

DevOps 구축 BOOTCAMP



Owen (Jepil Choi)

Back-End, DevOps Engineer.



SKILLSET

Front-end

Back-end

DevOps

Project management

Design

EDUCATION

- Hanyang Univ. (Major of Information system)

AJ NETWORKS

Architecture team

Next generation ERP systems DevOps project.

- Contributed to built and managed an IDC based to Micro Service architecture(MSA) with DC/OS
- Built and managed a CI/CD Pipeline with Jenkins CI and Gitlab CI
- Built and managed a Stress test environment with jMeter

Responsible for development the database metadata management tools

- Developed a RESTful backend with Node.js, Express, Knex.js and MySQL
- Developed a frontend with HTML5, Materialize CSS, LESS, jQuery, Vue.js and EJS

AJ Networks

2016 - 2018

PIKICAST

Development team

Pikicast web site(v2.0)

- Responsible for development the backend with PHP and Codeigniter.
- Management for web project.

[Click Link](#)

Pikicast

2014 - 2015

ADD2PAPER

Development team

- Maintained a printer driver and client
- Maintained an android client

Add2paper

2013 - 2014

DevOps

DevOps

데브옵스는 애플리케이션과 서비스를 빠른 속도로 제공할 수 있도록
조직의 역량을 향상시키는 문화 철학, 방식 및 도구의 조합

DevOps

개발 환경

DevOps

프로젝트 환경 구성, TDD, 빌드, 배포, ETC...

DevOps

운영 환경

DevOps

인프라 구성, 가용성, 모니터링, ETC...

DevOps

비즈니스의 발전에 빠르게 대응하기 위한 일련의 대응

강의 진행 방식

모든 과정은 실습 -> 개념 설명으로 진행합니다

질문은 facebook group에 달아주시면 상세히 답변드려요!

