개발 기본 설정 및 기본 이론 - 2

Web 개발 기초

2. Node.js / NPM 설치 및 기본 설정

3. Express.js 설치 및 기본 설정

4. MySQL 설치 및 phpmyadmin 설정

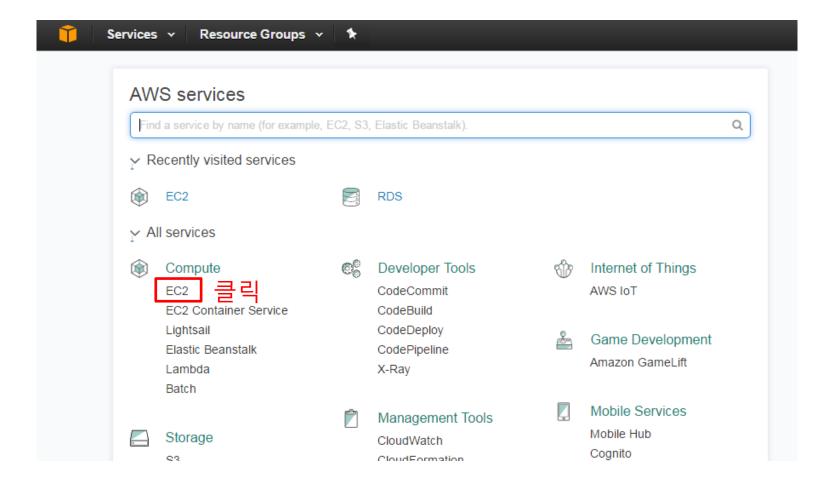


AWS EC2?

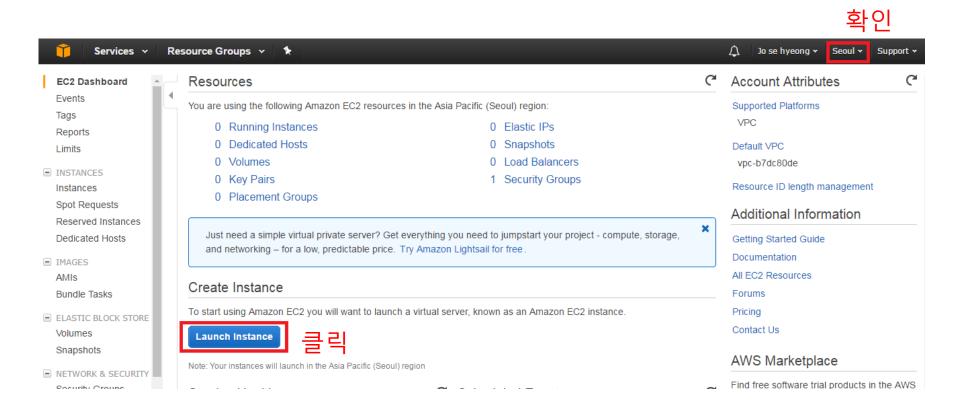
Amazon Elastic Compute Cloud의 약자

EC2 에는 1개의 컴퓨터 단위를 instance라고 부르는데, 이제 instance를 만든다.

Instance 생성



Instance 생성





선택

Instance 생성

Ubuntu로 생성한다.

Root device type: ebs Virtualization type: hvm

| | K < 1 to 2 | 29 of 29 AMIs > |
|------------------------------------|---|-----------------|
| I | Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type - ami-dac312b4 | Select |
| Amazon Linux Free tier eligible | The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. | 64-bit |
| | Root device type: ebs Virtualization type: hvm | |
| (4) | Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-44db152a | Select |
| Red Hat | Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type | 64-bit |
| Free tier eligible | Root device type: ebs Virtualization type: hvm | |
| 9 | SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-5060b73e | Select |
| SUSE Linux | SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems | 64-bit |
| Free tier eligible | Management, Web and Scripting, and Legacy modules enabled. | |
| | Root device type: ebs Virtualization type: hvm | |
| (| Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0500d06b | Select |
| Free tier eligible | Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical | 64-bit |
| 1 100 doi cligible | (http://www.ubuntu.com/cloud/services). | |



Instance 생성

Free Tier 로 선택!

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of cand networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your cor

| Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only) | | | | | | | | |
|---|-----------------|--------------------------------|-------------|----------------|-----------------------------|-------------------------|---------------------|--|
| | Family | Type | vCPUs (j) 🔻 | Memory (GiB) ~ | Instance Storage (GB) (j) v | EBS-Optimized Available | Network Performance | |
| | General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderat | |
| | General purpose | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | |
| | General purpose | t2.small | Free Lie | er 확인! | EBS only | - | Low to Moderat | |
| | General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderat | |
| | General purpose | t2.large | 2 | 8 | EBS only | - | Low to Moderat | |
| | General purpose | t2.xlarge | 4 | 16 | EBS only | - | Moderate | |
| | General purpose | t2.2xlarge | 8 | 32 | EBS only | - | Moderate | |
| | General purpose | m4.large | 2 | 8 | EBS only | Yes | Moderate | |
| | General purpose | m4.xlarge | 4 | 16 | EBS only | Yes | High | |



Instance 생성

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.



