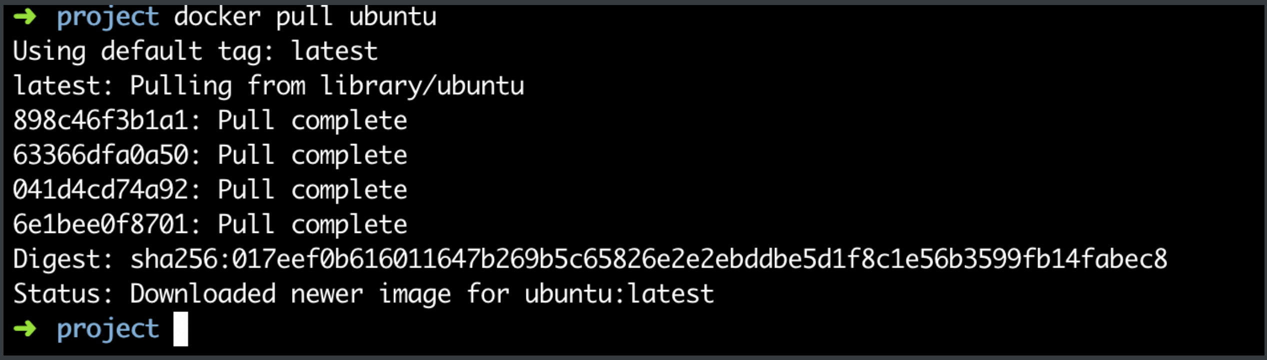
도커 응용~

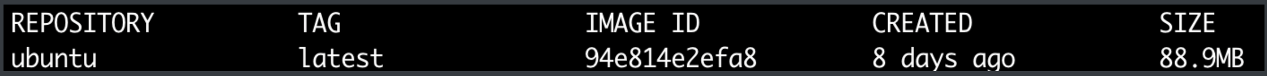
- 핵심 키포인트 : 웹서버, 데이터베이스를 도커 이미지화해서 컨테이너로 띄운 후 링크로 연결시켜 배포한다!

실습



```
➜  project docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
898c46f3b1a1: Pull complete
63366dfa0a50: Pull complete
041d4cd74a92: Pull complete
6e1bee0f8701: Pull complete
Digest: sha256:017eef0b616011647b269b5c65826e2e2ebddbe5d1f8c1e56b3599fb14fabec8
Status: Downloaded newer image for ubuntu:latest
➜  project
```

- docker pull ubuntu -> 우분투 이미지 가져오기



```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              94e814e2efa8        8 days ago          88.9MB
```

- docker images - > 이미지 확인

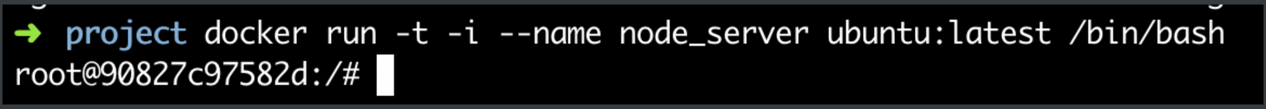
ubuntu 실행

- docker run
  - run 은 컨테이너를 생성하고 시작하는 명령어
  - create, start 명령어 두개를 한꺼번에 실행해줌
  - run으로 명령을 주었을 시에 생성은 되는데 시작이 안되면 명령 옵션을 잘 못 준것임
    - 컨테이너 볼륨 공유 설정을 잘못했을때
    - -p로 중복되는 포트 할당했을때
    - 단일 어플리케이션을 위한 컨테이너(mysql) 같은 경우 -e옵션 안준경우
  - docker run [option] [name] [command line] [기타..]

option 종류

- -c : —cpu-share : cpu 사용 얼마나 할건지 , 기본값 1024M
- -m : 메모리 제한 - 해당 메모리 넘어가면 자동종료 50m으로 하게되면 우분투 베이스 컨테이너도 종료

docker run –t –i —name node_server ubuntu:latest /bin/bash



```
➜ project docker run -t -i --name node_server ubuntu:latest /bin/bash
root@90827c97582d:/#
```

–t 랑 –i 가 뭐에요?

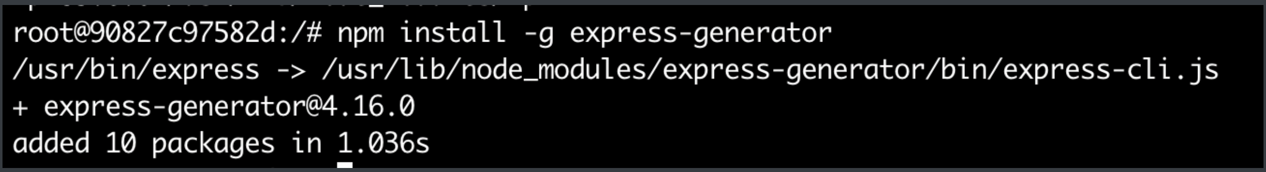
-i : interactive -> 해당 옵션을 안쓰게 되면 명령을 쳐도 보이지않음 -t : terminal 형식으로 실행되게 해줌 (tty)

컨테이너 exit 안걸고 나오기

CTRL Q + P

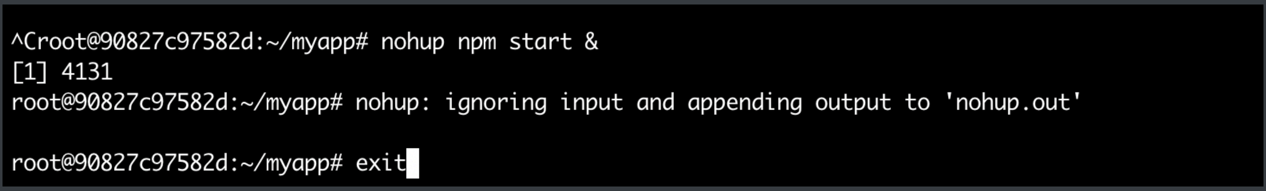
node 설치

```
apt-get update

apt-get install -y curl

curl -sL https://deb.nodesource.com/setup_9.x | bash -

apt-get install nodejs

npm install -g express-generator
```



