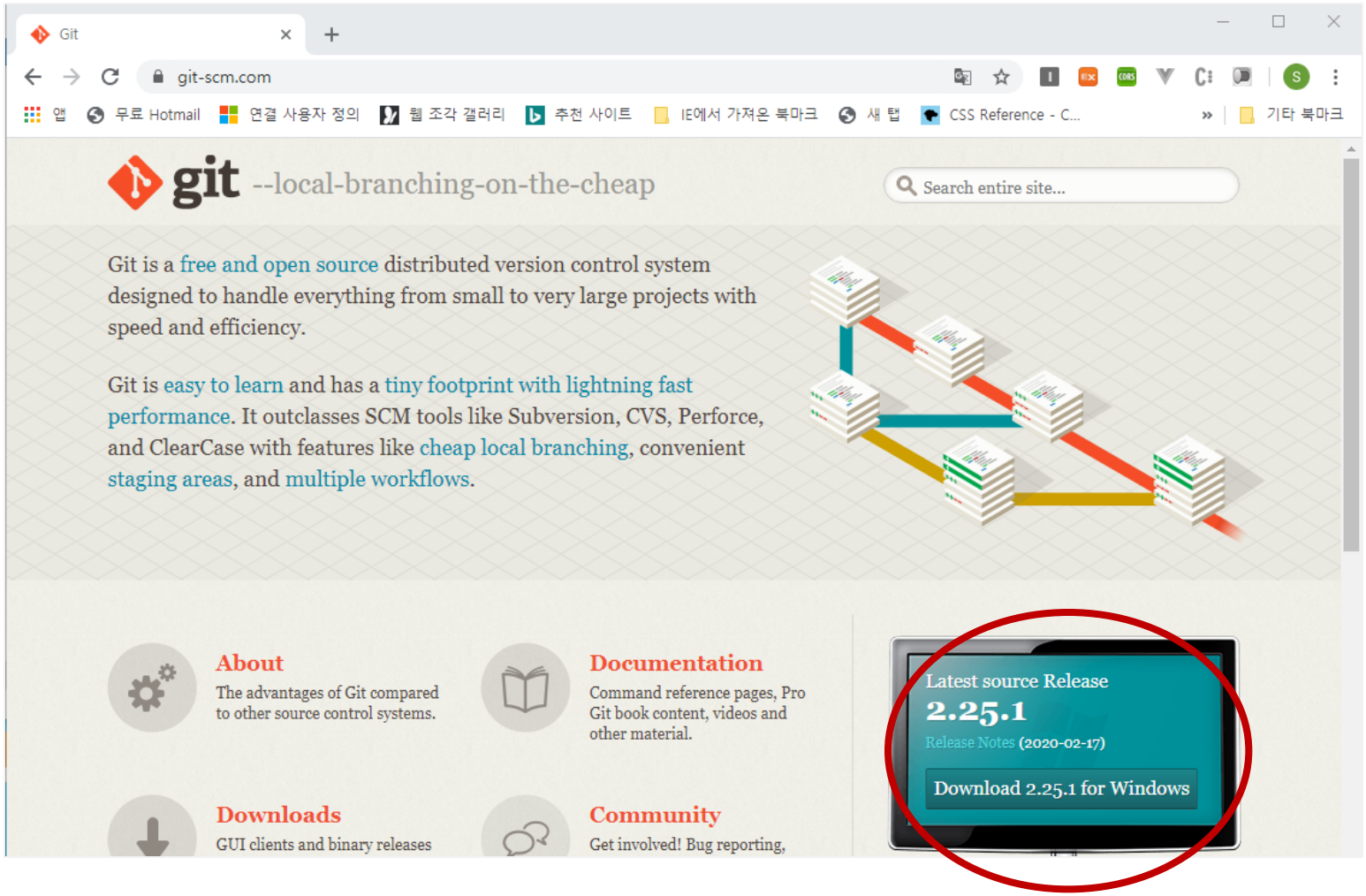
6장. mongodb Atlas cloud upload

순 서

I. 사전 작업

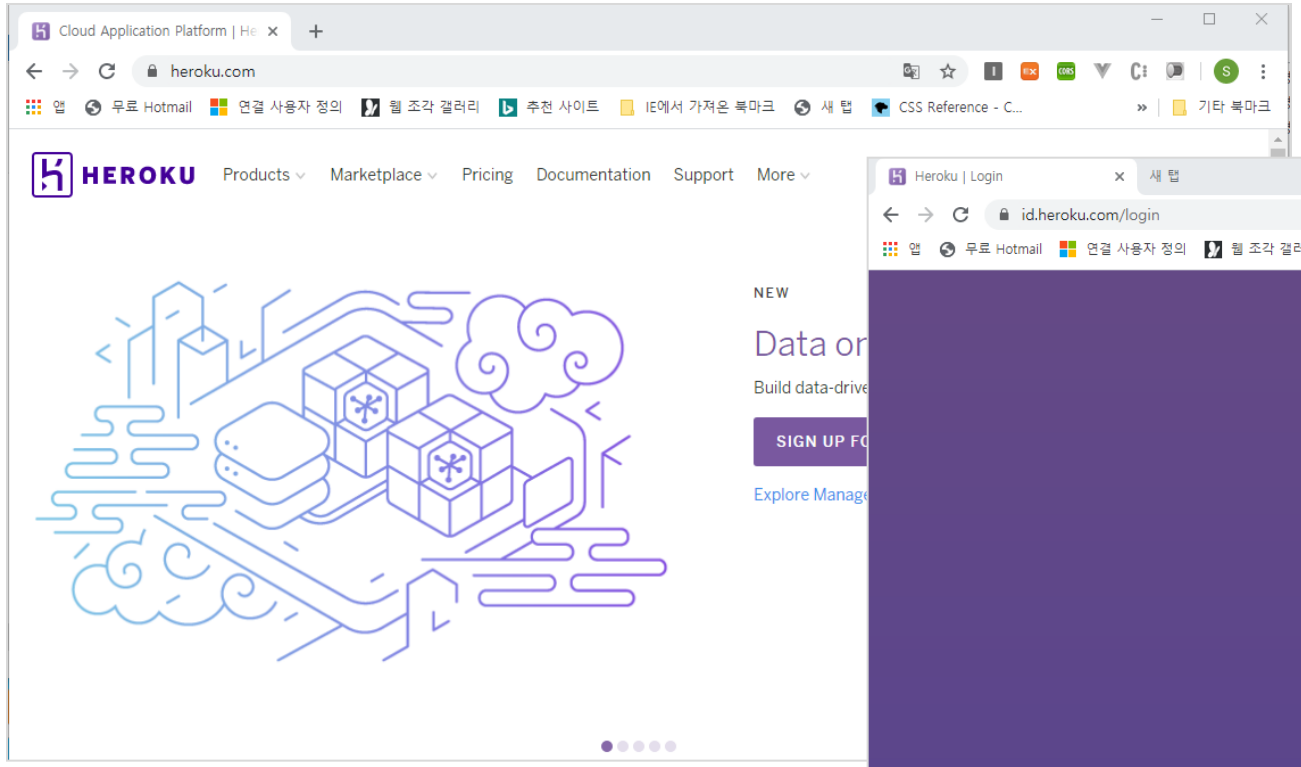
Preparing the Application - Git 설치

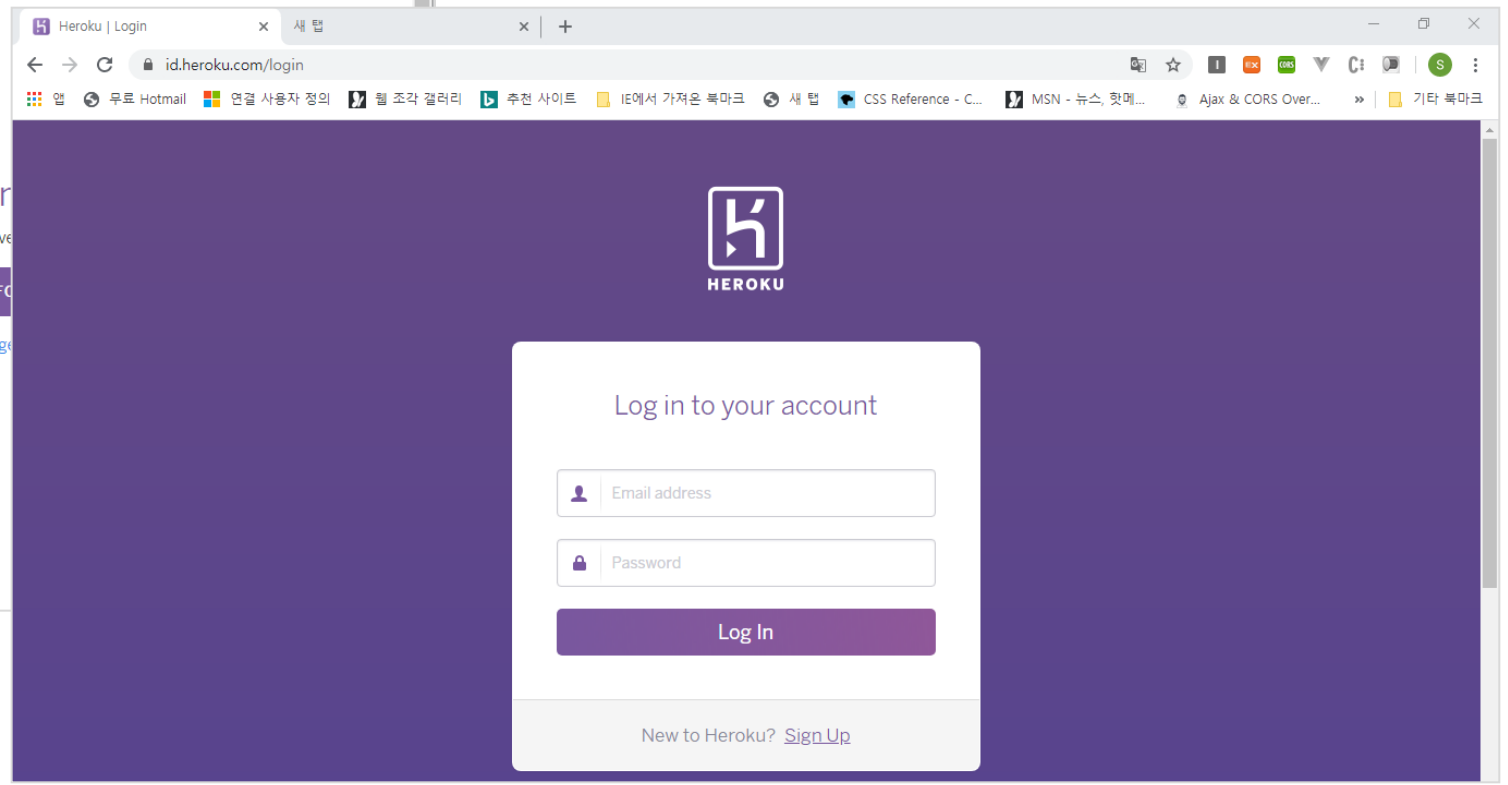
- Git 설치 : https://git-scm.com/downloads



Creating a Heroku account - Heroku에 Account 만들기

- 헤로쿠 - https://www.heroku.com/
- Heroku에 Account 만들기 - 인적사항기록 - (shyoon63@gmail.com /****@**23)

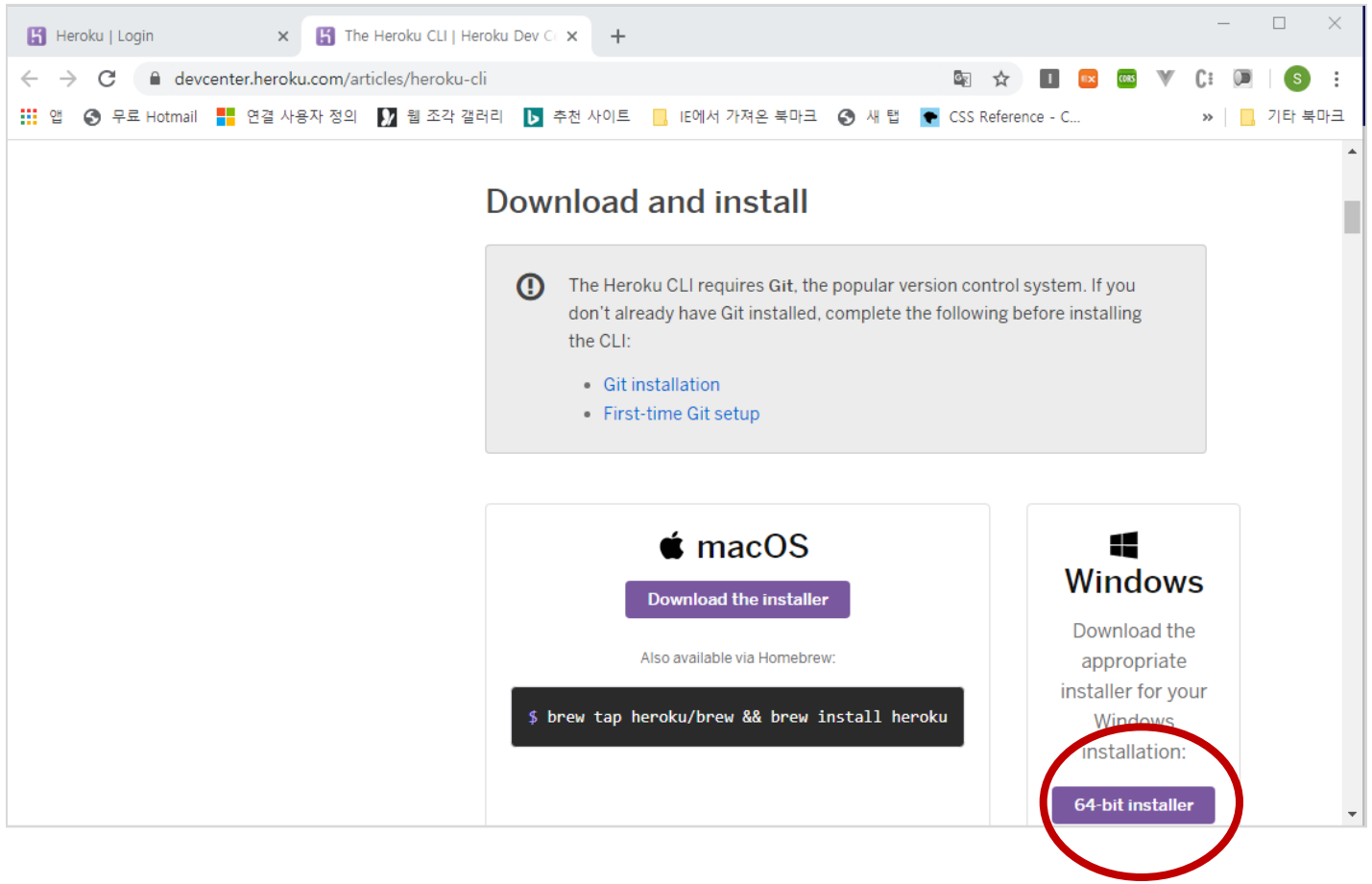




mail로 back해서
Link click하여 인증하기!!!

Installing the Heroku CLI

- Heroku 서비스를 사용하기 위해서는 Heroku CLI를 설치
- command line interface 가능하게 하는 것
- https://devcenter.heroku.com/articles/heroku-cli - Windows 64-bit Installer 선택 -> 설치



nodejs heroku hosting(1/2)

step1:

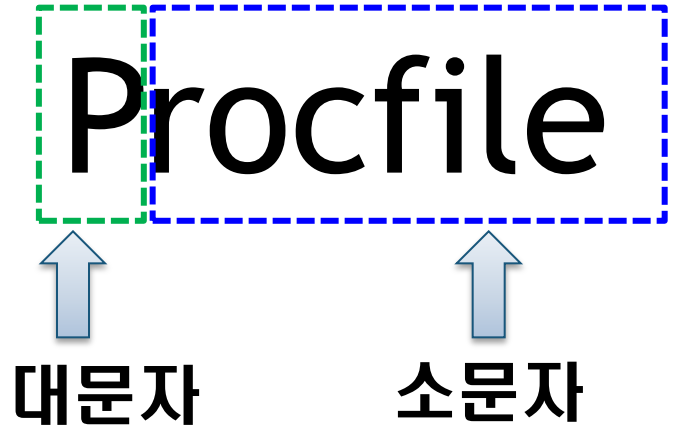
C:\nodejs\sunnysoft>

C:\nodejs\sunnysoft>Procfile <= 확장자 없이 대소문자 구분해서 만들기, 자동실행

web: npm start

Procfile

web: npm start



step2:

.gitignore 파일 만들기: <= heroku 에 모두 있으므로 무시하라는 의미

Dependency directory

node_modules

npm-debug.log

.gitignore

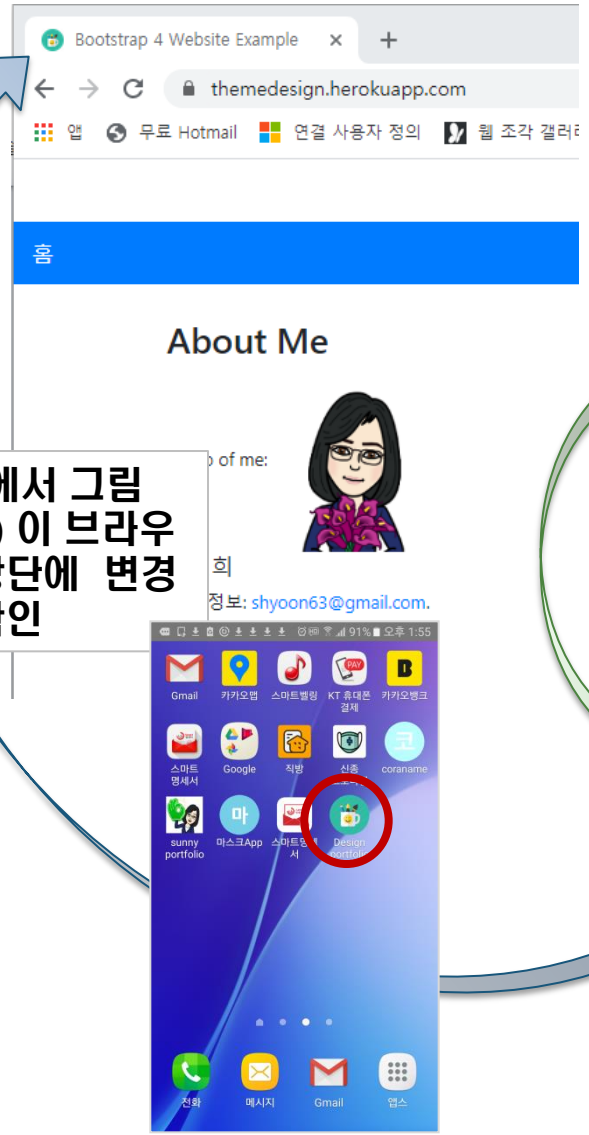
Dependency directory
node_modules

nodejs heroku hosting(2/2) - package.json 내용 추가

```
{
  "name": "express_example",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "start": "node app.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "cookie-parser": "^1.4.4",
    "cors": "^2.8.5",
    "errorhandler": "^1.5.1",
    "express": "^4.17.1",
    "express-error-handler": "^1.1.0",
    "express-session": "^1.17.0",
    "fs": "0.0.1-security",
    "multer": "^1.4.2",
    "path": "^0.12.7"
  }
}
```

"start": "node app.js",

index.html - 모바일화면에서 사용할 로고 만들고 홈화면 추가



로컬실행에서 그림(logo.png) 이 브라우저 좌측 상단에 변경된 것을 확인

- https://sunnyportfolio.herokuapp.com 사이트만들기
  - 구글 정책에서 https만 되도록 바꿈
  - heroku에 sunnyportfolio (app이름)만들기 – 학생 개인별로 다르게 만들기

- 아래와 같이 로그파일을 만든 후 /imgs에 저장하고 index.html 소스 수정
  - Android `<link rel="shortcut icon" href="./logo.png">`
  - Apple `<link rel="apple-touch-icon" href="./logo_apple.png">`

로고는 학생이 직접만들기

logo.png

Android 기기: 192 X 192 픽셀

logo_apple.png

Apple 기기: 180 X 180 픽셀

뒤에 배경이 있어야 없으면 검정색으로 표시

홈화면 추가

1)안드로이드Mobile phone – url 입력 후 메뉴(⋮)– 홈화면추가 – 이름 붙이기(Map Tracker)

2)아이폰Mobile phone – 사파리브라우저(나침판)–url 입력 후 –가운데 밑에 버튼(⬆) 누른 후– 홈화면추가

핸드폰 쿠키 삭제

- Chrome 앱의 경우
  1. Android 스마트폰 또는 태블릿에서 Chrome 앱 을 엽니다.
  2. 오른쪽 상단에서 더보기 를 탭합니다.
  3. 방문 기록 인터넷 사용 기록 삭제를 탭합니다.
  4. 상단에서 기간을 선택합니다. ...
  5. '쿠키 및 사이트 데이터'와 '캐시된 이미지 또는 파일' 옆의 체크박스를 선택합니다.