Improve your instances' security. Your security group, launch-wizard-1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

▼ AMI Details Edit AMI



Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0500d06b



Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type

Edit instance type

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|------------------|----------|-------|-----------------|--------------------------|----------------------------|------------------------|
| t2.micro | Variable | 1 | 1 | EBS only | - | Low to Moderate |

- Contribu Crausa

Edit cocurity aroune

Previous

Cancel







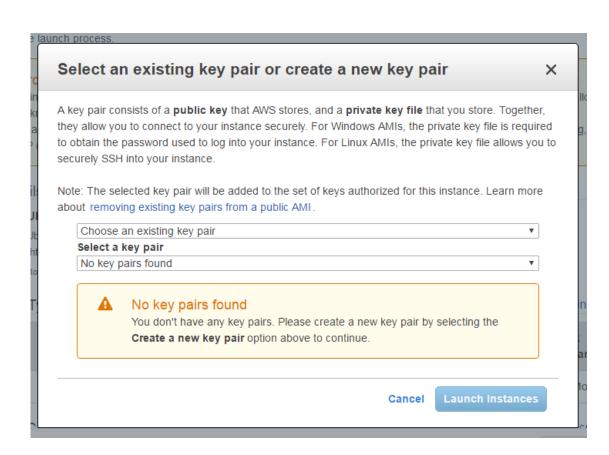
Instance 생성

여기서 key-pair를 생성하는데 접속하는데 있어 반드시 필요 한 것이므로 잘 간직하고 있어 야 한다.

순서

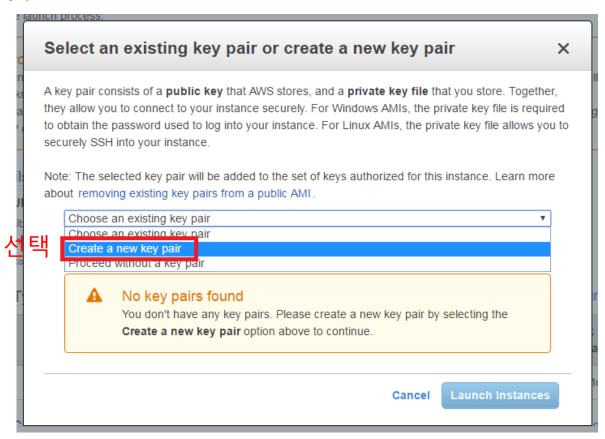
Create a new key pair 선택

- ➤ Download key pair 선택
- Launch Insatance 클릭



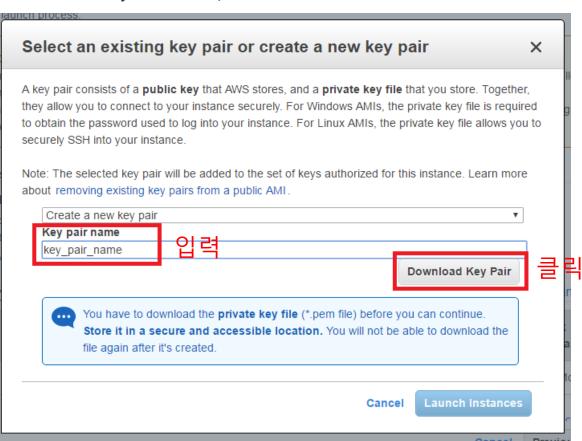
Instance 생성

Create a new key pair 선택



Instance 생성

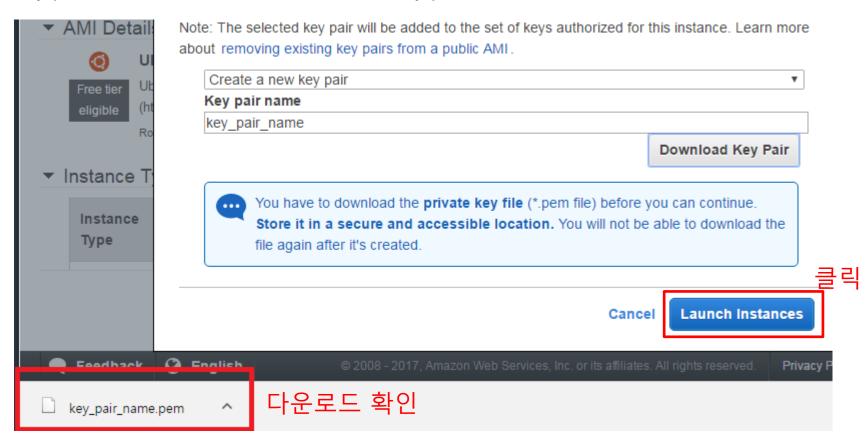
Key pair name 입력 후 Download Key Pair 클릭





Instance 생성

Key pair download 확인 / 원하는 위치에 key pair 파일 이동 하고 Launch Instance 클릭





Instance 생성

Launch Status



Initiating Instance Launches

Please do not close your browser while this is loading

Creating security groups... Successful

Authorizing inbound rules...