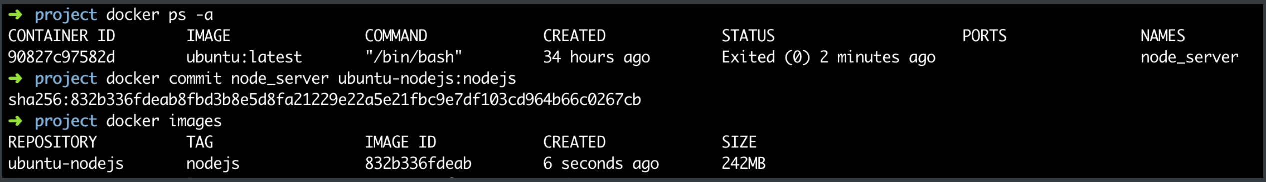
```
root@90827c97582d:/# npm install -g express-generator
/usr/bin/express -> /usr/lib/node_modules/express-generator/bin/express-cli.js
+ express-generator@4.16.0
added 10 packages in 1.036s
```

```
express myapp
cd myapp
npm install
```



```
^Croot@90827c97582d:~/myapp# nohup npm start &
[1] 4131
root@90827c97582d:~/myapp# nohup: ignoring input and appending output to 'nohup.out'

root@90827c97582d:~/myapp# exit
```

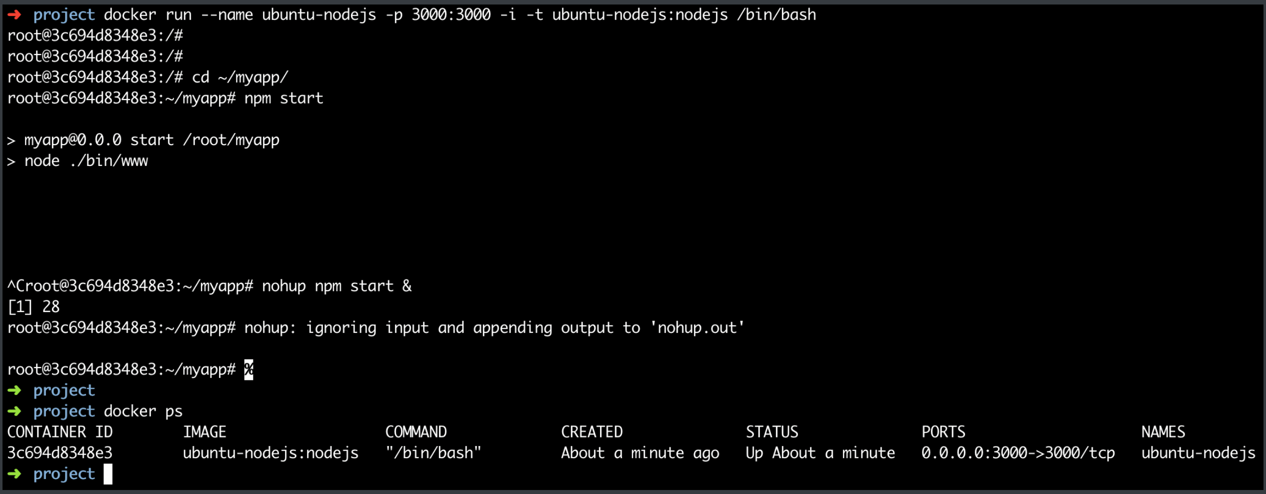
node 서버 이미지 생성



```
➜ project docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                     PORTS               NAMES
90827c97582d        ubuntu:latest       "/bin/bash"         34 hours ago        Exited (0) 2 minutes ago                       node_server
➜ project docker commit node_server ubuntu-nodejs:nodejs
sha256:832b336fdeab8fbd3b8e5d8fa21229e22a5e21fbc9e7df103cd964b66c0267cb
➜ project docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu-nodejs       nodejs              832b336fdeab        6 seconds ago       242MB
```