- iPhone, iPad 또는 iPod touch에서...
  1. 방문 기록과 쿠키를 삭제하려면 설정 > Safari로 이동하여
  2. '방문 기록 및 웹 사이트 데이터 지우기'를 탭합니다.

index.html – 주소창을 보이지 않도록 하여 앱으로 보이도록 하기

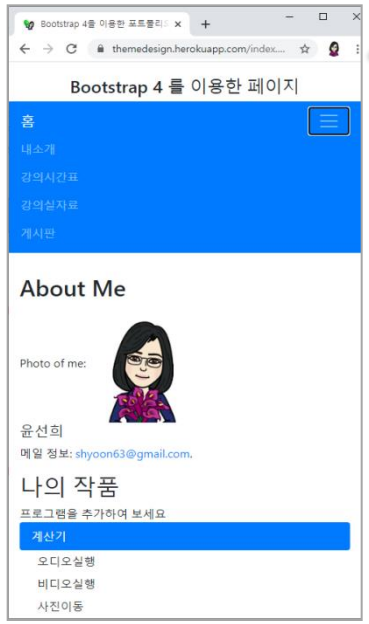
index.html

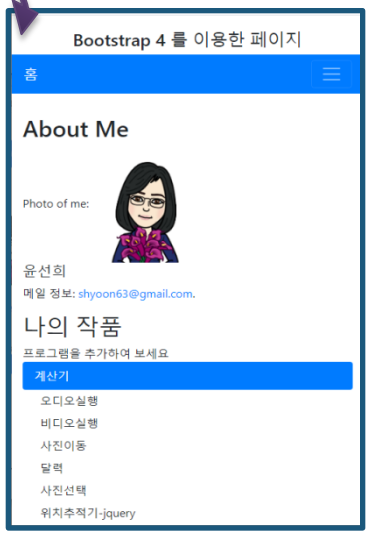
```
<meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1">
    <link rel="apple-touch-icon" sizes="180x180" href="imgs/logo.png" />
    <link rel="icon" type="image/png" href="imgs/logo.png" sizes="192x192"/>
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

모바일폰에서는 주소창이 나타나지 않도록 만들기

- Pt에서 복사하여 사용하면 보이지 않는 문자가 포함되어 실행이 안되는 경우 발생함!!!

데스크탑





모바일폰

몽고디비 설치

몽고디비설치- www.mongodb.com- software
-community server-오른쪽선택
-os:windowsx64x64 ms용선택-msi 파일 다운되면 설치-complete용으로 설치

환경변수-시스템변수 패스
-C:\Program Files\MongoDB\Server\4.2\bin
nodejs>database>local 디렉토리 만들어 놓기

새창에서 C:\nodejs> mongod --dbpath ./database/local

Git을 이용해서 heroku에 앱올리기 설명 - 빈칸없이 입력 - heroku에 upload됨

```
c:\>(base) heroku login
c:\>(base) git init
=>초기화

c:\>(base) heroku git:remote -a sunnyportfolio
c:\>(base) heroku git:remote -a sunnyboard
c:\>(base) heroku git:remote -a sunnysession
=>히로쿠에 올릴준비(heroku에 app이름  알려주기)

c:\>(base) git config --global user.email "shyoon63@gmail.com"
c:\>(base) git config --global user.name "shyoon63“
(Application Root $ git config -–global user.name "YOUR NAME")
=> 입력하라고하면 하기(email의   id = name)

c:\>(base) git add .
=>root 디렉터리 이하를 모두 git저장소에 추가한다.

