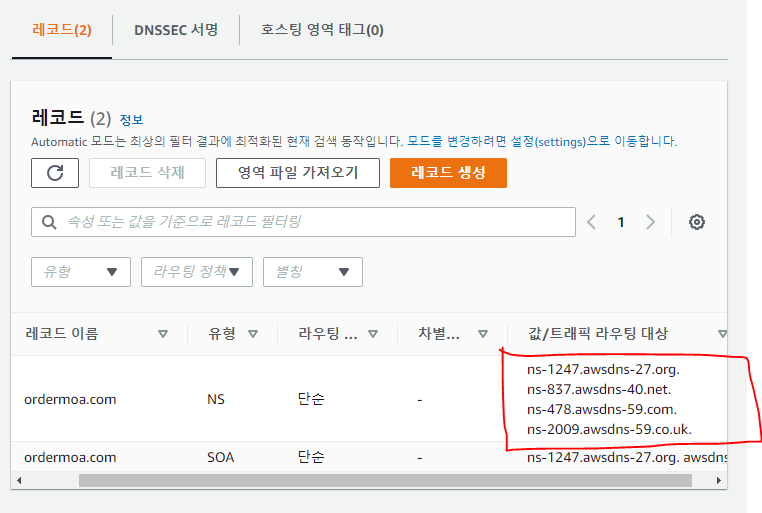
-> 아마존 AWS 우분투 SSH 연결 완료.



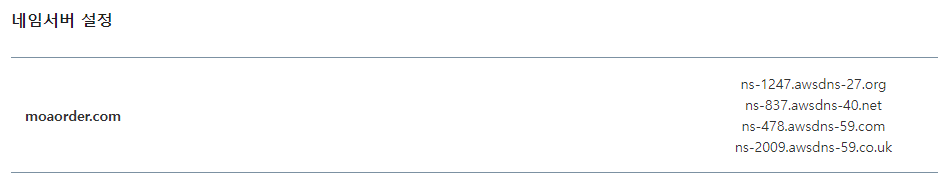
3. 도메인 연결 (유료)

-> 백서버, 프론트서버가 다른 IP를 가지면 도메인 연결을 해줘야함. 안하면 쿠키공유 안됨!

aws -> 라우트53 -> 호스팅 영역 -> 생성



가비아



**4. 탄력적 IP (유료) -> 아이피 고정 작업.**

EC2 -> 탄력적 IP

탄력적 IP 할당 -> 작업 - IP주소 연결 -> 인스턴스 선택

프론트, 백앤드 각각 생성

34.214.254.132 moaorder.com 프론트

35.80.237.123 moaorder.com 백

route53 -> 호스팅영역

레코드 생성 -> 유형: A (IPv4), 값: 각각 서버 IP주소

(백엔드) -> api.ordermoa.com



**프론트서버**

ssh -i "excuse-moa.pem" ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com

준비작업

npm i pm2

기존 주소 localhost로 되어있던 부분 수정.

깃 클론 (파일 복사)

git clone <https://github.com/moamoa-code/excuse-moa>

SSH에서 front\_rq 폴더로 이동.

노드 설치

sudo apt-get update

sudo apt-get install -y build-essential

sudo apt-get install curl

curl -sL <https://deb.nodesource.com/setup_14.x> | sudo -E bash --

sudo apt-get install -y nodejs

설치확인 명령어

node -v

npm -v

설치

npm i

빌드

npm run build

실행

npx pm2 start npm -- start

로컬에서 빌드 후 서버에 업로드 하기

리눅스 서버에서 권한 변경 ->

chmod -R 777 front\_rq

로컬에서 ->

scp -i "excuse-moa.pem" **.next.tar** [ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com**:~/excuse-moa/front\_rq**](mailto:ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com:~/excuse-moa/front_rq)

압축해제

tar -xvf .next.tar

**백서버**

ssh -i "excuse-moa.pem" [ubuntu@ec2-3-39-12-156.ap-northeast-2.compute.amazonaws.com](mailto:ubuntu@ec2-3-39-12-156.ap-northeast-2.compute.amazonaws.com)

git clone <https://github.com/moamoa-code/excuse-moa>

back 폴더로 이동

sudo apt-get update

sudo apt-get install -y build-essential

sudo apt-get install curl

curl -sL <https://deb.nodesource.com/setup_14.x> | sudo -E bash --

sudo apt-get install -y nodejs

npm i

MYSQL 설치 <https://www.tecmint.com/install-mysql-8-in-ubuntu/>

MYSQL 삭제 참고 <https://www.inflearn.com/questions/408891>

https://velog.io/@moorekwon/MySQL-%EC%99%84%EC%A0%84-%EC%82%AD%EC%A0%9C

PUBKEY 에러 해결 https://comcube.tistory.com/entry/Ubuntu-%EC%97%90%EC%84%9C-GPG-ERROR-NOPUBKEY-%EC%97%90%EB%9F%AC%EC%8B%9C-%ED%95%B4%EA%B2%B0%EB%B0%A9%EB%B2%95

~~sudo apt-get install -y mysql-server 🡨 5.7버전이 설치됨.~~

wget -c https://repo.mysql.com/mysql-apt-config\_0.8.13-1\_all.deb

sudo dpkg -i mysql-apt-config\_0.8.13-1\_all.deb

sudo apt-get update

sudo apt-get install mysql-server

접속

mysql -uroot -p

**백서버에 .env (sql비밀번호 등) 설정**

**vim .env**

mysql 루트 비밀번호 변경 (필요시)

sudo su

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '비밀번호'