피드백 많이 부탁드립니다

1주차 목표

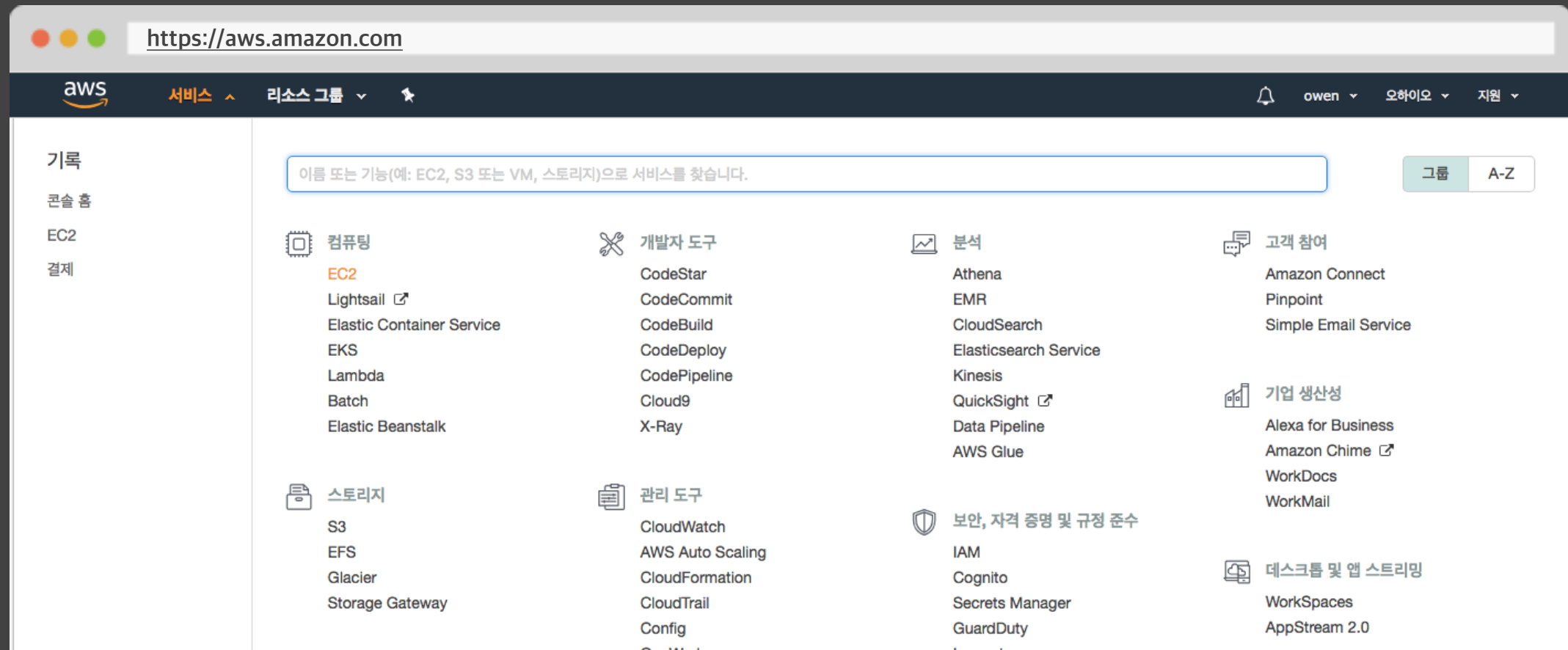
Web Server 설치 및 운영

서버 아키텍처 별 차이 비교

Docker를 간단히 써볼겁니다

AWS, CLI(명령어)와 친해져봅시다

AWS EC2 설치, 생성



1. <https://aws.amazon.com> 접속, 로그인
2. 상단 네비게이션 바에 서비스 탭 클릭 후 컴퓨팅(EC2) 클릭
3. 인스턴스 생성 하단 인스턴트 시작 버튼 클릭

AWS EC2 설치, 생성

https://aws.amazon.com

aws 서비스 리소스 그룹

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 1: Amazon Machine Image(AMI) 선택

[취소 및 종료](#)

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버, 애플리케이션)이 포함된 템플릿입니다. AWS, 사용자 커뮤니티 또는 AWS Marketplace에서 제공하는 AMI를 선택하거나, 자체 AMI 중 하나를 선택할 수도 있습니다.

빠른 시작

- 나의 AMI
- AWS Marketplace
- 커뮤니티 AMI

☐ 프리 티어만 ⓘ

	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-afd86dc1	선택
Amazon Linux 프리 티어 사용 가능	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.	64비트
	루트 디바이스 유형: ebs 가상화 유형: hvm	
	Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-ebc47185	선택
Amazon Linux 프리 티어 사용 가능	Amazon Linux AMI는 EBS 기반의 AWS 지원 이미지입니다. 기본 이미지에는 AWS 명령줄 도구, Python, Ruby, Perl 및 Java가 있습니다. 리포지토리에는 Docker, PHP, MySQL, PostgreSQL 및 기타 패키지가 포함됩니다.	64비트
	루트 디바이스 유형: ebs 가상화 유형: hvm	
	Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-3eee4150	선택
Red Hat 프리 티어 사용 가능	Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type	64비트
	루트 디바이스 유형: ebs 가상화 유형: hvm	

1. AMI 선택 - Amazon Linux 2 AMI (HVM), SSD Volume Type 선택
2. 상위 탭(6.보안 그룹 구성) 클릭

AWS EC2 설치, 생성

aws

서비스 리소스 그룹

owen 서울 지원

1. AMI 선택2. 인스턴스 유형 선택3. 인스턴스 구성4. 스토리지 추가5. 태그 추가6. 보안 그룹 구성7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용 하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 자세히 알아보기

보안 그룹 할당: ☒ 새 보안 그룹 생성 ☐ 기존 보안 그룹 선택

보안 그룹 이름:

설명:

유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	사용자 지정 0.0.0.0/0	예: 관리자 데스크톱용 SSH
HTTP	TCP	80	사용자 지정 0.0.0.0/0, ::/0	예: 관리자 데스크톱용 SSH

규칙 추가

경고

소스가 0.0.0.0/0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다.