c:\>(base) git commit -am "first"
=>git에 버퍼에 있는것을 stack에 올림-골라서 작업할 수 있도록 처리

c:\>(base) git push heroku master
=>heroku upload

c:\>(base) heroku open
=> 브라우저에서  자동실행
```

Deploying with git push

- step:
  - c:\>(base) heroku login

  - c:\>(base) git init
  - c:\>(base) heroku git:remote -a sunnyportfolio
  - c:\>(base) git config --global user.email "shyoon63@gmail.com"
  - c:\>(base) git config --global user.name "shyoon63"
  - c:\>(base) git add .
  - c:\>(base) git commit -am "first"
  - c:\>(base) git push heroku master

  - c:\>(base) heroku open

Deploying an Upgrade

- 내용 수정 후:

1. c:\>(base) git add .
2. c:\>(base) git commit -am "first"
3. c:\>(base) git push heroku master

c:\>(base) heroku open

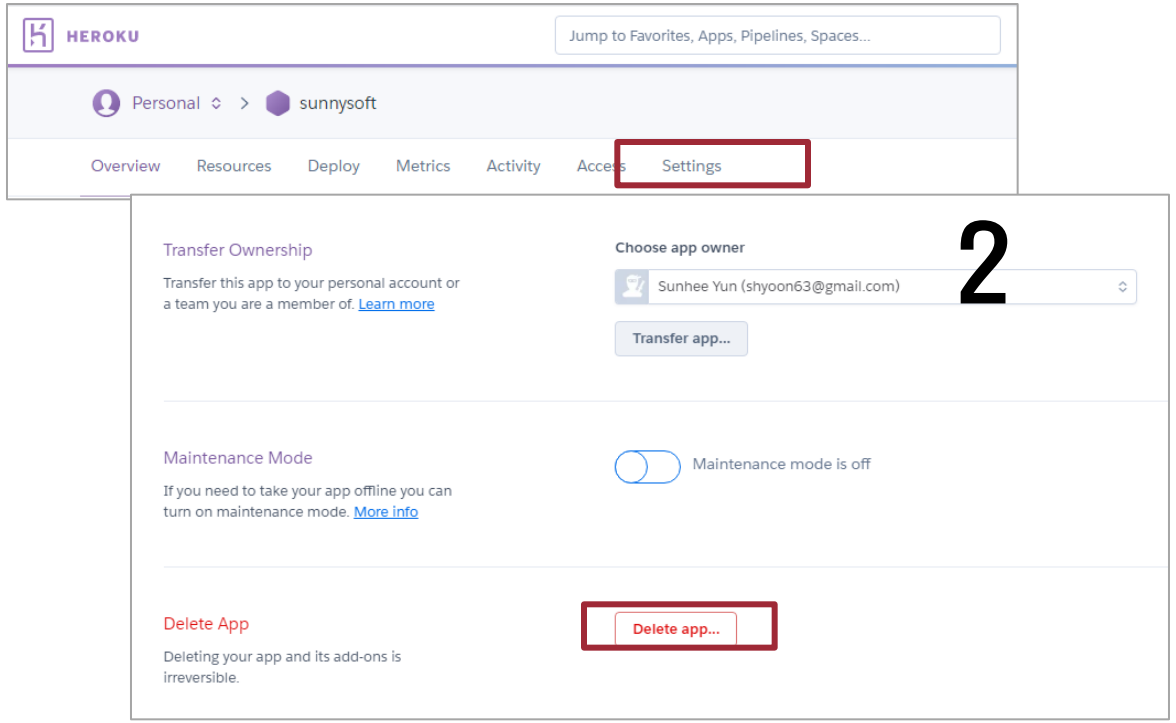
- 브라우저 자동샐행

1. heroku open
2. 또는 브라우저에서 https://sunnyportfolio.herokuapp.com

문제 발생시 해결 방법

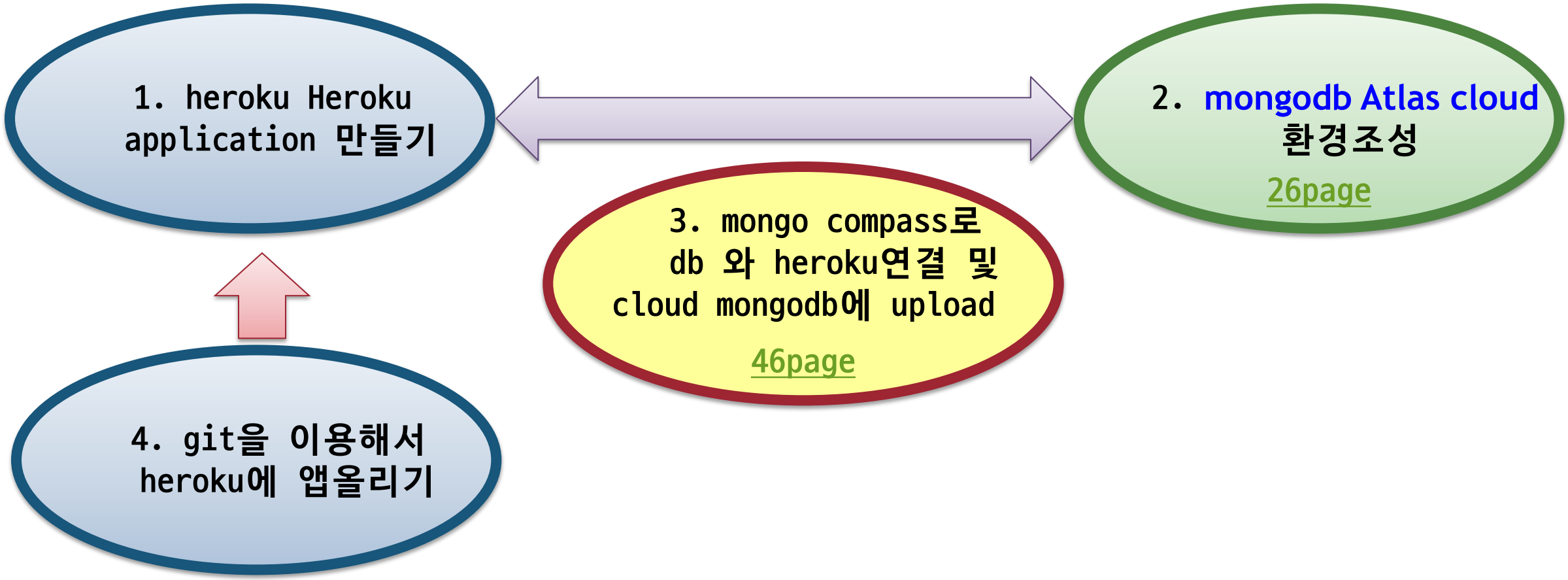
해결사항

1. git폴더 삭제
2. heroku app 삭제후 다시 만들고 처음부터 올리기



II. mongodb Atlas cloud 를 이용한 application upload

Upload 4단계

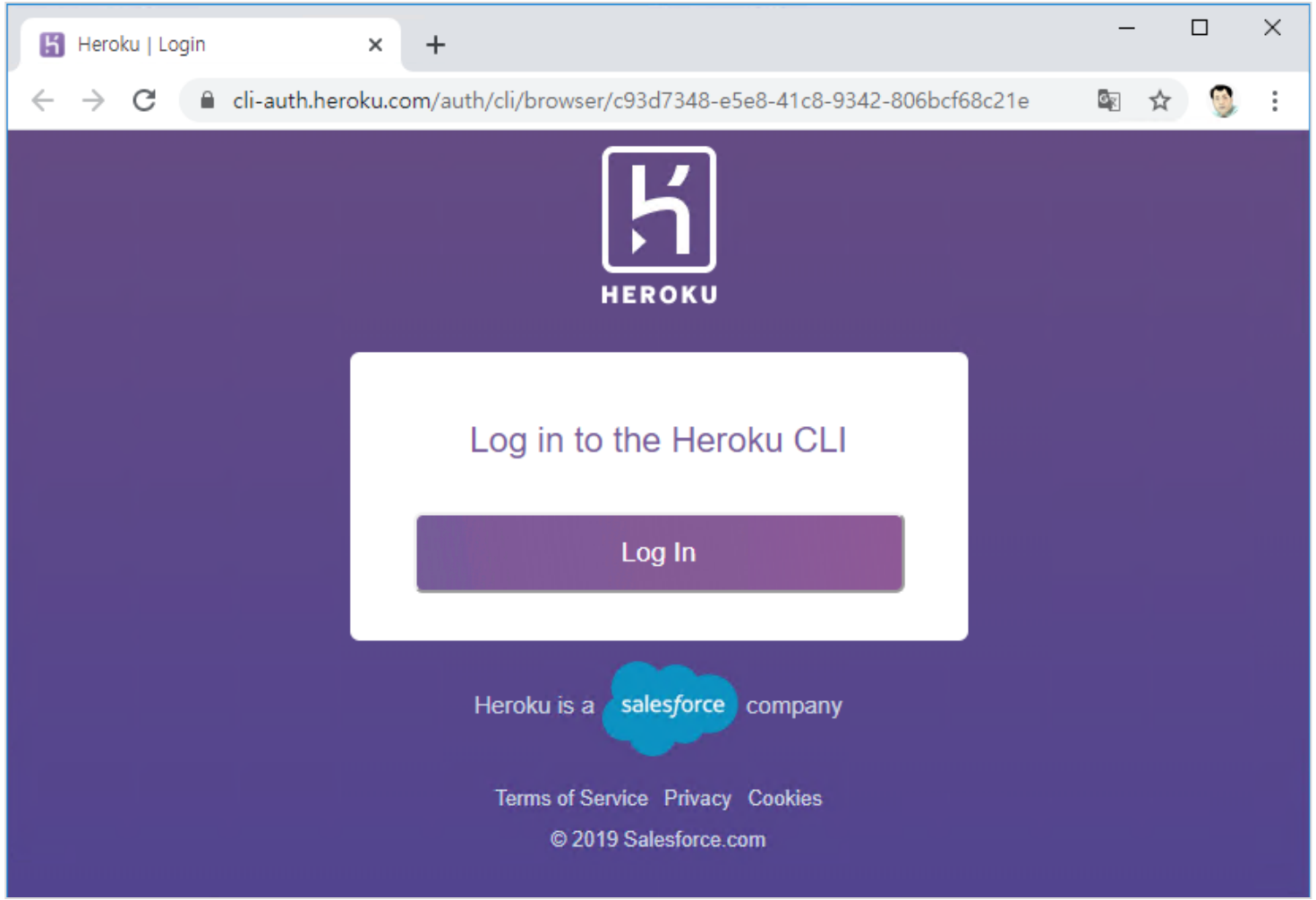

1. heroku Heroku application 만들기
2. mongodb Atlas cloud 환경조성
26page
3. mongo compass로 db 와 heroku연결 및 cloud mongodb에 upload
46page
4. git을 이용해서 heroku에 앱올리기

<실습> cloud system에 mongodb atlas를 활용한 application upload

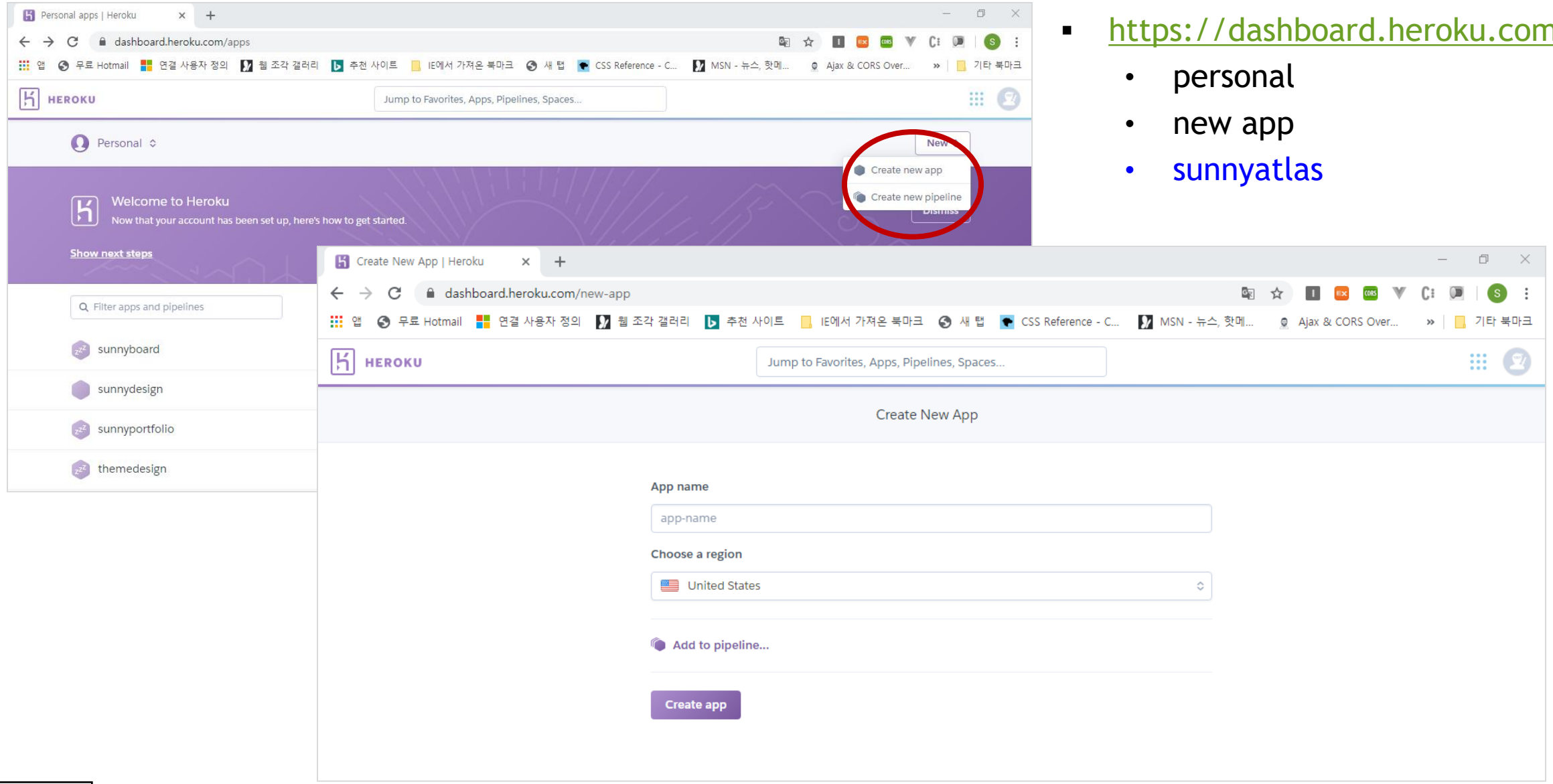
1. heroku Heroku application 만들기(20 slide)
2. mongodb Atlas cloud 환경 조성(26 slide)
3. mongo compass로 db 와 heroku연결 및 mongodb Atlas cloud 에 db upload(46 slide)
4. git을 이용해서 heroku에 앱올리기(69 slide)

1. heroku Heroku application 만들기

C:□(base)>heroku login