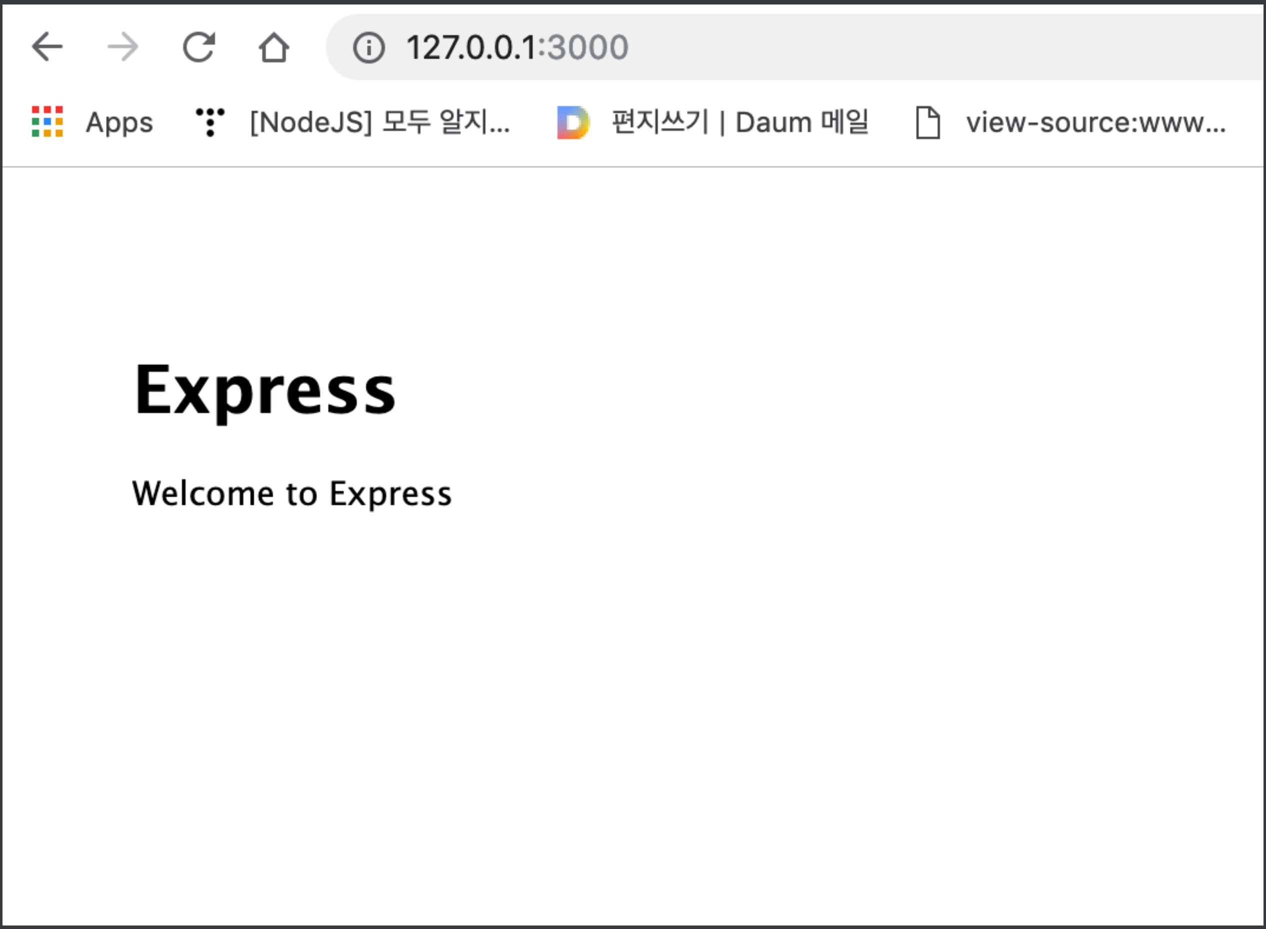
node 이미지로 포트 열어서 실행해보기

- 실행중인 상태로 컨테이너 빠져나오기 -> ctrl p + q



```
➜ project docker run --name ubuntu-nodejs -p 3000:3000 -i -t ubuntu-nodejs:nodejs /bin/bash
root@3c694d8348e3:/#
root@3c694d8348e3:/#
root@3c694d8348e3:/# cd ~/myapp/
root@3c694d8348e3:~/myapp# npm start

> myapp@0.0.0 start /root/myapp
> node ./bin/www





^Croot@3c694d8348e3:~/myapp# nohup npm start &
[1] 28
root@3c694d8348e3:~/myapp# nohup: ignoring input and appending output to 'nohup.out'

root@3c694d8348e3:~/myapp#
➜ project
➜ project docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED              STATUS              PORTS                    NAMES
3c694d8348e3        ubuntu-nodejs:nodejs   "/bin/bash"         About a minute ago   Up About a minute   0.0.0.0:3000->3000/tcp   ubuntu-nodejs
➜ project
```

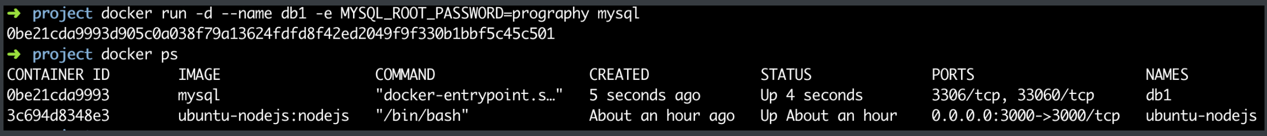
확인



Express

Welcome to Express

mysql 컨테이너 생성

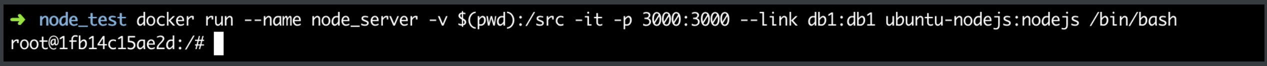


```
➜ project docker run -d --name db1 -e MYSQL_ROOT_PASSWORD=prography mysql
0be21cda9993d905c0a038f79a13624fdfd8f42ed2049f9f330b1bbf5c45c501
➜ project docker ps
CONTAINER ID        IMAGE                 COMMAND                  CREATED              STATUS              PORTS                    NAMES
0be21cda9993        mysql                 "docker-entrypoint.s…"   5 seconds ago        Up 4 seconds        3306/tcp, 33060/tcp      db1
3c694d8348e3        ubuntu-nodejs:nodejs  "/bin/bash"              About an hour ago    Up About an hour    0.0.0.0:3000->3000/tcp   ubuntu-nodejs
```

mysql-server 설치

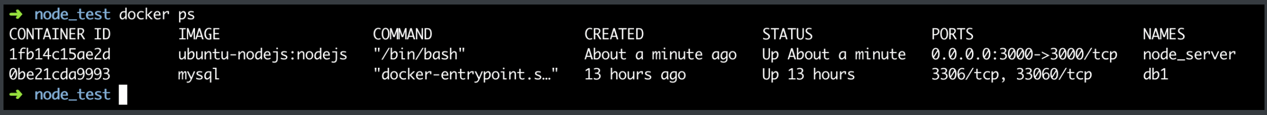
docker exec -it db1 bash apt update apt-get install mysql-server

