

# 삼성 청년 SW 아카데미

Node.js

## <알림>

본 강의는 삼성 청년 SW아카데미의 컨텐츠로  
보안서약서에 의거하여  
강의 내용을 어떠한 사유로도 임의로 복사,  
촬영, 녹음, 복제, 보관, 전송하거나  
허가 받지 않은 저장매체를  
이용한 보관, 제3자에게 누설, 공개,  
또는 사용하는 등의 행위를 금합니다.

# 1장. 클라이언트와 서버

# 챕터의 포인트

- 클라이언트와 서버
- IP 와 Port

## 클라이언트와 서버

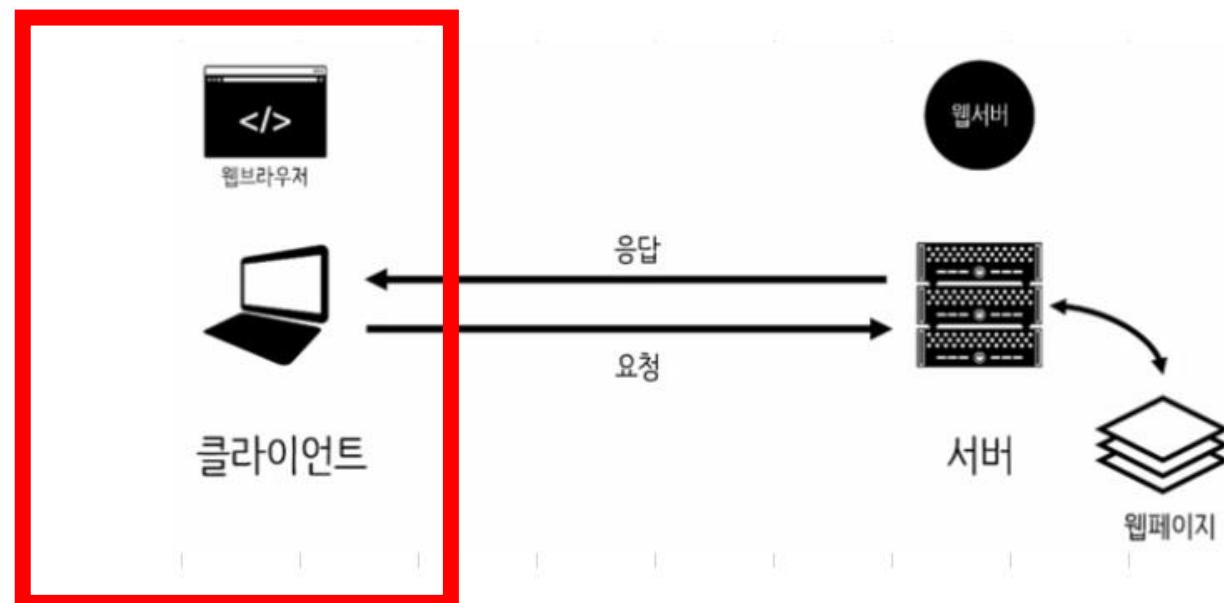
## 최종목표

- 클라이언트 요청 발생 시
- DB에 접근해서 CRUD 한 후
- 적절한 결과를 JSON으로 리턴하는
- Node.js Express REST API 서버 제작

우선, 클라이언트와 서버의 개념부터 알아보자.

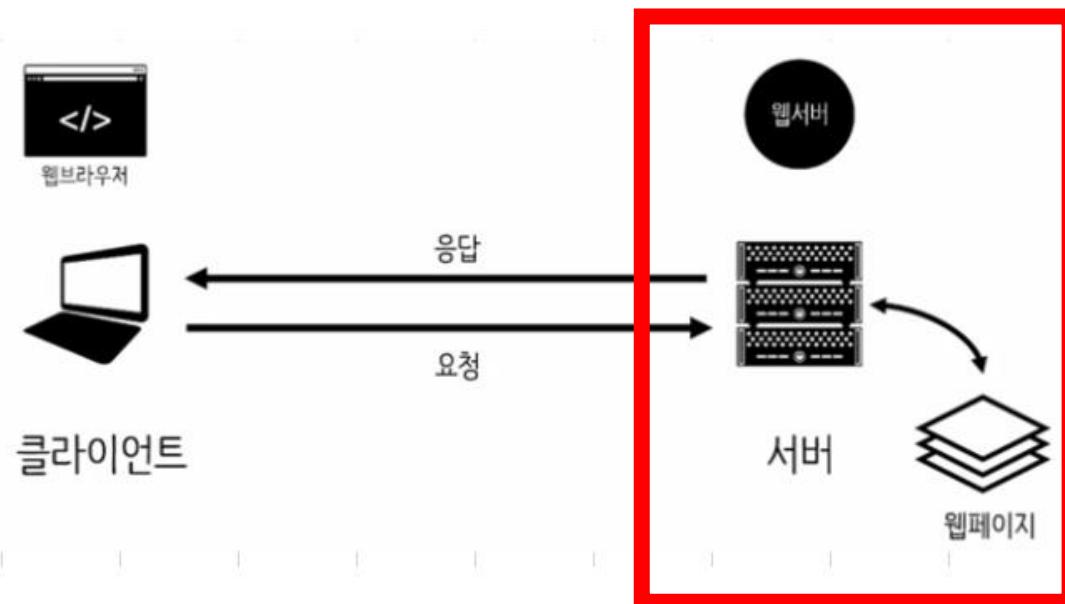
## 프론트엔드

- 웹에서는 클라이언트가 프론트엔드
- 사용자 눈에 보이고 상호작용하는 부분
- 서버를 통해 접속할 수 있는 애플리케이션이나 서비스
  - ex) 브라우저, 웹사이트
- 서버에게 요청 ( request ) 하면, 서버로부터 응답 ( response ) 을 받는다.
- 언어 및 기술: HTML, CSS, JavaScript, React.js, Vue.js ...



## 다른 말로 백엔드

- 사용자 눈에 보이지 않는 곳에서 작동하는 부분
- 클라이언트에게 정보나 서비스를 제공하는 시스템
  - ex) 웹서버, 데이터베이스 서버
- 클라이언트의 요청 ( request ) 을 받으면, 적절한 처리 후 클라이언트에게 응답 ( response )
- 언어 및 기술: Java, Python, Node.js, Server, DB, API ...



## 타워형, 랙형으로 나뉨

- 일반 데스크톱도 얼마든지 쓸만한 서버가 될 수 있으나,
- 서버 컴퓨터의 CPU (인텔 제온) 는 데스크톱 CPU 와는 다르게
- 멀티스레드 기능을 보강



## 각 서버는 역할이 존재

- Client 들에게 HTTP 문서를 전달해주는 컴퓨터: 웹서버
- Client 들의 DB 를 관리하는 서버 : DB 서버
- Client 들의 게임 데이터를 관리 & 처리 해주는 컴퓨터 : 게임 서버

# IP 와 Port

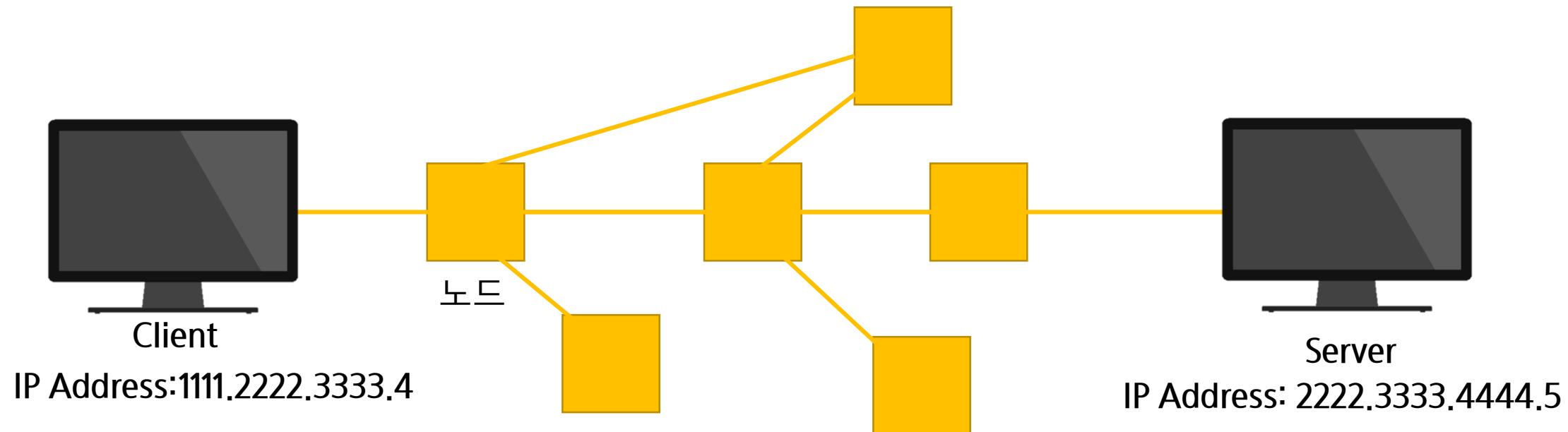
# IP(Internet Protocol)

- Protocol : 규칙
- 인터넷에서, 송신 호스트와 수신 호스트가 정보를 주고받는데 사용하는 규칙
  - Host : 네트워크에 연결된 컴퓨터 또는 주변기기를 의미 ex) PC, 프린터, 서버, 노트북, 스마트폰 ...
- 송신 호스트와 수신 호스트는 **동일한 프로토콜을 사용해야 통신 가능**



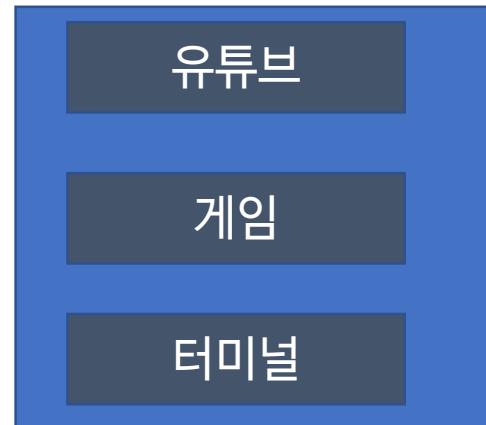
## IP Address

- 인터넷상에 있는 컴퓨터의 고유한 주소



## 예시

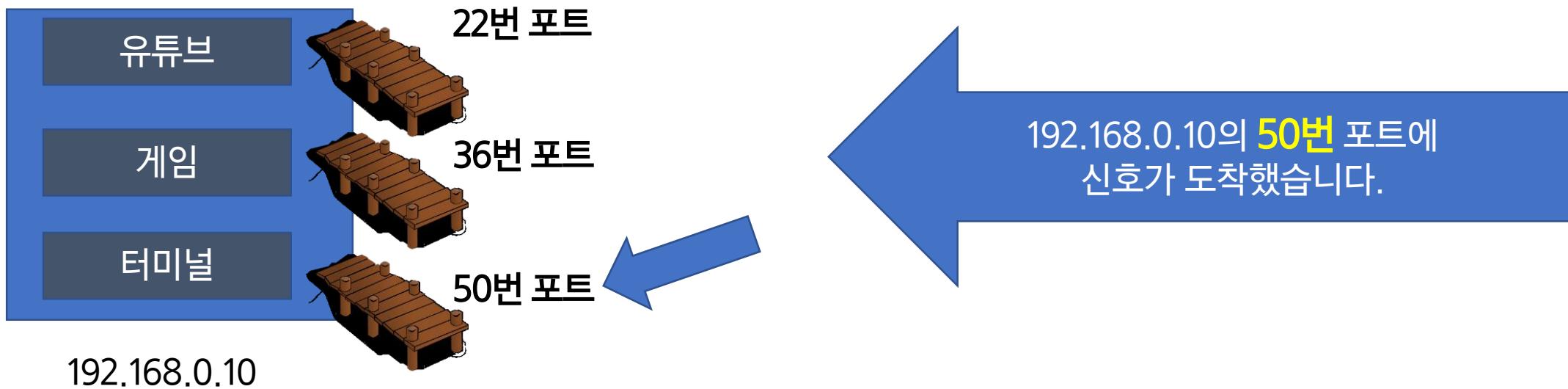
- 한 컴퓨터에서 통신을 하는 프로그램이 여러개가 있다.



컴퓨터  
(192.168.0.10)

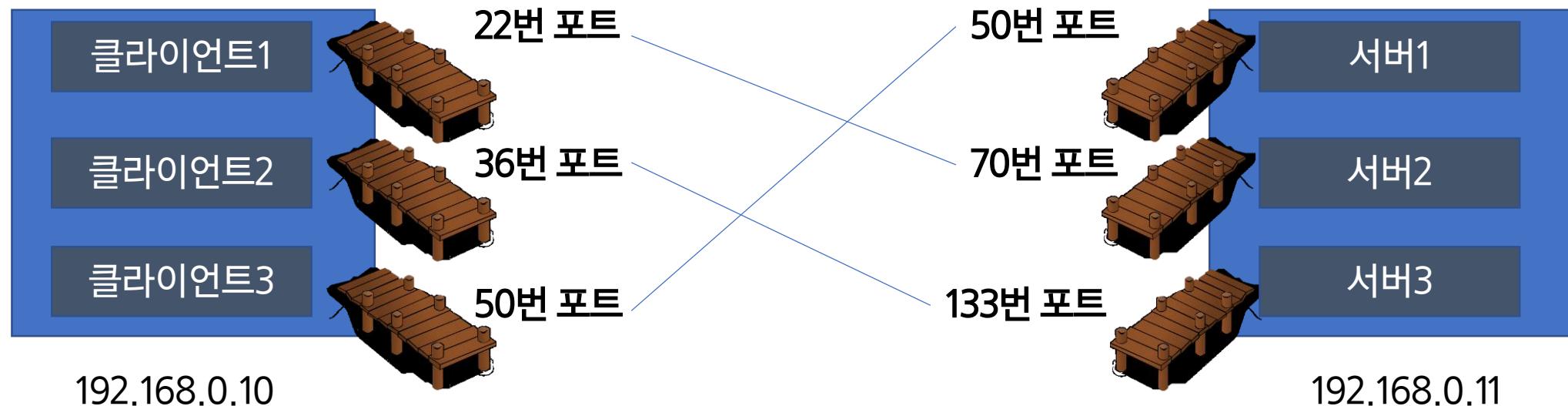
## Port

- 어떤 신호가 들어왔을 때,
- 어떤 프로그램에게 도착한 신호인지, 구분을 위한 번호



## IP 포트 번호

- IP + 포트번호 까지 알아야, 프로그램끼리 통신이 가능



## 22번 포트

- ssh, sftp 프로토콜을 쓸 때 자주 사용되는 포트 번호 (원격 접속 시 사용)

## 80번 포트

- http 프로토콜을 쓸 때 자주 사용되는 포트번호
- 80번 이외에도 8000, 8080, 8081 ... 등 사용
- (Node.js 수업 때는 8080, 8081 번 포트를 지정해서 쓸 것)

국제관리기구에서 정한 포트 번호이다.  
well-known port number 라고 하며,  
권장사항이다.

## Well-known Port Number (0 ~ 1023)

- Document에서 Well-known 은 예약된 이라는 뜻으로 쓰인다.
  - 80 / 443 : HTTP / HTTPS
  - 22 : FTP / SSH

## Registered Port (1024 ~ 49151)

- 여러 Tool이 쓰는 포트들
  - 3306 : MySQL
  - 8080 : HTTP 대체용

## Dynamic Port (49152 ~ 65535)

- 필요할 때마다 임의의 번호를 사용하는 곳

## 2장. 클라우드 서비스

# 챕터의 포인트

- 클라우드 서비스란
- 리눅스 개요

## 클라우드 서비스란

- cloud = 구름
  - 구름 속에 무엇이 있는지 몰라도, 얼마든지 사용자가 원하는 것을 꺼내어 사용
- 클라우드 서비스는 컴퓨터를 대여해주는 서비스
  - 그러나, 하나의 완성된 서버 컴퓨터를 빌려주는 게 아니라,  
고객 요청 시 적절한 컴퓨터 자원을 가져와 **가상의 컴퓨터**를 만들어 대여함
    - ex) CPU, 메모리, 디스크 ...
  - 정확하게 말하자면, 컴퓨터 자체가 아니라, **컴퓨터의 자원을 빌려오는 것**
  - 사용자는 초당 사용료 지급



## 아마존의 AWS 사업

- Black Friday, 추수감사절, 아마존 이벤트시 최고 트래픽 발생
- 트래픽 관리 노하우를 AWS로 운영 시작



## 전산실 운영 비용 아낄 수 있음

- 월 정기 비용 : 다량의 서버관리자 급여 / 전기비 / 사무실 임대료 / 회선비용
- 일시적 비용 : L4 스위치 / 방화벽 / 대당 서버 비용 / 랙 / 항온항습장치 / UPS

## 클라우드 이용시

- 월 정기 비용 : 한 명의 서버관리자 급여 / 시스템 사용료
- 일시적 비용 : 없음

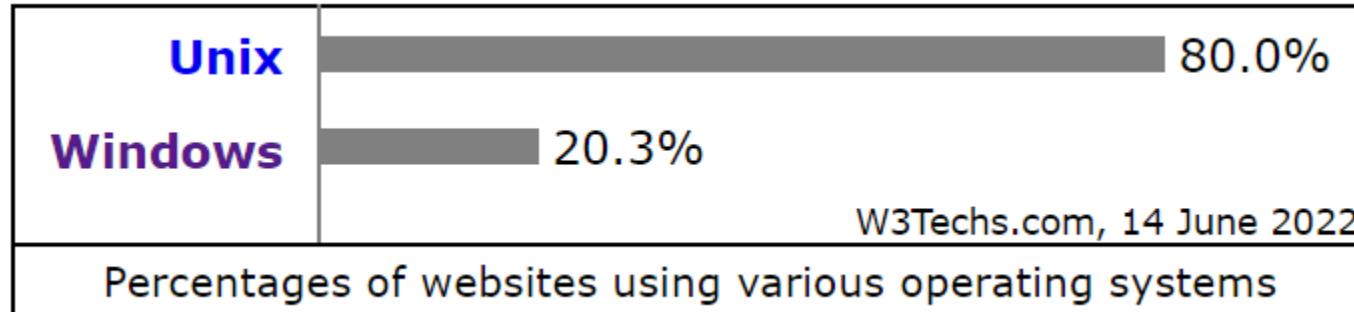
리눅스 개요

## 여러가지 운영체제

- Windows
  - PC시장(Personal Computer) 지배
- 리눅스
  - 서버
  - 임베디드 OS 시장 지배
- Android
  - 리눅스로 만들어진 Mobile OS
- MacOS
  - 유닉스로 만들어진 Apple OS

## 서버용

- 2022년 기준, Unix + Linux 가 서버의 약 80% 차지
- W3Techs.com 자료에는 Linux가 Unix 계열이기에, Unix로 표기가 되어있음
  - Unix : C 언어 제작자가 만든 OS, 이 계열로 대표적으로 iOS, macOS, Linux 등이 있음
  - Linux : Unix 계열 운영체제로, 리눅스토발즈가 Unix 를 토대로 만든 OS