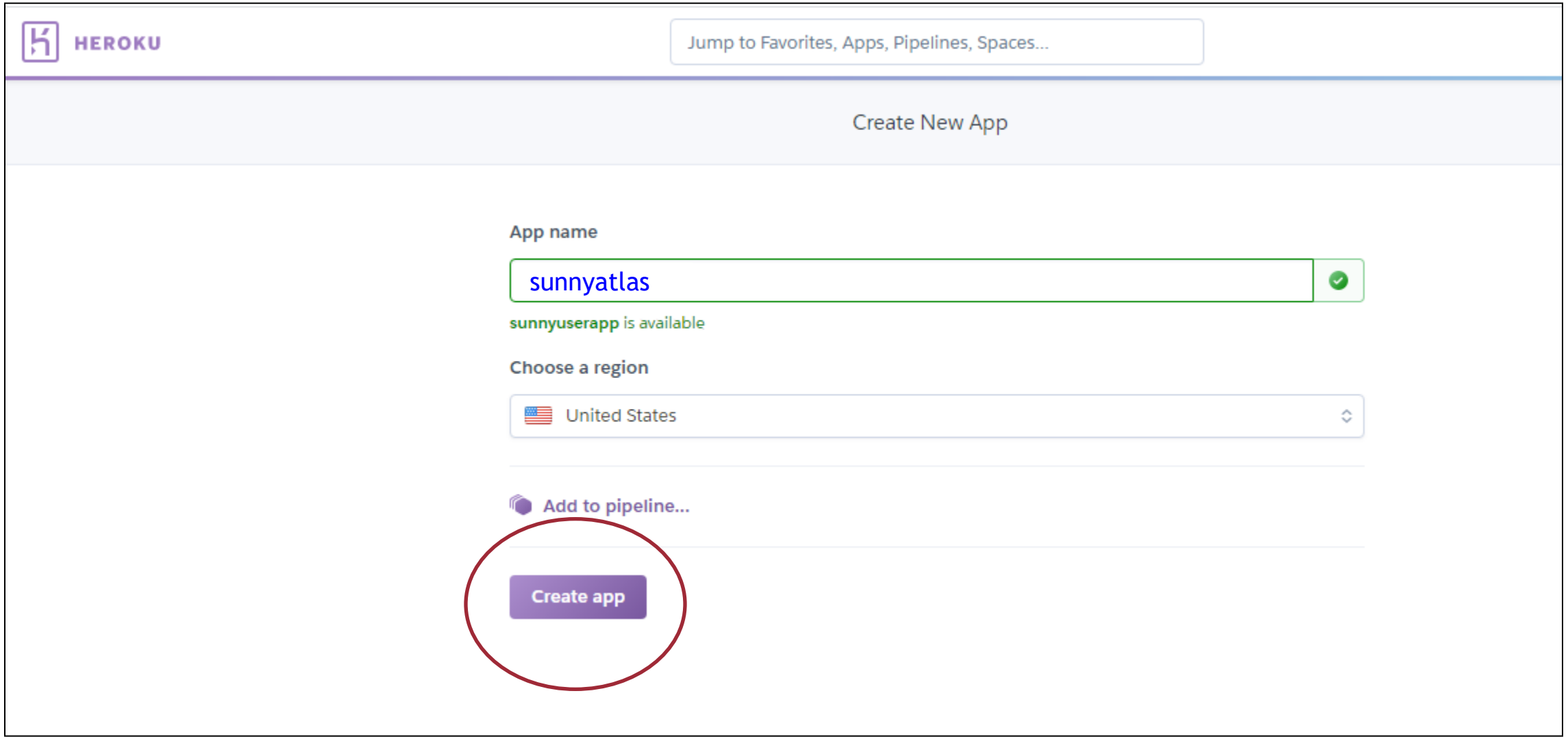


Creating an application



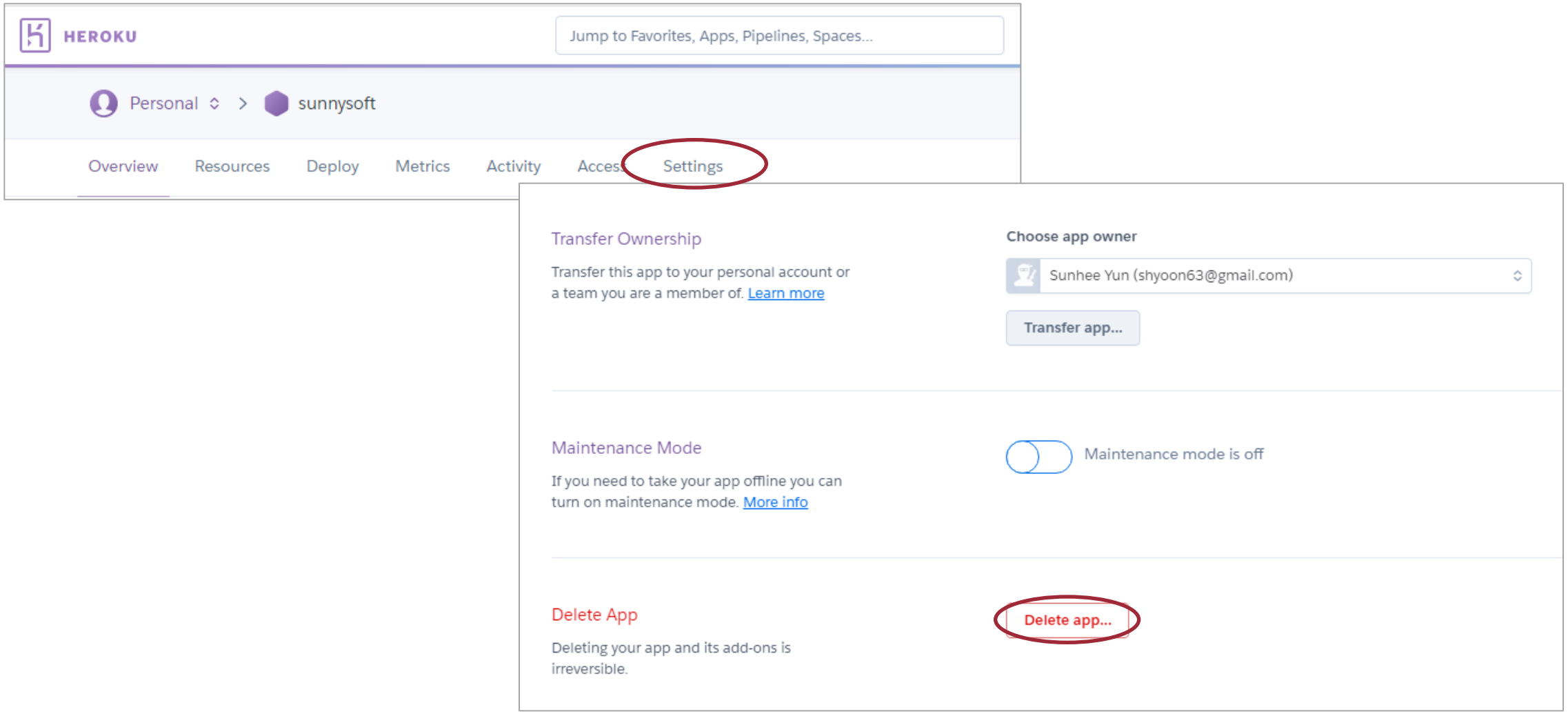
- https://dashboard.heroku.com/
  - personal
  - new app
  - sunnyatlas

heroku app 만들기



delete application – 문제 발생 시 지우고 다시 하기

- 앱을 선택후 settings – 맨아래 - delete application



2. mongodb Atlas cloud 환경조성

2. mongodb Atlas cloud 환경조성 요약

https://www.mongodb.com/cloud/atlas?jmp=docs
login (shyoon63@gmail.com /j**********3
구글 로그인후
Cluster=db
collection=table

step1) porject MyCloudAtlas 만들기=> project선택후

step2) Build a Cluster-free 선택
usa-virginia 선택-create cluster

step3) clusters-cluster0-connet
add your current ip address
add ip address
close

step4)왼쪽메뉴-network access-add ip address-
Whitelist Entry: 0.0.0.0/0
confirm

step5) db에 접근하기 위한것--dbuser 만들기
왼쪽메뉴-Database Access-Databaseusers
-Add New Database User - password
dbuser name : 폼내용: sunny5/***0414 - atlas admin 선택 -
AddUser 버튼

(왼쪽메뉴- settings -수정 가능- porject name :
MyCloudAtlas

step6) db 만들기 – 테이블을 계속 만든다면 step6 진행함
왼쪽메뉴-clusters-collections-add my own data

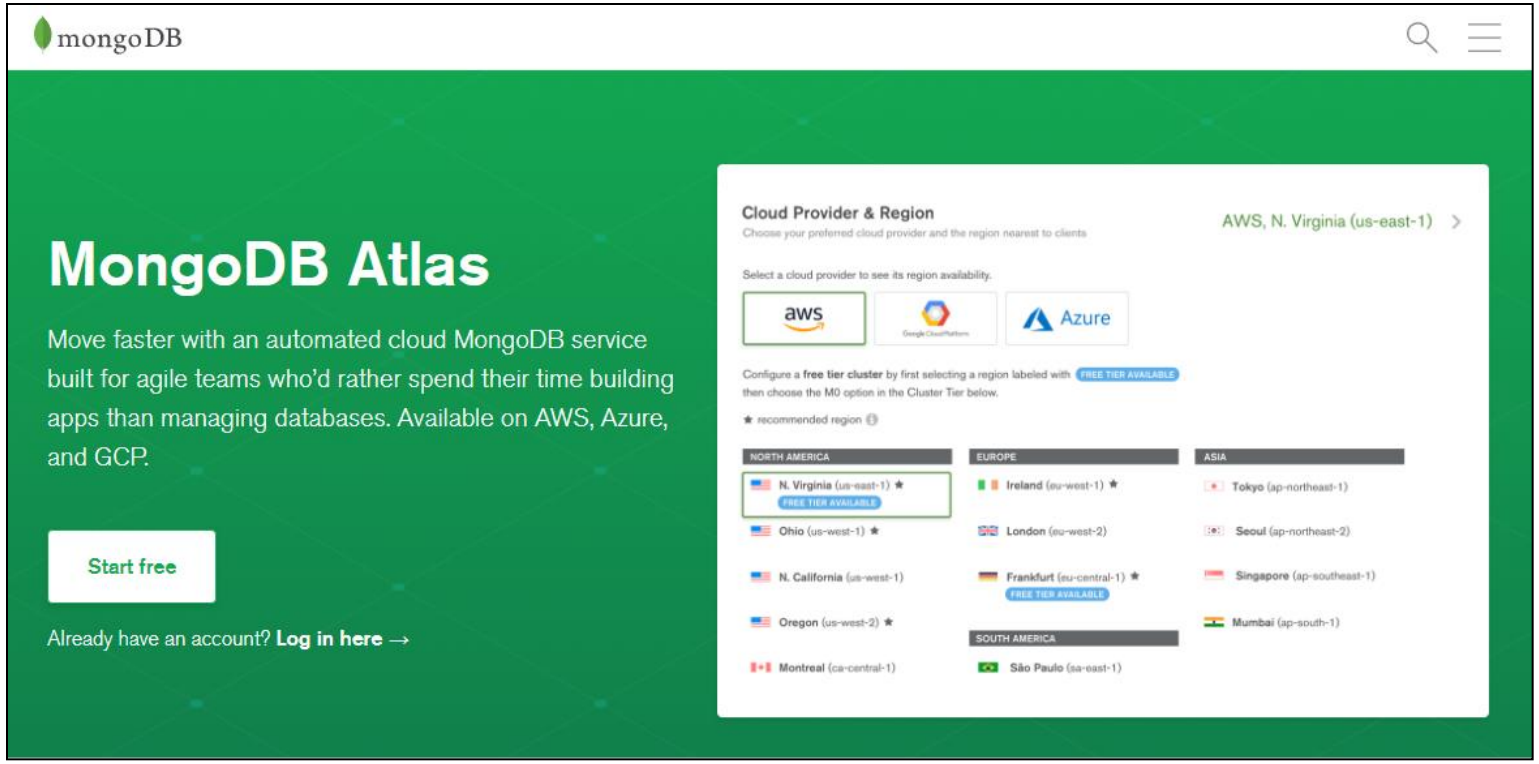
database name : myuserdb
collection name : users3

- cluster = db
- collection = table

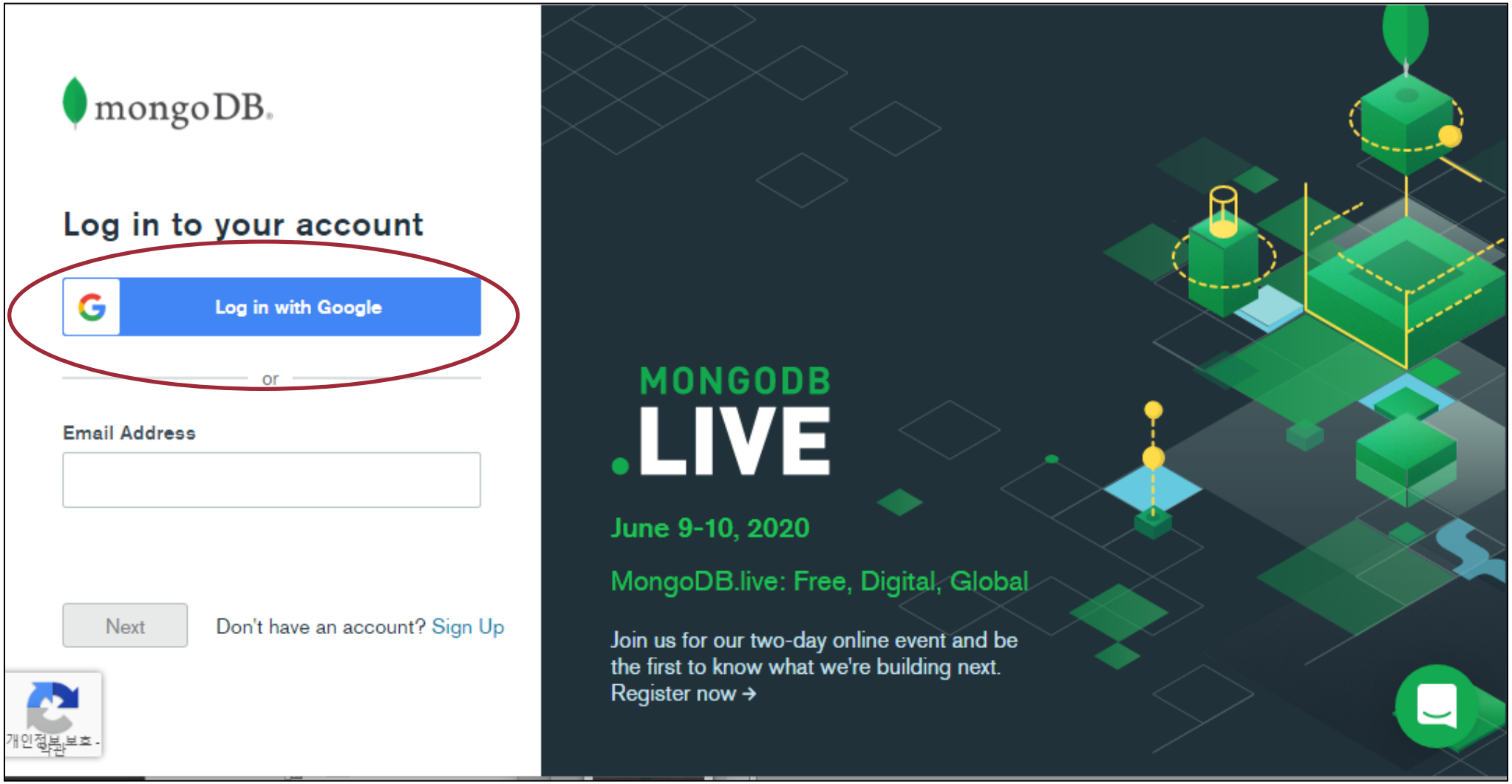
mongodb Atlas cloud : https://www.mongodb.com/cloud/atlas?jmp=docs

1. Navigate to Atlas to create your Atlas account.

You can register for an account on the MongoDB Atlas landing page.

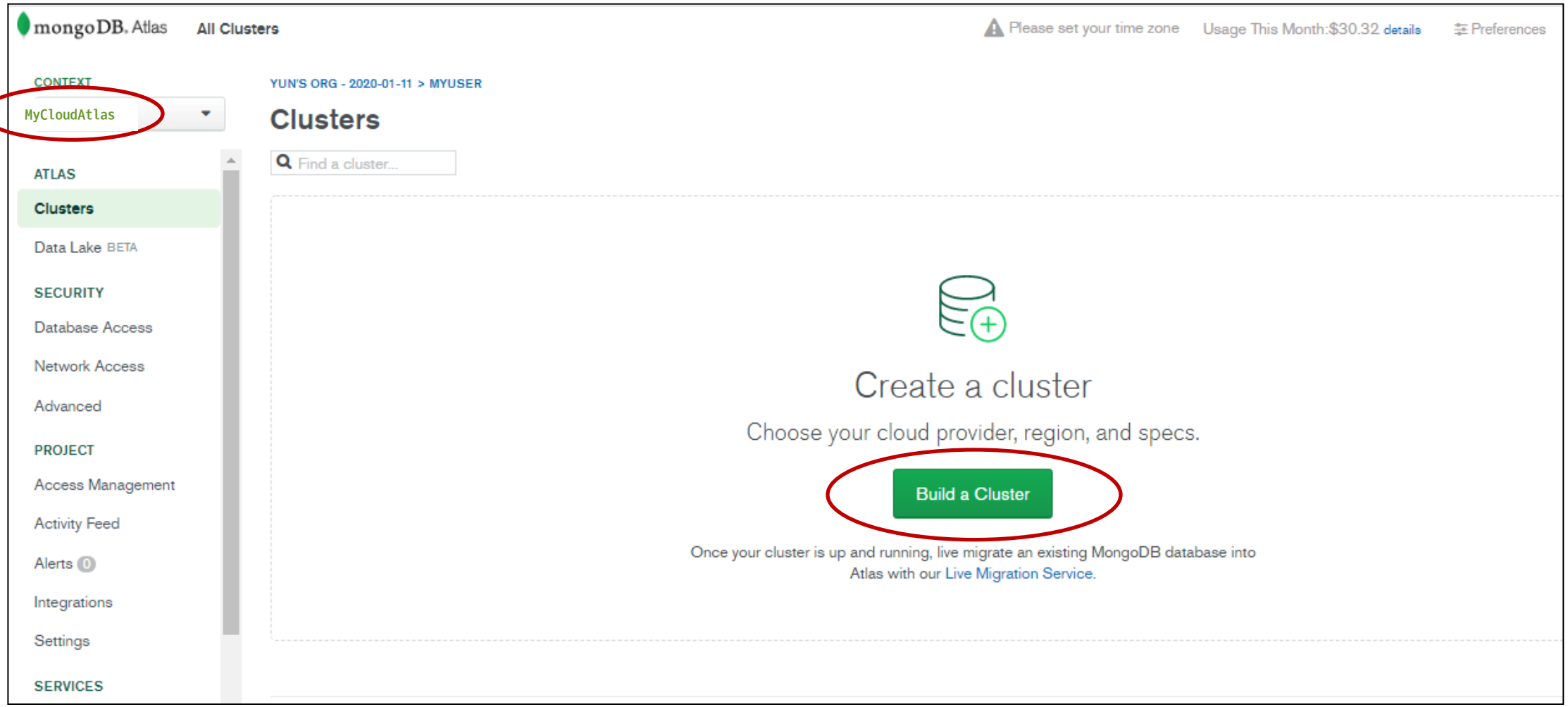


log in to your account

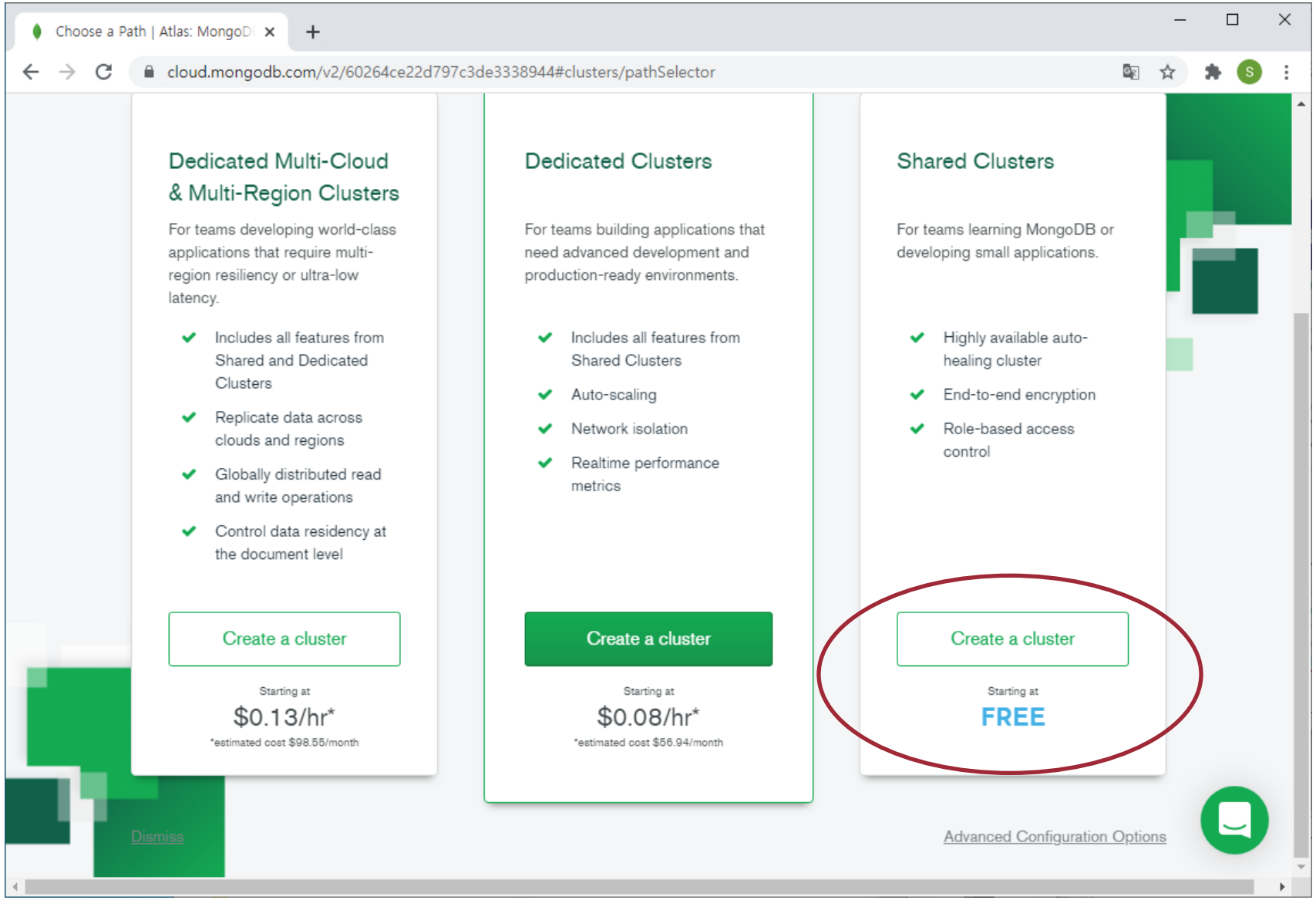


step1. porject MyUser 만들기

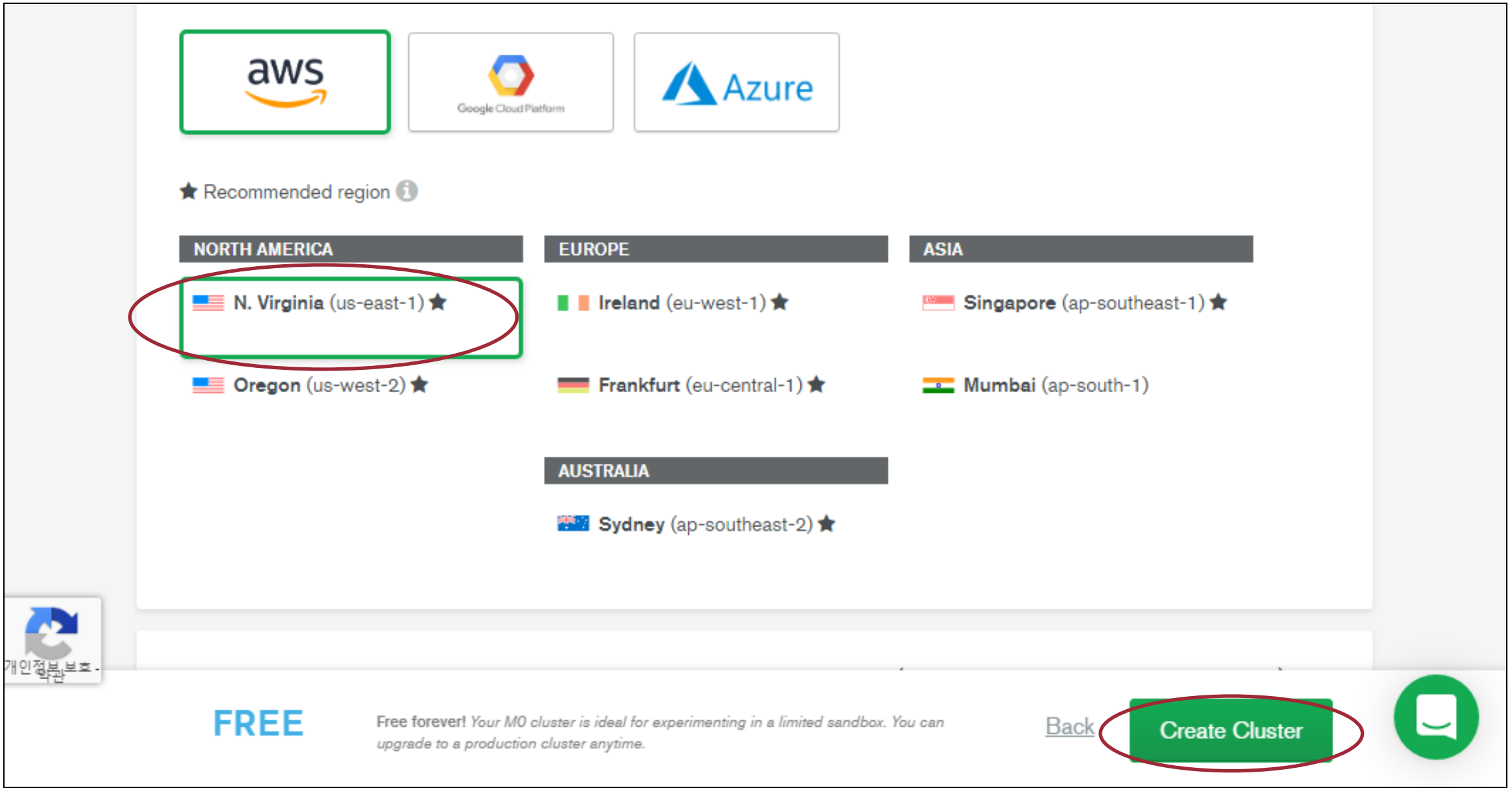
- Build a New Cluster