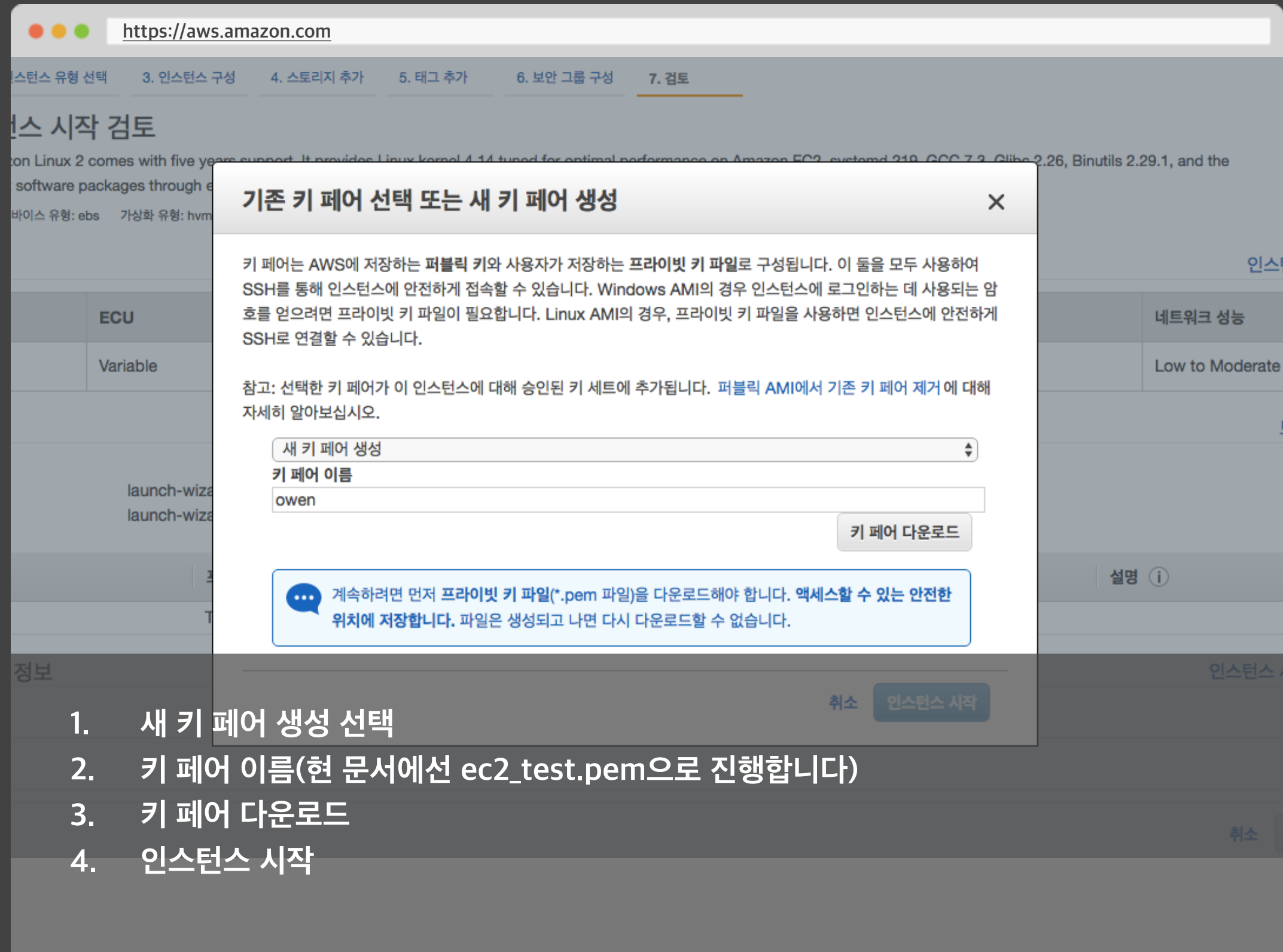
1. 새 보안 그룹 생성 선택(보안 그룹 이름, 설명 입력)