## Ubuntu

- 무료
- 리눅스 배포판 중 가장 널리 쓰이는 배포판
- 영국 캐노니컬에서 유지보수 중



## 스택

- 클라우드 서비스: AWS
- 운영체제: Ubuntu 18.04 LTS
- 서버: Node.js
- 데이터베이스: MySQL



# 3장. AWS

# 챕터의 포인트

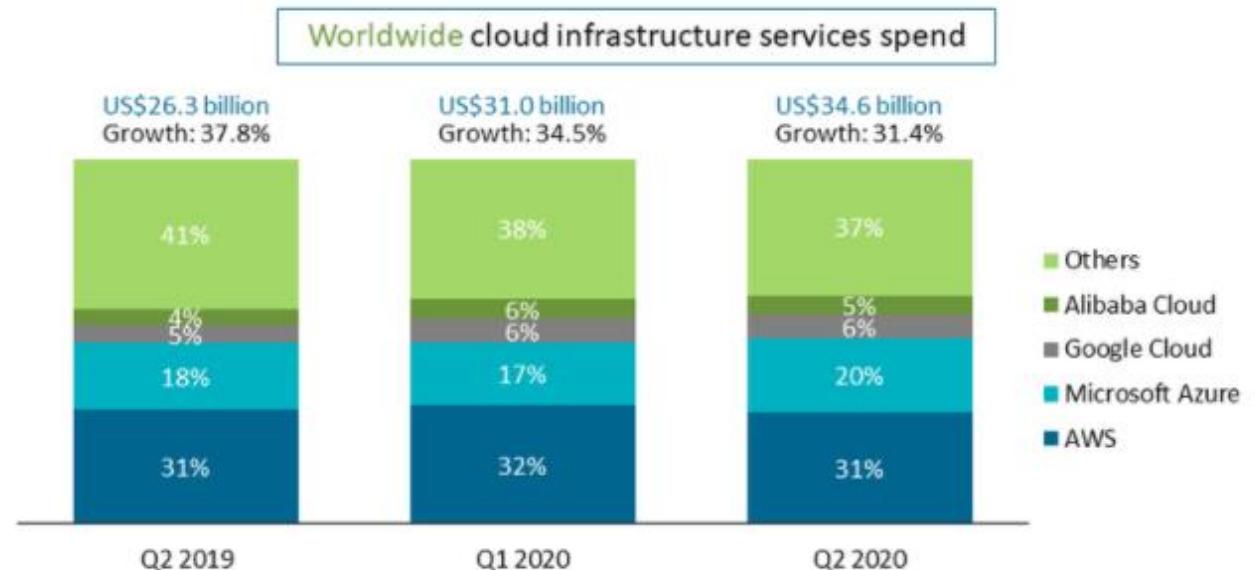
- AWS 개요
- AWS 인스턴스 생성
- SSH 연결

# AWS 개요

# Amazone Web Service

- AWS은 2016년 한국 리전 시작, 가용영역 4개 (전 세계에서 네 번째 규모)
- Google Clude Platform은 2020년 한국 리전 시작

Top four providers account for 63% of cloud spend

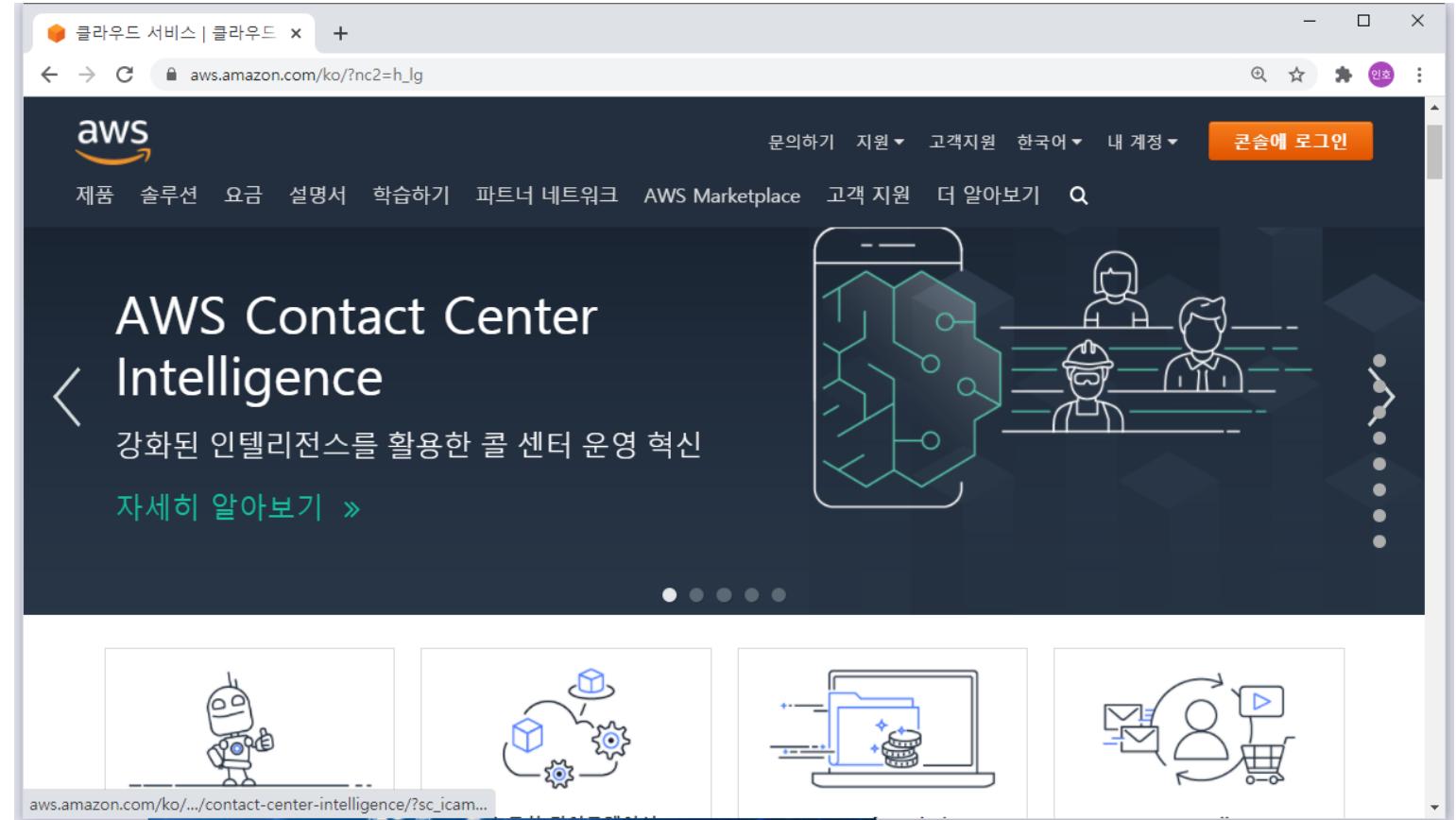


Source: Canalys estimates, July 2020

 canalys

## 신용카드 기입 필요

- 1 달러 테스트 결제가 된 후  
2 ~ 3 주 후 취소 & 환불 됨

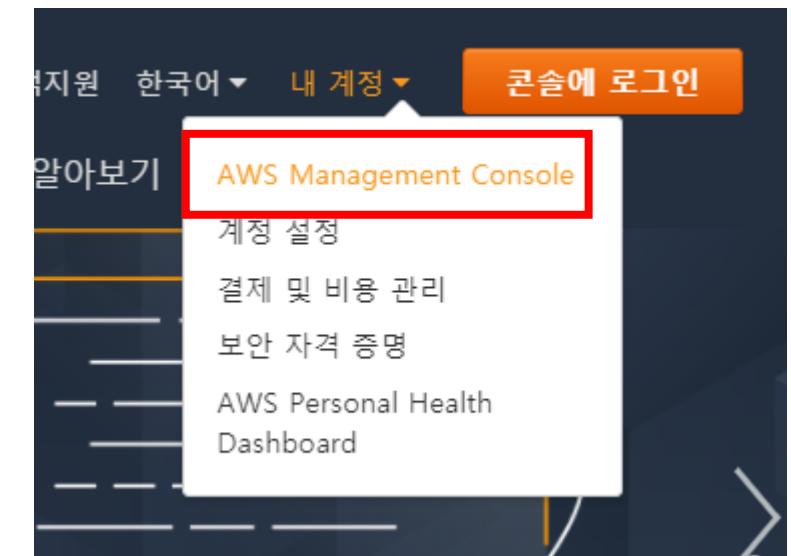
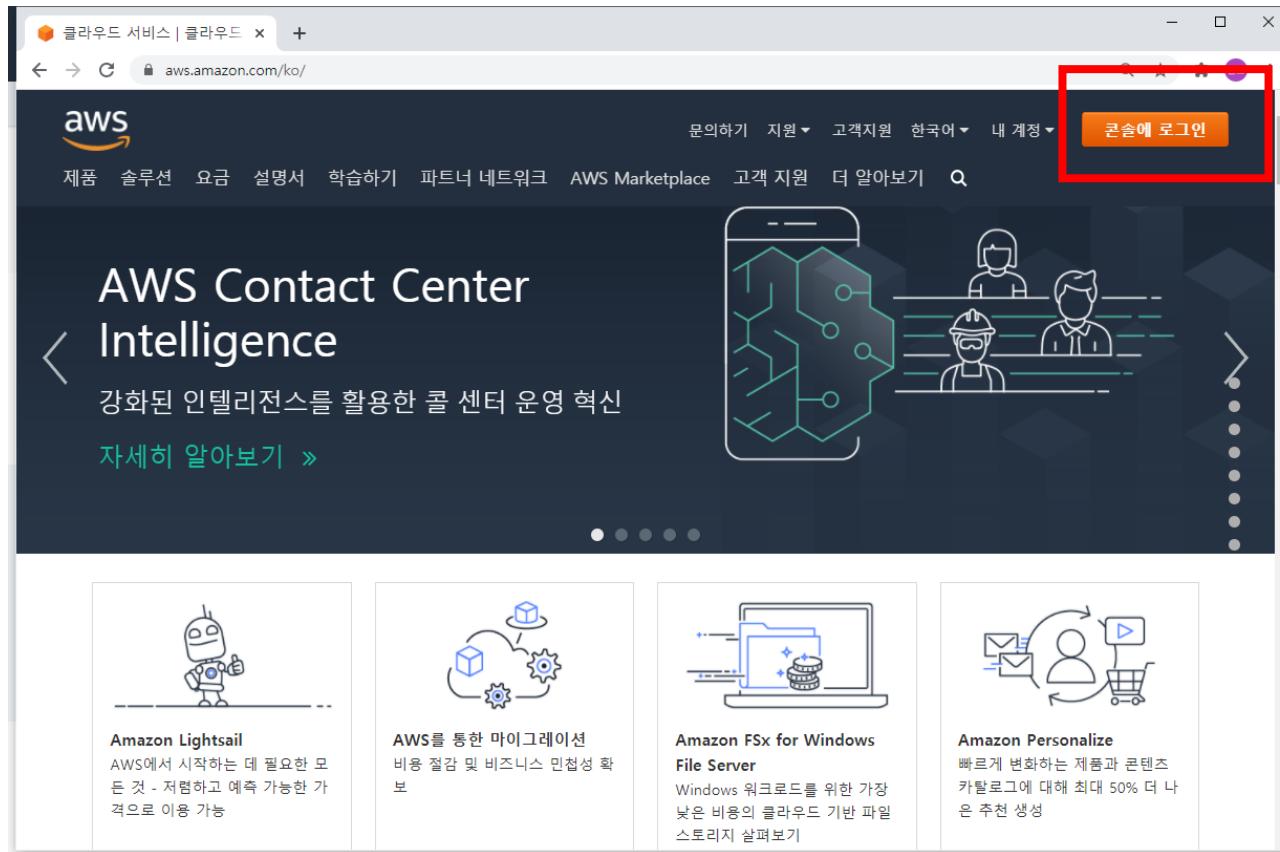


The screenshot shows the AWS Billing Management Console interface. The left sidebar has a tree view with nodes like 'Home', '결제' (Payment), '청구서' (Bills), '결제' (Payment), '크레딧' (Credit), '구매 주문' (Purchase Order), 'Cost & Usage Reports', 'Cost Categories', '비용 할당 태그' (Cost Allocation Tag), 'Free Tier', 'Billing Conductor', 'Cost Management', 'Cost Explorer', 'Budgets', 'Budgets Reports', '절감 계획' (Savings Plan), '기본 설정' (Basic Settings), '결제 기본 설정' (Payment Basic Settings), '결제 방법' (Payment Methods) which is highlighted in orange, '통합 결제' (Unified Payment), and '세금 설정' (Tax Settings). The main content area is titled '결제 방법' (Payment Methods) and shows a message: '이제 결제 프로필 기능을 사용할 수 있습니다.' (You can now use the payment profile feature). It also displays a 'VISA' logo under '결제 방법' and fields for '청구지 주소/기본 결제 방법 주소' (Billing Address/Billing Method Address) and '전화번호' (Phone Number). Below this, there's a section titled '결제 프로필' (Payment Profile) with a note about using it for AWS services and a '결제 프로필 보기' (View Payment Profile) button. At the bottom, there's a table with columns '신용카드' (Credit Card), '카드에 기입된 이름' (Name on Card), and '만료 날짜' (Expiration Date), showing a VISA card with a blurred number. Buttons at the bottom include '편집' (Edit), '삭제' (Delete), and '현재 기본값' (Current Default). A red box highlights the '카드 추가' (Add Card) button at the bottom left.

카드 추가 버튼을 클릭해 결제카드 등록

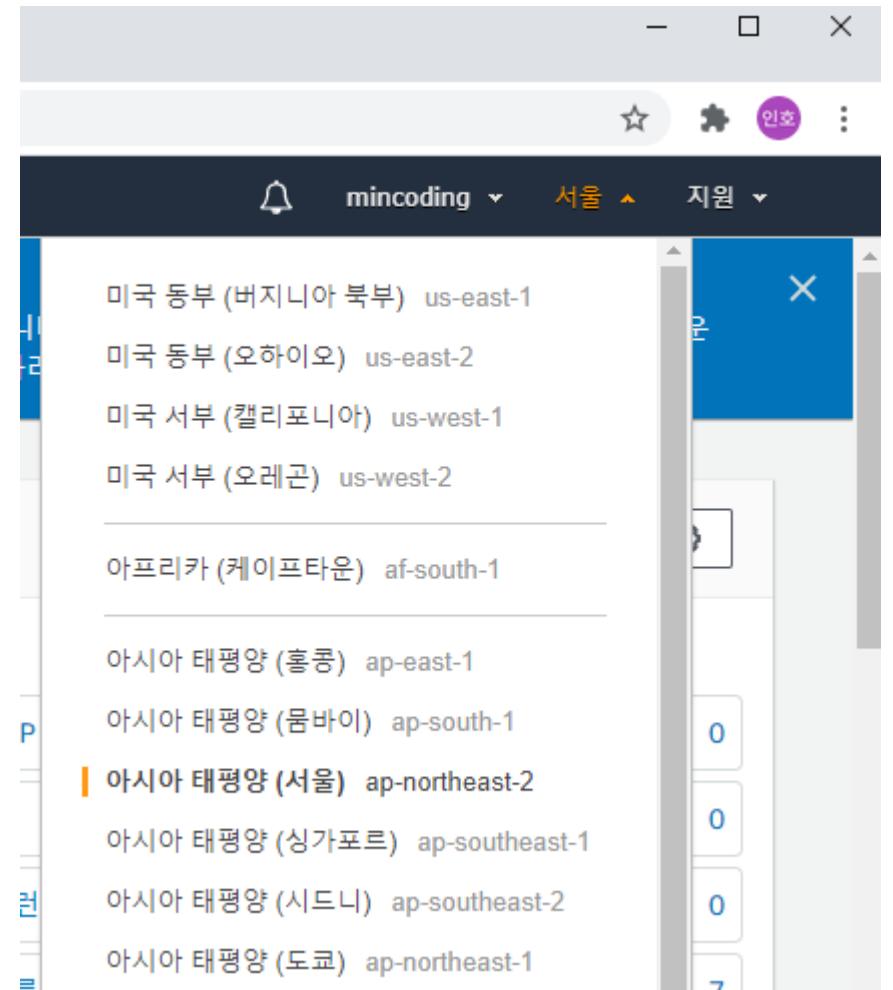
## AWS 서비스를 이용하려면 “콘솔에 로그인” 필요

- 모든 서비스는 콘솔에서 시작함
- AWS Management Console = 콘솔



## 서울 리전을 선택

- 서버가 운영될 지역을 선택
- 가까울 수록 속도가 빠름
- 리전별 가격이 다름



## 사용량 알림받기

- 프리티어는 1년간 무료제공
- 소멸시 알림이 되도록 체크

The screenshot shows the AWS Billing console's 'Basic Settings' page. On the left, there is a sidebar with various navigation options like '결제 기본 설정', '결제 방법', '통합 결제', and '세금 설정'. The main content area is titled '기본 설정' and contains two sections: '결제 기본 설정' and '비용 관리 기본 설정'. In the '비용 관리 기본 설정' section, there is a checkbox labeled '프리 티어 사용량 알림 받기' which is checked. Below it, there is a text input field for '이메일 주소' containing 'test@example.com'. This entire section is highlighted with a red box.

기본 설정

결제 기본 설정

이메일로 PDF 인보이스 받기

이 기능을 설정하면 이메일로 PDF 버전의 인보이스를 제공받을 수 있습니다. 인보이스는 보통 매달 첫 3일 이내에 제공됩니다.

비용 관리 기본 설정

프리 티어 사용량 알림 받기

이 기능을 설정하면 AWS 서비스가 AWS 프리 티어 사용량 제한에 도달하거나 초과한 경우 이메일로 알림을 수신합니다. 이 알림은 본 계정에 연결되어 있는 기본 이메일 주소 이외의 주소로 수신하고자 하는 경우에는 아래에 해당 이메일 주소를 지정하세요.

이메일 주소: test@example.com

결제 알림 받기

이 기능을 설정하면 자동으로 AWS 사용 요금 및 기본 요금을 모니터링하여 AWS 관련 소비를 쉽게 추적 및 관리할 수 있습니다. 결제 알림을 설정하면 요금이 특정 임계값에 도달할 때 이메일 알림을 받을 수 있습니다. 이 기본 설정은 한 번 설정하면 해제할 수 없습니다. [결제 알림 관리](#) 또는 [새로운 예산 기능을 사용해 보세요!](#)

▶ 세부 결제 보고서[레거시]

**기본 설정 저장**

## AWS 인스턴스 생성

# Elastic Compute Cloud (EC2)

- 가상컴퓨터
- AWS 의 대표 서비스
- 컴퓨터 1대 = 인스턴스

The screenshot shows the AWS EC2 console interface. On the left, a sidebar menu includes options like 'New EC2 Experience' (with a feedback link), 'EC2 대시보드 New', '이벤트 New', '태그', '제한', '인스턴스' (selected and highlighted with a red box), '인스턴스 유형', '시작 템플릿', '스팟 요청', 'Savings Plans', '예약 인스턴스', '전용 호스트 New', '용량 예약', and '이미지'. The main content area is titled '리소스' and displays resource counts: 0 running instances, 0 snapshots, and 5 key pairs. Below this is a section titled '인스턴스 시작' with a button labeled '인스턴스 시작 ▾'.