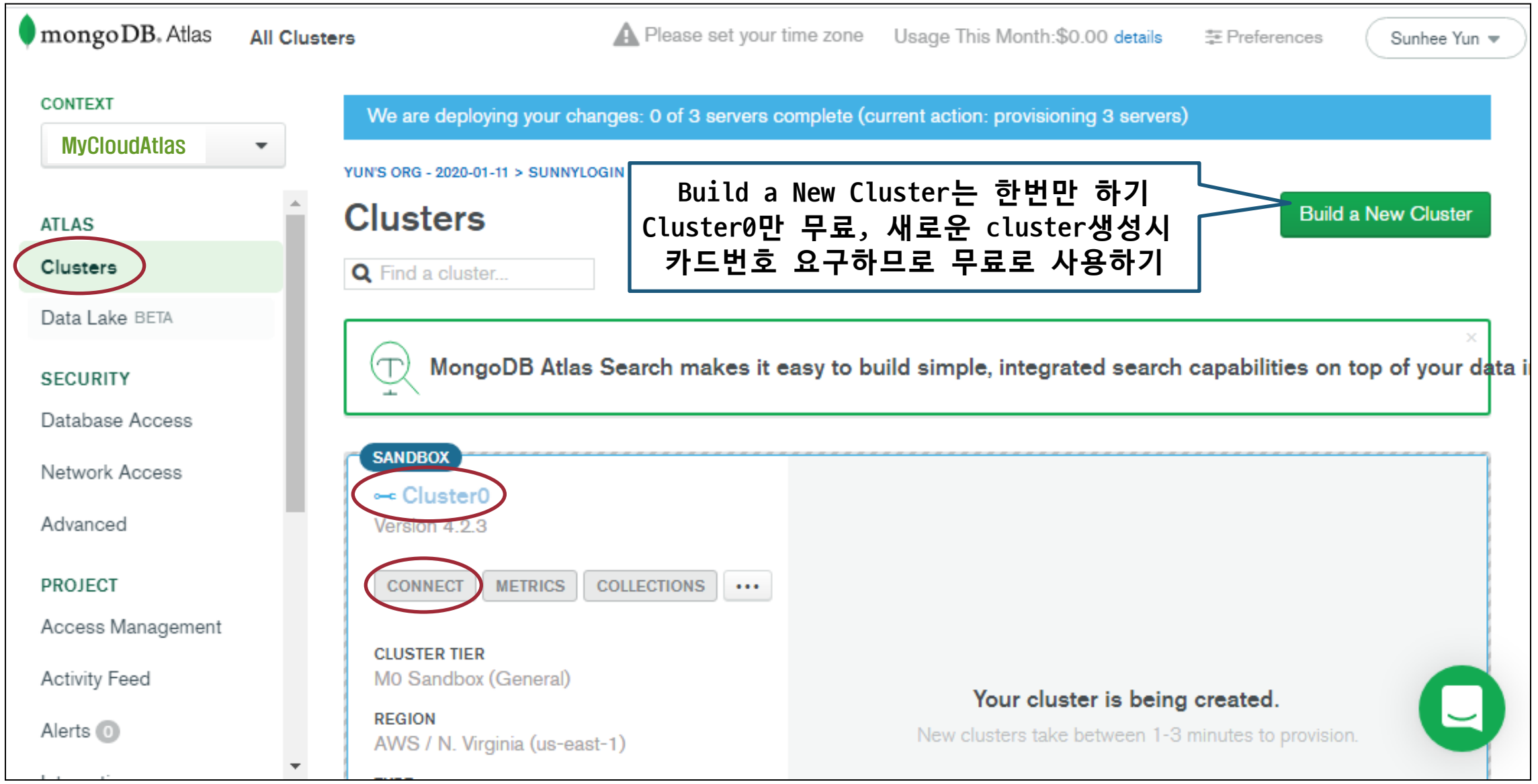
step2. Build a New Cluster - free 선택



usa-virginia

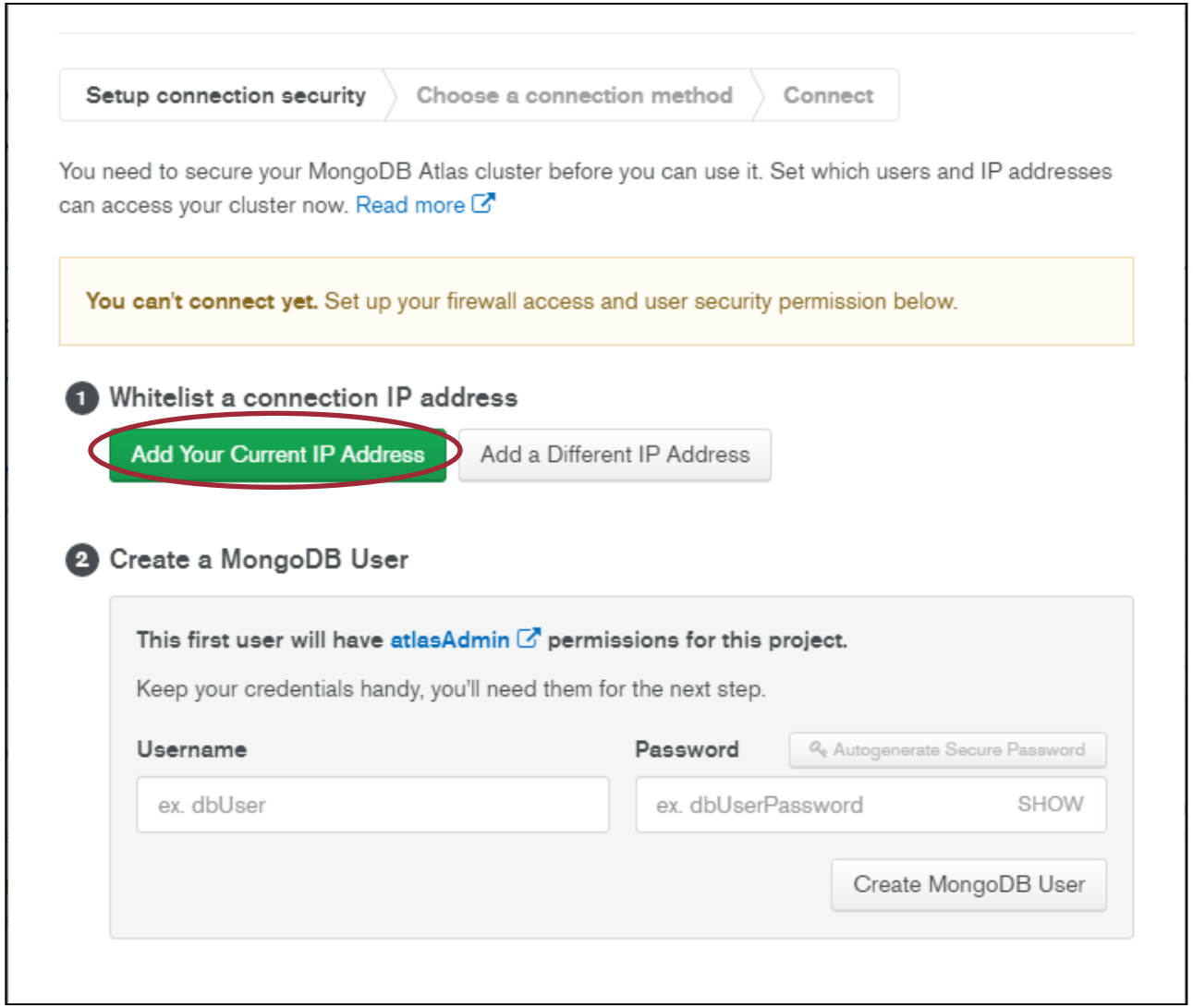


step3. clusters-cluster0-connet

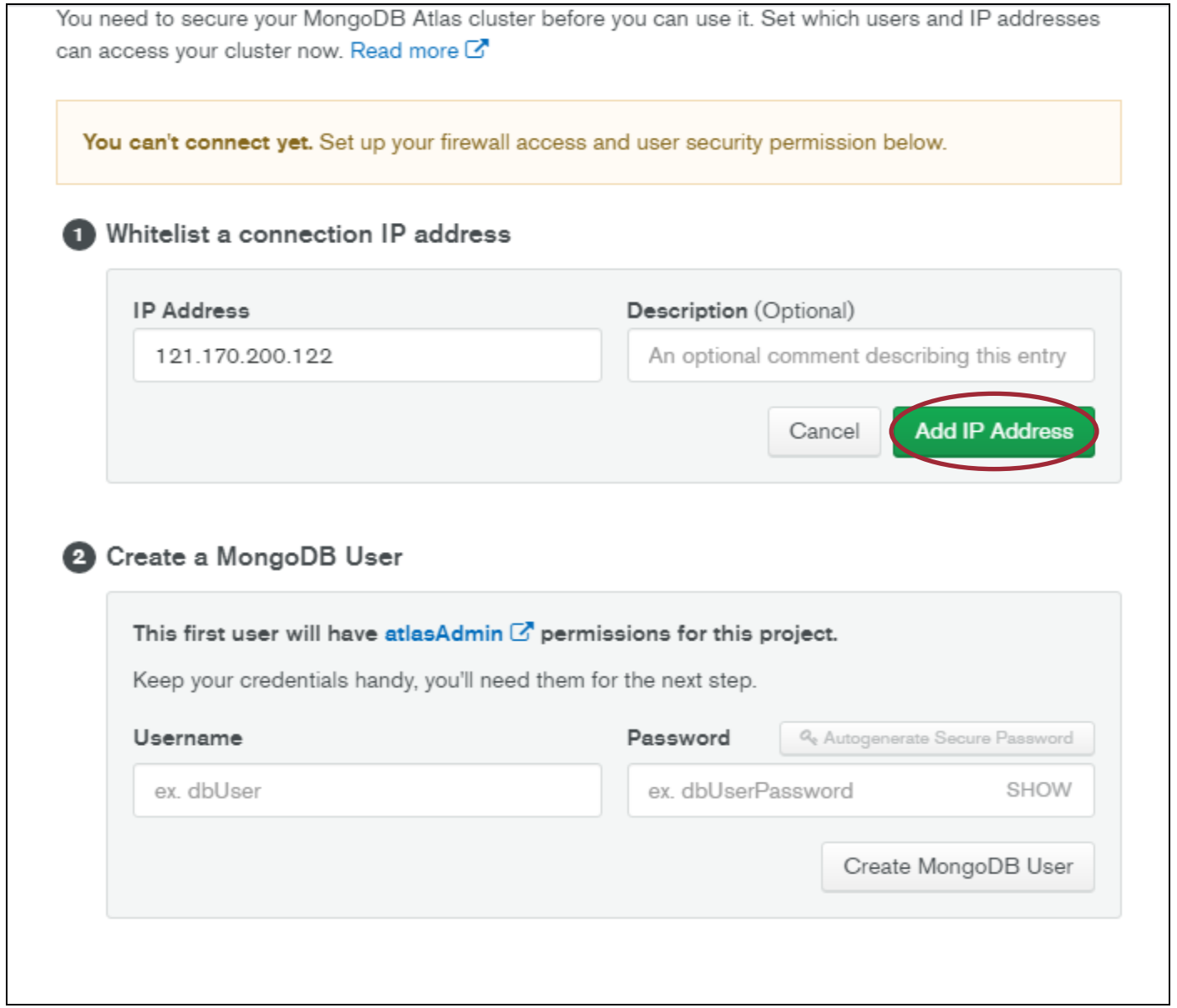


clusters-cluster0-connet

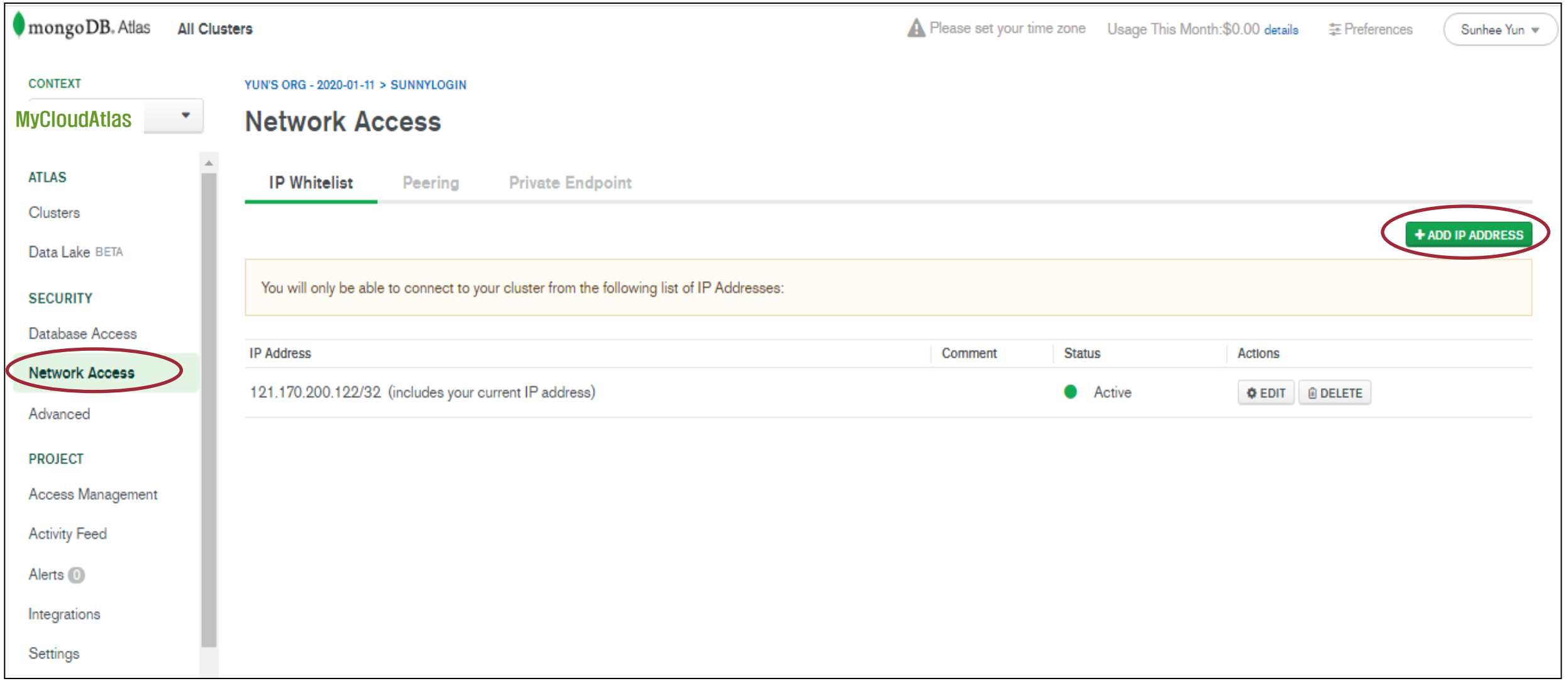
- add your current ip address
- add ip address
- close



add ip address

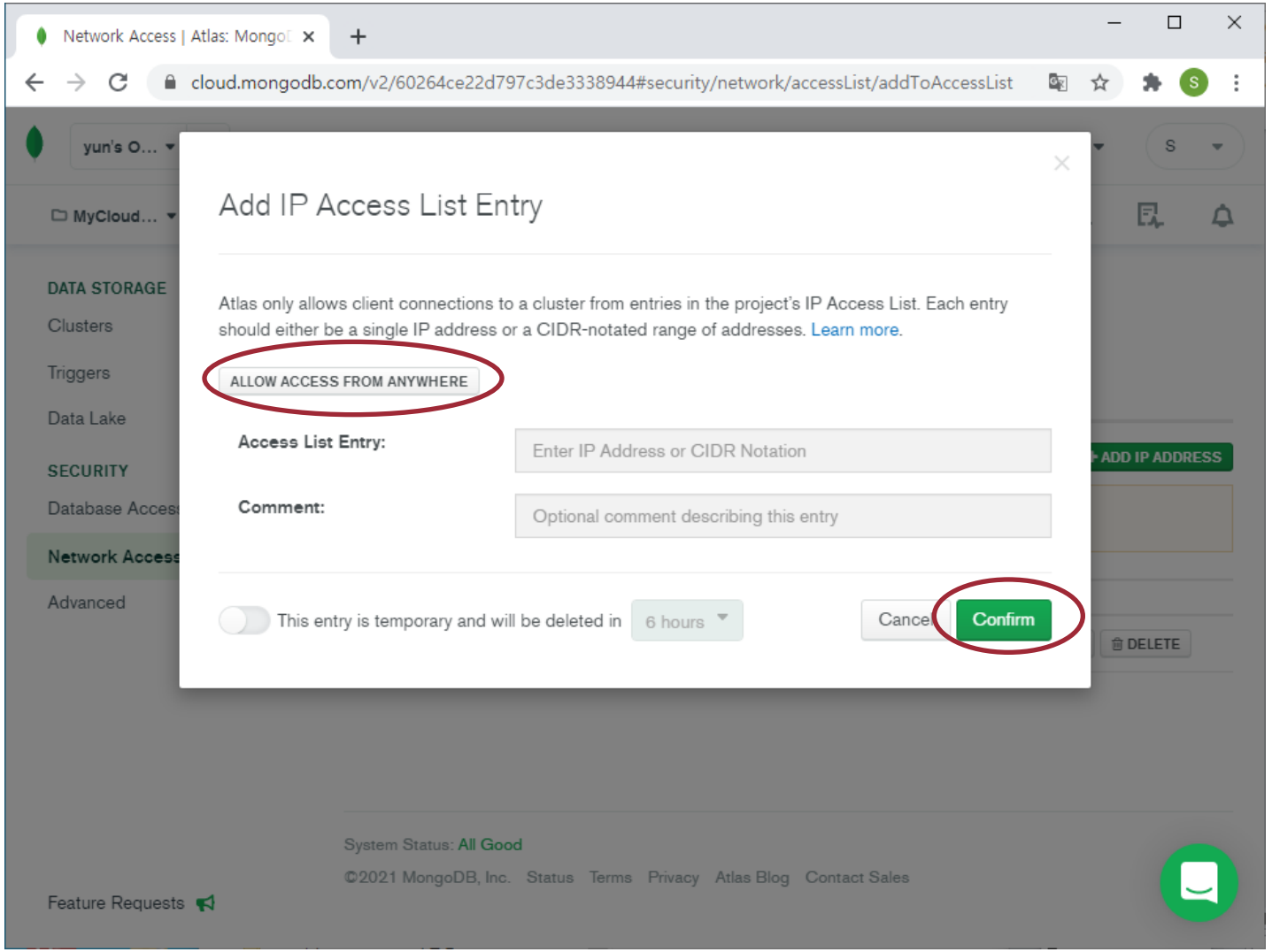


step4. 왼쪽메뉴 - network access-add ip address

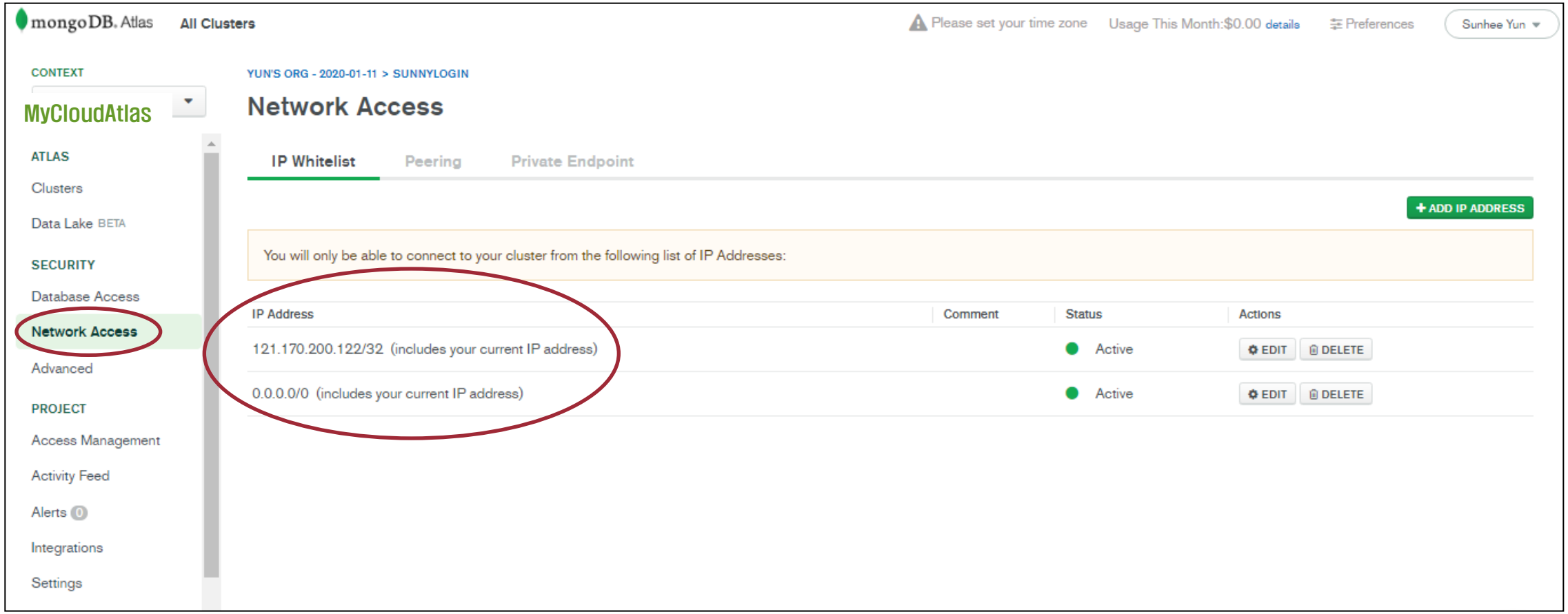


network access-add ip address

- Whitelist Entry: 0.0.0.0/0
- confirm

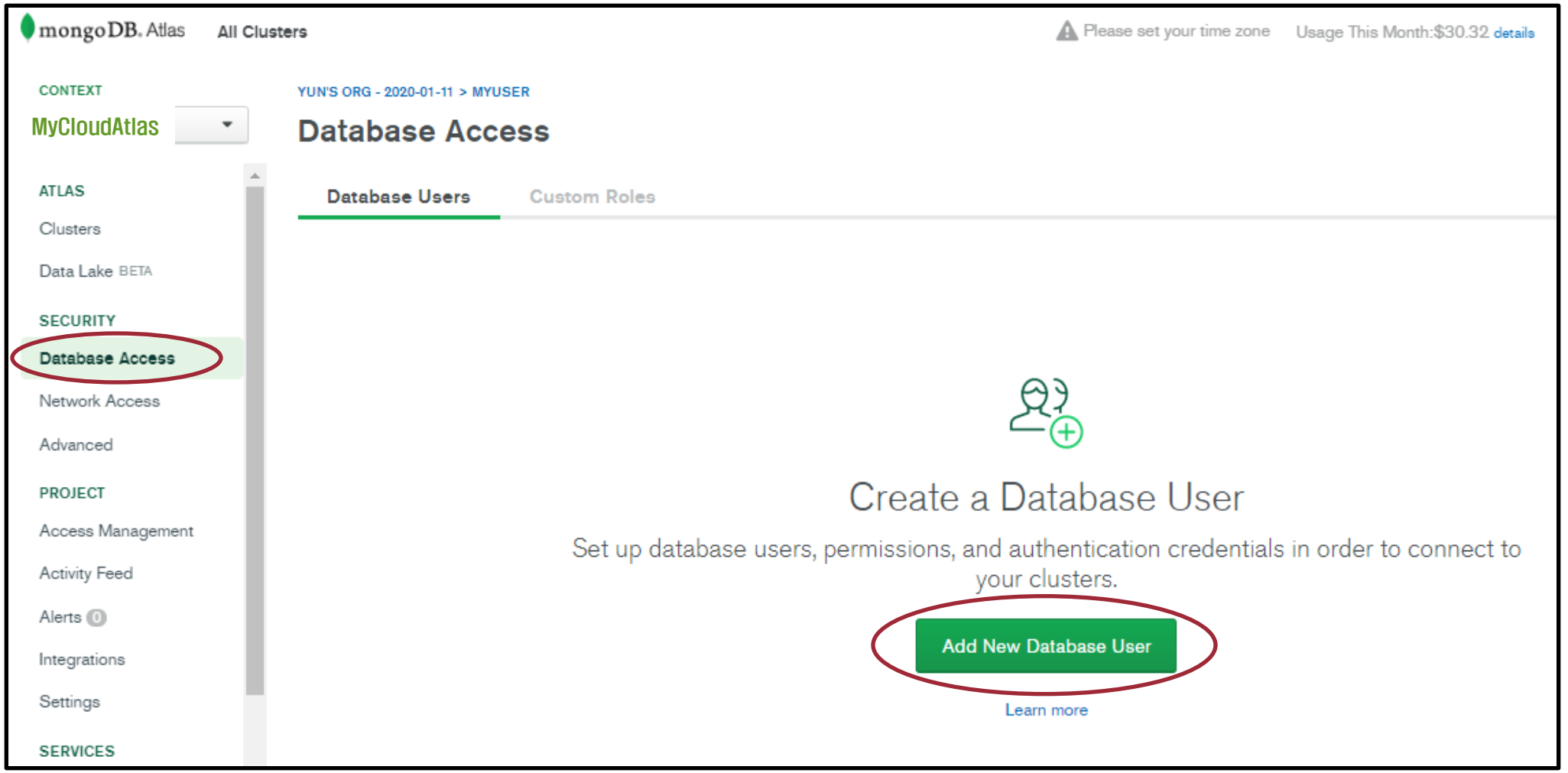


network access-add ip address 확인



step5. db에 접근하기 위한 것 - dbuser 만들기

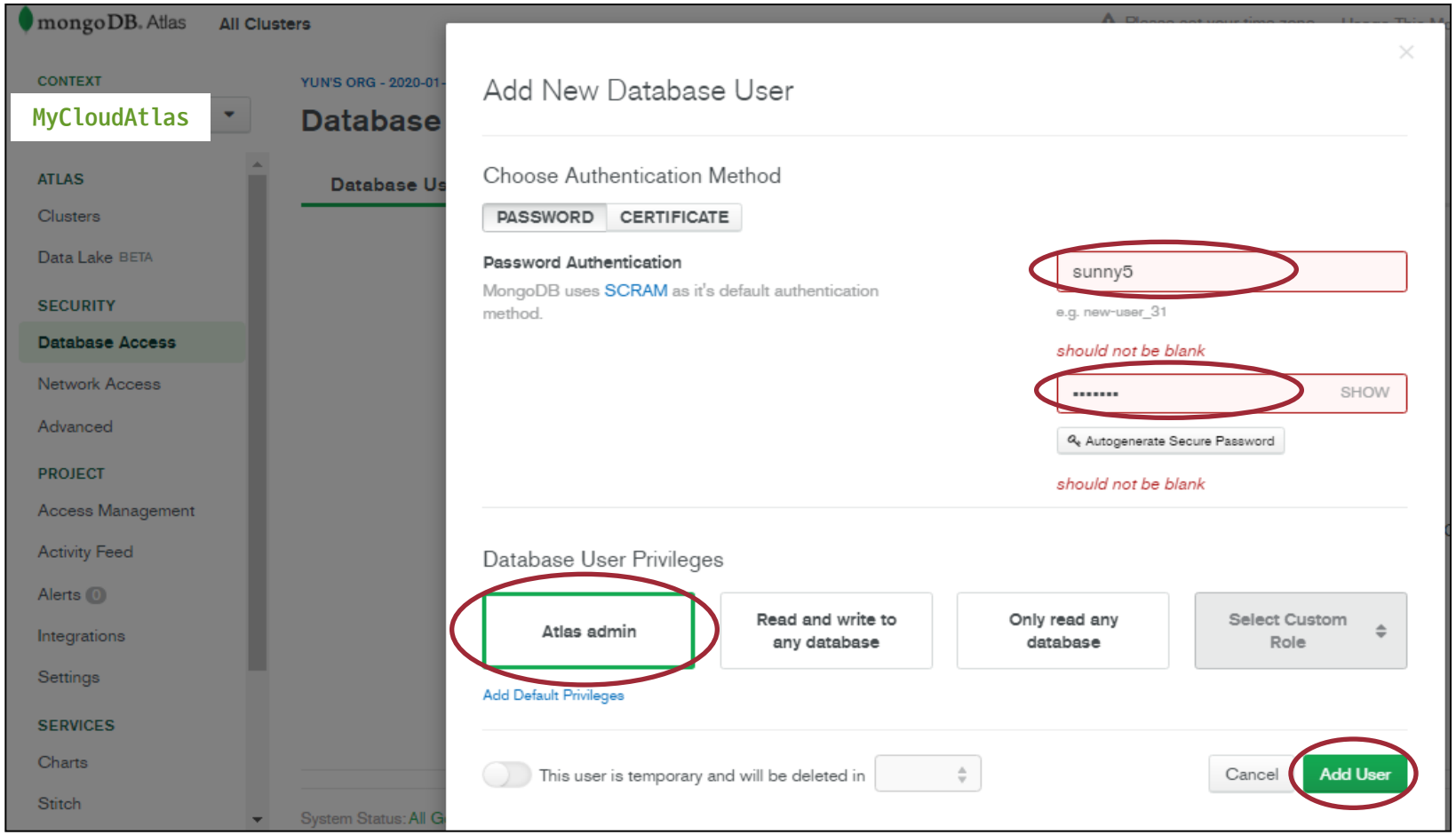
- Database Access-Databaseusers-add new database user



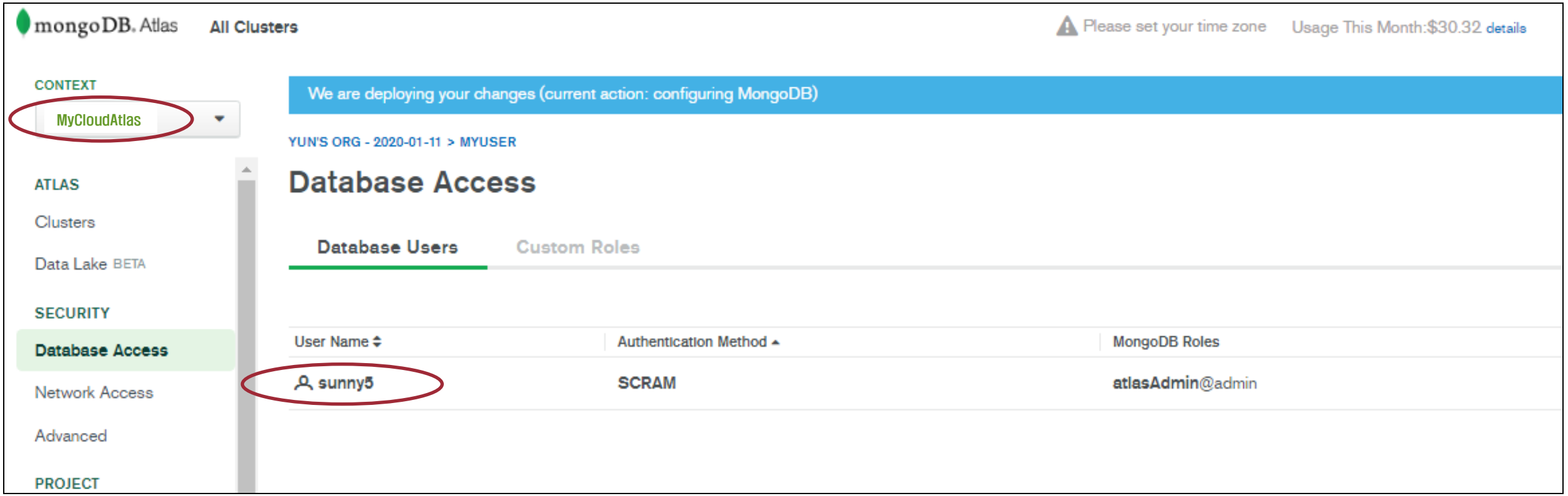
Add New Database user

- Add New Database
- username : sunny5/***0414
- Atlas admin - 폼내용 입력 - AddUser버튼
- MyCloudAtlas <= project이름

폼내용:
username: sunny5/***0414

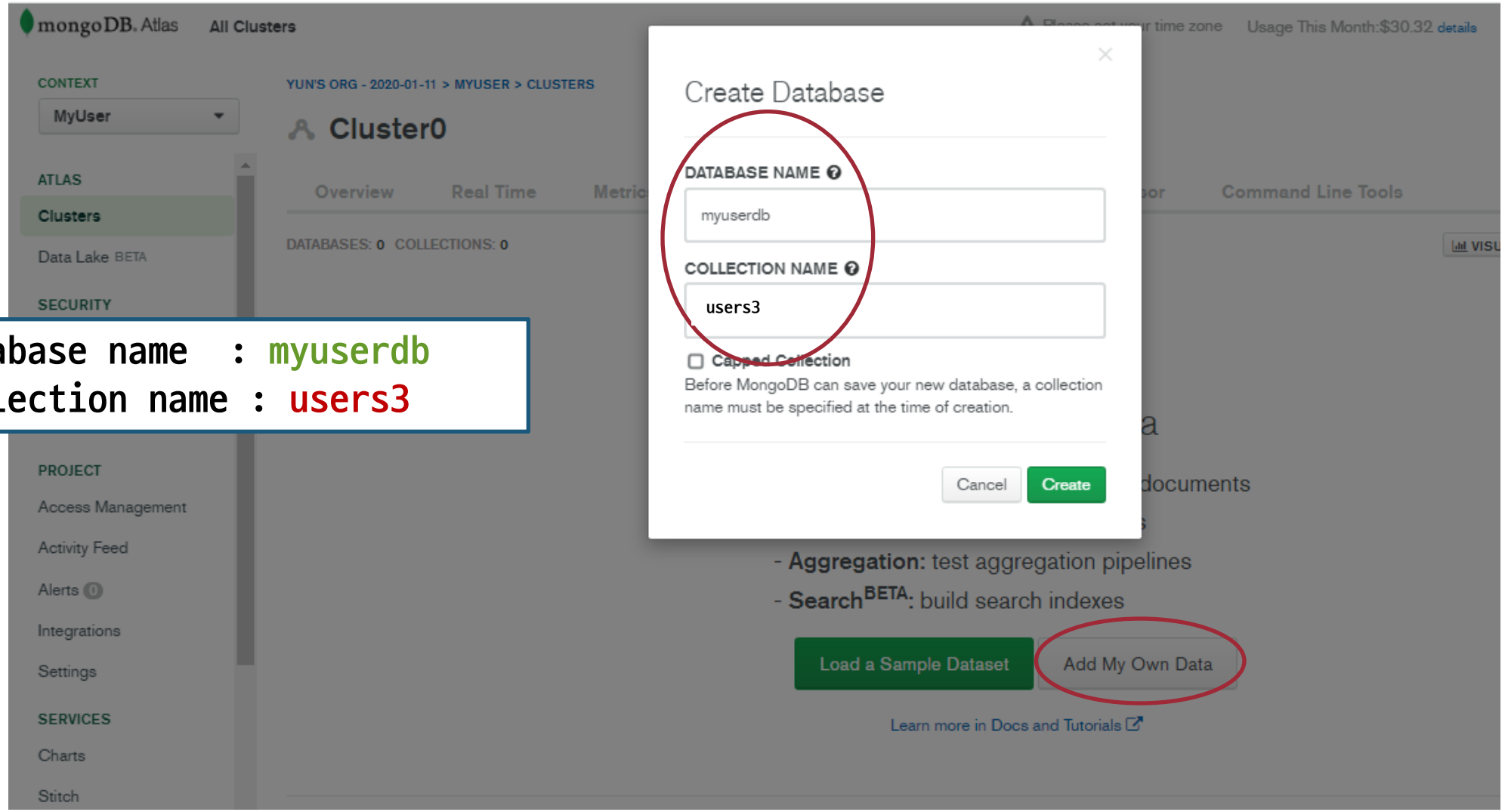


Add New Database user 확인



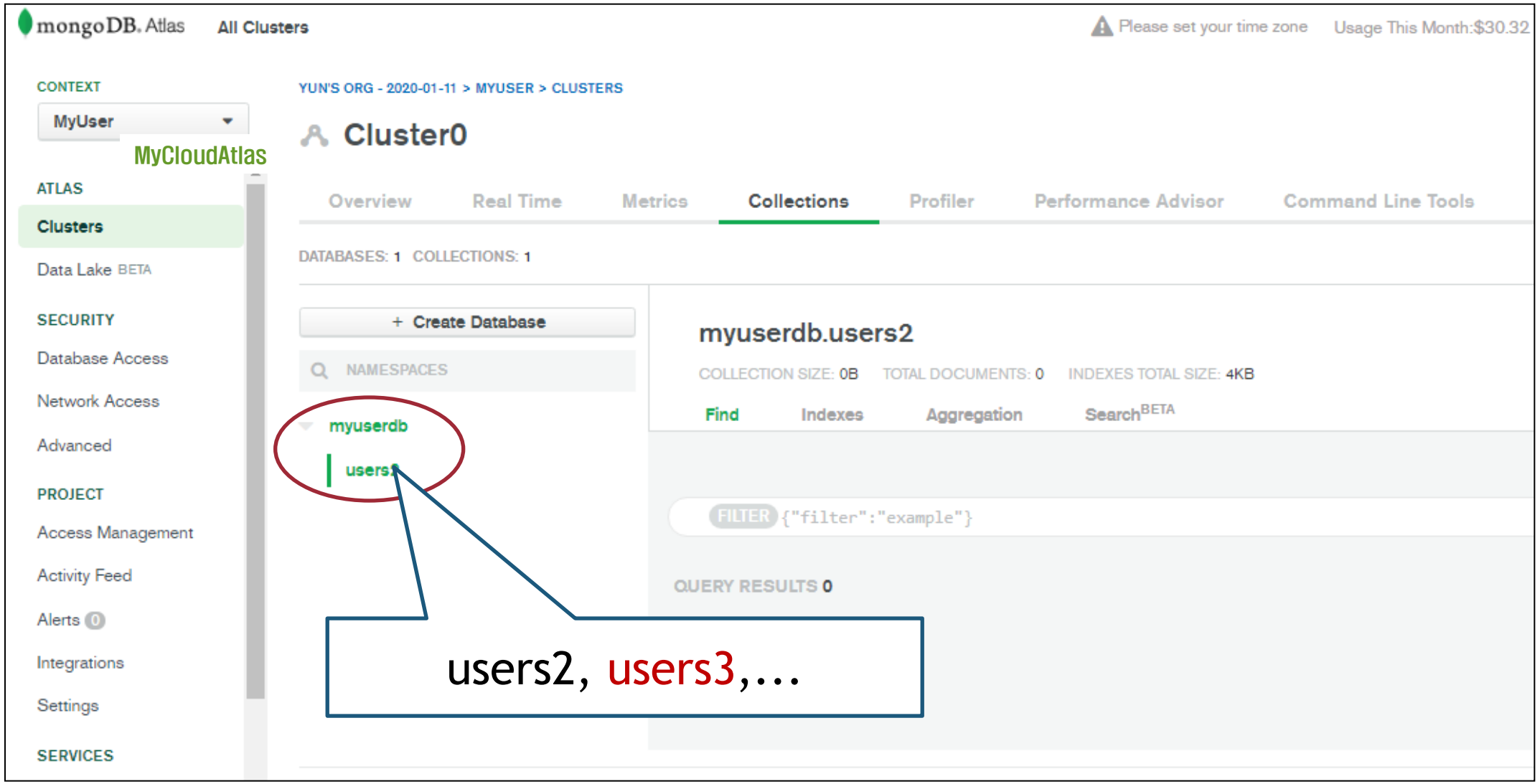
step6. db 만들기 => users2를 users3으로 수정하자!

- 왼쪽메뉴 clusters-collections-add my own data



database name : myuserdb
collection name : users3

db 만들기 확인

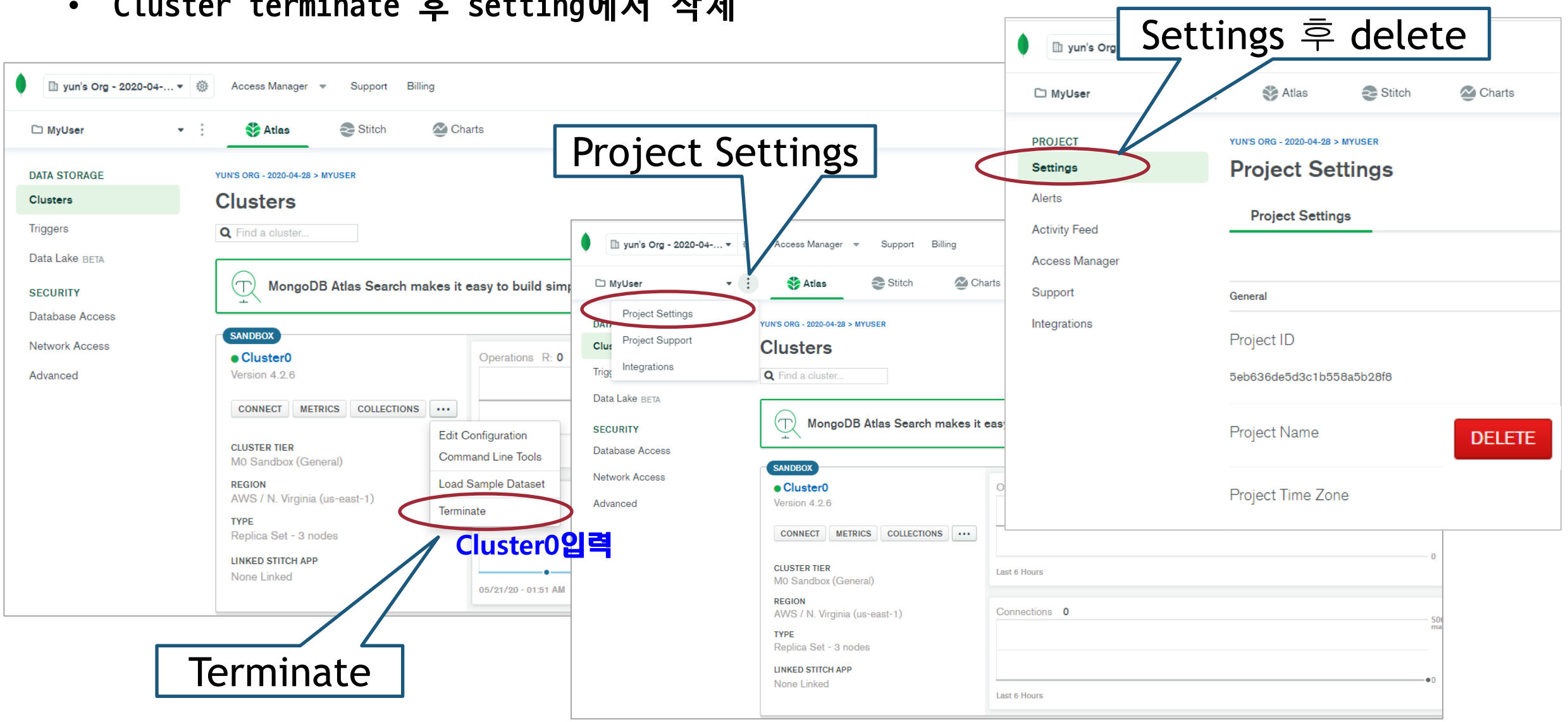

mongoDB. Atlas
All Clusters
Please set your time zone
Usage This Month:$30.32
CONTEXT
MyUser
MyCloudAtlas
ATLAS
Clusters
Data Lake BETA
SECURITY
Database Access
Network Access
Advanced
PROJECT
Access Management
Activity Feed
Alerts 0
Integrations
Settings
SERVICES
YUN'S ORG - 2020-01-11 > MYUSER > CLUSTERS