링크걸기 & 실행



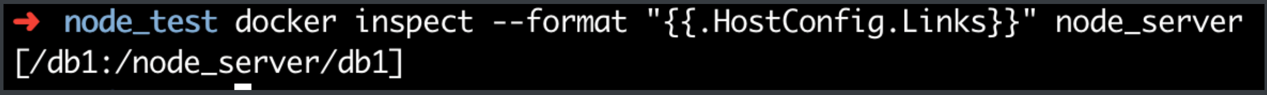
```
➜ node_test docker run --name node_server -v $(pwd):/src -it -p 3000:3000 --link db1:db1 ubuntu-nodejs:nodejs /bin/bash
root@1fb14c15ae2d:/#
```

- v 옵션 : volume 호스트의 directory 와 컨테이너의 디렉토리 위치 공유



```
➜ node_test docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED              STATUS              PORTS                    NAMES
1fb14c15ae2d        ubuntu-nodejs:nodejs   "/bin/bash"              About a minute ago   Up About a minute   0.0.0.0:3000->3000/tcp   node_server
0be21cda9993        mysql                  "docker-entrypoint.s…"   13 hours ago         Up 13 hours         3306/tcp, 33060/tcp      db1
➜ node_test
```

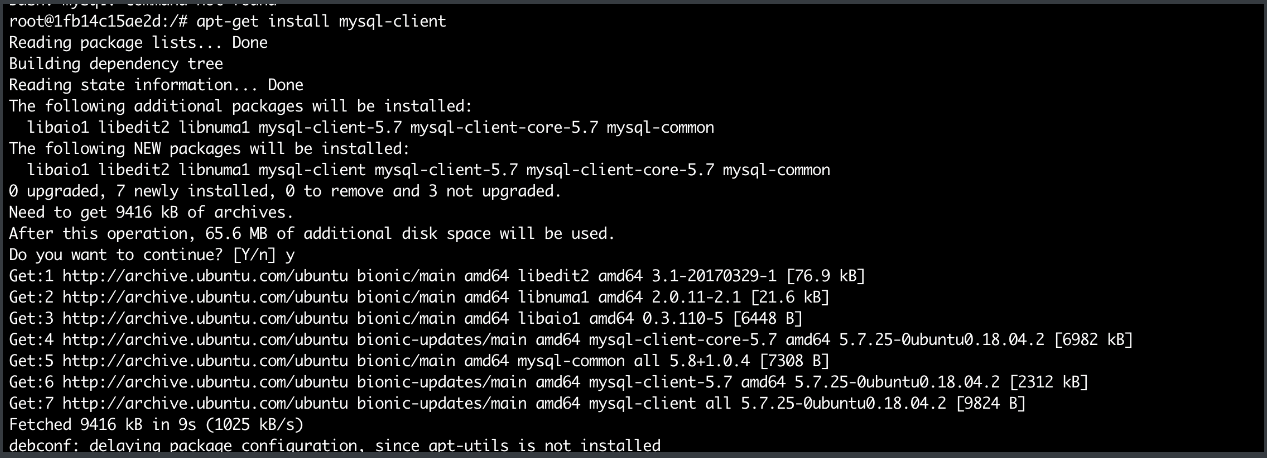
- 링크 확인



```
➜ node_test docker inspect --format "{{.HostConfig.Links}}" node_server
[/db1:/node_server/db1]
```