1. 리전 서울로 지정

새 위젯 출시 공지입니다. 콘솔 홈의 하단에서 찾을 수 있습니다.

최근에 방문한 서비스 정보

EC2

Database Migration Service

VPC

AWS Billing Conductor

AWS Organizations

Route 53

RDS

S3

2. EC2 클릭

AWS 시작

AWS 시작하기

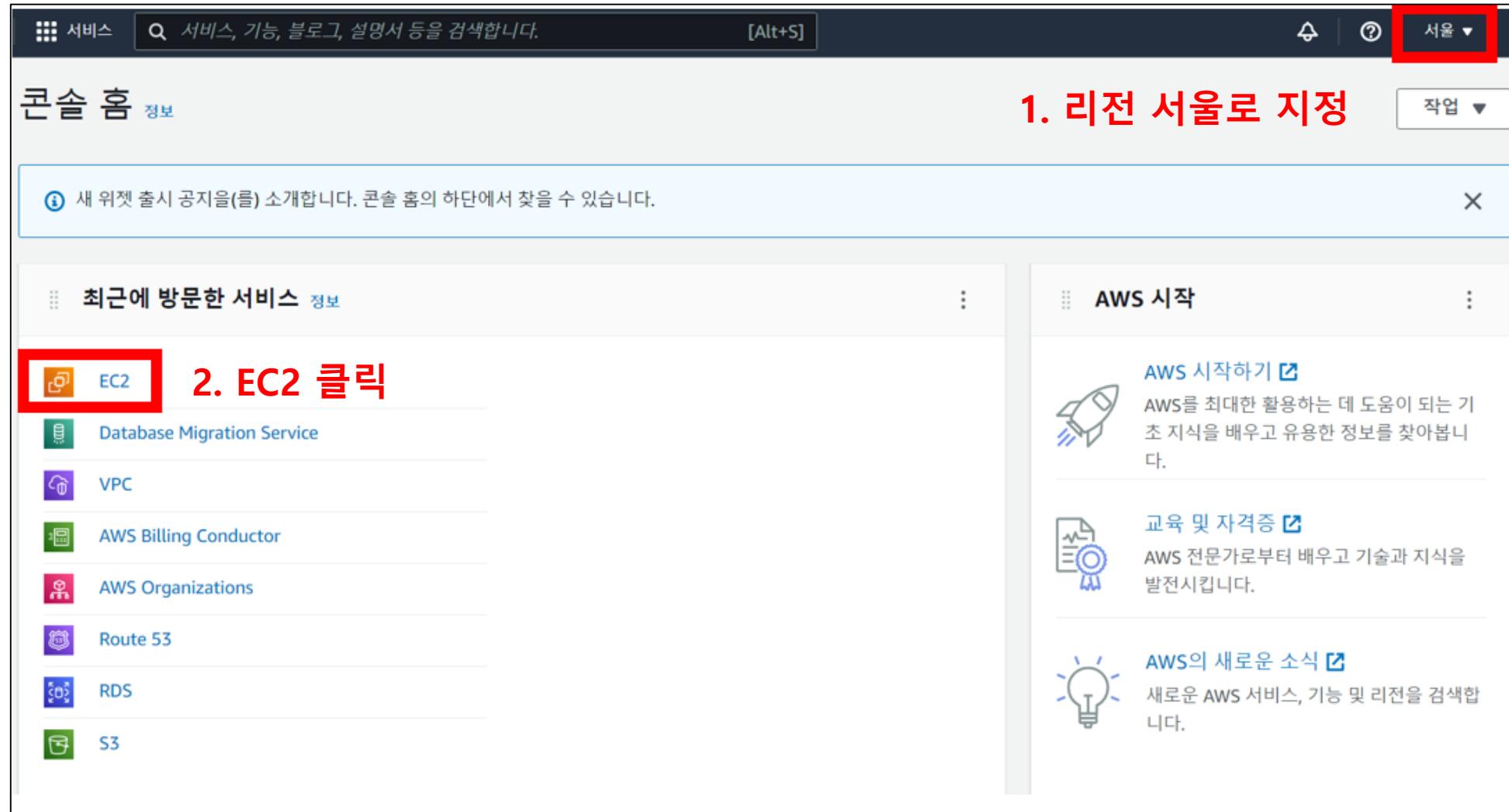
AWS를 최대한 활용하는 데 도움이 되는 기초 지식을 배우고 유용한 정보를 찾아봅니다.

교육 및 자격증

AWS 전문가로부터 배우고 기술과 지식을 발전시킵니다.

AWS의 새로운 소식

새로운 AWS 서비스, 기능 및 리전을 검색합니다.



## 리소스

아시아 태평양 (서울) 리전에서 다음 Amazon EC2 리소스를 사용하고 있음:

인스턴스(실행 중)	1	로드 밸런서	0	배치 그룹	0
보안 그룹	2	볼륨	1	스냅샷	1
인스턴스	1	전용 호스트	0	키 페어	1
탄력적 IP	1				

## 인스턴스 시작

시작하려면 클라우드의 가상 서버인 Amazon EC2 인스턴스를 시작하십시오.

**인스턴스 시작 ▾**

**서버 마이그레이션**

참고: 인스턴스는 아시아 태평양 (서울) 리전에서 시작됩니다.

## 서비스 상태

리전  
아시아 태평양 (서울)

상태  
**✓ 이 서비스가 정상적으로 작동 중입니다.**

EC2 > 인스턴스 > 인스턴스 시작

## 인스턴스 시작 정보

Amazon EC2를 사용하면 AWS 클라우드에서 실행되는 가상 머신 또는 인스턴스를 생성할 수 있습니다. 아래의 간단한 단계에 따라 빠르게 시작할 수 있습니다.

### 이름 및 태그 정보

이름

ssafy-8th-node-aws

추가 태그 추가

식별하기 쉽게, 이름을 지어주자.

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

최근 사용 Quick Start

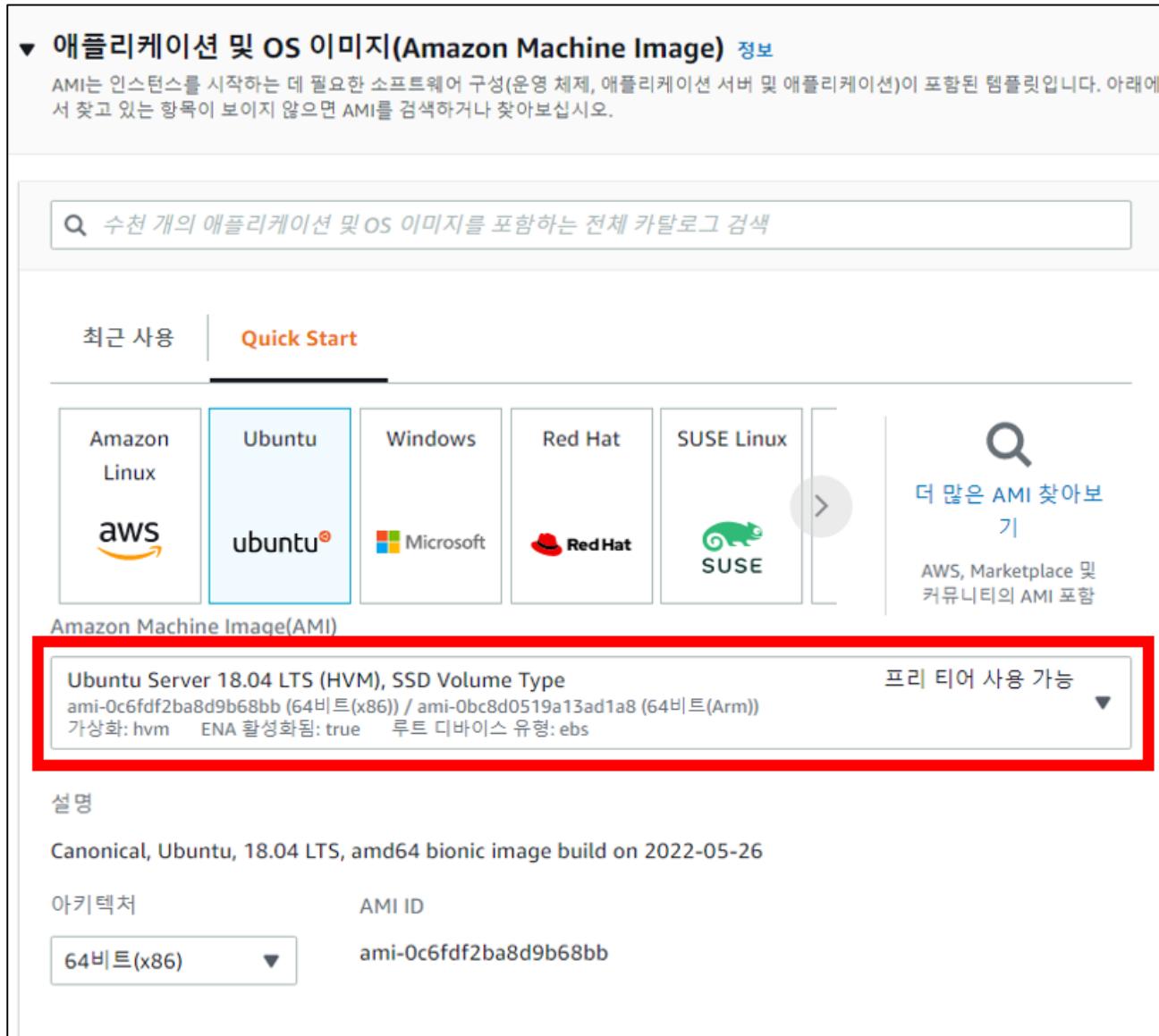
Amazon Linux Ubuntu Windows Red Hat SUSE Linux > 더 많은 AMI 찾아보기 AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type  
 ami-0c6fdf2ba8d9b68bb (64비트(x86)) / ami-0bc8d0519a13ad1a8 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

설명  
Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-05-26

아키텍처 AMI ID  
64비트(x86) ami-0c6fdf2ba8d9b68bb



프리티어 사용 가능한  
Ubuntu Server 18.04 LTS



## 키 페어 생성

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

아래에 키 페어의 이름을 입력합니다. 메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

키 페어 이름  
 키 파일 이름 지정

이름은 최대 255개의 ASCII 문자를 포함할 수 있습니다. 선택 또는 후행 공백은 포함할 수 없습니다.

키 페어 유형

RSA  
RSA 암호화된 프라이빗 및 퍼블릭 키 페어

ED25519  
ED25519 암호화된 프라이빗 및 퍼블릭 키 페어(Windows 인스턴스에는 지원되지 않음)

프라이빗 키 파일 형식

.pem  
OpenSSH와 함께 사용

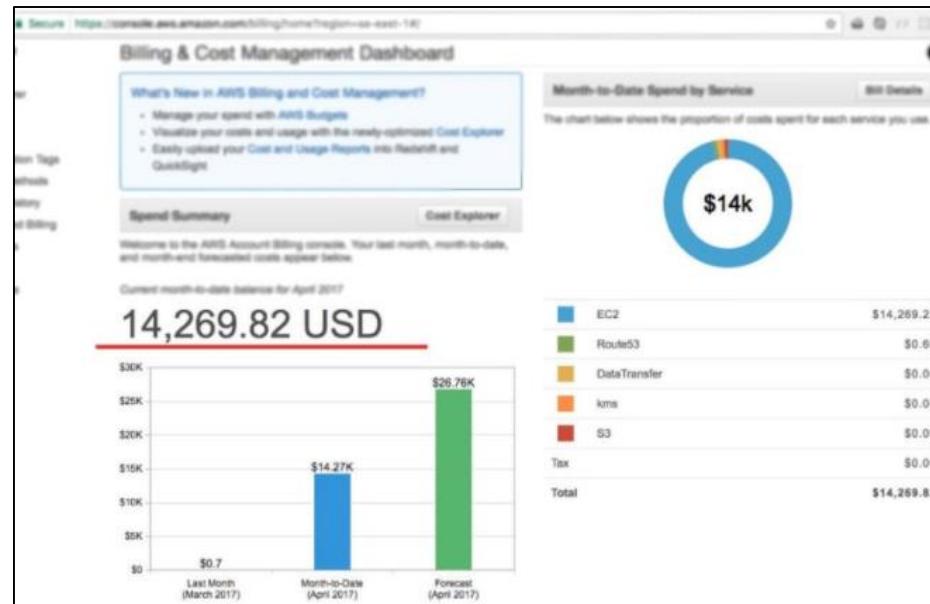
.ppk  
PuTTY와 함께 사용

취소 **키 페어 생성**

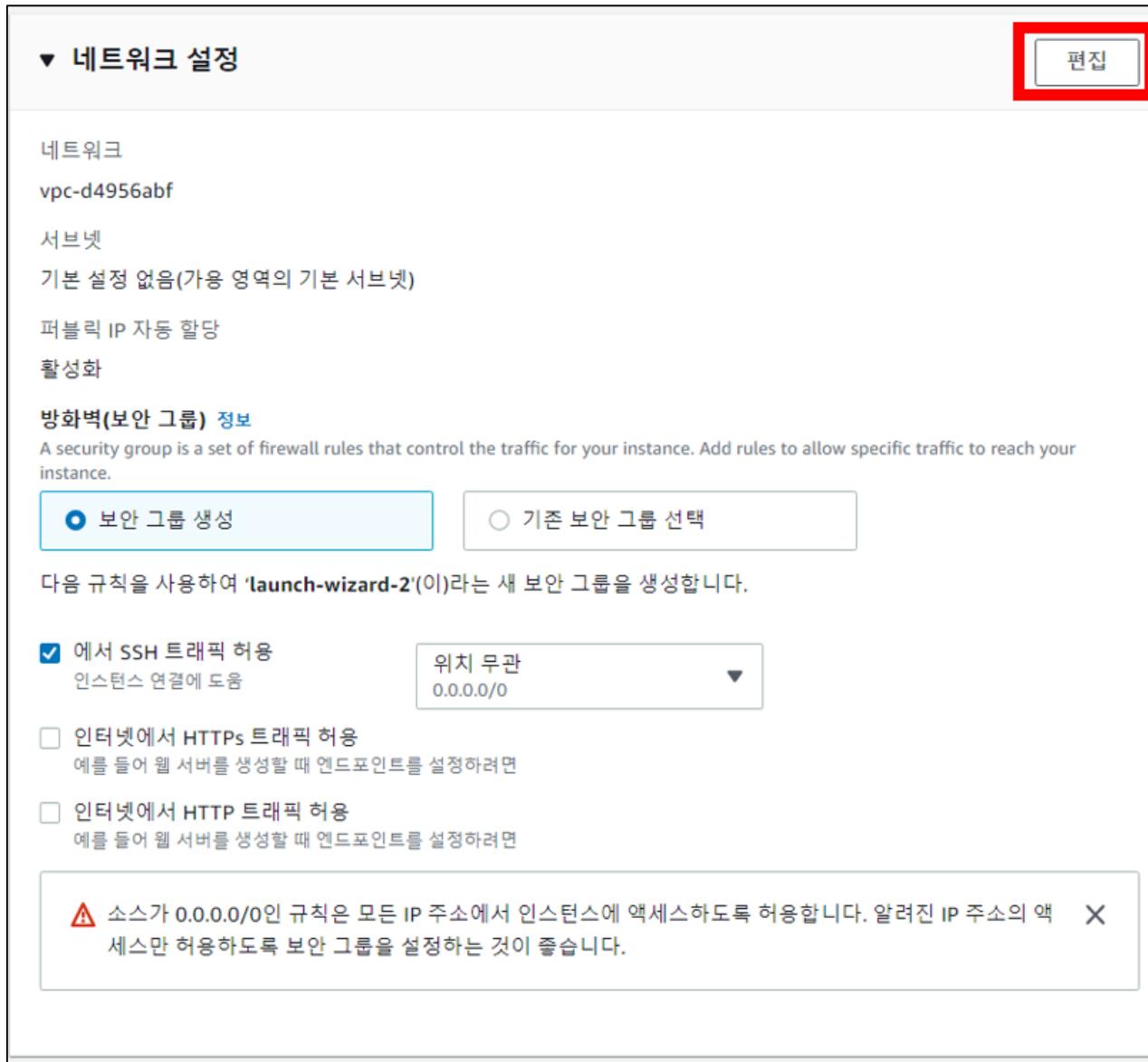
잃어버리지 않게 잘 저장

# 주의 : KEY 파일을 GitHub에 업로드하게되면 생기는 일

- ‘KEY’ 파일을 인터넷 업로드 하는 경우



<https://dev.to/juanmanuelramallo/i-was-billed-for-14k-usd-on-amazon-web-services-17fn>



보안 그룹 이름 설정을 위해  
편집 버튼 클릭

▼ 네트워크 설정

VPC - 필수 정보  
vpc-d4956abf (기본값)  
172.31.0.0/16

서브넷 정보  
기본 설정 없음 새 서브넷 생성

퍼블릭 IP 자동 할당 정보  
활성화

방화벽(보안 그룹) 정보  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

보안 그룹 생성     기존 보안 그룹 선택

보안 그룹 이름 - 필수  
ssafy-8th-node-security

이 보안 그룹은 모든 네트워크 인터페이스에 추가됩니다. 보안 그룹을 만든 후에는 이름을 편집할 수 없습니다. 최대 길이는 255자입니다. 유효한 문자는 a~z, A~Z, 0~9, 공백 및 \_-:/()#,@[]+=;&();!\$\*입니다.