Cluster0
Overview
Real Time
Metrics
Collections
Profiler
Performance Advisor
Command Line Tools
DATABASES: 1 COLLECTIONS: 1
+ Create Database
NAMESPACES
myuserdb
users1
myuserdb.users2
COLLECTION SIZE: 0B
TOTAL DOCUMENTS: 0
INDEXES TOTAL SIZE: 4KB
Find
Indexes
Aggregation
Search BETA
FILTER {"filter":"example"}
QUERY RESULTS 0
users2, users3,...

유료 되지 않도록 조심! - terminate하기, 필요 없는 것은 지우기

- clusters-cluster1이 있는가 확인
- cluster선택 ... 메뉴 아래에서 terminate 하기
- 그리고 해당 project를 setting에 가서 cluster1 이후는 모두 지우기

mongodb Atlas project 삭제

- Cluster terminate 후 setting에서 삭제



3. mongo compass로 db와 heroku연결 및 mongodb Atlas cloud 에 db upload

3. mongo compass로 db와 heroku연결 및 **mongodb Atlas cloud** 에 db upload (1/3)

- 몽고 dbatlas
clusters-connect-3번째 cennect using mongodb compass-I have MongoDb compass선택
copy 1.12(최신버젼) => 1.11 earlier

mongodb://sunny5:<password>@cluster0-shard-00-00.iczci.mongodb.net:27017,cluster0-shard-00-01.iczci.mongodb.net:27017,cluster0-shard-00-02.iczci.mongodb.net:27017/test?replicaSet=atlas-df8m6a-shard-0&ssl=true&authSource=admin

3. mongo compass로 db와 heroku연결 및 mongodb Atlas cloud 에 db upload (2/3)

-app.js

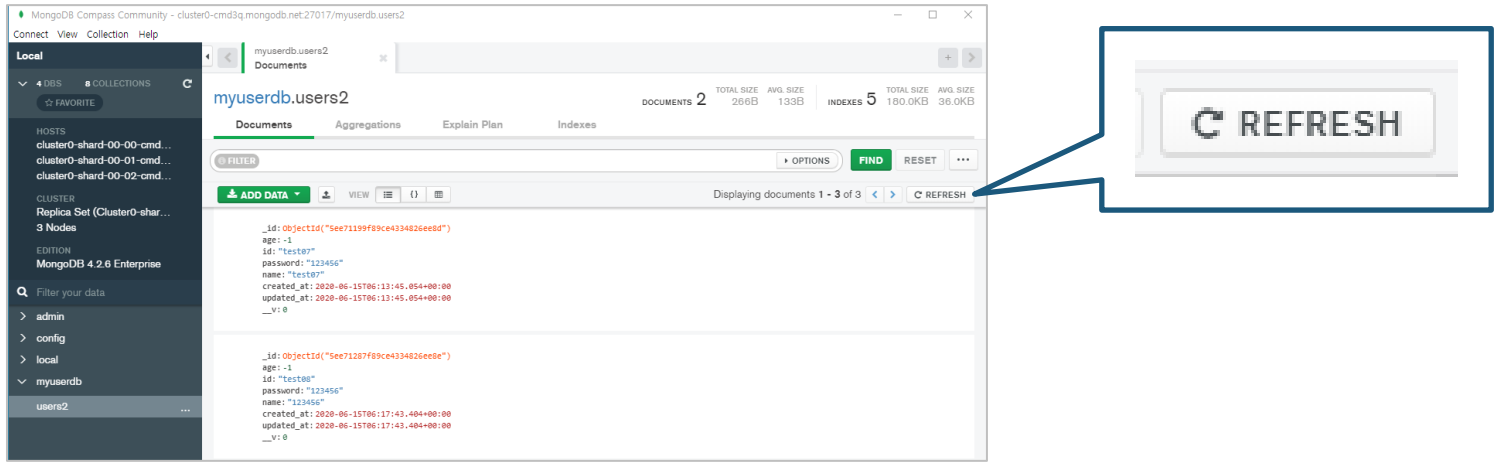
```
//데이터베이스에 연결
function connectDB() {
    // 데이터베이스 연결 정보
    //var databaseUrl = 'mongodb://localhost:27017/local';
    var databaseUrl = 'mongodb://sunny5:*******@cluster0-shard-00-00.iczci.mongodb.net:27017,cluster0-shard-00-01.iczci.mongodb.net:27017,cluster0-shard-00-02.iczci.mongodb.net:27017/myuserdb?replicaSet=atlas-df8m6a-shard-0&ssl=true&authSource=admin';