Instance 생성

Instance 생성 완료!

Launch Status

- Your instances are now launching
 - The following instance launches have been initiated: i-019eaa98eb894cdcf View launch log
- Get notified of estimated charges Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

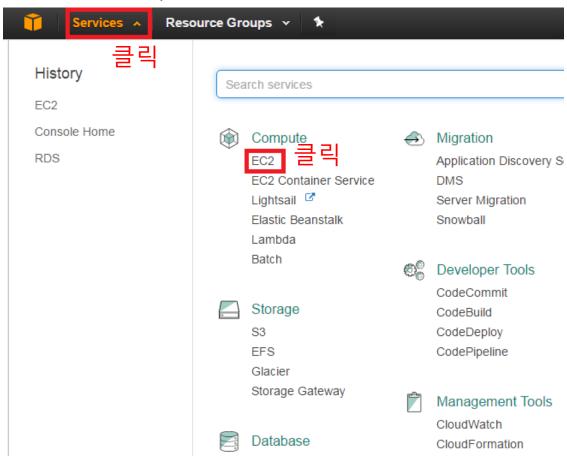
- Here are some helpful resources to get you started
- How to connect to your Linux instance
- Amazon EC2: User Guide
- Learn about AWS Free Usage Tier
- Amazon EC2: Discussion Forum

While your instances are launching you can also

Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
Create and attach additional EBS volumes (Additional charges may apply)

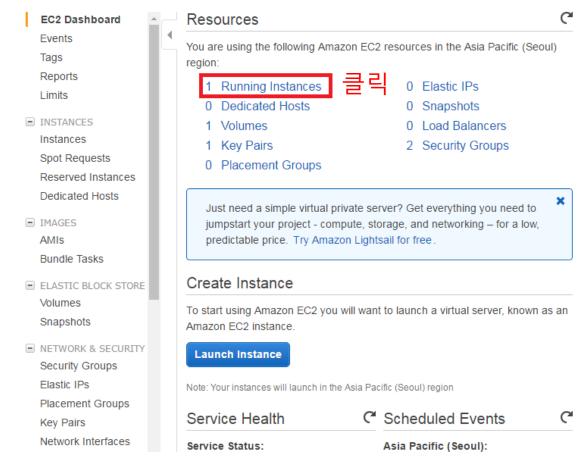
Instance 생성

Services > EC2 선택



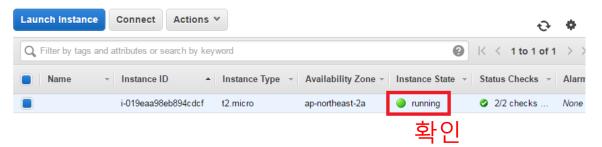
Instance 생성

Running Instances 클릭 (숫자 1 확인!)



Instance 생성

Running 상태 확인 (옆의 Status Checks에 initializing 이라고 뜨면 초기화 작업 중인 것으로 조금만 있으면 check 상태가 됩니다.)





Instance 생성

하단의 Description 에는 instance에 대한 정보들이 있습니다. 이 중에서 Public IP는 필요하기 때문에 확인합시다.







EC2와 연결

EC2 생성을 완료 했으니, 이제 서버와 연결을 해야 합니다.

서버와의 연결 방법은 Windows 와 Mac이 서로 다르니 나눠서 알아보기로 합니다.



EC2와 연결(Windows)

Windows는 서버와 연결하기 위해 따로 프로그램을 다운받아야 합니다.

PUTTY 다운로드:

http://www.chiark.greenend.org.uk/~ sgtatham/putty/latest.html

Or 구글 에서 putty 검색!

Putty.exe 와 puttygen.exe 를 각각 다운받습니다.