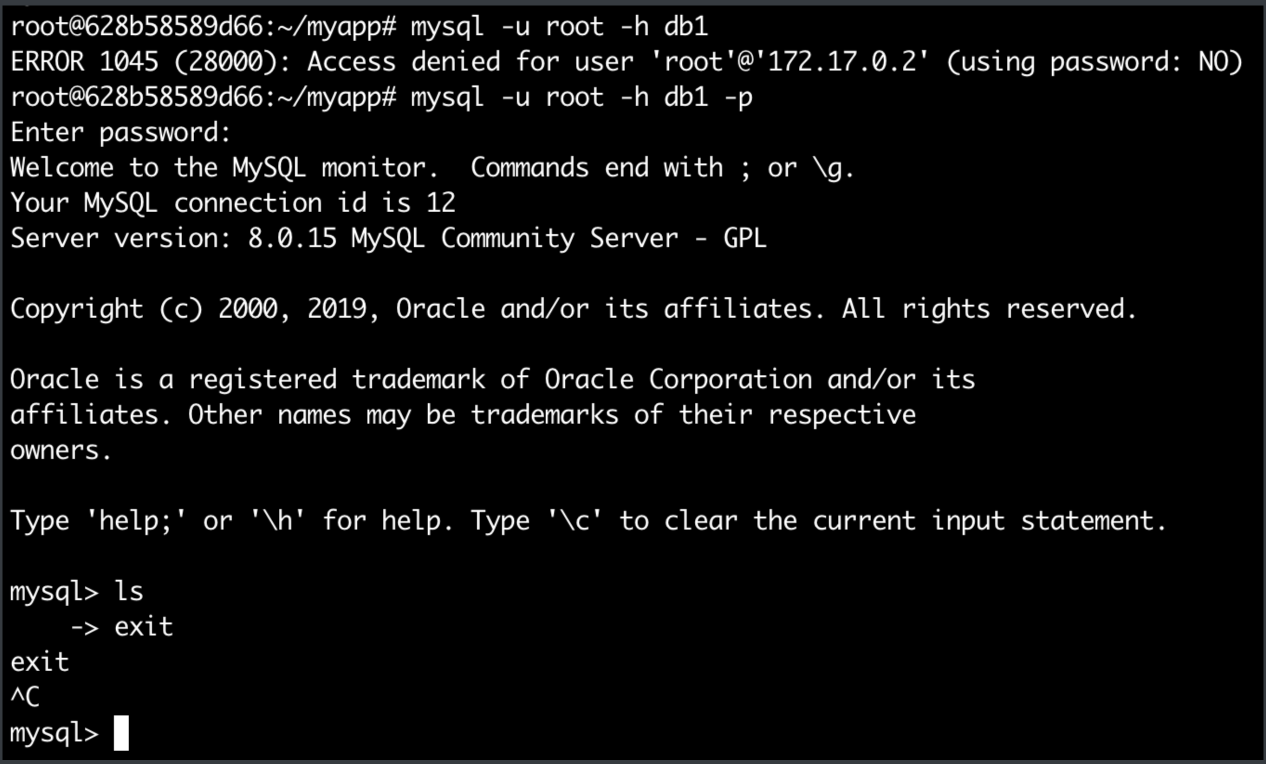
- 컨테이너끼리 통신가능함

mysql-client 설치



```
root@1fb14c15ae2d:/# apt-get install mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libedit2 libnuma1 mysql-client-5.7 mysql-client-core-5.7 mysql-common
The following NEW packages will be installed:
  libaio1 libedit2 libnuma1 mysql-client mysql-client-5.7 mysql-client-core-5.7 mysql-common
0 upgraded, 7 newly installed, 0 to remove and 3 not upgraded.
Need to get 9416 kB of archives.
After this operation, 65.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libedit2 amd64 3.1-20170329-1 [76.9 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libnuma1 amd64 2.0.11-2.1 [21.6 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 libaio1 amd64 0.3.110-5 [6448 B]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.25-0ubuntu0.18.04.2 [6982 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0.4 [7308 B]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-5.7 amd64 5.7.25-0ubuntu0.18.04.2 [2312 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client all 5.7.25-0ubuntu0.18.04.2 [9824 B]
Fetched 9416 kB in 9s (1025 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
```

접속, 테이블 생성



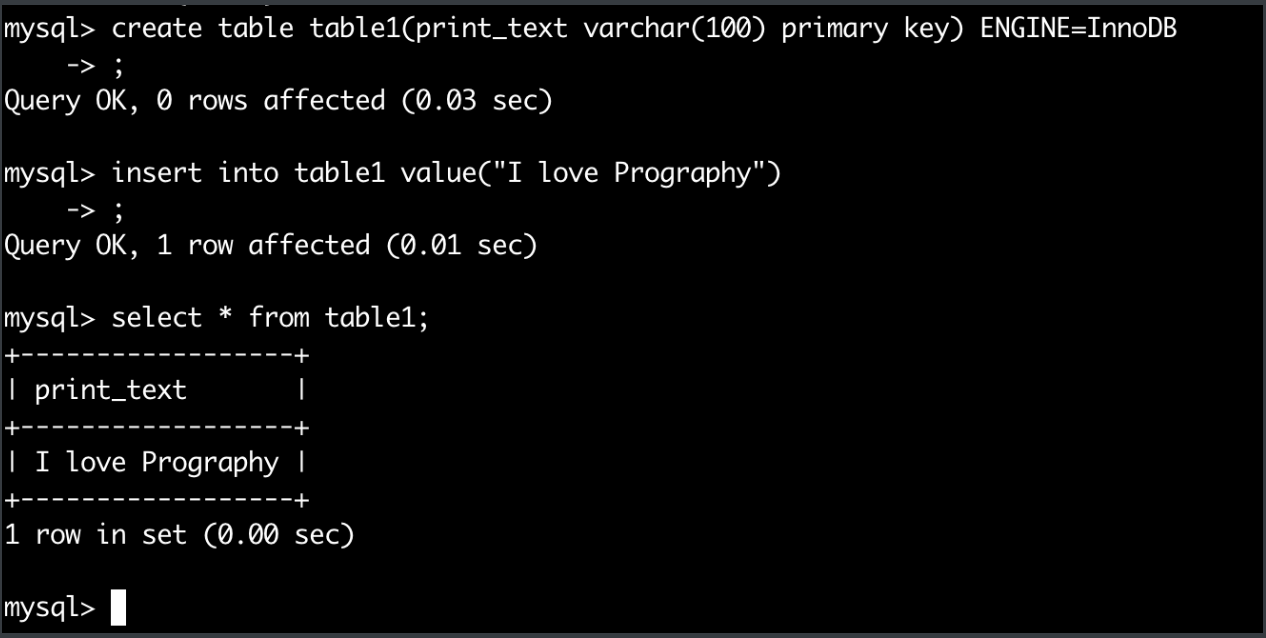
```
root@628b58589d66:~/myapp# mysql -u root -h db1
ERROR 1045 (28000): Access denied for user 'root'@'172.17.0.2' (using password: NO)
root@628b58589d66:~/myapp# mysql -u root -h db1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.15 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ls
    -> exit
exit
^C
mysql> 
```

Mysql 8버전은 현재 노드에서 연결 지원이 잘 안되어서 5.7 태그 버전을 사용하여 테스트

https://stackoverflow.com/questions/50373427/node-js-cant-authenticate-to-mysql-8-0