2. 규칙 추가(HTTP)

3. 검토 및 시작

취소 이전 검토 및 시작

AWS EC2 설치, 생성



https://aws.amazon.com

인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

인스턴스 시작 검토

on Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.2, Glibc 2.26, Binutils 2.29.1, and the software packages through e

바이스 유형: ebs 가상화 유형: hvm

ECU	Variable
-----	----------

launch-wiza
launch-wiza

키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

새 키 페어 생성

키 페어 이름
owen

키 페어 다운로드

계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.

취소 인스턴스 시작

1. 새 키 페어 생성 선택
2. 키 페어 이름(현 문서에선 ec2_test.pem으로 진행합니다)
3. 키 페어 다운로드
4. 인스턴스 시작

AWS EC2 설치, 생성

The screenshot shows the AWS Management Console interface. On the left, the navigation menu includes 'EC2 대시보드', '이벤트', '태그', '보고서', '제한', '인스턴스', 'Launch Templates', '스팟 요청', '예약 인스턴스', '전용 호스트', '이미지', 'AMI', '번들 작업', and 'ELASTIC BLOCK STORE'. The main content area displays a list of EC2 instances. The instance 'i-0797ae60d19139478' is highlighted, and its details are shown in the 'Details' tab. The instance is in the 'running' state, using the 't2.micro' instance type in the 'ap-northeast-2' region. The 'Public DNS (IPv4)' field is highlighted with a red box, showing the value 'ec2-13-125-216-230.ap-northeast-2.compute.amazonaws.com'. Other fields include 'Instance ID', 'Instance State', 'Instance Type', 'Elastic IP', 'Availability Zone', 'Security Groups', 'Private DNS', 'Private IP', and 'Secondary Private IP'.

Field	Value
인스턴스 ID	i-0797ae60d19139478
인스턴스 상태	running
인스턴스 유형	t2.micro
탄력적 IP	-
가용 영역	ap-northeast-2c
보안 그룹	launch-wizard-1. 인바운드 규칙 보기 . view outbound rules
퍼블릭 DNS(IPv4)	ec2-13-125-216-230.ap-northeast-2.compute.amazonaws.com
IPv4 퍼블릭 IP	13.125.216.230
IPv6 IP	-
프라이빗 DNS	ip-172-31-25-131.ap-northeast-2.compute.internal
프라이빗 IP	172.31.25.131
보조 프라이빗 IP	-

1. 인스턴트 상태(pending -> running) 확인
2. 설명 탭에서 퍼블릭 DNS 복사

NginX 설치, 실행

1. ec2-user@ip-172-31-25-131:~ (bash)

a1@1ui-MacBook-Air:~\$

```
sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소} ↵
```

1. ssh 접속 (앞서 복사한 DNS 주소) - ex) `sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@ec2-13-125-251-146.ap-northeast-2.compute.amazonaws.com`
2. (처음 접속 시) yes 입력

참고 https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html

NginX 설치, 실행

1. ec2-user@ip-172-31-25-131:~ (bash)

a1@1ui-MacBook-Air:~\$

`sudo su` ↵

`amazon-linux-extras install nginx1.12` ↵

`nginx` ↵

1. `sudo su` 명령어로 root 권한으로 변경
2. nginx1.12 설치
3. nginx 실행
4. 앞서 복사한 DNS 주소로 웹 브라우저에서 접속(NginX 실행 확인)