    // 데이터베이스 연결
    console.log('데이터베이스 연결을 시도합니다.');
    mongoose.Promise = global.Promise;
    //mongoose.connect(databaseUrl);
    mongoose.connect(databaseUrl, {useMongoClient:true});

    database = mongoose.connection;
```

3. mongo compass로 db와 heroku연결 및 mongodb Atlas cloud에 db upload (3/3)

- http://localhost:3000/adduser.html에서 데이터 입력 후 mongo compass에서 확인하여보기

- mongodb compass에서 db 내용 확인
시작 => mongodb compass 확인
'mongodb://localhost:27017/local';
 => mongodb://sunny5:*******@cluster0-shard-00-00.iczci.mongodb.net:27017,cluster0-shard-00-01.iczci.mongodb.net:27017,cluster0-shard-00-02.iczci.mongodb.net:27017/myuserdb?replicaSet=atlas-df8m6a-shard-0&ssl=true&authSource=admin
- connect 클릭

=> mongodb atlas cloud db내용을 보여줌
=> 현재 프로그램은 로컬을 사용하고 있으니 atlas cloud db를 사용하는것으로 변경함



오류설명1> 최신버전 사용시 => 구버전으로 대체 - config.js

```
db_url: 'mongodb+srv://sunny5:***0414@cluster0.udznn.mongodb.net/myuserdb',
```

```
서버가 시작되었습니다. 포트 : 3000
init() 호출됨.
connect() 호출됨.
uncaughtException 발생함 : Error: Invalid mongodb uri "mongodb+srv://sunny5:sol0414@cluster0.udznn.mongodb.net/myuserdb". Must begin with "mongodb://"
서버 프로세스 종료하지 않고 유지함.
Error: Invalid mongodb uri "mongodb+srv://sunny5:sol0414@cluster0.udznn.mongodb.net/myuserdb". Must begin with "mongodb://"
    at muri (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\muri\lib\index.js:28:11)
    at NativeConnection.Connection.openUri (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongoose\lib\connection.js:766:18)
    at Mongoose.connect (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongoose\lib\index.js:262:17)
    at connect (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\database\database.js:27:11)
    at Object.database.init (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\database\database.js:16:2)
    at Server.<anonymous> (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\app.js:200:11)
    at Object.onceWrapper (events.js:427:28)
    at Server.emit (events.js:321:20)
    at emitListeningNT (net.js:1343:10)
    at processTicksAndRejections (internal/process/task_queues.js:83:21)
```

=>

```
module.exports = {
  server_port: 3000,
  //db_url: 'mongodb://localhost:27017/local',
  db_url: 'mongodb://sunny5:***0414@cluster0-shard-00-00-u6kid.mongodb.net:27017,cluster0-shard-00-01-
u6kid.mongodb.net:27017,cluster0-shard-00-02-u6kid.mongodb.net:27017/myuserdb?replicaSet=Cluster0-shard-
0&ssl=true&authSource=admin',
```

오류설명 2> mongoose의 버전 4.11.0 이후버전 속성값 바꾸어주어야 함 - database.js

```
// 데이터베이스 연결 : config의 설정 사용
mongoose.Promise = global.Promise;  // mongoose의 Promise 객체는 global의 Promise 객체 사용하도록 함

// mongoose.connect(config.db_url, {useMongoClient:true, useNewUrlParser: true, useUnifiedTopology: true});
// mongoose.connect(config.db_url, {useMongoClient:true});
mongoose.connect(databaseUrl,{useNewUrlParser: true, useUnifiedTopology: true});
database.db = mongoose.connection;
```

```
(node:4312) DeprecationWarning: `openSet()` is deprecated in mongoose >= 4.11.0, use `openUri()` instead, or set the `useMongoClient` option if using `conn
ect()` or `createConnection()`. See http://mongoosejs.com/docs/4.x/docs/connections.html#use-mongo-client
connect() 호출됨.
uncaughtException 발생함 : TypeError: Cannot read property 'db_url' of undefined
서버 프로세스 종료하지 않고 유지함.
TypeError: Cannot read property 'db_url' of undefined
    at NativeConnection.connect (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\database\database.js:26:26)
    at NativeConnection.emit (events.js:321:20)
    at NativeConnection.set (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongoose\lib\connection.js:120:12)
    at C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongoose\lib\connection.js:696:24
    at C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongoose\lib\drivers\node-mongodb-native\connection.js:226:21
    at C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongodb\lib\db.js:236:14
    at ReplSet.<anonymous> (C:\nodejs\ViewExample+uploads+bootstrap(login,listuser)\node_modules\mongodb\lib\replset.js:362:9)
    at Object.onceWrapper (events.js:428:26)
    at ReplSet.emit (events.js:321:20)
```

adduser.html 실행시 발생오류

=>

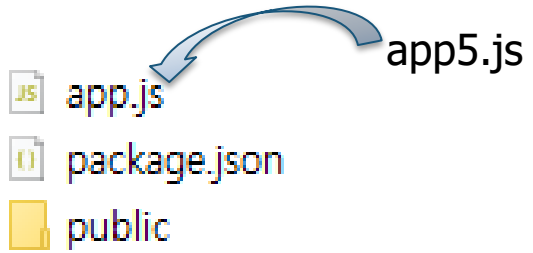
```
// mongoose.connect(config.db_url, {useMongoClient:true, useNewUrlParser: true, useUnifiedTopology: true});
mongoose.connect(config.db_url, {useMongoClient:true});
// mongoose.connect(databaseUrl,{useNewUrlParser: true, useUnifiedTopology: true});
database.db = mongoose.connection;
```

III. 실습 1
local host, mongodb Atlas cloud 사용

<실습보고서11> app5.js - 사용자 추가 및 로그인– users3, localhost, localdb, bootstrap 적용안됨
<실습보고서12> localhost, mongodb Atlas cloud 사용, 로고 만들기, bootstrap
<실습보고서13> sunnyatlas.herokuapp.com, mongodb Atlas cloud 사용, bootstrap(모바일)

<실습1>localhost - http://localhost:3000

1. DababaseExample>에서 app5.js, package.json, public을 복사하여
   DatabaseExample_sunnyatlas_app5> 붙여넣기
2. > DatabaseExample_sunnyatlas_app5> npm install



3. > DatabaseExample_sunnyatlas_app5> app.js 에서(DababaseExample>의 app5.js)
   - app.use('/', static(path.join(__dirname, 'public'))); 로 수정
   - app.js 새로 수정하여 경로이동하게 만들기
4. app.js에 맞게 public> *.html 만들기

작업순서

1. loclhost test 완료 (localdb 사용)=> localhost, localdb 사용(6.database_example에서 했음)
2. heroku app 만들기
3. 26 slide mongodb atlas환경조성 요약
4. 46 slide mongo compass로 db와 heroku연결 및 db cloud 에 upload
   (mongodb://sunny5:***0414@cluster0-shard-~~~=admin)
5. mongocompass로 mongodb atlas확인하기(atlas db 사용) => local host, mongodb atlas사용
6. ~~application heroku upload~~
7. ~~https://sunnyatlas.herokuapp.com 확인 => heroku applicaiont, mongodb atlas사용~~

<실습1-1> app.js (1/4)

```
//데이터베이스에 연결
function connectDB() {
    // 데이터베이스 연결 정보
    //var databaseUrl = 'mongodb://localhost:27017/local';
    var databaseUrl = 'mongodb://sunny5:***0414@cluster0-shard-00-00.99rnw.mongodb.net:27017,cluster0-shard-00-01.99rnw.mongodb.net:27017,cluster0-shard-00-02.99rnw.mongodb.net:27017/myuserdb?replicaSet=atlas-napmvz-shard-0&ssl=true&authSource=admin';

    // 데이터베이스 연결
    console.log('데이터베이스 연결을 시도합니다.');
    mongoose.Promise = global.Promise;  // mongoose의 Promise 객체는 global의 Promise 객체 사용하도록 함
    //mongoose.connect(databaseUrl);
    mongoose.connect(databaseUrl, {useMongoClient:true});

    database = mongoose.connection;
```

<실습1-2> app.js (2/4)

```
// 조회된 레코드가 있으면 성공 응답 전송
        if (docs) {
            console.dir(docs+'last');

            // 조회 결과에서 사용자 이름 확인
            var username = docs[0].name;
            res.redirect('/login_success.html');
            // res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});
            // res.write('<h1>로그인 성공</h1>');
            // res.write('<div><p>사용자 아이디 : ' + paramId + '</p></div>');
            // res.write('<div><p>사용자 이름 : ' + username + '</p></div>');
            // res.write("<br><br><a href='/public/login.html'>다시 로그인하기</a>");
            // res.end();

        } else {  // 조회된 레코드가 없는 경우 실패 응답 전송
            res.redirect('/login_fail.html')
            // res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});
            // res.write('<h1>로그인  실패</h1>');
            // res.write('<div><p>아이디와 패스워드를 다시 확인하십시오.</p></div>');
            // res.write("<br><br><a href='/login.html'>다시 로그인하기</a>");
            // res.end();
        }
```

<실습1-3> app.js (3/4)

```
// 결과 객체 있으면 성공 응답 전송
            if (addedUser) {
                console.dir(addedUser);
                res.redirect('/adduser_success.html');
                // res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});
                // res.write('<h2>사용자 추가 성공</h2>');
                // res.end();
            } else {  // 결과 객체가 없으면 실패 응답 전송
                res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});
                res.write('<h2>사용자 추가  실패</h2>');
                res.end();
            }
```

<실습1-4> app.js (4/4)

```
        if (results) {  // 결과 객체 있으면 리스트 전송
            console.dir(results);

            res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});
            res.write('<hr>');
            res.write('<h2>사용자 리스트</h2>');
            res.write('<hr>');
            res.write('<div><ul>');

            for (var i = 0; i < results.length; i++) {
                var curId = results[i]._doc.id;
                var curName = results[i]._doc.name;
                res.write('    <li>#' + i + ' : ' + curId + ', ' + curName + '</li>');
            }

            res.write('</ul></div>');
            res.write("<br><br><button><a href='/adduser.html'>사용자추가</a>");
            res.write("<button><a href='/login.html'> 로그인</a>");

            res.end();
        } else {
```

<실습1-5> login.html, 로그 만들기

```
<!DOCTYPE html><html> <head><meta charset="UTF-8">
    <title>로그인 테스트</title>
    <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1">
    <link rel="apple-touch-icon" sizes="180x180" href="img/cloud.webp" />
    <link rel="icon" type="image/png" href="img/cloud.webp" sizes="192x192"/>
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <!--추가1-->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
    <!--추가1-->
  </head>
  <body>
    <hr/>
    <h4>  로그인</h3>
    <hr/>
    <form method="post" action="/process/login">
      <br>
        <label class="btn btn-info">아  이  디 </label>
      <input type="text" size ="12" name="id" ></input>
      <br> 
      <label class="btn btn-info">비밀번호</label>
      <input type="password" size ="12" name="password" ></input>
      <br><br>    
      <input type="submit" value="확  인" class="btn btn-info"></input>
        <a href='/index.html' class="btn btn-info">메인페이지</a>
    </form></body></html>
```

로고, 주소창 숨기기, **bootstrap** 연결부분 동일하게 적용

<실습1-6> login_success.html

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
        <title>로그인 성공</title>
```

로고, 주소창 숨기기,
bootstrap 연결부분
동일하게 적용

```
    </head>
    <body>
        <hr/>
        <h4>  로그인 성공</h3>
        <hr/>
        <br>
          <a href='/adduser.html' class="btn btn-info">사용자추가</a>
         <a href='/listuser.html' class="btn btn-info">사용자리스트</a>
         <a href='/index.html' class="btn btn-info">메인페이지</a>
     </body>
</html>
```

<실습1-7> login_fail.html

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
        <title>로그인 실패</title>
        로고, 주소창 숨기기,
        bootstrap 연결부분
        동일하게 적용

   </head>
    <body>
        <hr/>
        <h4>  로그인 실패</h3>
        <hr/>
        <br>
          <a href='/login.html' class="btn btn-info">다시로그인</a>
         <a href='/index.html' class="btn btn-info">메인페이지</a>
    </body>
</html>
```

<실습1-8> adduser.html

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
      <title>사용자추가</title>
```

로고, 주소창 숨기기,
bootstrap 연결부분
동일하게 적용

```
   </head>
<body>
   <hr/>
   <h4>  사용자추가</h3>
   <hr/>
   <form method="post" action="/process/adduser">
        <table>
         <tr>
            <td>  <label class="btn btn-info">아 이 디  </label></td>
            <td><input  type="text" name="id" /></td>
         </tr>
         <tr>
            <td>  <label class="btn btn-info">비밀번호</label></td>
            <td><input  type="password" name="password" /></td>
         </tr>
         <tr>
            <td>  <label class="btn btn-info">사용자명</label></td>
            <td><input  type="text" name="name" /></td>
         </tr>
      </table><br>   
        <input class="btn btn-info" type="submit" value="전송" name="" />
       <a href='/index.html' class="btn btn-info">메인페이지</a>
   </form></body></html
```

<실습1-9> adduser_success.html

```
<!DOCTYPE html><html><head> <meta charset="UTF-8">
      <title>사용자추가 성공</title>
```

로고, 주소창 숨기기,
bootstrap 연결부분
동일하게 적용

```
   </head>
   <body>
      <hr/>
      <h4>  사용자추가 성공</h3>
      <hr/>
      <br>
        <a href='/adduser.html' class="btn btn-info">사용자추가</a>
       <a href='/listuser.html' class="btn btn-info">사용자리스트</a>
       <a href='/index.html' class="btn btn-info">메인페이지</a>
   </body>
</html>
```