Alternative binary files

The installer packages above will provide all of these (except PuTTYtel), but you can d

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

| putty.exe (th | e SSH and Telnet client itself) | l Bit 확인 후 | . 다운 |
|---------------|---------------------------------|--------------------|-------------|
| 32-bit: | <u>putty.exe</u> | <u>(or by FTP)</u> | (signature) |
| 64-bit: | <u>putty.exe</u> | (or by FTP) | (signature) |

pscp.exe (an SCP client, i.e. command-line secure file copy)

| 32-bit: | pscp.exe | (or by FTP) | (signature) |
|---------|----------|-------------|-------------|
| 64-bit: | pscp.exe | (or by FTP) | (signature) |

psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)

| 32-bit: | psftp.exe | (or by FTP) | (signature) |
|---------|-----------|-------------|-------------|
| 64-bit: | psftp.exe | (or by FTP) | (signature) |

puttytel.exe (a Telnet-only client)

| 32-bit: | <u>puttytel.exe</u> | (or by FTP) | <u>(signature)</u> |
|---------|---------------------|-------------|--------------------|
| 64-bit: | puttytel.exe | (or by FTP) | (signature) |

plink.exe (a command-line interface to the PuTTY back ends)

| 32-bit: | plink.exe | (or by FTP) | <u>(signature)</u> |
|---------|-----------|-------------|--------------------|
| 64-bit: | plink.exe | (or by FTP) | (signature) |

pageant.exe (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)

| 32-bit: | pageant.exe | (or by FTP) | (signature) |
|---------|-------------|-------------|-------------|
| 64-bit: | pageant.exe | (or by FTP) | (signature) |

| puttygen.exe | (a | RSA | and | DSA | key | generation | utility) |
|--------------|----|-----|-----|-----|-----|------------|----------|
| | | | | | | | |

| 32-bit: | <u>puttygen.exe</u> |
|---------|---------------------|
| 64-bit: | <u>puttygen.exe</u> |

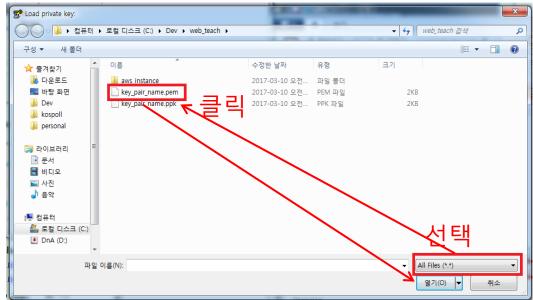
| ľ | t | 옥 | - (| \Box | ļ |
|-----|----|------|-----|--------|---|
| r l | οу | FTP) | | | |



EC2와 연결(Windows)

먼저 puttygen.exe를 실행하여, Load를 클릭후, 확장자를 All files로 변경하고, 아까 AWS에서 받은 key pair 파일(.pem) 을 선택합니다.





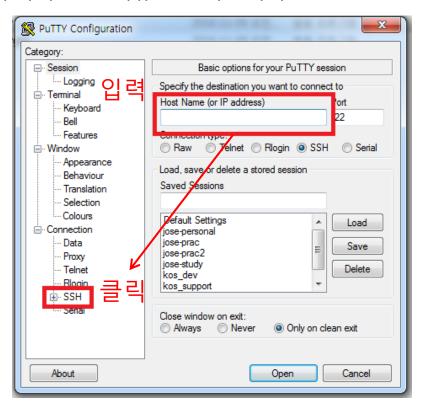
EC2와 연결(Windows)

Save private key 를 클릭해 저장합니다. 이름은 바꿀 수도 있는데 그냥 그대로 저장합니다. 확장자가 .ppk 인 파일이 생성됩니다.

| PuTTY Key Generato | | - | X |
|---|--|--------------------------------------|---------------------------------------|
| File Key Conversion | s Help | | |
| Key Public key for pasting ir | to OpenSSH authorize | d_keys file: | |
| Ei0rF8lQ3KDj3SfkKxU +iNoLSfQBBloBA9mPJ | AAADAQABAAABAQC OopiJRLJNxjbC2gmGX; OuwUy4YAYQ69BBSn II2X4XA/G6Tp8dRanh | g20wnmWMnylllNBc n6K65xudHYCNmVFn | IniRRR2g6YVjZU+ 🗏 vPb7lPKYA9ktUOmt |
| Key fingerprint: | ssh-rsa 2048 8f:ab:c9:08:f6:61:e0:58:bd:c3:35:90:1e:ef:85:33 | | |
| Key comment: | imported-openssh-key | | |
| Key passphrase: | | | |
| Confirm passphrase: | | | |
| Actions | | | |
| Generate a public/priva | te key pair | | Generate |
| Load an existing private key file Load | | | Load |
| Save the generated key Save public key Save private key | | | Save private key |
| Parameters | | | 크리 |
| Type of key to generate SSH-1 (RSA) | e: SSH-2 RSA | ⊚ S: | SH-2 DSA |
| Number of bits in a gen | erated key: | | 2048 |