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

AWS EC2

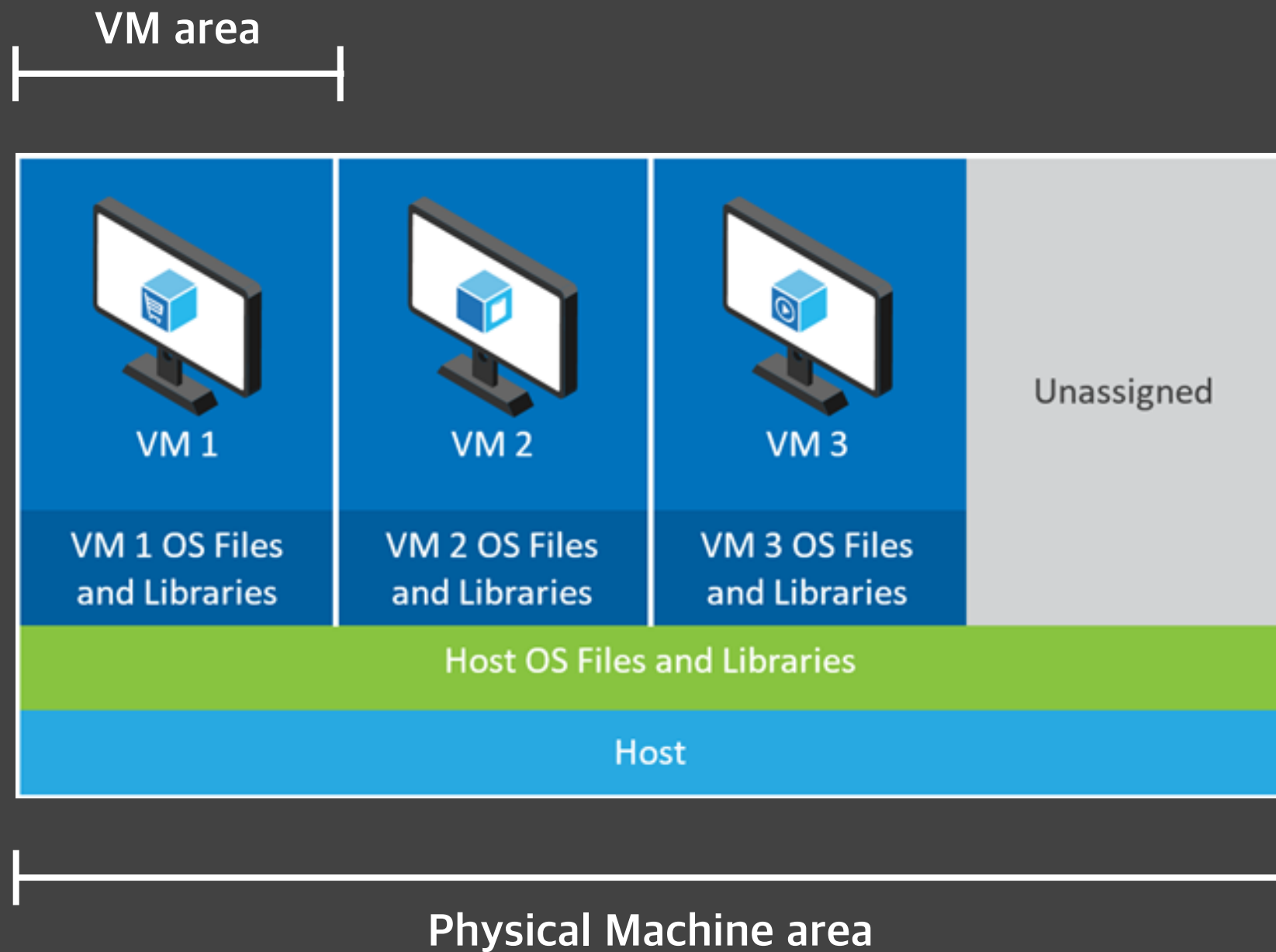
- AWS에서 제공하는 가상 서버(Virtual Machine, **VM**)
- Amazon Linux, Cent OS, Ubuntu, Windows 등 다양한 OS 제공
- 원하는 사양을 관리자가 직접 선택할 수 있음
- 필요할 때마다 사용 / 중지가 가능

Virtual Machine

가상 머신은 "실제 컴퓨터의 효율적이고 고립된 복제물"로 정의했다. 현재는 "실제 하드웨어와 직접적인 통신이 없는 가상 컴퓨터"를 가리킨다.