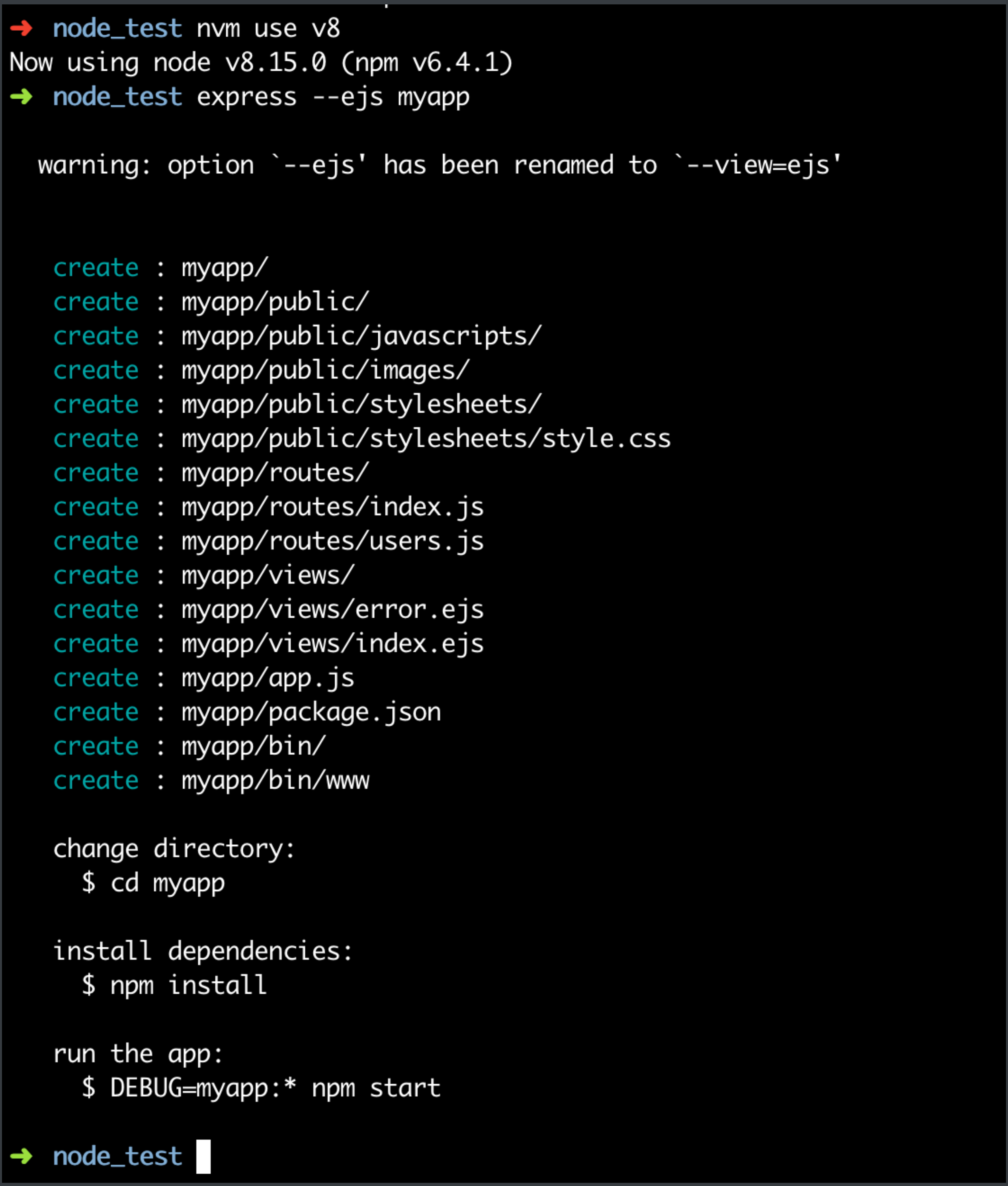
create database test1;

use test1;