<실습1-10> listuser.html

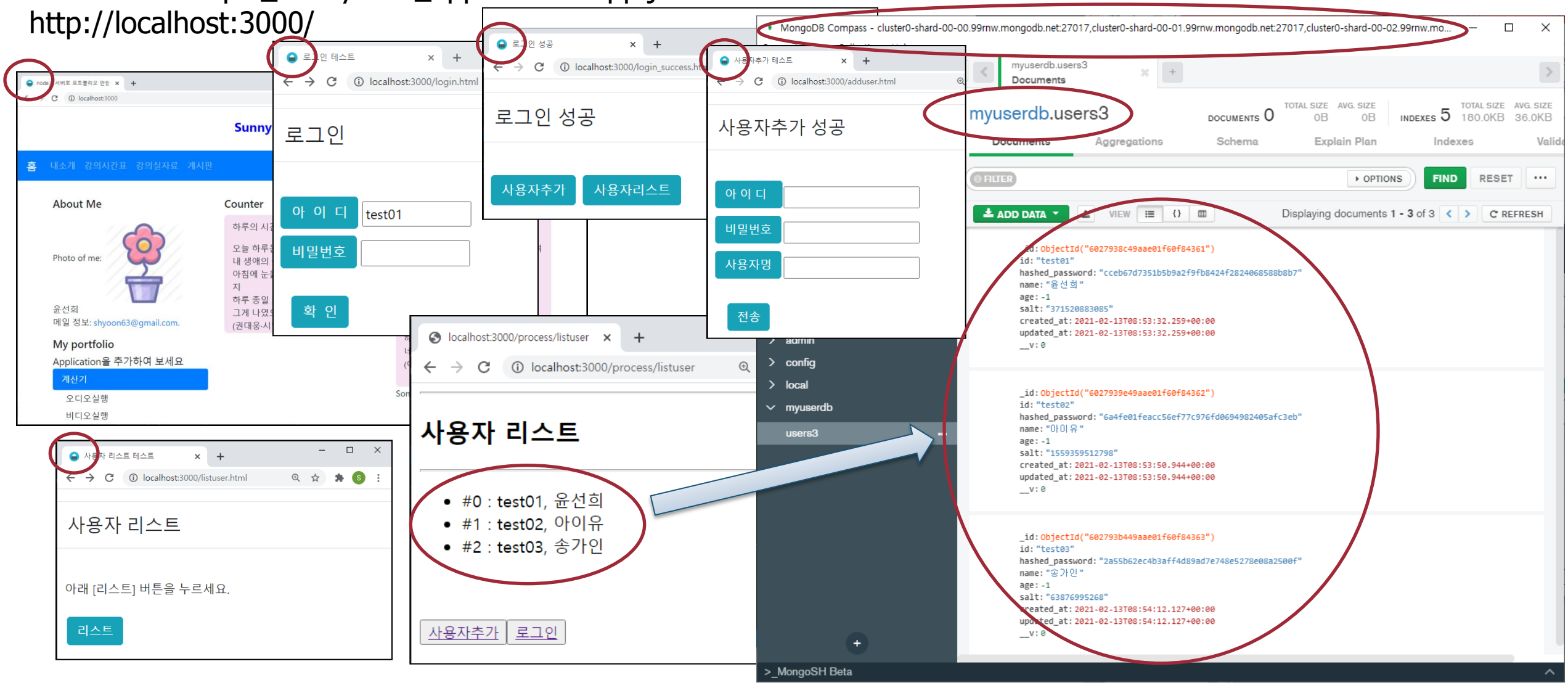
```
<!DOCTYPE html><html><head><meta charset="UTF-8">
        <title>사용자 리스트 테스트</title>
```

로고, 주소창 숨기기,
bootstrap 연결부분
동일하게 적용

```
    </head>
<body>
    <hr/>
    <h4>  사용자 리스트</h3>
    <hr/>
    <br>
    <form method="post" action="/process/listuser">
        <table>
            <tr>
                <td><label></label>  아래 [리스트] 버튼을 누르세요.</label></td>
            </tr>
        </table>
          <br>   <input class="btn btn-info" type="submit" value="리스트" name="" />
         <a href='/index.html' class="btn btn-info">메인페이지</a>
     </form>
</body>
</html>
```

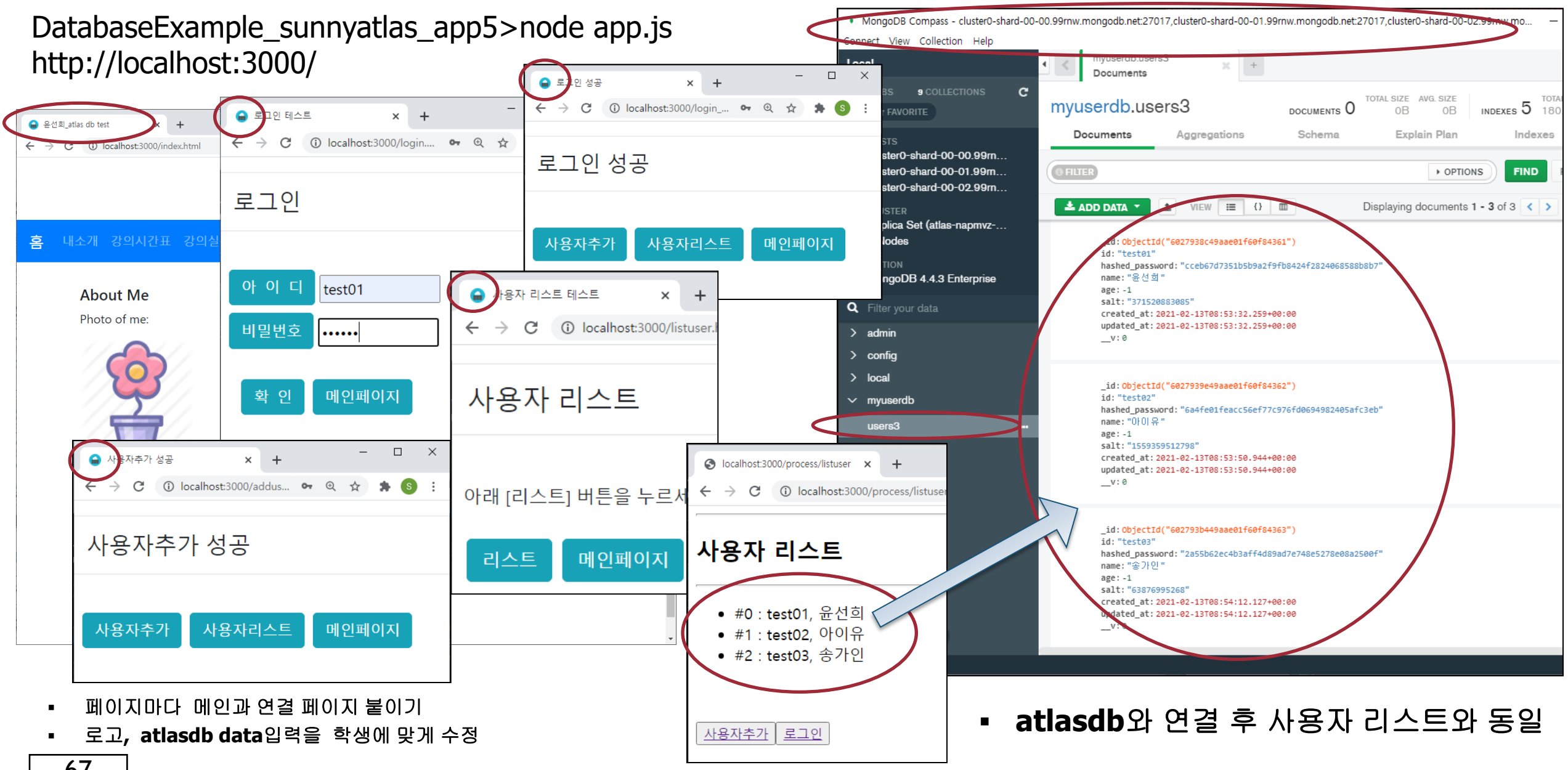
<실습1-10> http://localhost:3000

DatabaseExample_sunnyatlas_app5>node app.js
http://localhost:3000/



<실습보고서12> http://localhost:3000 - atlas cloud mongodb 사용, 로고 만들기

DatabaseExample_sunnyatlas_app5>node app.js
http://localhost:3000/



- 페이지마다 메인과 연결 페이지 붙이기
- 로고, **atlasdb data**입력을 학생에 맞게 수정

- **atlasdb**와 연결 후 사용자 리스트와 동일

4. git을 이용해서 heroku에 앱올리기

III. 실습2
sunnyatlas.herokuapp.com, mongodb Atlas cloud 사용

<실습보고서11> app5.js - 사용자 추가 및 로그인– users3, localhost, localdb, bootstrap 적용안됨
<실습보고서12> localhost, mongodb Atlas cloud 사용, 로고 만들기, bootstrap
<실습보고서13> sunnyatlas.herokuapp.com, mongodb Atlas cloud 사용, bootstrap(모바일)

git으로 올리기 사전 작업

1. package.json의 node 항목 지우기

2. Procfile 파일 만들기 <= 확장자 없이 대소문자 구분해서 만들기, 자동실행
c:□(base) Procfile
```
web: npm start
```

3. .gitignore 파일 만들기 <= heroku 에 모두 있으므로 무시하라는 의미
c:□(base) .gitignore
```
# Dependency directory
node_modules
npm-debug.log
```

4. package.json내용 수정
```
 "scripts": {
    "start": "node app.js",
    "test": "echo □"Error: no test specified□" && exit 1"
  },
```

5. logo 만들기

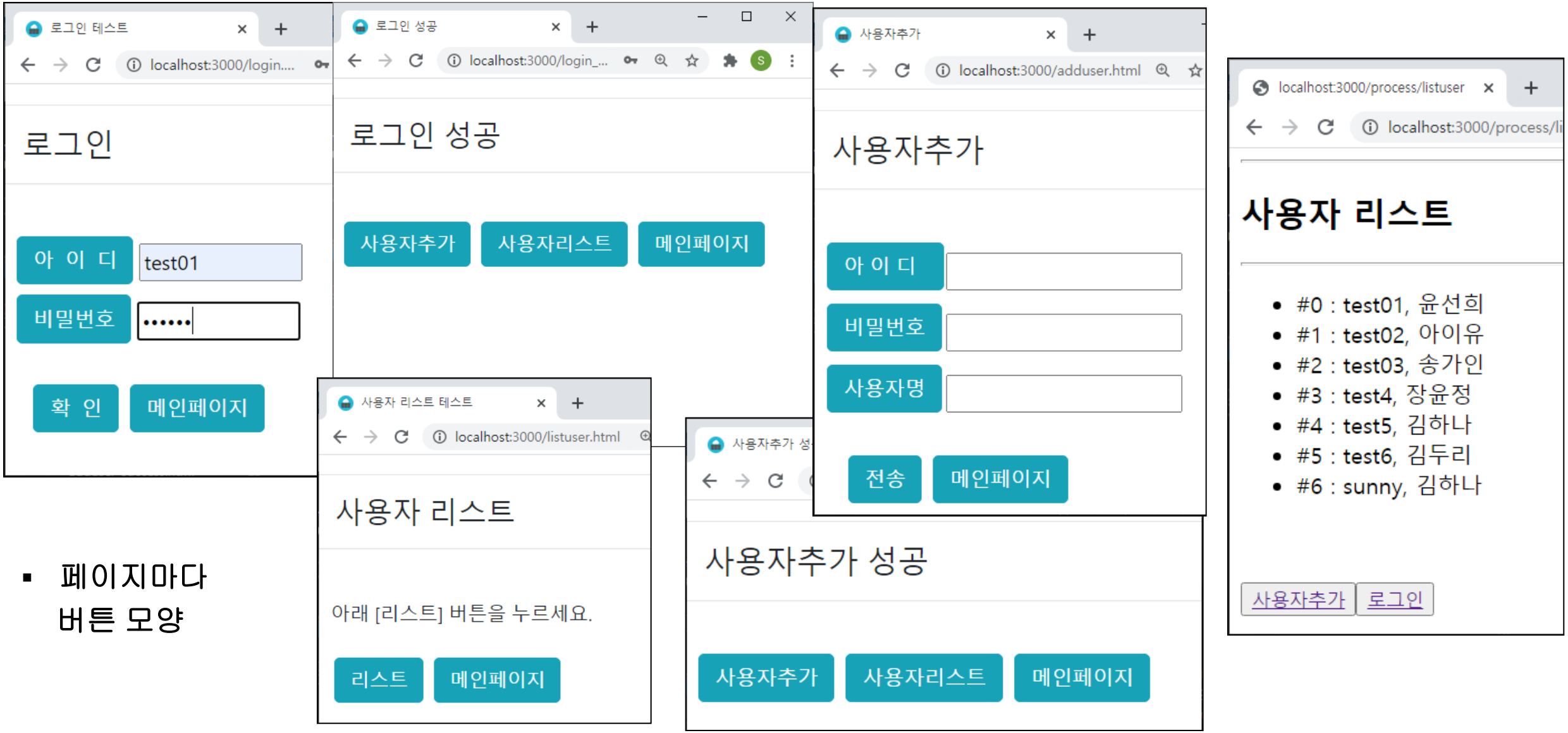
logo icon만들기 와 주소창 숨기기 => 모바일 바탕화면에 바로가기 만들기

index.html

```
<meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1">
    <link rel="apple-touch-icon" sizes="180x180" href="../public/images/logo.png" />
    <link rel="icon" type="image/png" href="../public/images/logo.png" sizes="192x192"/>
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
```

```
// for Node.js
// layout.pug:
link(rel="shortcut icon", href="/images/logo.png")
link(rel="apple-touch-icon" href="/images/logo.png")
```

버튼 확대 참고



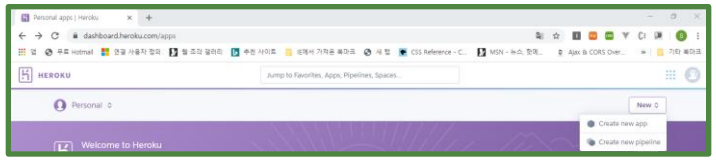
- 페이지마다 버튼 모양

<실습2>cloud system - http://sunnyatlas.herokuapp.com

- 작업순서

1. ~~loclhost test 완료 (localdb 사용)=> localhost, localdb 사용~~
2. ~~heroku app 만들기~~
3. ~~26 slide mongodb atlas환경조성 요약~~
4. ~~46 slide mongo compass로 db와 heroku연결 및 db cloud 에 upload (mongodb://sunny5:***0414@cluster0-shard-~~~=admin)~~
5. ~~mongocompass로 mongodb atlas 확인하기(atlas db 사용) => local host, atlas db사용~~
6. application heroku upload
7. https://sunnyatlas.herokuapp.com 확인 => heroku application, mongodb atlas사용

- heroku app 만들기



- 준비파일들

Procfile

.gitignore

```
"scripts": {
  "start": "node app.js",
  "test": "echo \"Error: no t
},
```

- heroku upload step

```
$ heroku login
$ cd my-project/

$ git init
$ heroku git:remote -a sunnyatlas

$ git add .
$ git commit -am "make it better"
$ git push heroku master

$ heroku open
  또는 브라우저에서
https://sunnyatlas.herokuapp.com
```

- 내용 수정 후:

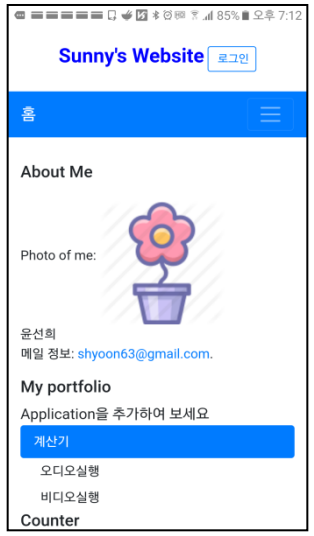
```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

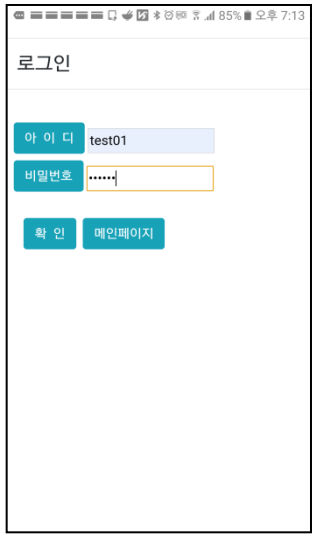
- 오류 해결사항

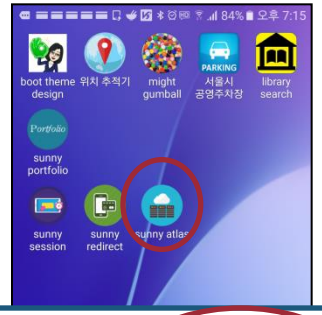
1. git폴더 삭제
2. heroku app 삭제후 처음부터 올리기

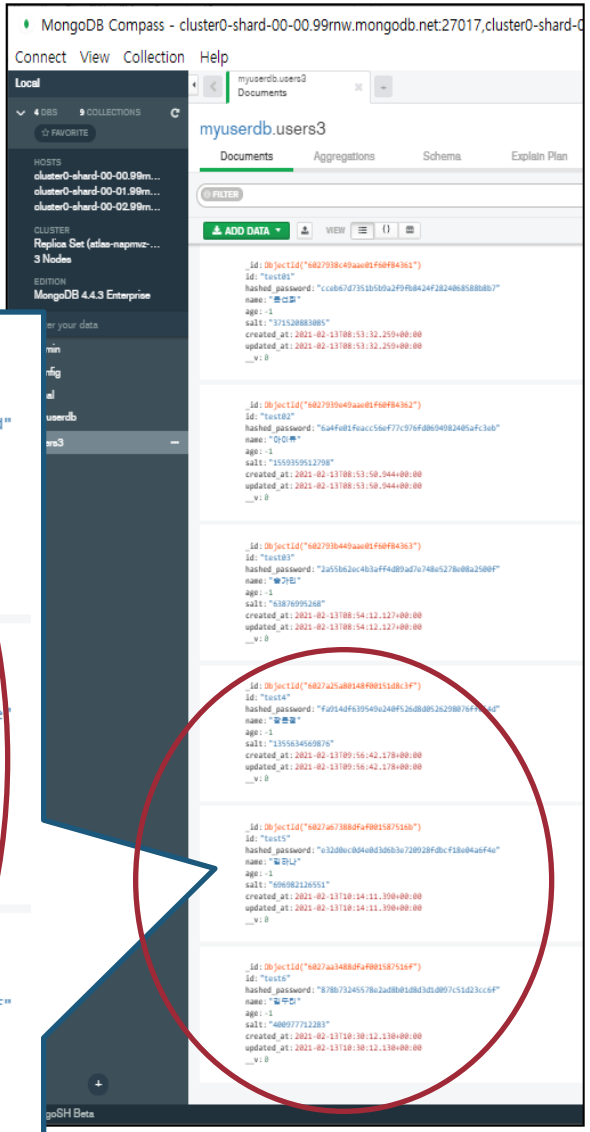
- 페이지마다 메인과 연결 페이지 붙이기

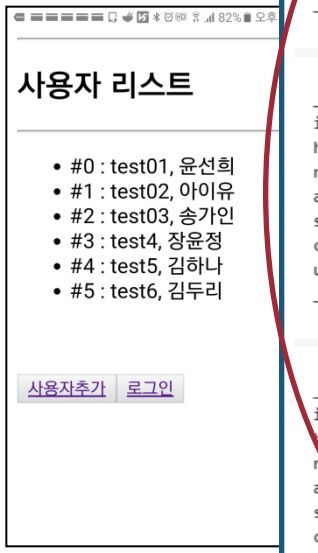
<실습보고서13> http://sunnyatlas.herokuapp.com - heroku - atlas mongodb 사용, mobile용











- <실습보고서> 형식 참조

Q & A