참고 https://ko.wikipedia.org/wiki/%EA%B0%80%EC%83%81_%EB%A8%B8%EC%8B%A0

Virtual Machine



물리 머신 위에 독립적 OS
실행이 가능한 논리적인 가상
OS를 생성하여 실행

- 하나의 서버에서 여러 OS를 실행
- 자원(CPU, Memory 등)의 공유

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

Amazon Linux2를 기반으로 하는 AWS AMI를 활용하여
EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

Amazon Linux2를 기반으로 하는 AWS AMI를 활용하여
EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

Amazon Linux2

- AWS에서 제공하는 차세대 Linux 운영 체제
- Amazon Linux Extras 리포지토리를 통해 최신 소프트웨어 패키지 제공
- Container 환경에 최적화 된 OS

참고 <https://aws.amazon.com/ko/about-aws/whats-new/2017/12/introducing-amazon-linux-2/>

Amazon Linux2

- AWS에서 제공하는 차세대 Linux 운영 체제
- **Amazon Linux Extras** 리포지토리를 통해 최신 소프트웨어 패키지 제공
- Container 환경에 최적화 된 OS

참고 <https://aws.amazon.com/ko/about-aws/whats-new/2017/12/introducing-amazon-linux-2/>

Amazon Linux2

```
1. ec2-user@ip-172-31-25-131:~ (bash)
a1@lui-MacBook-Air:~$
```

sudo su ↵

amazon-linux-extras install nginx1.12 ↵

nginx ↵

Linux 별 차이

- Cent OS, Amazon Linux, Ubuntu, Debian, Etc...
- 목적에 맞는 Linux 사용 (가장 좋은 Linux는 없습니다)
- 저희 수업에선 Amazon Linux, Cent OS를 주로 사용합니다

참고 <http://korean-daeddo.blogspot.com/2016/11/blog-post.html>

Amazon Linux2를 기반으로 하는 **AWS AMI**를 활용하여
EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

AWS AMI

aws

서비스 ▾ 리소스 그룹 ▾

owen ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 1: Amazon Machine Image(AMI) 선택

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버, 애플리케이션)이 포함된 템플릿입니다. AWS, 사용자 커뮤니티 또는 AWS Marketplace에서 제공하는 AMI를 선택하거나, 자체 AMI 중 하나를 선택할 수도 있습니다.

취소 및 종료

빠른 시작

나의 AMI

AWS Marketplace

커뮤니티 AMI

☐ 프리 티어만 ⓘ

Amazon Linux

프리 티어 사용 가능

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-afd86dc1

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

루트 디바이스 유형: ebs 가상화 유형: hvm

선택

64비트

Amazon Linux

프리 티어 사용 가능

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-ebc47185

Amazon Linux AMI는 EBS 기반의 AWS 지원 이미지입니다. 기본 이미지에는 AWS 명령줄 도구, Python, Ruby, Perl 및 Java가 있습니다. 리포지토리에는 Docker, PHP, MySQL, PostgreSQL 및 기타 패키지가 포함됩니다.