설명 - 필수 정보  
ssafy-8th-node-security created 2022-06-13

인바운드 보안 그룹 규칙  
▼ 보안 그룹 규칙 1 (TCP, 22, 0.0.0.0/0)

제거

식별 가능한 이름 및 설명 지정

▼ 요약

인스턴스 개수 [정보](#)

1

소프트웨어 이미지(AMI)

Canonical, Ubuntu, 18.04 LTS, ... [더 보기](#)  
ami-0c6fdf2ba8d9b68bb

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

새 보안 그룹

스토리지(볼륨)

1개의 볼륨 – 8GiB

취소

인스턴스 시작



오른쪽 인스턴스 시작 버튼 클릭

# 실행중인 인스턴스 확인

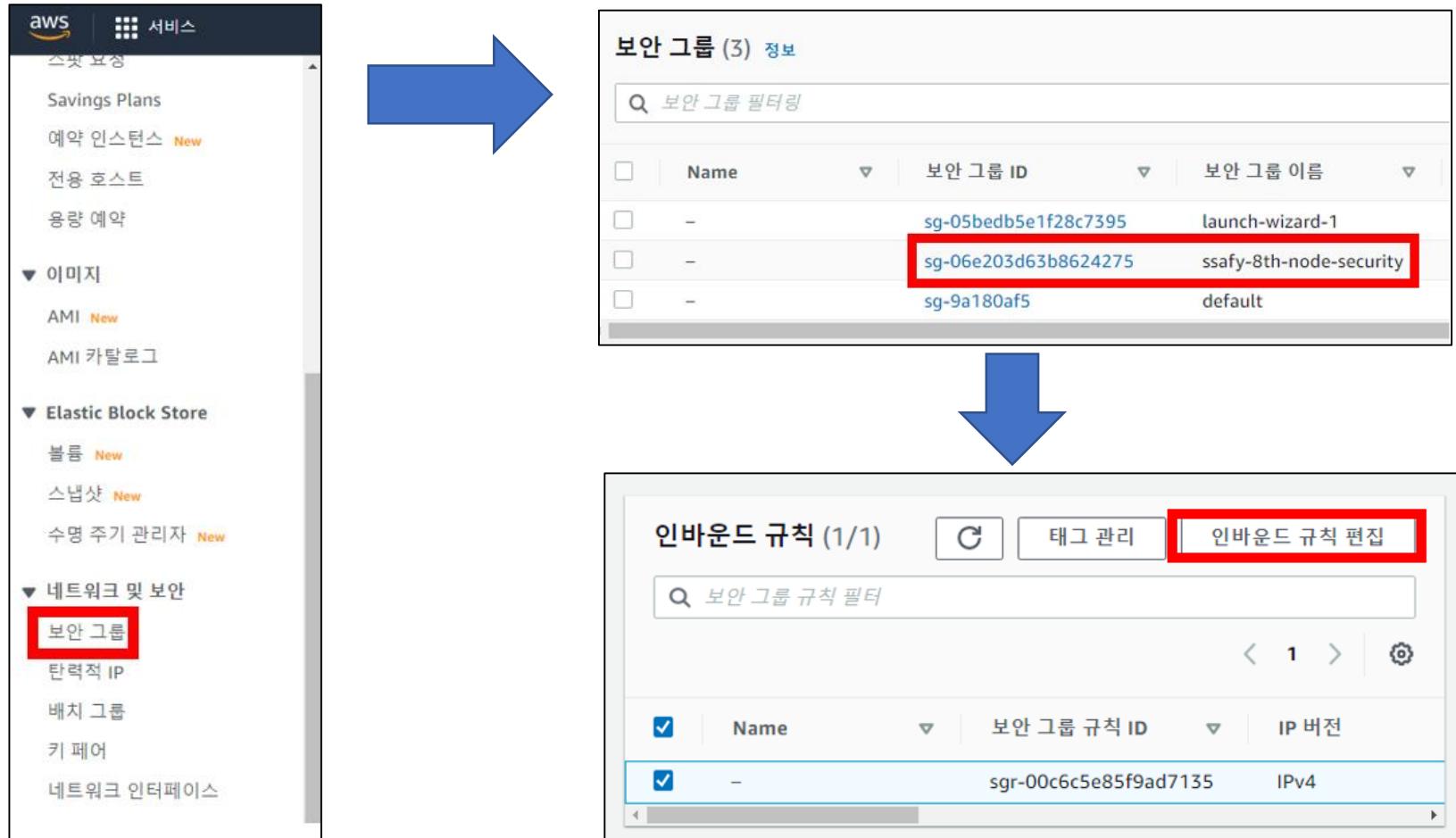
Confidential

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like 'EC2 대시보드', 'EC2 글로벌 보기', '이벤트', '태그', '제한', and '인스턴스' (which is expanded). Under '인스턴스', the '인스턴스 New' button is highlighted with a red box. The main area displays a table of instances:

	Name	인스턴스 ID	인스턴스 상태	
<input type="checkbox"/>	jony-server	[REDACTED]	실행 중	
<input type="checkbox"/>	ssafy-8th-node-aws	[REDACTED]	실행 중	

Below the table, a message reads: "방금 만든 인스턴스가 실행중인것이 확인됨"

## 원활한 실습을 위해, 보안그룹을 미리 지정하자



## 인바운드 규칙 편집 정보

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

## 인바운드 규칙 정보

## 보안 그룹 규칙 ID

sgr-0833b982a6141e005

## 유형 정보

MySQL/Aurora

## 프로토콜 정보

TCP

## 포트 범위 정보

3306

## 소스 정보

사용자 지정

## 설명 - 선택 사항 정보

0.0.0.0 X

sgr-0ecd74faf6dc40e0b

사용자 지정 TCP

TCP

8081

Anywhere-...

sgr-0e218e9d7ded0a4f2

사용자 지정 TCP

TCP

8000

Anywhere-...

sgr-0c17d86f2634929fa

HTTP

TCP

80

Anywhere-...

sgr-0e4eb33a27dc498a7

사용자 지정 TCP

TCP

8080

Anywhere-...

sgr-00c6c5e85f9ad7135

SSH

TCP

22

Anywhere-...

-

모든 ICMP - IPv4

ICMP

전체

Anywhere-...

## 보안그룹 설정

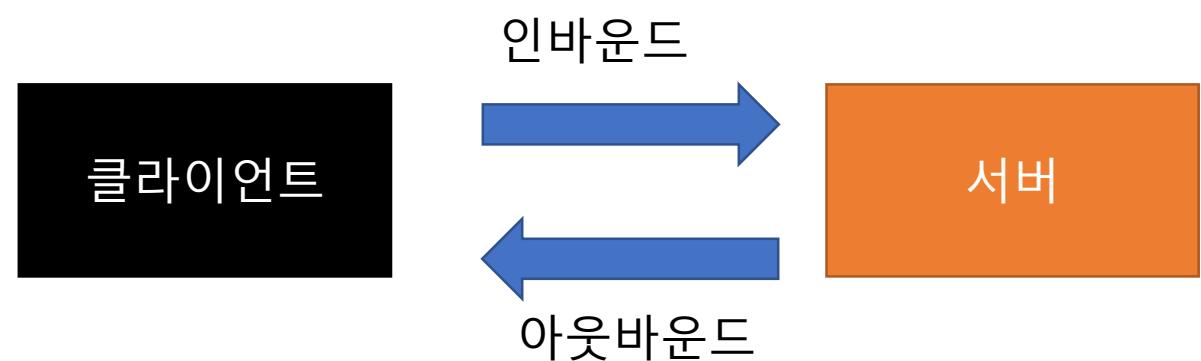
- SSH (22번 포트) : 원격 연결
- MySQL/Aurora (3306 포트) : MySQL 원격접속
- HTTP (80번 포트) : HTTP 통신
- 사용자 지정 TCP (8000, 8080, 8081) : HTTP 예비 포트
- 모든 ICMP - IPv4: PING 테스트
- 모두 **Anywhere-IPv4** 로 지정

## • 인바운드

- 클라이언트가 자신의 서버 데이터에 들어올 수 있는 규칙
- 서버 내부로 들어오는 것
- 서버에 업로드할 때

## • 아웃바운드

- 서버에서 나갈 수 있는 데이터에 대한 규칙
- 서버에서 다운로드 할 때



## 탄력적 IP

- 인스턴스를 중지하게 되면 다시 시작할 때 IP가 변경
- IP 가 바뀌는 불편함에 대한 개선점으로 탄력적 IP를 활용
- 탄력적 (Elastic) IP
  - AWS 리전 당 총 5개 무료로 사용 가능한 고정 IP
  - 추가요금 없으나, 만들어놓고 사용하지 않으면 요금 발생

## SSH 연결

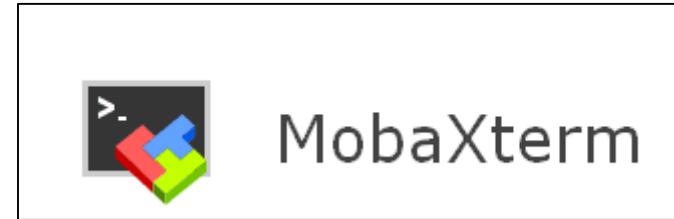
## 원격 접속 터미널

- Putty : 가장 널리 사용됨
- MobaXterm : 편리한 오픈소스 터미널



## 클라이언트 프로그램

- 터미널 프로그램 = mobaXterm
- 키보드 입력을 서버에 전달



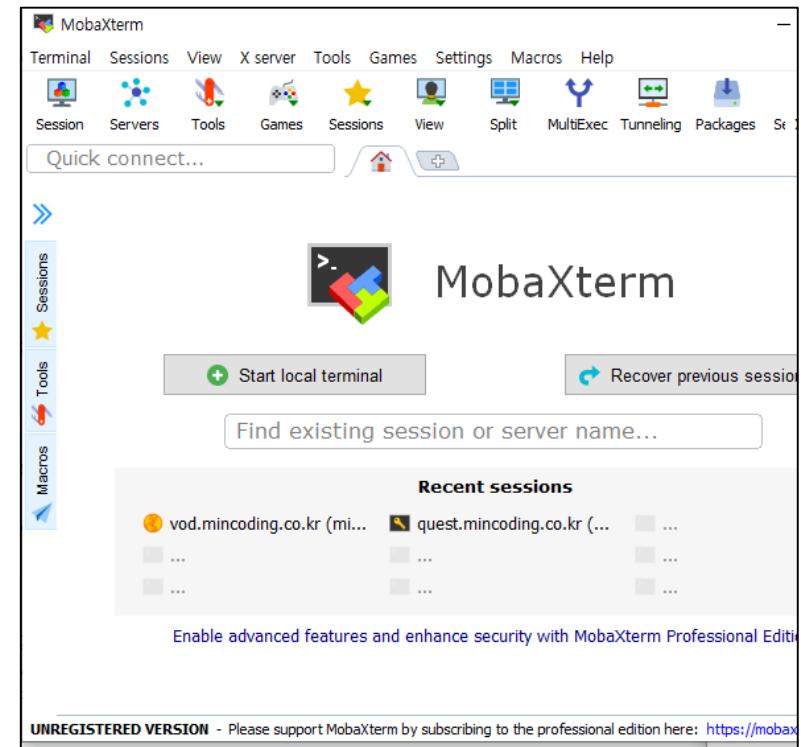
## 서버 프로그램

- 클라이언트의 키보드 입력 → 원격지 PC에 입력해주는 프로그램
- 결과 화면을 클라이언트에 전달한다.

Ubuntu Server 엔 기본적으로 openssh-server 가 설치되어 있다

## MobaXterm

- <https://mobaxterm.mobatek.net/>
- 다운 받아서 설치하기



The screenshot shows the MobaXterm website homepage. At the top, there is a navigation bar with links: Home, Demo, Features, Download (which is highlighted with a red box), Plugins, Help, Contact, and social media icons for Facebook, Twitter, and Google+. To the right of the navigation bar are links for Customer area and Buy.

The main content is divided into two sections:

- Home Edition** (Free):
  - Full X server and SSH support
  - Remote desktop (RDP, VNC, Xdmcp)
  - Remote terminal (SSH, telnet, rlogin, Mosh)
  - X11-Forwarding
  - Automatic SFTP browser
  - Master password protection
  - Plugins support
  - Portable and installer versions
  - Full documentation
  - Max. 12 sessions
  - Max. 2 SSH tunnels
  - Max. 4 macros
  - Max. 360 seconds for Tftp, Nfs and Cron
- Professional Edition**:
  - \$69 / 49€ per user\***
  - \* Excluding tax. Volume discounts [available](#)
  - Every feature from Home Edition +
    - Customize your startup message and logo
    - Modify your profile script
    - Remove unwanted games, screensaver or tools
    - Unlimited number of sessions
    - Unlimited number of tunnels and macros
    - Unlimited run time for network daemons
    - Enhanced security settings
    - 12-months updates included
    - Deployment inside company
    - Lifetime right to use

At the bottom of each section is a large red button with white text: "Download now" for the Home Edition and "Subscribe online / Get a quote" for the Professional Edition. There are also payment method icons (PayPal, Visa, Mastercard) next to the Professional Edition button.

 MobaXterm

Home Demo Features **Download** Plugins Help Contact [!\[\]\(f7a218c7df7c2e0e4c82de1ea36eb835\_img.jpg\)](#) [!\[\]\(d175cd91094f09b10874cb8c902a8480\_img.jpg\)](#) [!\[\]\(907e36c2ba8ea315fc407f0724fbfc3d\_img.jpg\)](#) [!\[\]\(a9a2979be6c749d5b384f5a3afee0322\_img.jpg\)](#)

[Customer area](#) [Buy](#)

## MobaXterm Home Edition

Download MobaXterm Home Edition (current version):

 [MobaXterm Home Edition v21.5  
\(Portable edition\)](#)

 [MobaXterm Home Edition v21.5  
\(Installer edition\)](#)

Download previous stable version: [MobaXterm Portable v21.4](#) [MobaXterm Installer v21.4](#)

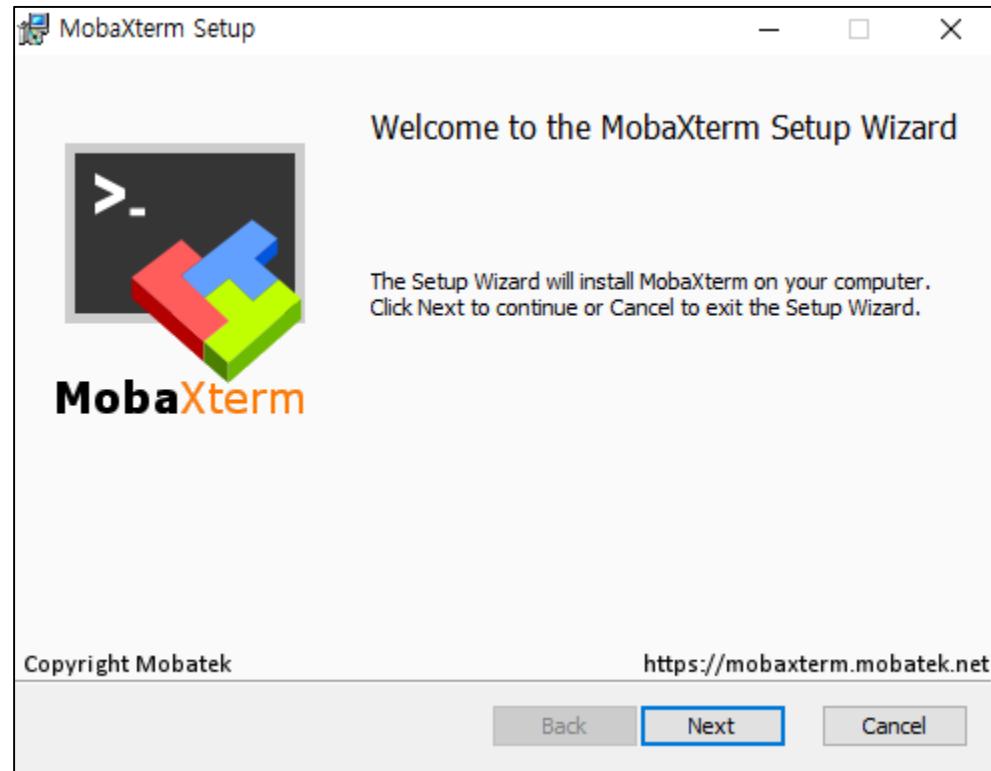
You can also get early access to the latest features and improvements by downloading MobaXterm Preview version:

[MobaXterm Preview Version](#)

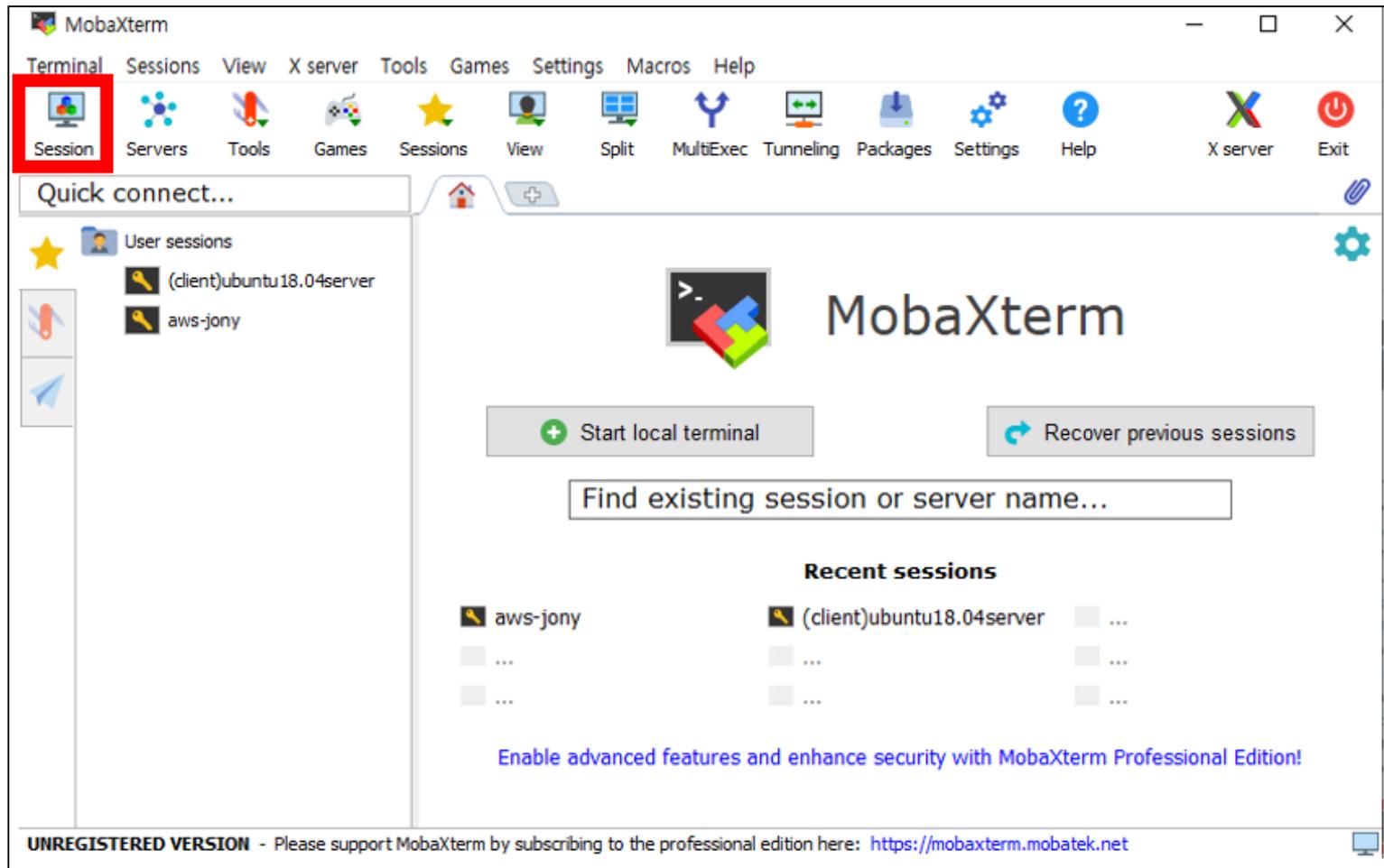
By downloading MobaXterm software, you accept [MobaXterm terms and conditions](#)

You can download the third party plugins and components sources [here](#)