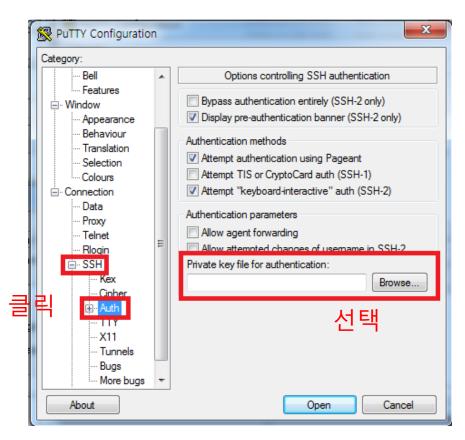
EC2와 연결(Windows)

.ppk 파일도 만들었으니, 이제 접속을 해보도록 하겠습니다. Putty.exe 파일을 켜고 Host Name 에 아까 확인한 public ip를 적어줍니다. 그리고 좌측 메뉴에서 SSH 메뉴로 들어갑니다.



EC2와 연결(Windows)

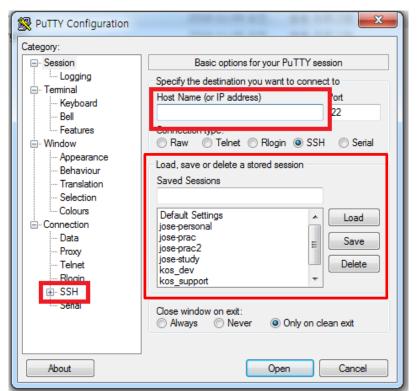
SSH > Auth 로 들어간 다음, Private key file for authentication 에서 Browse를 누르고 아까 생성한 ppk 파일을 선택합니다.



EC2와 연결(Windows)

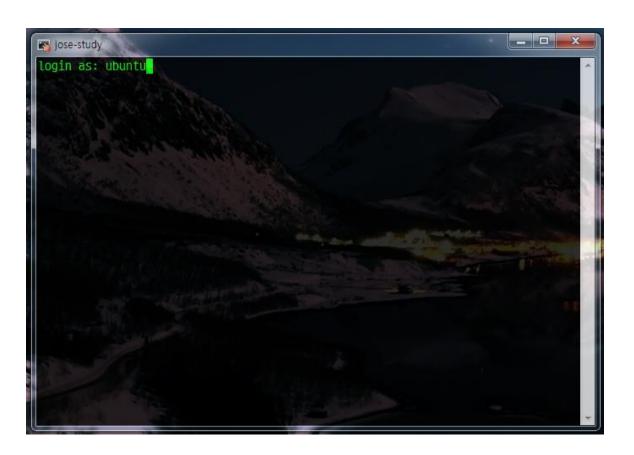
Tip . Key pair 선택 후 open을 누르면 바로 접속이 됩니다. 하지만 다음 번 접속 시 다시 public ip를 입력해야 하는 불편함이 있습니다. 그래서 Saved Sessions 에 이름을 입력 하여 Save를 해두면 다음번에는 더블클릭 또는 Load > open을 통해 편하게 접속이 가

능합니다.



EC2와 연결(Windows)

Login as 에 ubuntu 라고 입력



EC2와 연결(Windows)

접속 완료!



EC2와 연결(Mac)

맥은 조금더 간단하다. 먼저 아까 저장한 pem 파일에 권한을 수정해주고 chmod 400 web-study.pem

그리고 바로 접속 ssh -i "web-study.pem" ubuntu@52.78.180.125

```
seojungwon — -bash — 80×24

Last login: Tue Sep 27 11:18:30 on ttys002

seojungwonui-MacBook-Air:~ seojungwon$ chmod 400 ceos-study.pem
```

```
seojungwon — -bash — 80×24

Last login: Tue Sep 27 11:18:30 on ttys002

seojungwonui-MacBook-Air:~ seojungwon$ ssh -i "ceos-study.pem" ubuntu@52.78.180.

125
```



외부 접속을 위한 설정

이제 Atom 에서도 접속을 하기 위해서 ubuntu 에 몇 가지 작업을 해줘야 한다.

계정의 비밀번호를 만들고, 비밀번호로 로그인이 가능하도록 해줘야 한다. (이 과정을 거치면 key pair 없는 컴퓨터에서도 서버로 로그인이 가능해진다)

외부 접속을 위한 설정

비밀번호 설정

sudo passwd <username> (ex: sudo passwd ubuntu)

비밀번호 두번 입력(비밀번호 짧게 하면, 금방 해킹 당해요)

ubuntu@ip-172-31-10-83:~\$ sudo passwd ubuntu

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

ubuntu@ip-172-31-10-83:~\$

1



외부 접속을 위한 설정

이제 password-login을 허용 하기 위한 세팅 sudo vi /etc/ssh/sshd_config

내려가다 보면, PasswordAuthentication : no 로 되어있는 부분이 있는데, 이걸 yes로 바꾼다.