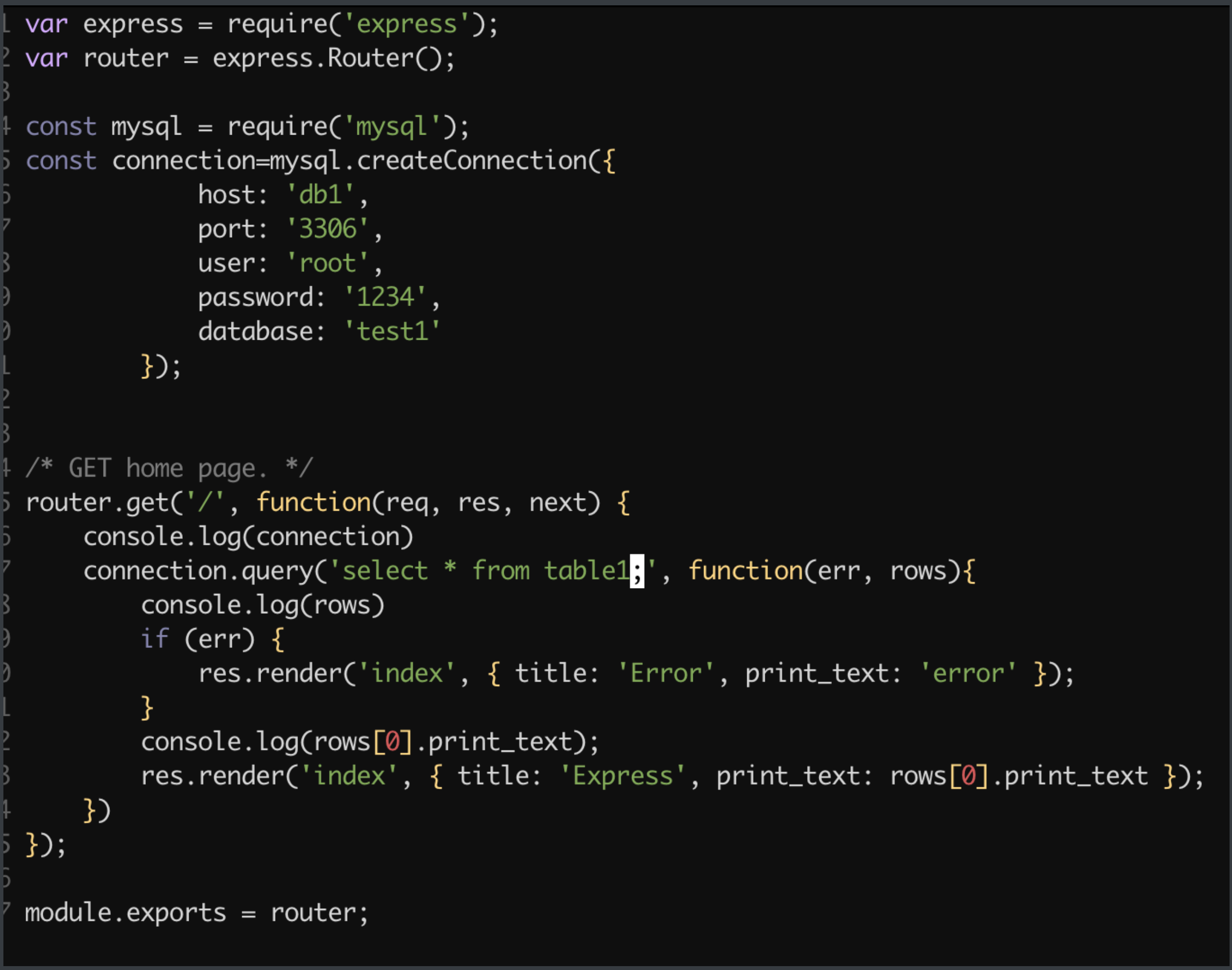


```
mysql> create table table1(print_text varchar(100) primary key) ENGINE=InnoDB
    -> ;
Query OK, 0 rows affected (0.03 sec)

mysql> insert into table1 value("I love Prography")
    -> ;
Query OK, 1 row affected (0.01 sec)

mysql> select * from table1;
+------------------+
| print_text       |
+------------------+
| I love Prography |
+------------------+
1 row in set (0.00 sec)

mysql> 
```

나와서 로컬 $(pwd) 경로이동

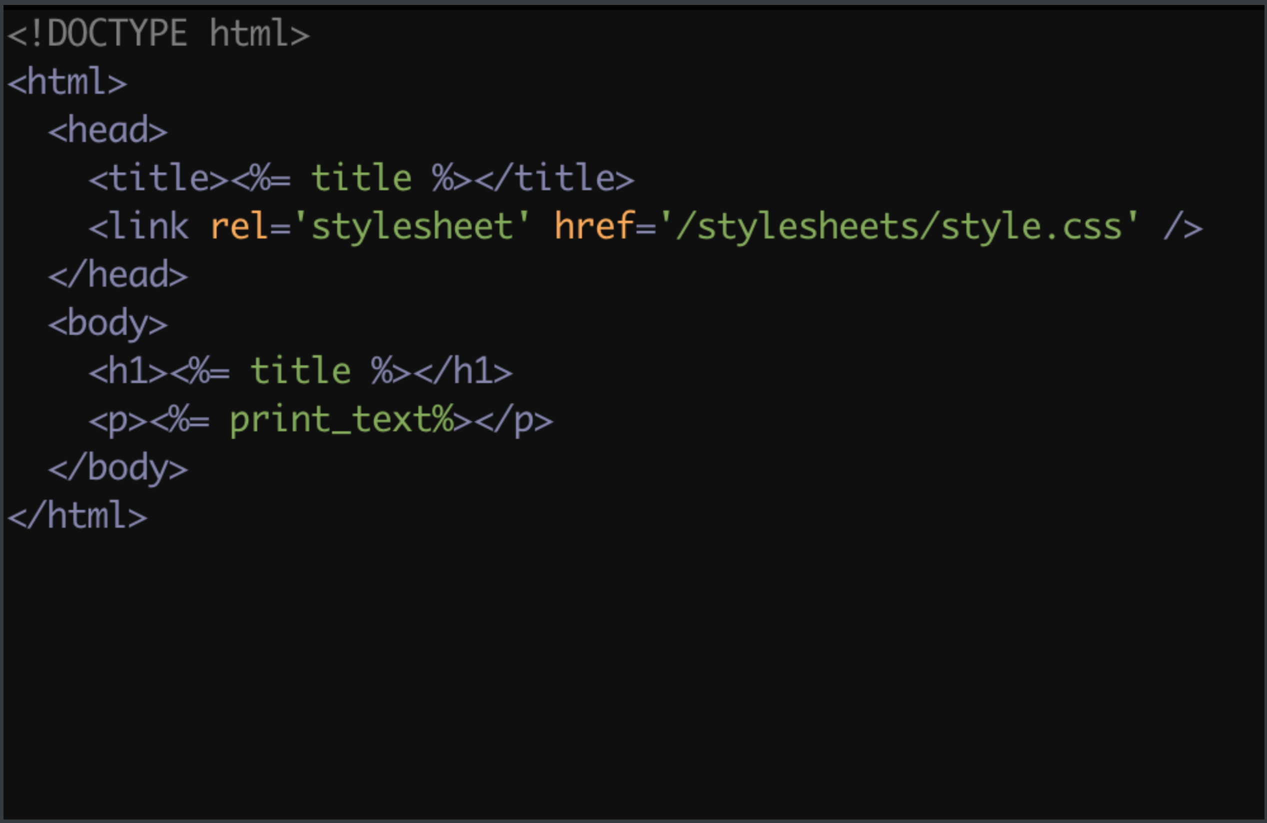

```
➜  node_test nvm use v8
Now using node v8.15.0 (npm v6.4.1)
➜  node_test express --ejs myapp

  warning: option `--ejs' has been renamed to `--view=ejs'


   create : myapp/
   create : myapp/public/
   create : myapp/public/javascripts/
   create : myapp/public/images/
   create : myapp/public/stylesheets/
   create : myapp/public/stylesheets/style.css
   create : myapp/routes/
   create : myapp/routes/index.js
   create : myapp/routes/users.js
   create : myapp/views/
   create : myapp/views/error.ejs
   create : myapp/views/index.ejs
   create : myapp/app.js
   create : myapp/package.json
   create : myapp/bin/
   create : myapp/bin/www

   change directory:
     $ cd myapp

   install dependencies:
     $ npm install

   run the app:
     $ DEBUG=myapp:* npm start

➜  node_test █
```