## 설치 진행



# 내 Windows PC에서 AWS 인스턴스에 연결해보자



인스턴스 (1/2) 정보

G 연결 인스턴스 상태 ▾

인스턴스 시작 ▾

검색

퍼블릭 IPv4 ... 탄력적 IP

42.26.24 42.26.24

연결하고 싶은 AWS 인스턴스 퍼블릭 IP

Session settings

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser M

유저네임은 ubuntu

Basic SSH settings

Remote host: 42.26.24 Specify username: ubuntu Port: 22

Advanced SSH settings Terminal settings Network settings Bookmark settings

X11-Forwarding Compression Remote environment: Interactive shell

Execute command: Do not exit after command ends

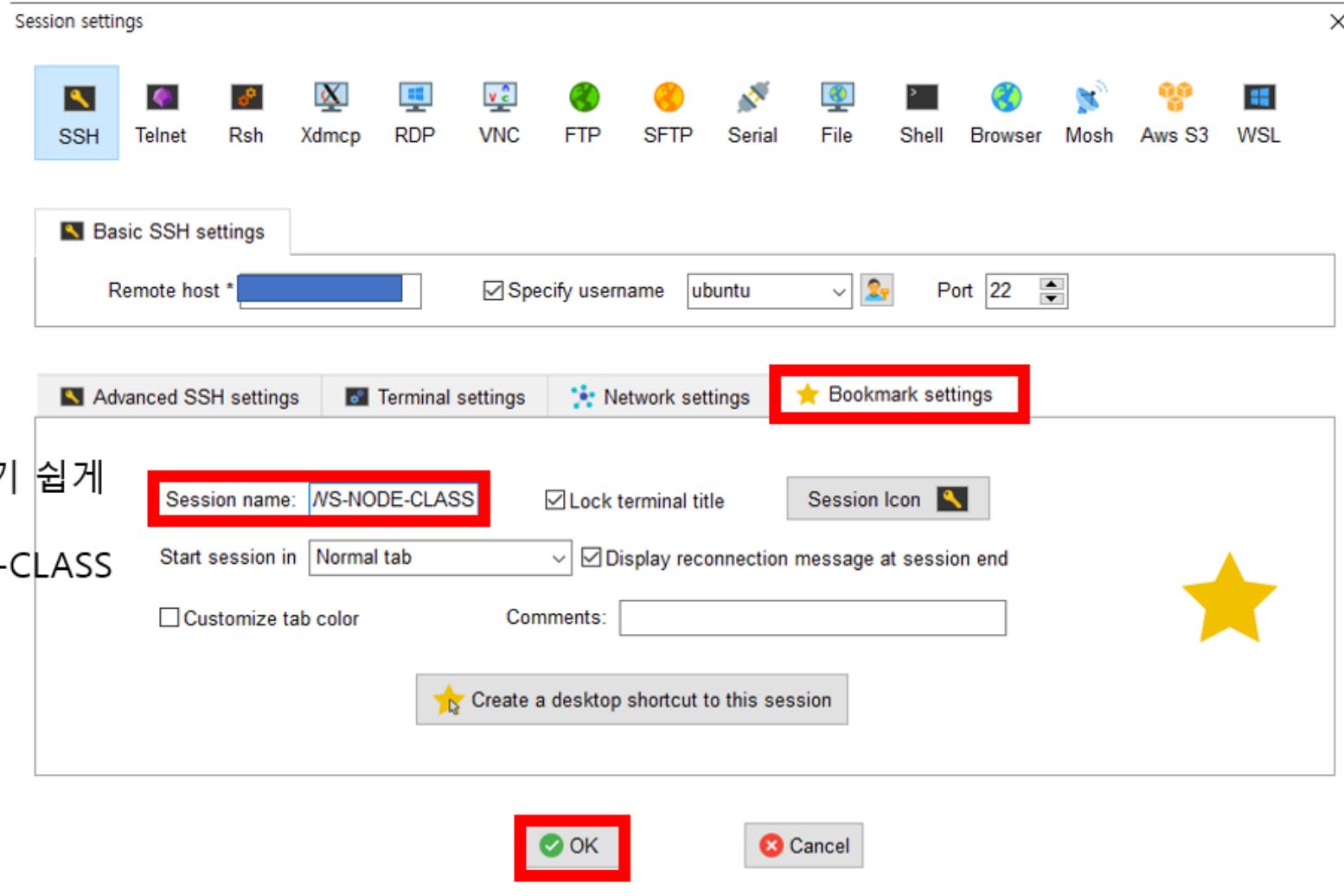
SSH-browser type: SFTP protocol Follow SSH path (experimental)

Use private key C:\Users\mincoding\Documents\ssh-key.pem Adapt locales on remote server

Execute macro at session start: <none>

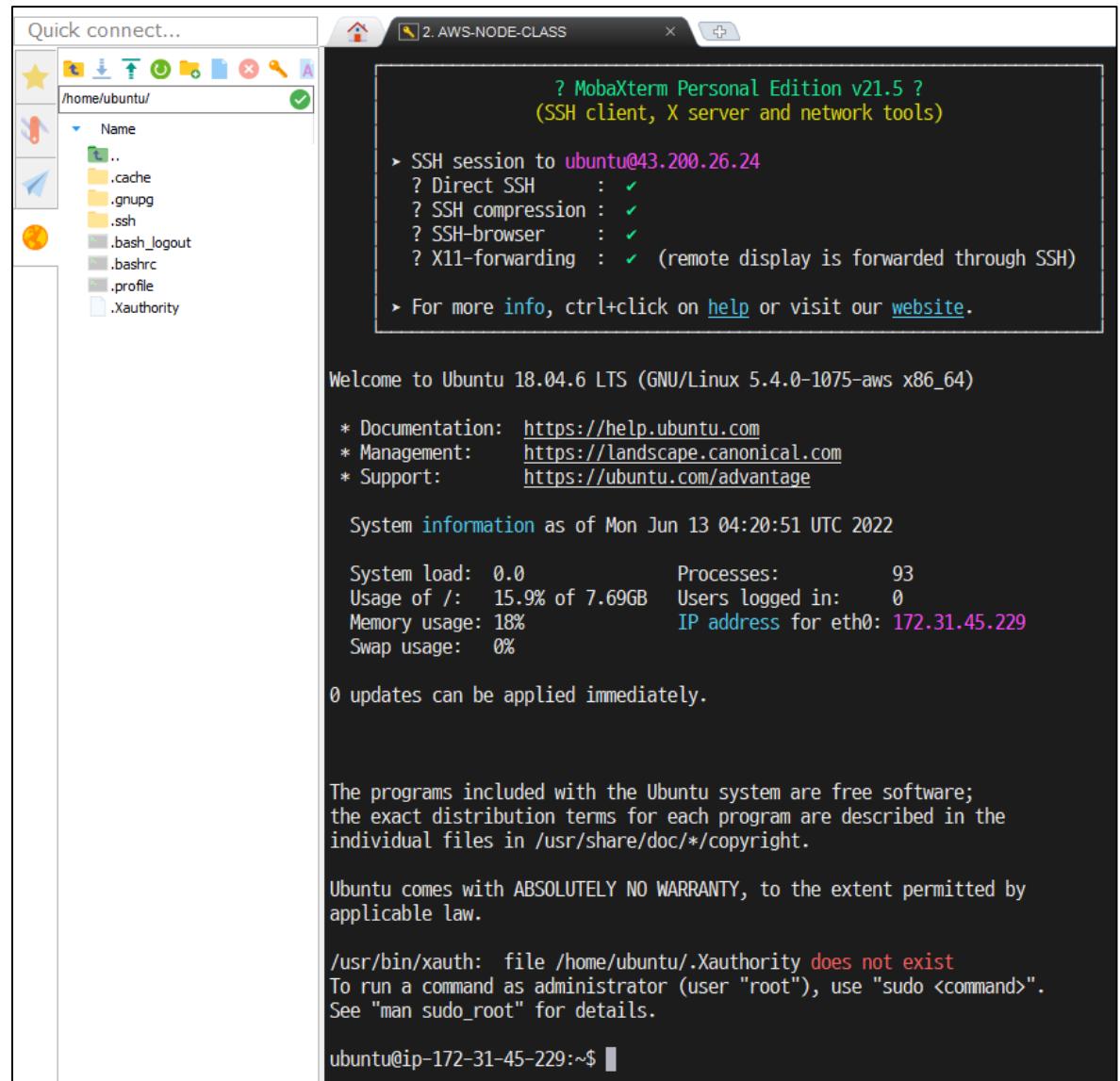
OK Cancel

아까 저장한 Key



## 접속 성공

- Windows에서, AWS Instance에 원격접속 성공

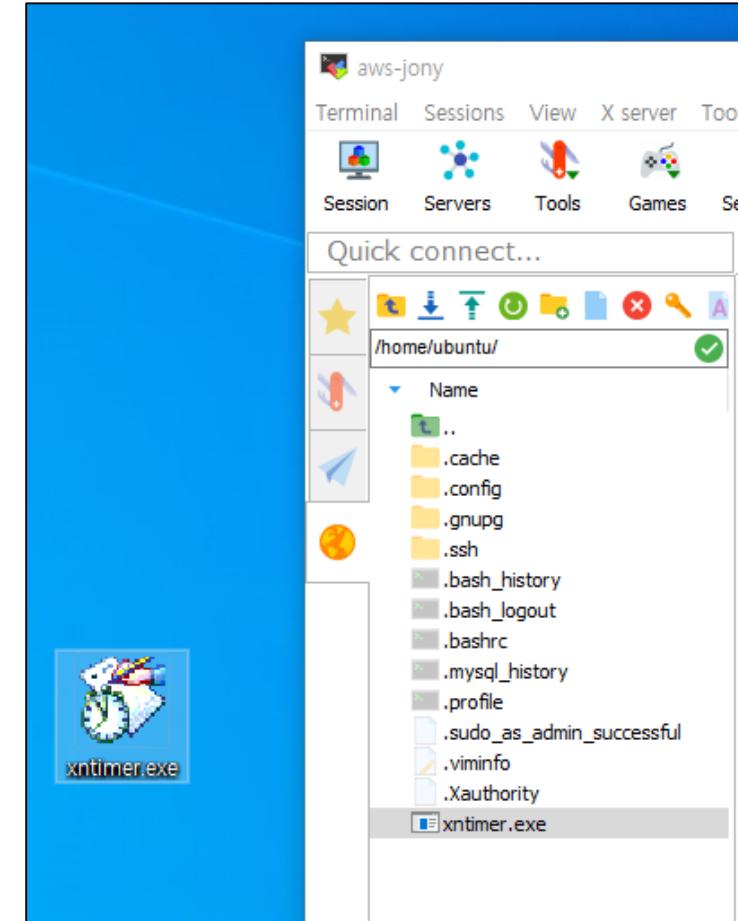


## 한국 시간대인 KST로 변경

- \$ sudo timedatectl set-timezone Asia/Seoul
- \$ date

## 파일 업로드

- 서버에 파일을 업로드하고, 다운로드해보자.
- MobaXterm 왼쪽 네비게이션에
- 마우스 드래그로 업로드/다운로드 가능



## 4장. 기본적인 리눅스 명령어

# 챕터의 포인트

- 기본 명령어
- 파일관리 명령어

# 기본 명령어

## ls -al

- a 옵션 : all, 숨김파일까지 출력
- l 옵션 : list, 리스트 형태로 상세 보기

## 주의사항

- !!이라고 입력해도 되지만, ls -al을 쓰자.
- !!은 임베디드 리눅스에서 지원안하는 경우 많음

```
ubuntu@ip-172-31-45-229:~$ ls -al
total 44
drwxr-xr-x 6 ubuntu ubuntu 4096 Jun 13 04:39 .
drwxr-xr-x 3 root   root   4096 Jun 13 02:11 ..
-rw----- 1 ubuntu ubuntu   62 Jun 13 04:20 .Xauthority
-rw-r--r-- 1 ubuntu ubuntu  220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr  4 2018 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jun 13 04:20 .cache
drwx----- 3 ubuntu ubuntu 4096 Jun 13 04:20 .gnupg
-rw-r--r-- 1 ubuntu ubuntu  807 Apr  4 2018 .profile
drwx----- 2 ubuntu ubuntu 4096 Jun 13 02:11 .ssh
drwxrwxr-x 5 ubuntu ubuntu 4096 Jun 13 04:39 .vscode-server
-rw-rw-r-- 1 ubuntu ubuntu 183 Jun 13 04:39 .wget-hsts
ubuntu@ip-172-31-45-229:~$ █
```

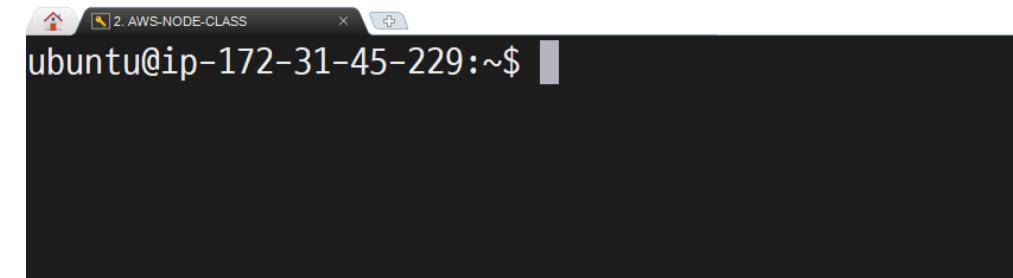
## pwd

- 현재 디렉토리 확인
  - 리눅스 CLI 환경에서는 폴더가 아닌 “디렉터리”라는 명칭을 사용

```
ubuntu@ip-172-31-45-229:~$ pwd  
/home/ubuntu
```

## clear

- 화면 지우기



## cd

- **cd ..**
  - 이전 디렉토리로 이동하기
- **cd “디렉토리 이름”**
  - 해당 디렉토리로 이동하기
- **cd ~**
  - 홈 디렉토리로 이동
- **cd -**
  - 이전 디렉토리로 되돌아가기



```
ubuntu@ip-172-31-45-229:~$ ls
ubuntu@ip-172-31-45-229:~$ cd ..
ubuntu@ip-172-31-45-229:/home$ ls
ubuntu
ubuntu@ip-172-31-45-229:/home$ cd ..
ubuntu@ip-172-31-45-229:$ ls
bin etc initrd.img.old lost+found opt run srv usr vmlinuz.old
boot home lib media proc sbin sys var
dev initrd.img lib64 mnt root snap tmp vmlinuz
ubuntu@ip-172-31-45-229:$ █
```

## 복사 붙여넣기 단축키

- 터미널 창에서는 Ctrl + C / Ctrl + V 가 아니다.
  - Ctrl + Insert (복사)
  - Shift + Insert (붙여넣기)
  - 임베디드 리눅스 환경, 모두 동일
  - 윈도우에서도 사용 가능

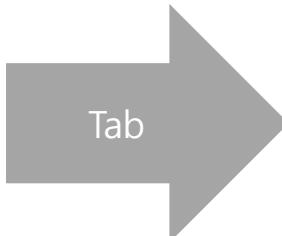
## 리눅스 명령어와 친해지기(제한시간 1분)

1. home directory 이동 후, 전체 디렉토리 확인
2. 현재 디렉토리가 어디인지 확인한다.
3. root directory 이동 후, 전체 디렉토리 확인
4. 화면을 모두 지운다.
5. /usr/bin 으로 이동한다.
6. 현재 디렉토리에 있는 파일 중 yes 파일의 생성 날짜 정보를 확인한다. (ls 사용)
7. 화면을 모두 지운다.
8. 이전 디렉토리로 이동한다.
9. 현재 디렉토리가 어디인지 출력한다.
10. 터미널을 종료한다. (exit 명령어)

Tab을 누르면 타이핑을 다 안해도 자동 완성이 된다.

- 후보가 하나일때는 Tab 한번 누르면 자동완성 됨
- 후보가 여러개일때는 Tab Tab 눌러야 후보 목록이 나옴

```
ubuntu@ip-172-31-45-229:/usr/bin$ cd /
ubuntu@ip-172-31-45-229:$ ls
bin  initrd.img   media  run   tmp
boot initrd.img.old  mnt   sbin  usr
dev   lib          opt    snap  var
etc   lib64        proc   srv   vmlinuz
home lost+found   root   sys   vmlinuz.old
ubuntu@ip-172-31-45-229:$ ls l
lib/      lib64/    lost+found/
ubuntu@ip-172-31-45-229:$ ls lo█
```



```
ubuntu@ip-172-31-45-229:/usr/bin$ cd /
ubuntu@ip-172-31-45-229:$ ls
bin  initrd.img   media  run   tmp
boot initrd.img.old  mnt   sbin  usr
dev   lib          opt    snap  var
etc   lib64        proc   srv   vmlinuz
home lost+found   root   sys   vmlinuz.old
ubuntu@ip-172-31-45-229:$ ls l
lib/      lib64/    lost+found/
ubuntu@ip-172-31-45-229:$ ls lost+found/█
```

```
ubuntu@ip-172-31-45-229:$ cd /
ubuntu@ip-172-31-45-229:$ ls
bin  initrd.img   media  run   tmp
boot initrd.img.old  mnt   sbin  usr
dev   lib          opt    snap  var
etc   lib64        proc   srv   vmlinuz
home lost+found   root   sys   vmlinuz.old
ubuntu@ip-172-31-45-229:$ ls l█
```



```
ubuntu@ip-172-31-45-229:$ cd /
ubuntu@ip-172-31-45-229:$ ls
bin  initrd.img   media  run   tmp
boot initrd.img.old  mnt   sbin  usr
dev   lib          opt    snap  var
etc   lib64        proc   srv   vmlinuz
home lost+found   root   sys   vmlinuz.old
ubuntu@ip-172-31-45-229:$ ls l
lib/      lib64/    lost+found/
ubuntu@ip-172-31-45-229:$ ls l█
```

## 파일관리 명령어

## 파일을 다루는 기본 명령어

- 파일 생성 / 삭제
- 디렉토리 생성 / 삭제
- 파일 이동 / 이름 변경
- 파일 복사

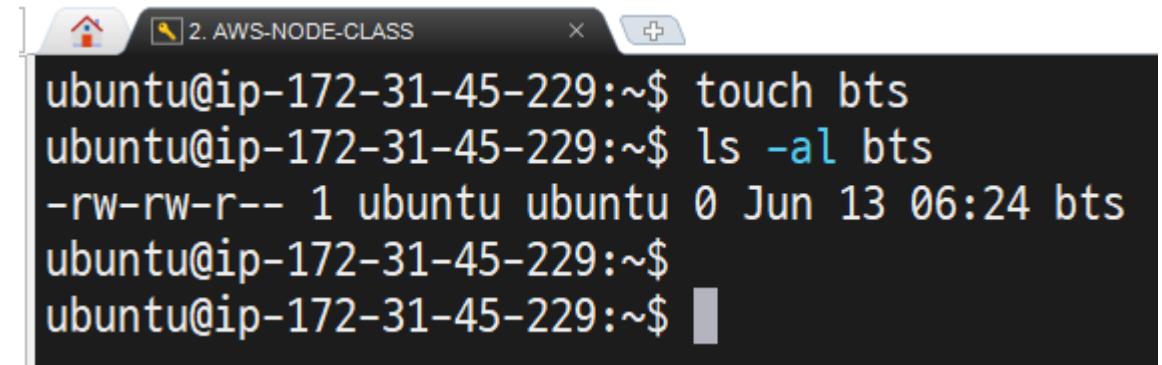
## touch [파일명]

- 새로운 빈 파일을 생성한다.
- 이미 있는 파일이라면, 변경된 시간을 현재 시간으로 업데이트 한다.
- 예시 : touch ./bts

./ 는 현재 디렉토리를 의미한다.  
생략해도 되지만,  
가독성 측면에서 파일명임을 나타낸다.