Vi 명령어 알아둬야 할 것. 수정할 때 i 누르고 시작, 그러면 하단에 insert 라고 표시가 되고, 그때 지우고 입력한 다음. Esc를 눌러서 insert모드를 취소하고 :wq를 입력해서 저장 후 종료

```
# To enable empty passwords, change to yes (NOT RECOMMENDED)

PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with # some PAM modules and threads)

ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords

PasswordAuthentication no

# Kerberos options

# Kerberos Authentication no
```

외부 접속을 위한 설정

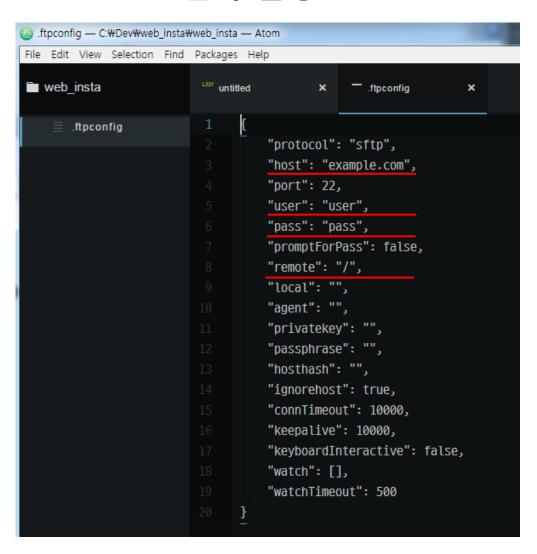
저장하고 빠져나왔으면, ssh재시작 sudo service ssh restart

```
ubuntu@ip-172-31-10-83:~$ sudo service ssh restart
ssh stop/waiting
ssh start/running, process 1759
ubuntu@ip-172-31-10-83:~$
```

이제 ppk 파일 없이도 바로 접속가능!

이제 Atom 에서 연결해보자!

Atom 으로 접속 설정



이 화면이 기억이 날지 모르겠다.

Atom 설치 후 처음 만든 폴더를 열고 .ftpconfig 를 열면 좌측과 같 은 화면이 열린다.

이제 밑줄그은 부분을 채워 넣도 록 하자!

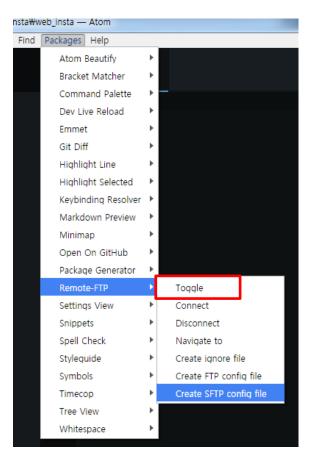
Host에는 public ip를 User 에는 ubuntu를 Pass 에는 아까 설정한 비번을 Remote 에는 /home/

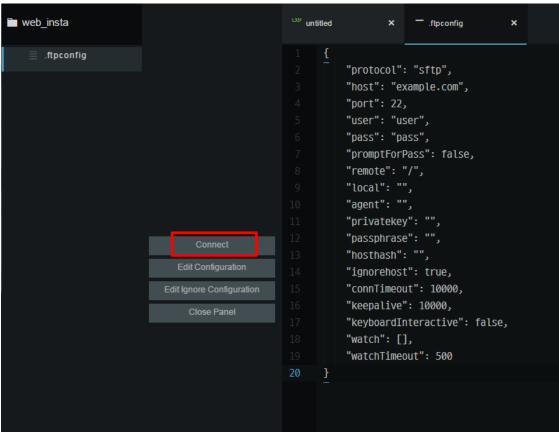
이라고 적고 저장하자.

Atom 으로 접속 설정

Packages > Remote-FTP > Toggle 선택하면 우측과 같은 메뉴가 뜬다.

여기서 이제 Connect를 누르면

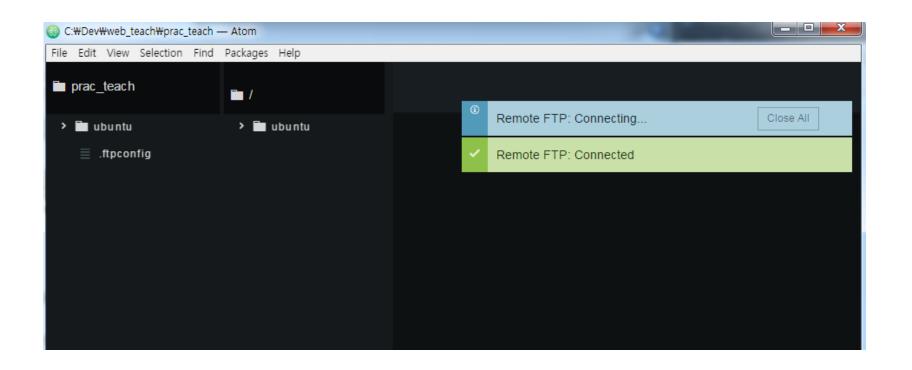






Atom 으로 접속 설정

이렇게 서버와 연결된 것을 확인 할 수 있다!