router/index.js 수정



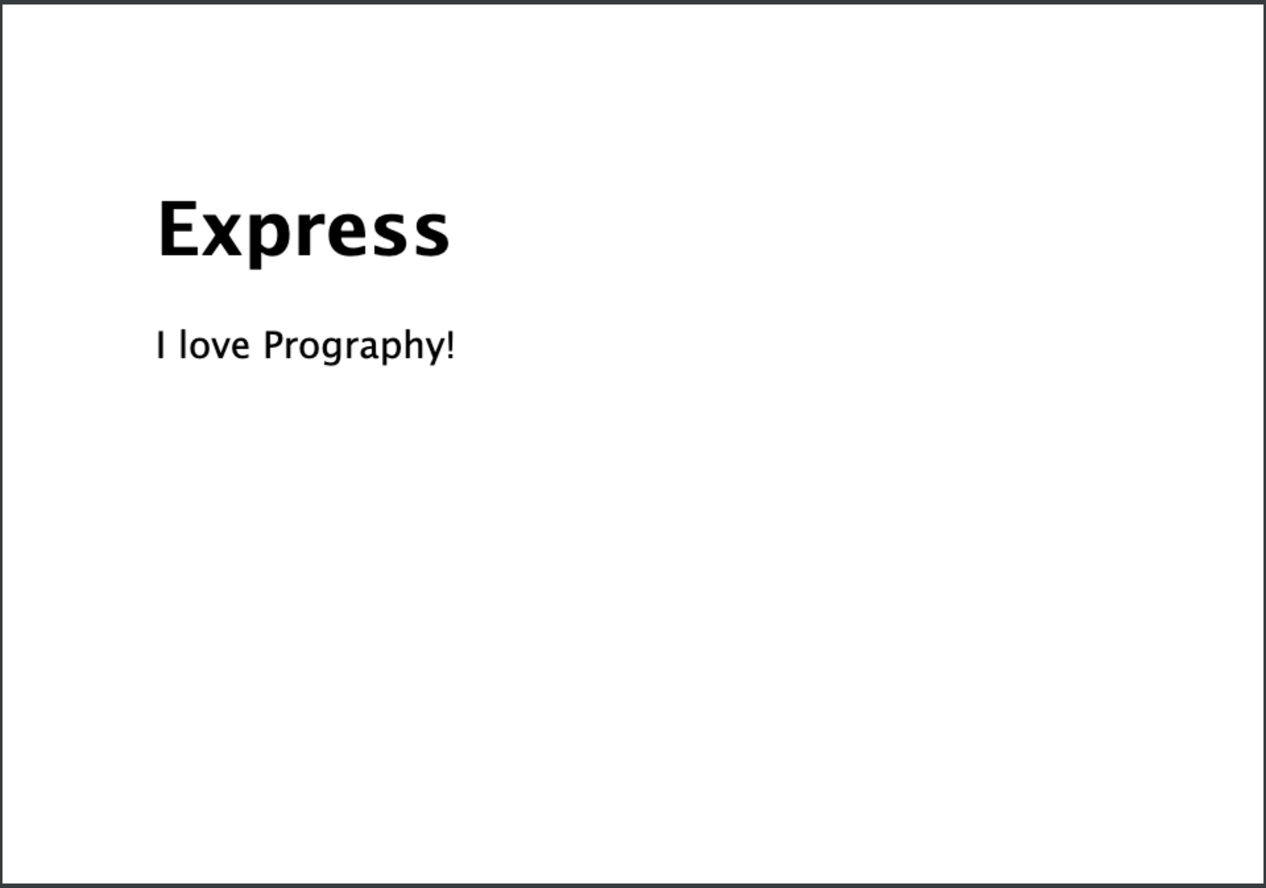
```
var express = require('express');
var router = express.Router();

const mysql = require('mysql');
const connection=mysql.createConnection({
            host: 'db1',
            port: '3306',
            user: 'root',
            password: '1234',
            database: 'test1'
        });


/* GET home page. */
router.get('/', function(req, res, next) {
    console.log(connection)
    connection.query('select * from table1;', function(err, rows){
        console.log(rows)
        if (err) {
            res.render('index', { title: 'Error', print_text: 'error' });
        }
        console.log(rows[0].print_text);
        res.render('index', { title: 'Express', print_text: rows[0].print_text });
    })
});

module.exports = router;
```

views/index.ejs 수정



```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p><%= print_text%></p>
  </body>
</html>
```

실행



Express

I love Prography!

끝~~

결론

Dockerfile을 이용하자

- 해당 프로그램을 실행시키기위한 환경을 한눈에 볼 수 있다.
- 설치 순서를 통해 어떤 프로그램이 돌아가는지 확인할 수 있다.
- 소스파일의 명시 - 어떤 소스파일이 뭔지 알 수 있음
- 환경변수, 포트등 환경변수 확인 가능