## rm [파일명]

- 파일을 삭제한다.
- 예시 : rm ./bts



```
ubuntu@ip-172-31-45-229:~$ touch bts
ubuntu@ip-172-31-45-229:~$ ls -al bts
-rw-rw-r-- 1 ubuntu ubuntu 0 Jun 13 06:24 bts
ubuntu@ip-172-31-45-229:~$
```

홈 디렉토리에서 실습해보기

a, b, c 파일을 생성하자

- TIP : 리눅스는 윈도우와 달리 확장자 개념이 없다.

파일을 각각 touch 해주자.

- 수정시간을 현재 시간으로 업데이트 한다.

파일을 삭제하자.

## mkdir “디렉토리명”

- 디렉토리가 생성된다.
  - 예시 : mkdir a b c : a, b, c 디렉토리가 생성된다.
- -p 옵션 : 디렉토리 하위메뉴 까지 모두 한꺼번에 생성
  - 예시 : mkdir -p ./aaa/bbb/ccc/ddd

```
ubuntu@ip-172-31-45-229:~$ rmdir ./aaa
rmdir: failed to remove './aaa': Directory not empty
ubuntu@ip-172-31-45-229:~$ █
```

## rmdir “디렉토리명”

- 디렉토리를 삭제한다.
- 디렉토리 안에 파일이 있으면 삭제가 되지 않는다.
- rmdir 대신, rm -r “경로” 를 사용하면 디렉토리 내부 파일까지 모두 삭제 가능
  - rmdir 대신 사용하기 더 간편한 rm -r 을 사용하자.

```
ubuntu@ip-172-31-45-229:~$ rm -r ./aaa
ubuntu@ip-172-31-45-229:~$ █
```

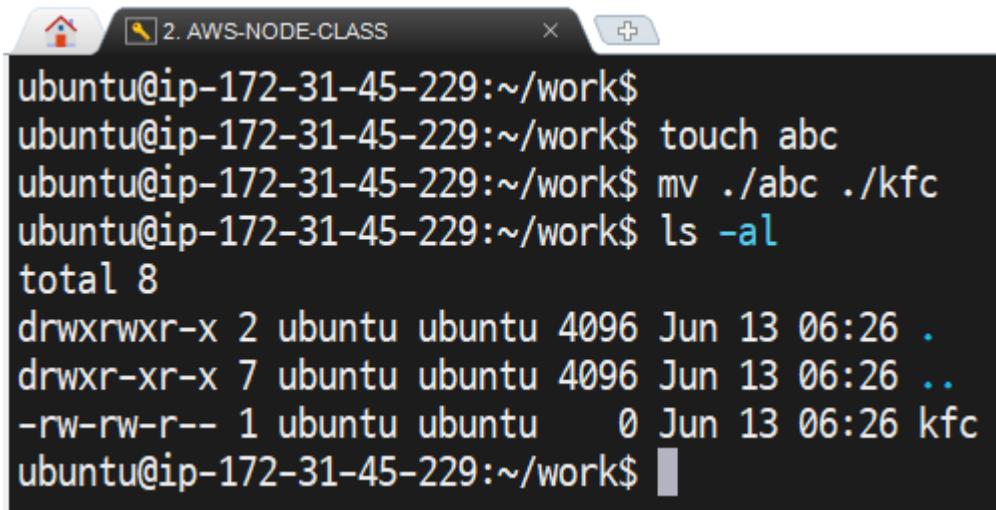


## 생성 삭제를 연습 (제한시간 1분)

1. 홈 디렉토리로 이동 후 현재 디렉토리 확인하기
2. 현재 디렉토리에서 ./a/b/c 디렉토리 생성
3. ./a/b/c 디렉토리로 이동하기
4. 빈 파일 두 개 “kfc”, “bts” 생성
5. 생성된 파일 확인하기
6. 상위 디렉토리로 이동하기
7. 빈 파일 “mc” 하나 생성
8. 상위 디렉토리로 이동하기
9. 현재 디렉토리 내부에 있는 파일 모두 삭제 (`rm -r ./*` : 현재 디렉토리에서 전체 삭제)
10. 홈 디렉토리로 이동
11. a 디렉토리 삭제
12. 화면 지우기

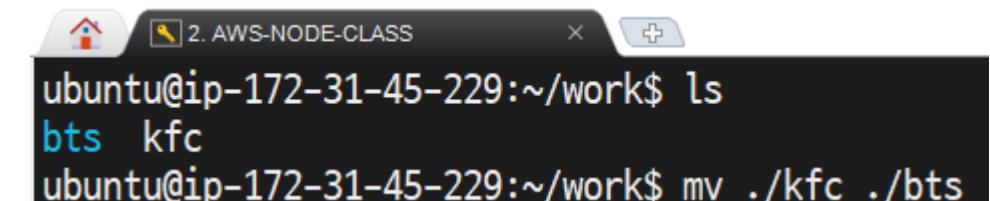
## mv 명령어

- 이동 : mv [파일] [옮길 곳]
- 이름변경 : mv [파일] [파일이름]



```
ubuntu@ip-172-31-45-229:~/work$ touch abc
ubuntu@ip-172-31-45-229:~/work$ mv ./abc ./kfc
ubuntu@ip-172-31-45-229:~/work$ ls -al
total 8
drwxrwxr-x 2 ubuntu ubuntu 4096 Jun 13 06:26 .
drwxr-xr-x 7 ubuntu ubuntu 4096 Jun 13 06:26 ..
-rw-rw-r-- 1 ubuntu ubuntu     0 Jun 13 06:26 kfc
ubuntu@ip-172-31-45-229:~/work$
```

mv로 이름 바꾸는 방법



```
ubuntu@ip-172-31-45-229:~/work$ ls
bts  kfc
ubuntu@ip-172-31-45-229:~/work$ mv ./kfc ./bts
```

mv로 옮기는 법

## 이름 변경 및 파일이동 (제한시간 1분)

1. work directory 하나 생성 후 이동
2. mc, bbq, kfc 파일 생성
3. shop 디렉토리 생성
4. mc와 bbq를 shop 디렉토리로 옮김
5. kfc를 bts로 파일이름 변경
6. shop 디렉토리로 이동
7. bbq를 상위 디렉토리로 옮기기
8. mc를 상위 디렉토리에 옮기면서 파일이름을 good 으로 수정
9. ls -al 로 work 디렉토리 내용 확인
10. work directory 삭제

```
ubuntu@ip-172-31-45-229:~/work$ ls  
bbq  bts  good  shop
```

work directory  
최종 결과

## cp 명령어

- cp [파일] [경로] : 한 파일을 경로로 복사하기
- -r 옵션 : 디렉토리 복사하기

디렉토리 복사는 rm과 마찬가지로 -r 옵션을 붙이자.

## 미션 수행하기 (제한시간 1분)

1. abc1 파일 생성
2. abc1 파일을 사용하여 abc2 파일 복사
3. chocho/haha 디렉토리 생성
4. abc1, abc2 를 chocho 로 이동시키기
5. chocho 를 tacktack 으로 디렉토리 복사
6. chocho 를 tacktack 안으로 이동시키기
7. tacktack 디렉토리 제거하기

## 5장. 나의 첫 웹서버

# 챕터의 포인트

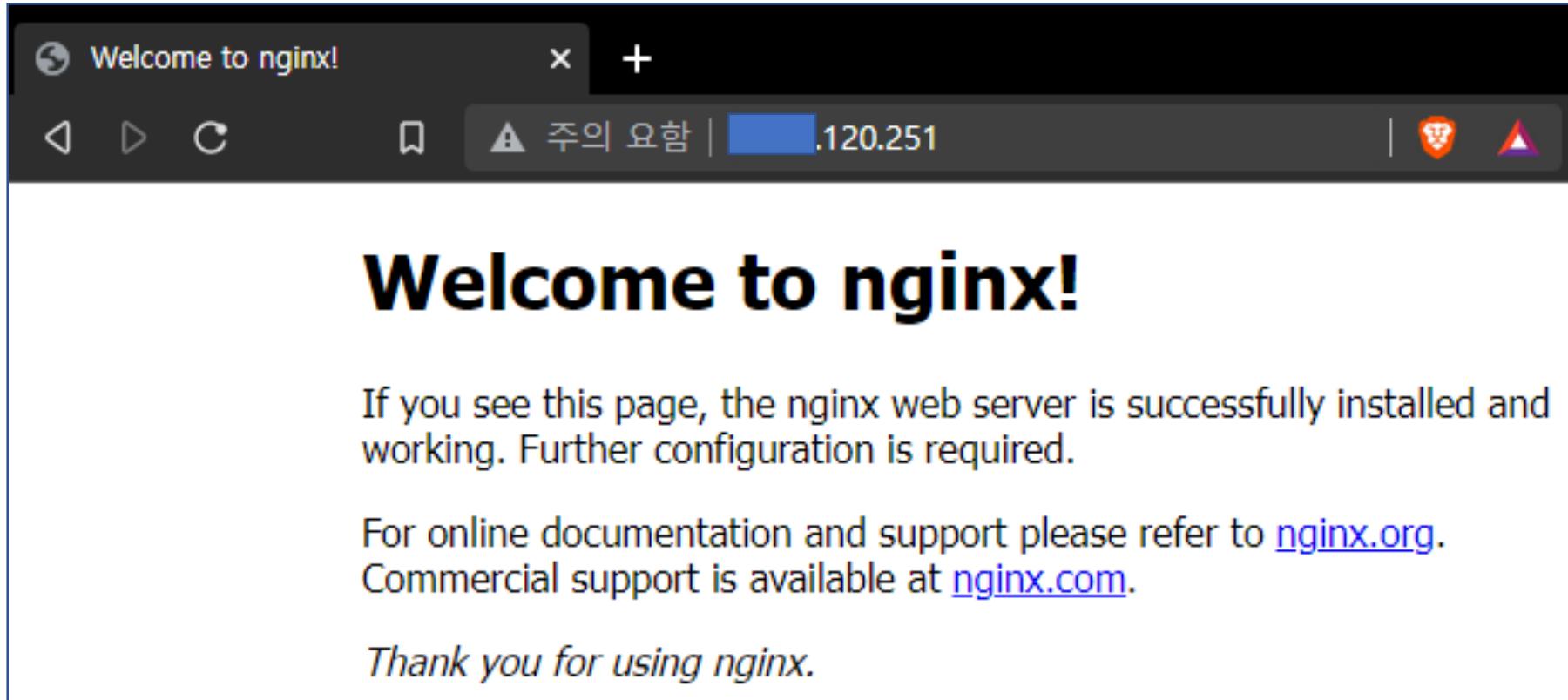
- 간단한 nginx 서버
- 도메인 연결
- 인스턴스 중지와 종료

## 간단한 nginx 서버

## NGINX 서버 구축하기

- 다른 사람의 웹사이트로 도메인 접속 => 웹서버와 친해지기
- MobaXterm 을 사용해 nginx 설치
  - \$ sudo apt update
    - sudo는 관리자 권한을 의미
    - apt(advvanced packaging tool) 로써 리눅스의 패키지 관리 도구
  - \$ sudo apt install nginx
- \$ nginx -v
  - 설치 후 nginx 버전 확인

```
ubuntu@ip-172-31-40-222:~/back-end$ nginx -v
nginx version: nginx/1.14.0 (Ubuntu)
```

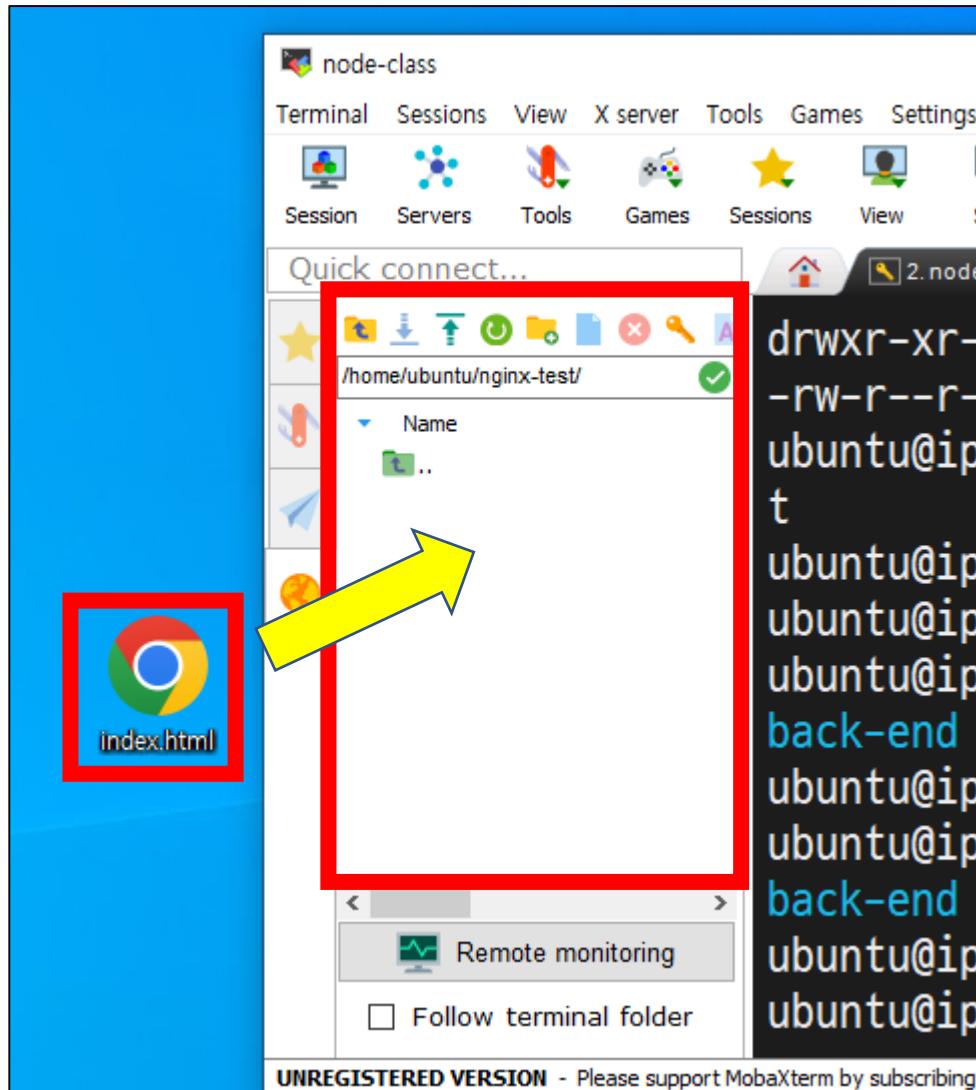


AWS 인스턴스 IP 입력 시, nginx 소개 페이지가 뜬다면 성공

## 디렉토리 생성

- 홈디렉터리에, 서비스를 위한 디렉터리 하나 생성
- \$ mkdir ~/nginx-test
- \$ cd ~/nginx-test
- 간단한 index.html 을 windows 에서 작성
- pwd로 경로를 파악한 후 복사(Ctrl + Insert)

```
1  <!DOCTYPE html>
2  <html lang="en">
3  | <head>
4  | | <meta charset="UTF-8" />
5  | | <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6  | | <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7  | | <title>nginx 테스트</title>
8  | </head>
9  | <body>
10 | | <h1>nginx 테스트</h1>
11 | </body>
12 </html>
```



드래그 앤 드롭

즉, 윈도우의 파일을 AWS 인스턴스로 이동 !

# nano

- 리눅스 용 명령 텍스트 에디터
- 일반 텍스트 에디터와 유사하게 사용 가능

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.php index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
    }
}
```

**^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location    M-U Undo  
^X Exit    ^R Read File    ^Y Replace    ^U Paste    ^J Justify    ^Y Go To Line    M-E Redo**

## vi/vim

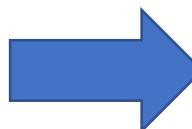
- 임베디드 개발자가 가장 많이 사용하는 텍스트 에디터
- 입문시 난이도가 있는편 (단축키에 익숙해져야 한다.)
- 임베디드 수업 때 본격적으로 사용할 예정



## 만든 프로젝트 디렉터리를 nginx에 연결

- nginx 설치시 /etc/nginx 라는 디렉터리가 자동으로 생성된다.
- sudo nano /etc/nginx/sites-available/default
  - 기존 /var/www/html 으로 되어있는경로를 생성한 디렉토리로 변경한다.
    - /home/ubuntu/nginx-test
  - 복사한 디렉토리 붙여넣기 (shift + insert)
  - 붙여 넣은 후 Ctrl + O로 작성 후 Ctrl + X로 나가기

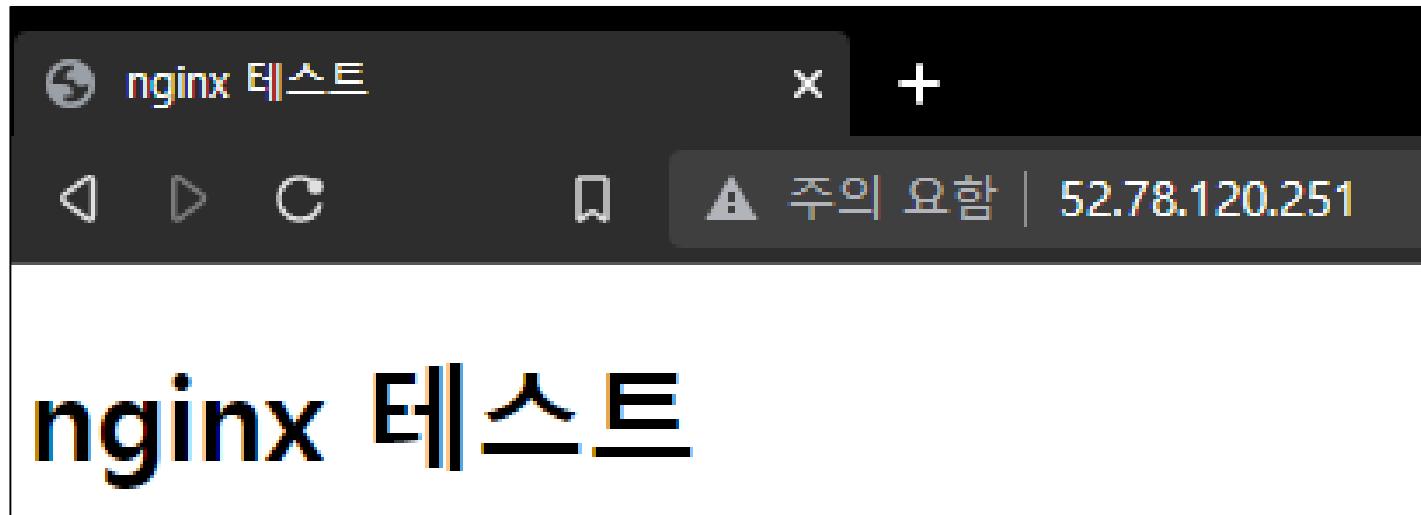
```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    # listen [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
    #  
    # Read up on ssl_ciphers to ensure a secure configuration.  
    # See: https://bugs.debian.org/765782  
    #  
    # Self signed certs generated by the ssl-cert package  
    # Don't use them in a production server!  
    #  
    #include snippets/ssl-nginx.conf;  
  
    root /var/www/html;  
  