2. Node.js / NPM 설치 및 기본 설정

Node.js 및 NPM 설치

- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo apt-get install nodejs
- \$ sudo apt-get install npm
- \$ sudo In -s /usr/bin/nodejs /usr/bin/node

아래 두 명령어로 설치된 node.js 와 npm의 버전을 확인할 수 있다.

```
$ node -v
$ npm -v
```

```
      ubuntu@ip-172-31-14-194:~$ node -v
      Reserved instances

      v4.2.6
      ■ .bash_logc

      ubuntu@ip-172-31-14-194:~$ npm -v
      Dedicated Hosts

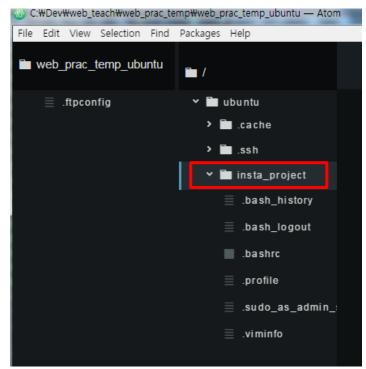
      3.5.2
      ■ .bashrc

      ubuntu@ip-172-31-14-194:~$ []
      AMIs
```

Express.js 설치

먼저 프로젝트 폴더를 만든다.

Atom 의 ubuntu 폴더에 insta_project 라는 폴더를 만든다.



Express.js 설치

프로젝트 폴더 내에 package.json 파일을 만들고. 주어진 내용을 복붙한다.

https://github.com/loofoomj/node_inst a/blob/master/package.json

이때 name과 author는 각자 알아서 변경.

```
📅 package.json
1

✓ ■ ubuntu

                                 "name": "insta_project",
   > i .cache
                                 "version": "1.0.0"
   > 🗎 .ssh
                                 "main": "index.js";

▼ insta_project

                                 "scripts": {
                                   "test": "echo \"Error: no test specified\" && exit 1".
        package.json
                                    "start": "node server.js"
      bash history
      _ .bash logout
                                 "author": "jose"
                                 "license": "ISC"
      .bashrc
      ______.profile
                                   "body-parser": "^1.17.1",
                                    "connect-flash": "^0.1.1",
        .sudo_as_admin_:
                                   "cookie-parser": "^1.4.3",
      _viminfo
                                    "date-utils": "^1.2.21",
                                    "ejs": "^2.5.6",
                                    "express": "^4.15.2",
                                    "express-mysgl-session": "^1.2.0";
                                    "express-session": "^1.15.1",
                                   "fs": "0.0.1-security",
                                    "morgan": "^1.8.1",
                                    "multer": "^1.3.0";
                                    "mysql": "^2.13.0",
                                   "passport": "^0.3.2";
                                    "passport-local": "^1.0.0"
                                 "devDependencies": {},
                                 "keywords" []
                                 "description": ""
```



Express.js 설치

터미널(win-putty/mac-terminal)에서

\$ cd insta_project(또는 각자 만든 프로젝트 폴더 이름)

\$ sudo npm install

하면 아래와 같이 설치가 되고, Atom 에서 refresh를 해보면 node_modules라는 폴더가 생성된 것을 확인할 수 있다.

```
ubuntu@ip-172+31-14-194:~$ cd insta_project/"date-upuntu@ip-172-31-14-194:~/insta_project$ lspackage.json"ejs"ubuntu@ip-172-31-14-194:~/insta_project$ sudo npm install17insta_project@1.0.0 /home/ubuntu/insta_project"expression of the project of
```

4

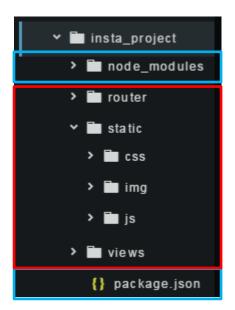
Express.js 기본 디렉토리 구조

Express.js Framework를 이용하여 생성할 프로젝트의 기본 구조는 오른쪽에 보이는 형태로 구성 한다.