루트 디바이스 유형: ebs 가상화 유형: hvm

선택

64비트

Red Hat

프리 티어 사용 가능

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-3eee4150

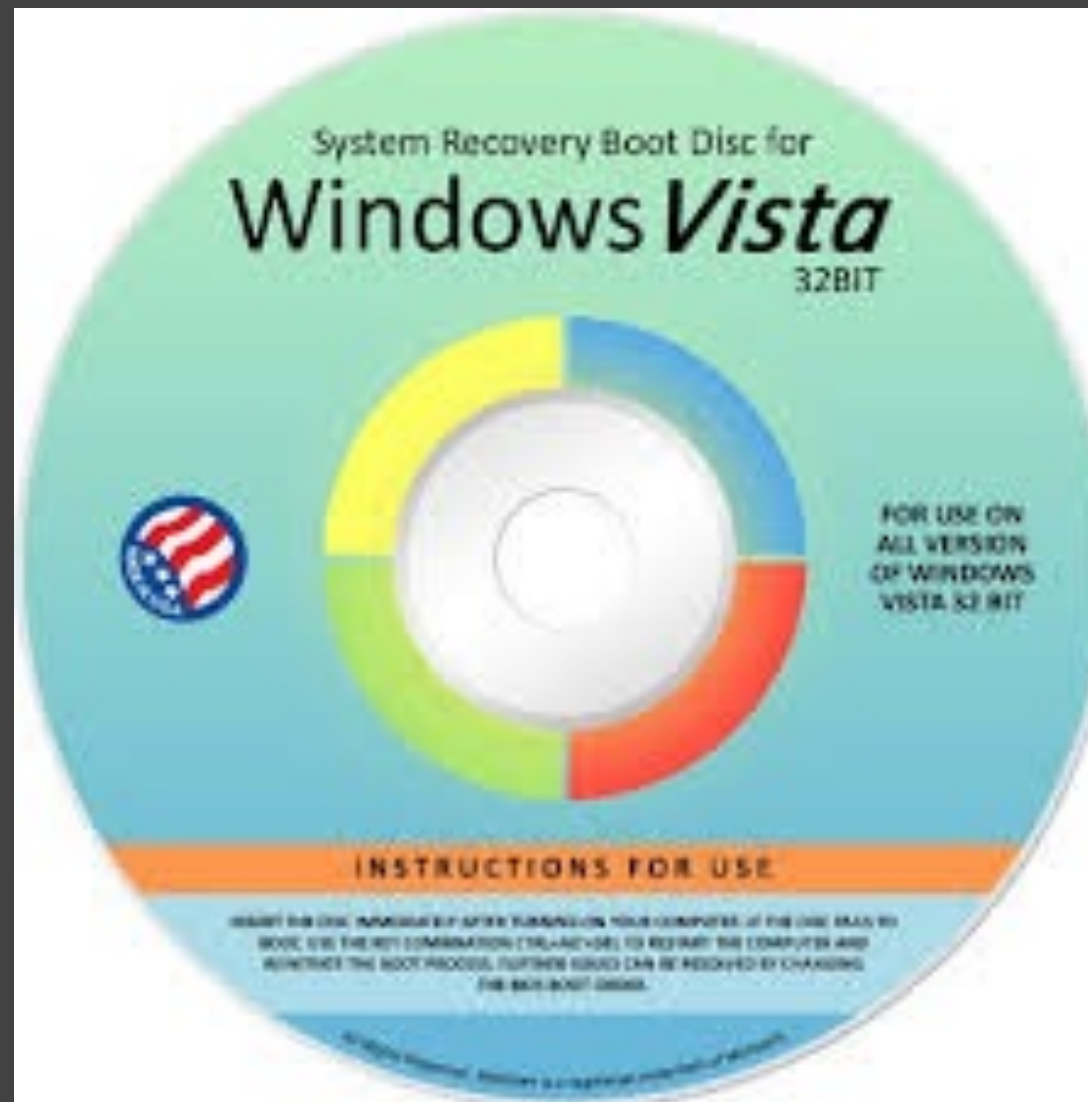
Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