    # Add index.php to the list if you are using PHP  
    index index.php index.html index.htm index.nginx-debian.html;  
  
    server_name _;
```



```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    # listen [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
    #  
    # Read up on ssl_ciphers to ensure a secure configuration.  
    # See: https://bugs.debian.org/765782  
    #  
    # Self signed certs generated by the ssl-cert package  
    # Don't use them in a production server!  
    #  
    #include snippets/ssl-nginx.conf;  
  
    root /home/ubuntu/nginx-test;  
  
    # Add index.php to the list if you are using PHP  
    index index.php index.html index.htm index.nginx-debian.html;  
  
    server_name _;
```

## 디렉토리 연결이 끝났다면 명령어 실행

- sudo service nginx reload
- EC2 IP 주소를 브라우저 주소창에 입력해서 확인

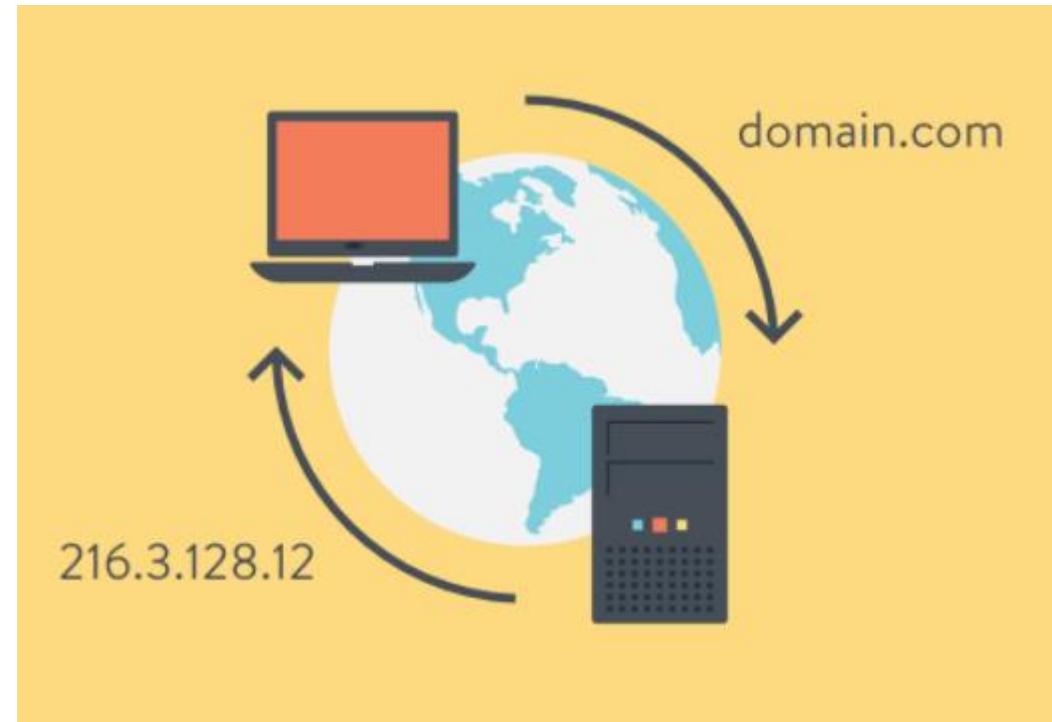


IP 입력 시, 성공적으로 테스트파일이 보인다.

## 도메인 연결

## DNS(Domain Name System)

- 사람이 이해하기 쉬운 도메인 주소를 IP로 변환
- 인터넷상의 전화번호부



## 무료 도메인 적용하기

- 나의 AWS IP 에 무료 도메인을 적용해보자
- <https://내도메인.한국>
- 회원가입 진행



# 원하는 키워드를 검색해보고 도메인 선택

## 일반 도메인 검색

.p-e.kr / .o-r.kr / .n-e.kr / .r-e.kr / .kro.kr /

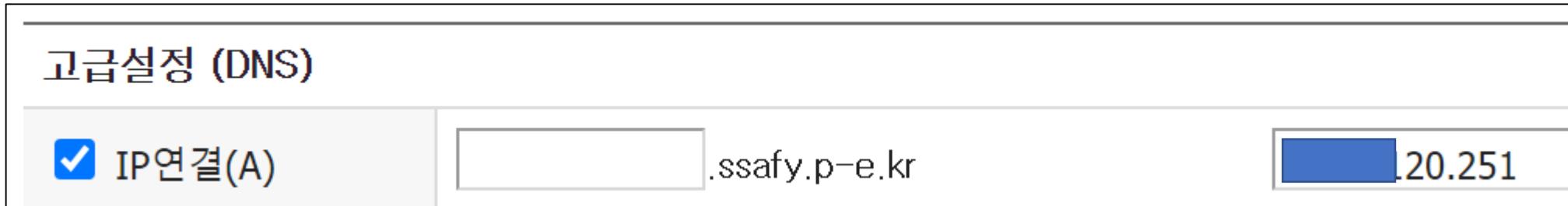
## ※ 도메인 검색 결과 ※

- |   |              |
|---|--------------|
| 1 | ssafy.p-e.kr |
| 2 | ssafy.o-r.kr |
| 3 | ssafy.n-e.kr |
| 4 | ssafy.r-e.kr |
| 5 | ssafy.kro.kr |

- |                                  |      |
|----------------------------------|------|
| <input type="radio"/>            | 등록하기 |
| <input checked="" type="radio"/> | 등록불가 |

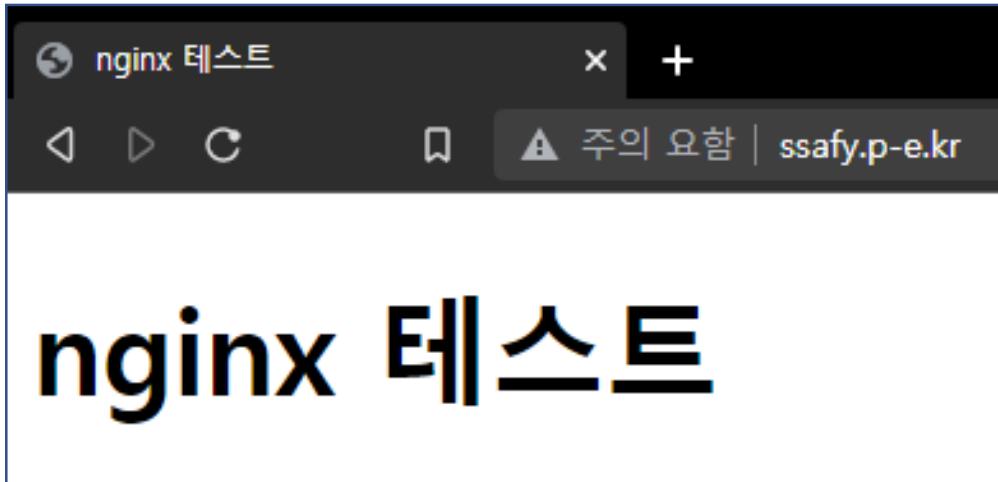
# 고급설정 AWS

- 인스턴스 아이피를 기입



- 수정하기 누른 후, 브라우저 주소창에
- 부여받은 도메인을 입력해보자





도메인으로 AWS 인스턴스에 접속 성공!



## 인스턴스 중지와 종료

- EC2 중지 vs EC2 종료

- 종료: 완전 삭제
- 정지: 사용중지, 과금중지

- 탄력적 IP

- 탄력적 IP를 설정하면 인스턴스를 중지한 후 재시작해도 IP가 변하지 않는다.
- 사용하지 않을때는 중지 버튼을 누르자

## 인스턴스 종료 = 과금 종료

- 목록에 없어지지 않아도,  
종료됨이라고 뜨면 과금종료
- 목록은 차후에 자동으로 삭제됨

인스턴스 (1/6) 정보

Name	인스턴스 ID	인스턴스 상태
–	i-0583818c7bd377d3f	실행 중
–	i-095f9a95130bb2b72	실행 중
–	i-03d158422e294fe60	실행 중
–	i-0f928013a775ceb3b	실행 중
–	i-0b0bdbf60c9337623	실행 중
<input checked="" type="checkbox"/>	i-05033a46683d2c073	

인스턴스 시작  
템플릿으로 인스턴스 시작  
Migrate a server  
연결  
인스턴스 중지  
인스턴스 시작  
인스턴스 재부팅  
인스턴스 최대 절전 모드  
인스턴스 종료

세부 정보 | 보안 | 네트워킹 | 스토리지

▼ 인스턴스 요약 정보

## 탄력적 IP 유의 사항

- 인스턴스 종료 후 탄력적 IP 미사용 시 요금이 부과
- 해당 탄력적 IP 갱신 or 제거 하기(릴리스 작업)

The screenshot shows the AWS Management Console interface for managing elastic IPs. On the left, a sidebar menu under '네트워크 및 보안' (Network & Security) has '탄력적 IP' (Elastic IP) selected, indicated by a red box. The main content area is titled '탄력적 IP 주소 (1/1)' and shows a single entry in a table:

Name	할당된 IPv4 주소	유형
-	■■■.120.251	퍼블릭 IP

A red box highlights the IP address '■■■.120.251'. Below the table, the text 'IP 클릭' (Click IP) is visible.

## 탄력적 IP 릴리스

- 연결 해제 이후 릴리즈버튼 활성화
- 해제 후 릴리스



## 나의 포트폴리오 사이트 자랑하기

- **인스턴스 삭제** 후 모든과정을 처음부터 다시 진행하기
- WEB PROJECT 첫 주간 마지막날에 진행한 PJT를 서버에 업로드 하기
- 업로드가 완료되었다면 친구의 포트폴리오 페이지에 도메인 접근하기

### About



David. Moms  
IoT Developer  
(Web + Embedded)

안녕하세요. 언제나 달려가는 개발자 David.Moms 입니다.  
함께 일하시고 싶으시다면 언제든 연락주셔도 됩니다.

010-9360-2485  
inho.choi@mincoding.co.kr  
경기도 성남시 분당구 정자동  
mincoding.co.kr

# 6장. Node.js 시작

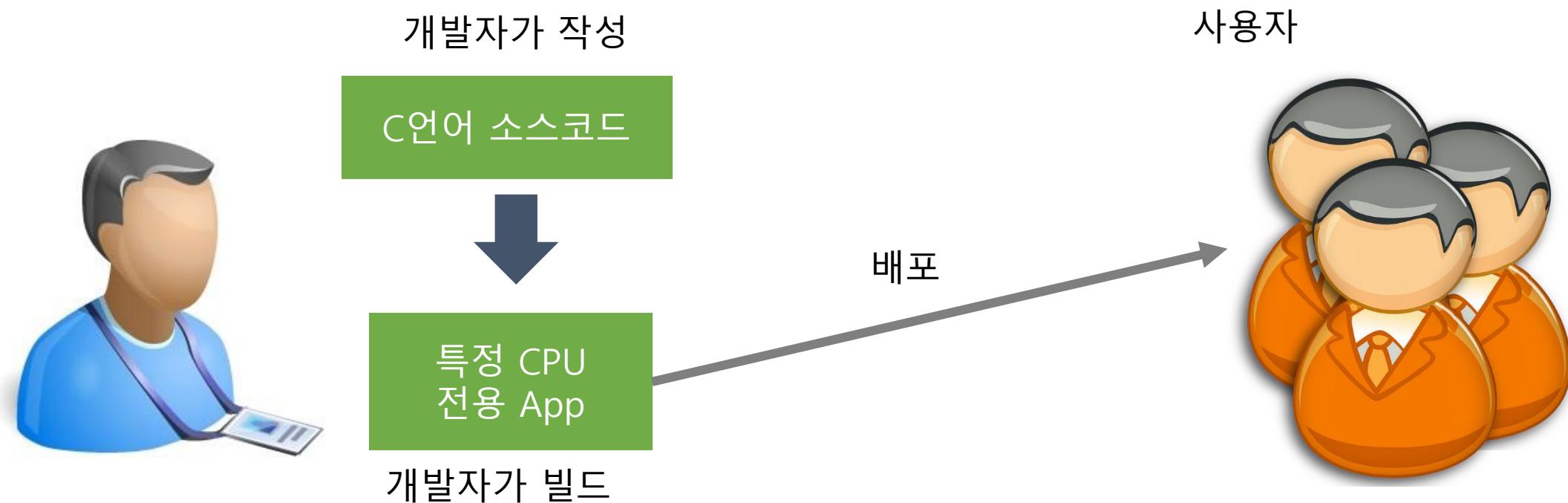
# 챕터의 포인트

- Node.js 이해
- Node.js 설치하기
- Node.js 활용 예
- 모듈

# Node.js 이해

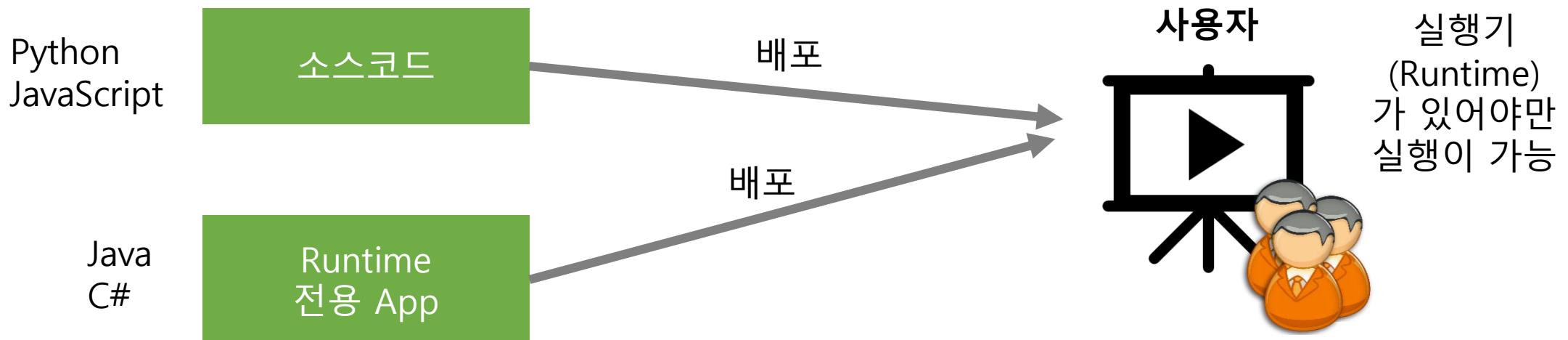
## 컴파일 언어

- 기존 컴파일 언어는 CPU가 알아들을 수 있는 2 진수 명령어로 번역 후, 수행하는 방식
- 번역 완료된 2 진수 명령어들은, 실행기 없이 App 동작 가능하다.



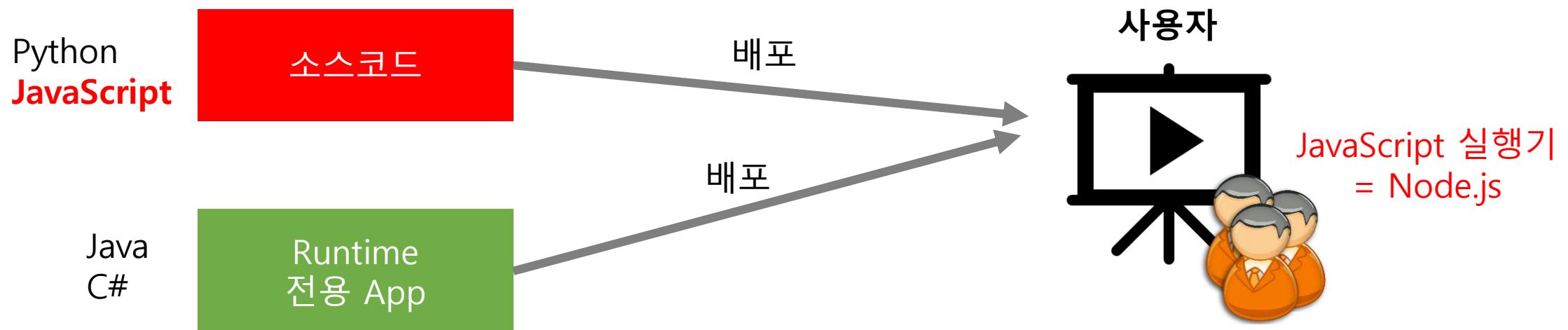
## Runtime Environment

- 실행에 필요한 것들은 모아둔 환경을 “Runtime 환경”이라고 하며, 줄여서 Runtime이라고 함
- Runtime은 하나의 Run System이며, “실행기”이다.
- 특정 언어는 실행기가 있어야만, App 동작이 가능하다.



## Node.js

- JavaScript의 Runtime (실행기, Runtime Environment)



## JavaScript를 Python 처럼 사용할 수 있다.

- JavaScript는 웹브라우저 전용 언어가 아니다.
- Node.js라는 Runtime만 설치 해 두면, JavaScript로 만든 App들을 사용 가능 함

### Python Source code

```
1 from sympy import (
2     TimeDifferencingWrapper, NetCDFMonitor)
3 import climt
4 from datetime import timedelta
5
6 # Define model timestep in minutes
7 model_timestep = timedelta(minutes=1)
8
9 # Create Components
10 radiation = climt.RRTMGLongwave()
11 convection = climt.EmanuelConvection()
12 boundary_layer = TimeDifferencingWrapper(
13     climt.SimplePhysics())
14 time stepper = GFSDynamicalCore(
15     [radiation, convection, boundary_layer])
```

Runtime 이름 : Python

### JavaScript Source code