파란색 사각형 > 미리 만들어 놓은 package.json 파일 및 npm install 시 생성되는 node_modules

빨간색 사각형 > 사용하기 용이하게 직접 만드는 폴더들

폴더: router / static – css, img,js / views 총 6개



Express.js 기본 디렉토리 구조

Router – 서버 쪽에서 처리할 내용을 적는 파일이 담길 폴더

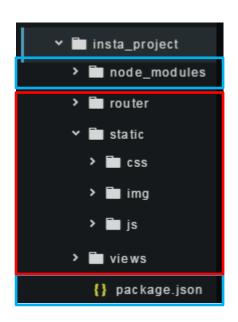
Static – 정적 파일을 담는 폴더

> css : css 파일들을 담을 폴더

> img : 이미지 파일들을 담을 폴더

> js : javascript 파일들을 담을 폴더

Views - 사용자에게 보여질 템플릿(HTML) 파일을 담을 폴더



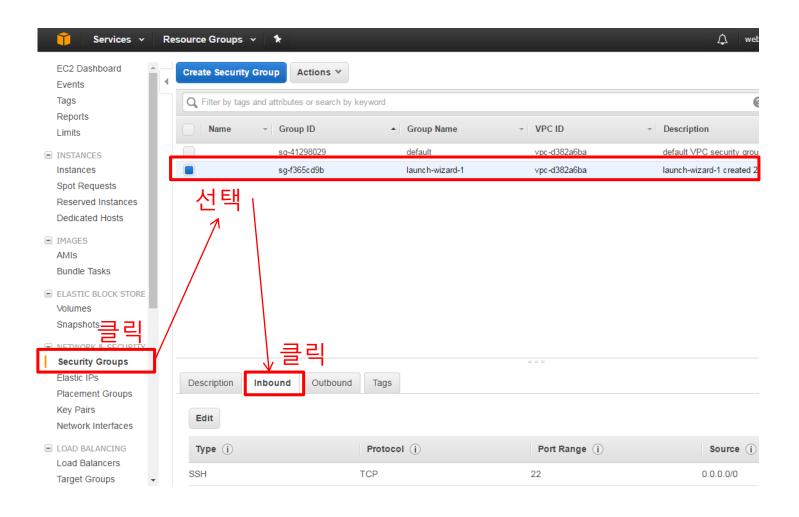
MySQL / phpmyadmin 설치

- \$ sudo apt-get install apache2
- \$ sudo apt-get install php libapache2-mod-php php-xml php-gd php-mysql
- \$ sudo apt-get install mysql-server mysql-client
- >> 비번 쓰라고 나오는데 이제부터 비밀번호는 처음 ubuntu 비번과 같이 통일! (까먹을 수도 있으니까)
- \$ sudo apt-get install phpmyadmin
- \$ sudo apt-get install php-mbstring php-gettext
- \$ sudo service apache2 restart

이제 AWS로 가서 사용할 포트를 열어줘야한다.

MySQL / phpmyadmin 설치

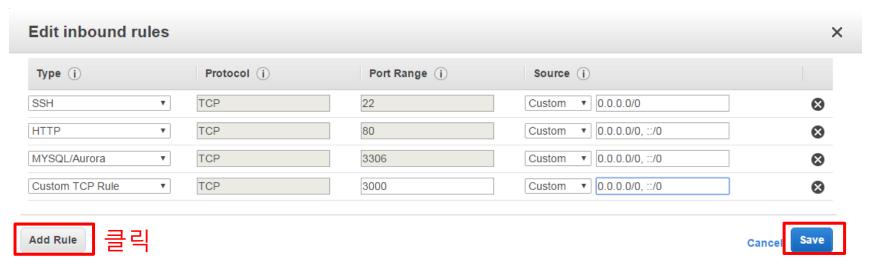
AWS instance 화면에서 Security Groups > launch-wizard-1 선택 > Inbound 클릭



MySQL / phpmyadmin 설치

Edit 를 눌러서 나온 화면에서

Add Rule 을 눌러서 아래와 같이 채우고 Save 클릭





MySQL / phpmyadmin 설치

이제 phpmyadmin 으로 접속 하기 위한 설정을 한다.

터미널로 돌아와서

\$ sudo vi /etc/apache2/apache2.conf

ubuntu@ip-172-31-14-194:~/insta_project\$ sudo vi /etc/apache2/apache2.conf

입력하고, 파일이 열리면 맨 아래로 내려간다.

그 상태에서 키보드 i를 누르고,

Include /etc/phpmyadmin/apache.conf

라고 입력 후 esc를 눌러 insert 모드를 종료하고 :wq를 입력해 빠져나온다.

```
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
Include /etc/phpmyadmin/apache.conf
222,1 Bot
```

MySQL / phpmyadmin 설치

아파치 재시작 \$ sudo service apache2 restart

> ubuntu@ip-172-31-14-194:~/insta_project\$ sudo vi /etc/apache2/apache2.conf ubuntu@ip-172-31-14-194:~/insta_project\$ sudo service apache2 restart

인터넷 브라우저에서 http://본인 public ip/phpmyadmin/ 로 접속!

사용자명은 root 비밀번호는 계속 쓰던 그거



다음 시간에는...

- 1. Node.js 기본 연결
- 2. HTML / CSS 1