**포트번호 변경**

**vim app.js**

**app.listen부분에서 포트부분 80(AWS에서 준 값)으로 변경**

**.env 파일 생성**

**COOKIE\_SECRET=excuse-moa**

**DB\_PASSWORD=비번**

db생성

npx sequelize db:create

서버 background로 실행하기 (터미널 종료시 서버 종료 방지)

sudo node app $ <- 안좋은 방법

pm2 설치.

npm i pm2

vim package.json

"start": "pm2 start app.js" 로 수정

끄기: npx pm2 kill

모니터: npx pm2 monit

리스트: sudo npx pm2 list

재시작: sudo npx pm2 reload all

실행 확인

sudo npm start (80번 포트는 항상 sudo로 실행해야 함)

sudo npm start && sudo npx pm2 monit 모니터링 동시 실행

git pull에서 Aborting 에러 날경우

git reset --hard

**아마존 S3에 이미지 업로드**

[https://s3.console.aws.amazon.com/s3/home?region=ap-northeast-2#](https://s3.console.aws.amazon.com/s3/home?region=ap-northeast-2)

버킷만들기

버킷 정책 편집

|  |
| --- |
| {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "addPerm",  "Principal": "\*",  "Effect": "Allow",  "Action": [  "s3:GetObject",  "s3:PutObject"],  "Resource": [  "arn:aws:s3:::excuse-moa/\*"]  }  ]  } |

액세스키 만들기

프로필-> 보안 자격 증명 -> 액세스 키 생성 -> 키 파일 다운로드

/back

npm i multer-s3 aws-sdk

.env에 키 추가

S3\_ACCESS\_KEY\_ID=

S3\_SECRET\_ACCESS\_KEY=

이미지 업로드 부분 router 수정

const multerS3 = require('multer-s3');

const AWS = require('aws-sdk');

// 파일 업로드, 이름 변경 처리 + 아마존 S3

AWS.config.update({

  accessKeyId: process.env.S3\_ACCESS\_KEY\_ID,

  secretAccessKey: process.env.S3\_SECRET\_ACCESS\_KEY,

  region: 'ap-northeast-2', // 서울

});

const upload = multer({

  storage: multerS3({

    s3: new AWS.S3(),

    bucket: 'excuse-moa',

    key (req, file, cb) {

      const ext = path.extname(file.originalname);

      cb(null, `original/${Date.now()}\_${path.basename(file.originalname).substring(0,5)+ext}`)

    }

  }),

  limits: { fileSize: 2 \* 1024 \* 1024 }, // 2MB

});

// 사진 업로드

router.post('/image', isLoggedIn, upload.array('image'), (req, res, next) => {

  res.json(req.files[0].location); // S3는 location

});

프론트에서 이미지 부분 주소 변경. -> S3의 location이 주소이기 때문에 backurl이 필요없음

**HTTPS 적용**

앤진엑스 nginx 이용 (리버스 프록시)

기존 front -> 443/80 요청 -> next (80포트) (https: 443, 80 요청을 443으로 리다이랙트)

적용후 front -> 443/80 요청 -> nginx (80포트) -> next (3060포트) (https: 443)

아마존 AWS front에서

apt-get instal eninx

vim etc/nginx/nginx.conf

server {

server\_name **excusemoa.com**

listen 80

**Let's Encrypt 무료 SSL 인증서 (3개월마다 갱신)**

wget <https://dl.eff.org/certbot-auto>

**버그**

메인화면 공지사항 바로 로드안됨 해결

공지사항수정 - 판매자 정보 바로 로드 안됨. 해결 SSR문제 - (useEffect로 땜빵)

DDNS 해결

프로그램 이용 <https://openart.tistory.com/278>

가비아

ssh -i moaorder1.pem [gabia\_user@139.150.72.115](mailto:gabia_user@139.150.72.115)

**moaorder.com**

리전 -> 오레곤

프론트

ssh -i "moaorder-aws.pem" [ubuntu@ec2-34-214-254-132.us-west-2.compute.amazonaws.com](mailto:ubuntu@ec2-34-214-254-132.us-west-2.compute.amazonaws.com)  
.next 복사

scp -i "moaorder-aws.pem " **.next.tar** [[ubuntu@ec2-34-214-254-132.us-west-2.compute.amazonaws.com](mailto:ubuntu@ec2-34-214-254-132.us-west-2.compute.amazonaws.com)**:~/excuse-moa/front\_rq**](mailto:ubuntu@ec2-3-36-36-223.ap-northeast-2.compute.amazonaws.com:~/excuse-moa/front_rq)

tar -xvf .next.tar

백

ssh -i "moaorder-aws.pem" ubuntu@ec2-35-80-237-123.us-west-2.compute.amazonaws.com

소개

원두커피 주문관리 시스템

원두커피 주문 관리에 최적화된 시스템.

간편하게 원두커피를 주문해 보세요.

판매자 - 구매자 통합

판매자, 구매자 회원 모두 간편하게 사용 가능합니다.

편리한 인터페이스 사용

KIOSK, 온라인 쇼핑몰과 비슷한 친숙한 인터페이스

주문목록 기록

생산자

판매자

구매자

-판매자-

고객 관리

판매자는 구매고객을 생성, 등록할 수 있습니다.

제품 주문

판매자가 생성한 고객이 직접 제품주문이 가능하며

판매자가 구매자를 선택해 발주가 가능합니다.

고객 맞춤 관리

구매자별로 노출 가능한 제품, 공지사항을 설정할 수 있습니다.