```
wrapper = function(tag){
    return function(v){return '<' + tag + '>' + }
}

initArr = function(arr){
    return Array.isArray(arr) ? arr : []
}

var people = [
    ['김민수', '3', '아들'],
    ['임선영', '39', '엄마'],
    ['김태완', '43', '아빠']
]
```

Runtime 이름 : Node.js

## JavaScript로 만든 소스코드 동작 방법

### 1. 웹브라우저에서 JavaScript 코드 수행

웹브라우저 안에 실행 환경이 구축 되어 있음 == JavaScript Engine

Chrome에는 “V8”이라는 이름의 엔진이 사용되고 있음

### 2. 웹브라우저 “밖에서” JavaScript 코드 수행 :

Node.js 필요 = JavaScript의 실행기가 필요하기 때문

JavaScript Engine 이상의 동작 환경을 제공 함 ( 외부 모듈 로딩 / Thread 등 동작 가능)

# 서버에서, Node.js 이외의 기술 비교

Confidential



기술명	언어	장점	단점
Spring	Java	대한민국에서 가장 많이 쓰임 정부 사업은 오로지 Spring 으로 진행 (전자정부 프레임워크) 객체 지향 개발에 특화됨	난이도 어려움 Java 언어의 난해한 가독성 간단한 프로젝트시엔 지나치게 복잡함
Django	Python	Python 의 간결한 문법. 가독성 좋음 일반적인 사이트에 필요한 모듈 기본 제공 ex) 인증, admin ... 간단한 REST API 를 만들기에 적합 (Django REST Framework)	파이썬의 느린 성능 사용률이 낮음 => 취업 힘듦 Django ORM 강요
Node.js	JavaScript	개발자에게 높은 자유도 부여 ex) ORM 강제 안함 클라이언트와 동일한 언어인 JS 사용 싱글스레드, 비동기에 기반한 빠른 처리속도	자유도가 너무 높아 직접 설계해야 할 일이 많음 JS 의 비동기에 익숙해져야

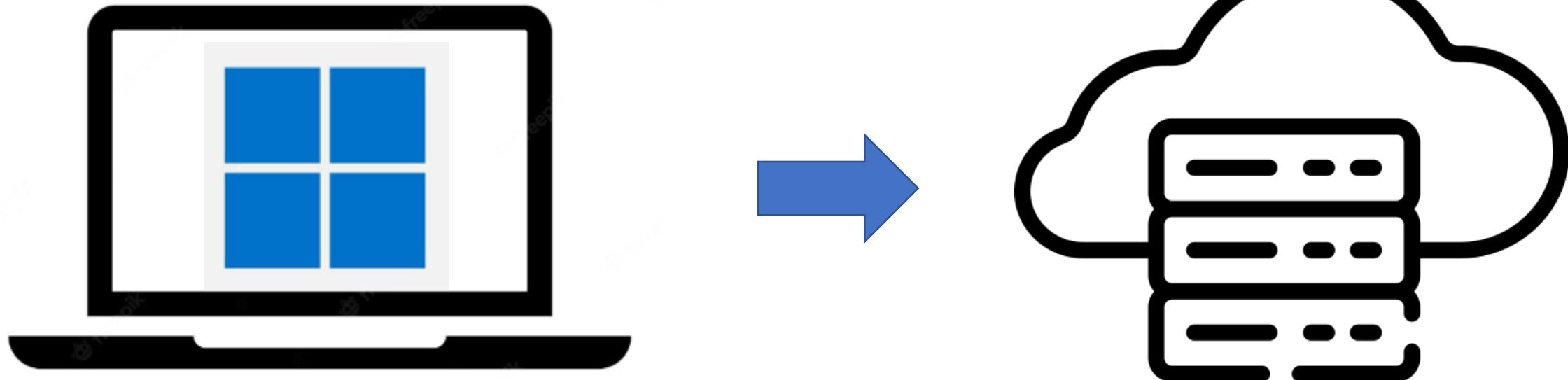
# Node.js 설치하기

## 프리티어용 인스턴스 유형( t2.micro ) 스펙

- CPU 1 vCPU
- 메모리 1 GiB
- SSD 8 GiB
- 라즈베리 파이보다 낮은 스펙으로써 서비스 배포 목적으로 사용해야한다

## AWS 서버 활용하기

- 개발은 윈도우, 배포는 AWS
- 즉, 윈도우에서 Node.js 를 설치해 개발 예정



## LTS (롱텀서포트)

- 가장 안정적인 버전
- 3년간의 유지보수, 업데이트 지원

## Current

- 현재 개발 중인 버전
- 최신 기술 사용 가능

The screenshot shows the official Node.js website. At the top, there's a dark header bar with the Node.js logo and a navigation menu in Korean: 홈 | ABOUT | 다운로드 | 문서 | 참여하기 | 보안 | 뉴스 | CERTIFICATION. Below the header, a green banner states: "Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다." A dark green box contains the text: "Changing the End-of-Life Date for Node.js 16 to September 11th, 2023". In the center, there are two download buttons: "다운로드 - Windows (x64)" with "16.15.1 LTS" (highlighted with a red border) and "18.3.0 현재 버전". At the bottom, there are links for "다른 운영 체제 | 변경사항 | API 문서" for each version.

Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

Changing the End-of-Life Date for Node.js 16 to September 11th, 2023

다운로드 - Windows (x64)

16.15.1 LTS  
안정적, 신뢰도 높음

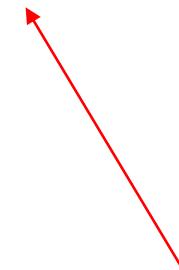
18.3.0 현재 버전  
최신 기능

다른 운영 체제 | 변경사항 | API 문서      다른 운영 체제 | 변경사항 | API 문서

LTS 일정은 여기서 확인하세요

## 명령 프롬프트 창 수행 후 명령어 입력

- node - v
- npm - v



노드 패키지 매니저  
파이썬의 pip와 같은 역할을 하는 툴

```
명령 프롬프트
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mincoding>node -v
v16.15.1

C:\Users\mincoding>npm -v
npm WARN config global`--global`, `--local` are
```

## Node를 수행 후 Hello World를 출력

- CMD 창에서 node 입력하면 Node Shell이 수행된다.

```
C:\Users\minco>node
```

```
Welcome to Node.js v16.15.1.  
Type ".help" for more information.  
> console.log("Hello world");  
Hello world  
undefined  
>
```

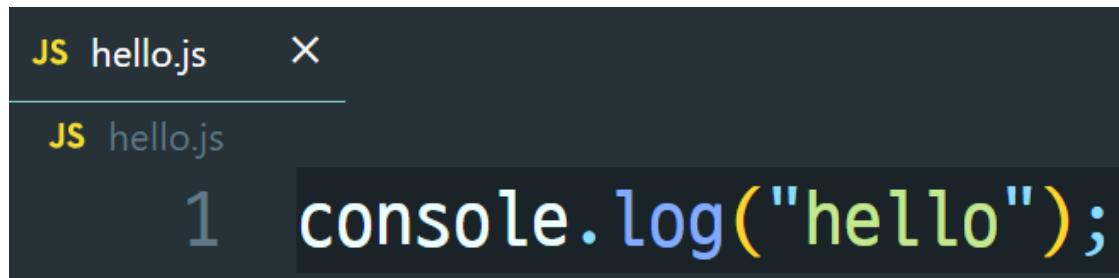
# REPL은 Node의 Shell을 뜻한다.

- 사용자 명령을 읽고, 수행 후 출력하는 동작을 반복하는 터미널을 뜻함
- Read(읽고) Eval(해석하고) Print(출력하고) Loop (반복하고)의 줄임말
- Python Shell 과 같은 역할.

```
> let a = 100;
undefined
> console.log(a + 200);
300
undefined
> "HI MINCO" + "HAHAHA"
'HI MINCOHAHAHA'
> const sum = (a, b)=> a + b;
undefined
> sum(35, 40);
75
>
```

## vscode에 파일 생성하기

- `hello.js` 파일에 `console.log("hello")` 입력
- vscode 터미널을 켠다
  - 상단 메뉴중 터미널클릭 -> 새 터미널 열기 or Ctrl + Shift + `
  - 터미널에서 `node hello.js` 입력후 결과 확인하기



```
JS hello.js ×  
JS hello.js  
1 console.log("hello");
```



```
C:\Users\mincoding\Desktop\node-test>node hello.js  
hello
```

## Javascirpt로 사칙 연산 계산기 만들기

- JavaScript로 사칙연산 계산기를 만든 후,

### Node.js로 작동시켜보기

- 파일명: calc.js
- 입력: **하드코딩**
  - ex) a = 2, b = 4
- 출력
  - 오른쪽 참고
- 조건
  - 화살표 함수를 사용해 객체 calc를 생성
    - calc.add(a, b) 형태 식으로 코드를 작성

```
C:\Users\mincoding\Desktop\node-test>node calc.js  
주어진 수: 2 4  
덧셈 결과: 6  
뺄셈 결과: -2  
곱셈 결과: 8  
나눗셈 결과: 0.5
```

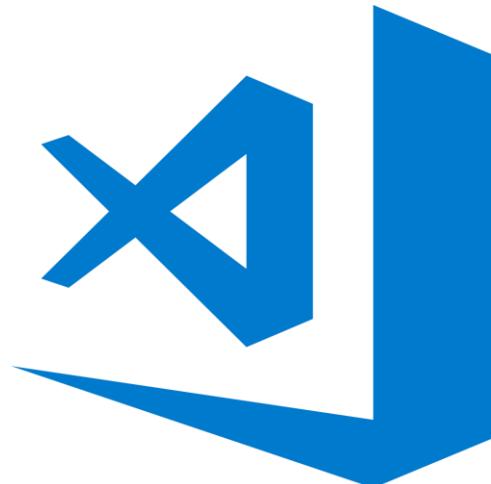
## Node.js 활용 예

## Node.js란

- 지금까지 경험했듯 Node.js 는 브라우저 바깥에서 사용하는 JavaScript
- Node.js 를 써야만 하는 분야
  - 서버 ( REST API, SQL DB 연결 )
  - 모바일 앱
  - 데스크톱 앱

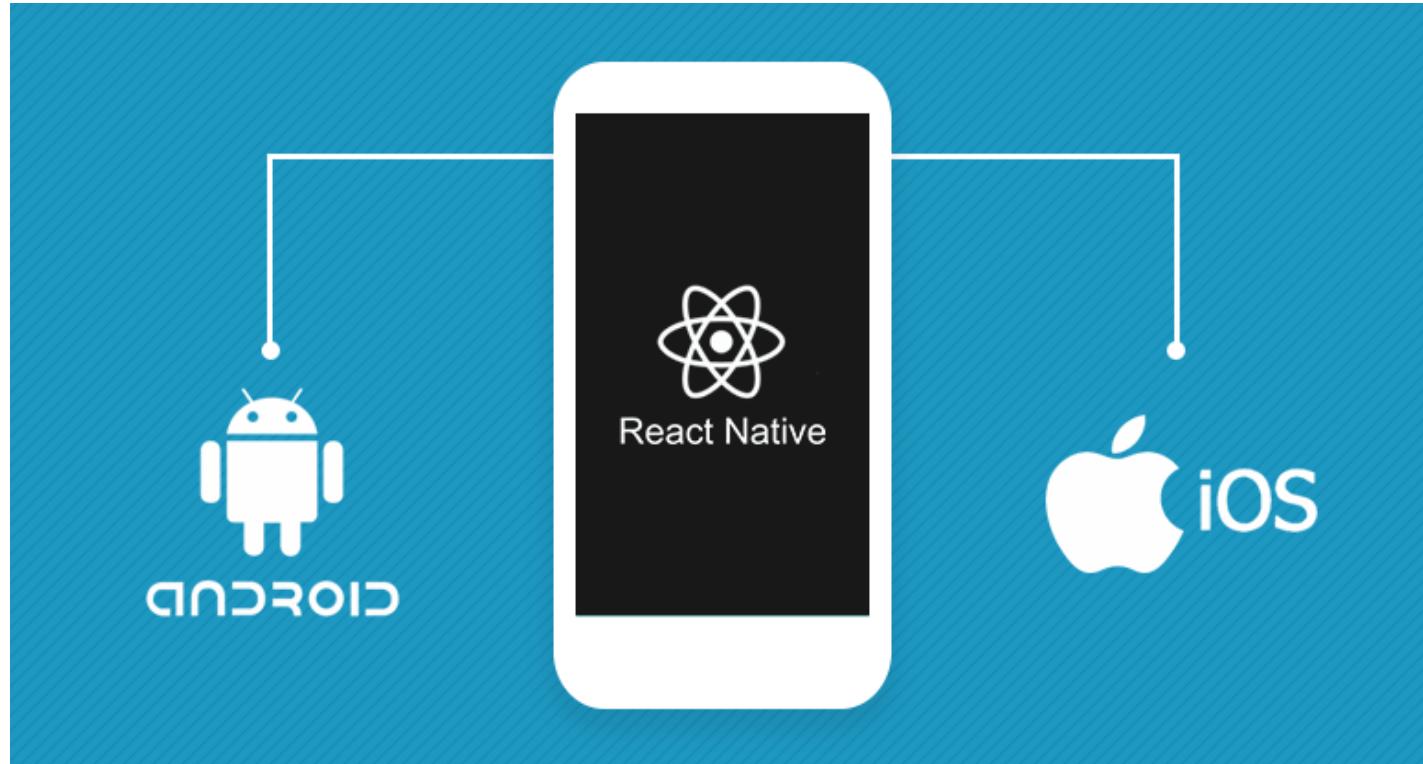
## Desktop App 개발

- **Electron Framework** 으로 개발한다.
  - App 1개에 Node.js + Chromium 브라우저를 포함되어 있음  
즉, App에 Node.js Runtime이 내포되어, 사용자는 따로 Node.js 설치 필요 없음



## Mobile App 개발

- React-Native : iOS + Android App을 동시에 개발 가능
- 인스타, 페이스북이 React-Native로 개발 되었음



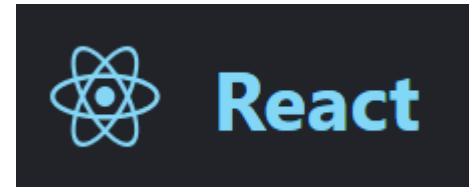
## Web 개발 : Server Side

1. Node.js 에는 간단한 웹 서버가 내장되어 있다. (테스트 용도로만 사용)
2. **Express.js** 를 통해 HTTP protocol 로 서버의 JavaScript 메서드를 호출할 수 있음  
→ REST API 제작 가능

Express

## Web 개발 : Client Side

- Vue.js / React / Angular를 통해 비동기방식 웹사이트를 쉽게 만들 수 있음
- 비동기 웹사이트 제작으로, 서버 부하를 줄이기 위해 사용



수업 예정



# 모듈

## 객체로 묶인 맴버들을, 변수로 분해할 수 있는 문법

- destructing - 객체
- name1 / name2 / run 변수에 철수, 영희, **run함수가** 각각 대입된다.

```
1 const friends = {  
2   name1: "철수",  
3   name2: "영희",  
4   run: (word) => console.log(word, "가자"),  
5 };  
6  
7 const { name1, name2, run } = friends;  
8  
9 console.log(name1); // 철수  
10 console.log(name2); // 영희  
11 run("하핳"); // 하핳 가자
```

## hi.js에서 lib.js 모듈을 불러오기

- lib.js : module.exports라는 Object 객체에 넣어둔다.
- hi.js : require 메서드를 사용하면, 해당 파일의 module.exports 객체에 접근할 수 있다.

```
JS lib.js      X  JS hi.js
JS lib.js > ...
1 const friends = {
2   name1: "철수",
3   name2: "영희",
4   run: (word) => console.log(word, "가자"),
5 };
6
7 module.exports = friends;
```

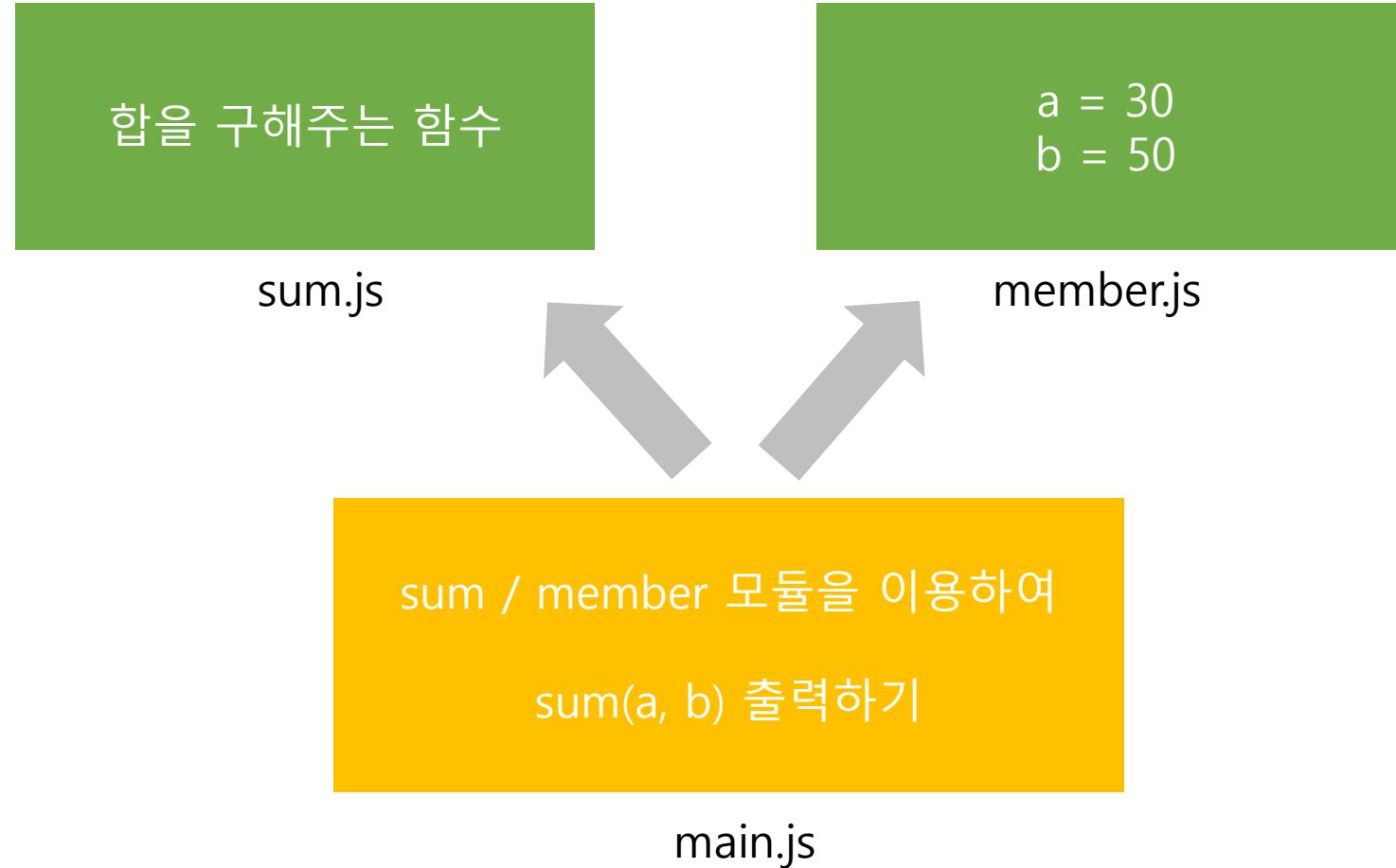
```
1 const { name1, name2 } = require("./lib");
2 ...
3 console.log(name1, name2);
```

js 확장자는 생략이 가능하다.

```
C:\Users\mincoding\Desktop\node-test>node hi.js
철수 영희
```

## 간단한 모듈을 만들어보자

- sum.js
- member.js
- main.js



내일  
방송에서  
만나요!

삼성 청년 SW 아카데미