루트 디바이스 유형: ebs 가상화 유형: hvm

선택

64비트

AWS AMI



AWS AMI

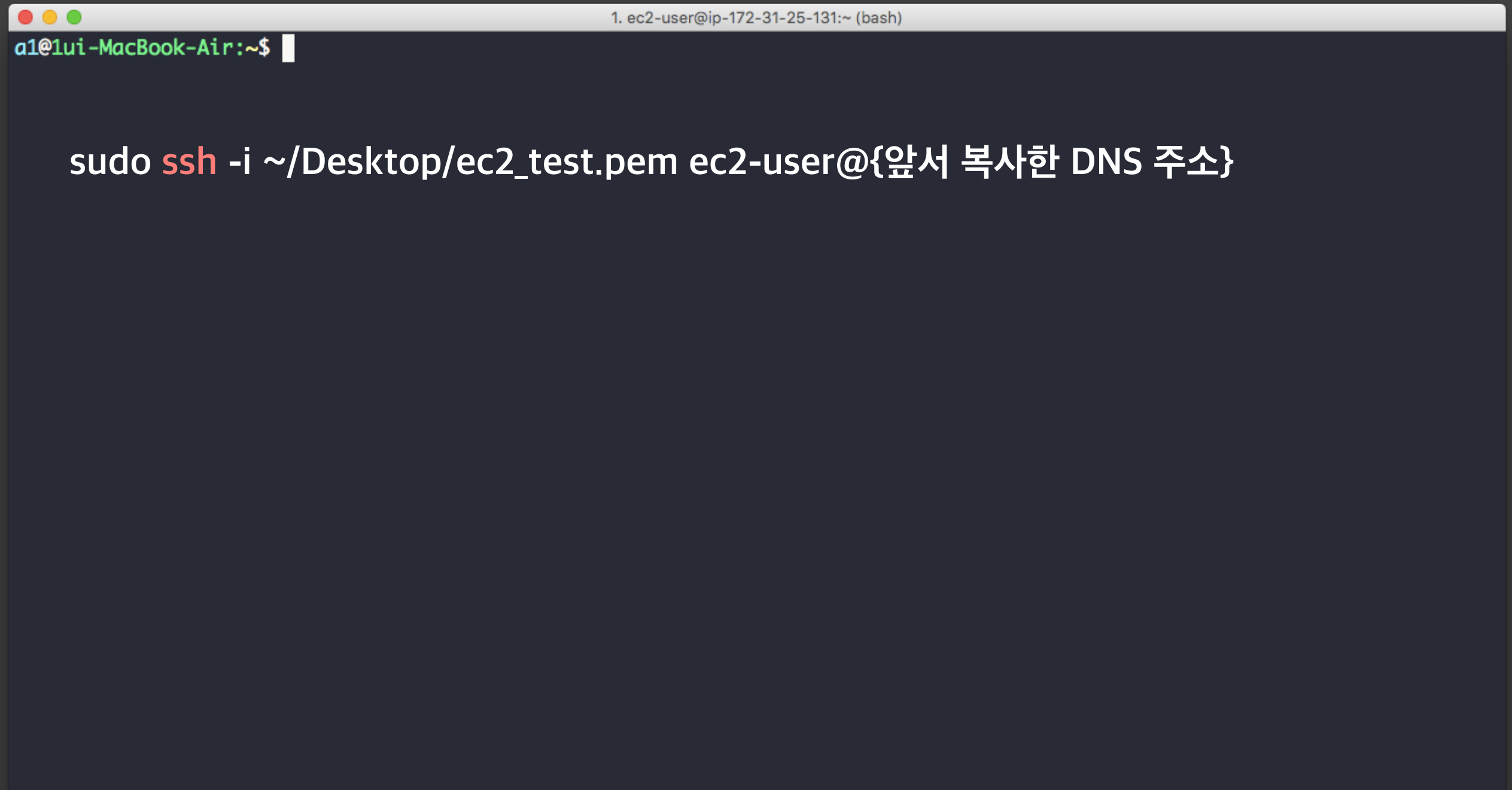
- 서버 인스턴스의 기반이 되는 이미지
- 윈도우 CD, iso와 같은 개념
- Amazon에서 제공하는 빅데이터, 웹서비스 등 각 기능 별로 빌드구워낸 한 이미지를 사용하거나 다른 사람이 공유하는 이미지도 사용 가능

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

EC2를 생성하고 SSH 클라이언트로 접속하여 해당 VM에 웹 서버를 설치 후 외부에서 확인하였습니다.

EC2를 생성하고 **SSH 클라이언트**로 접속하여 해당 VM에 웹 서버를 설치 후 외부에서 확인하였습니다.

SSH



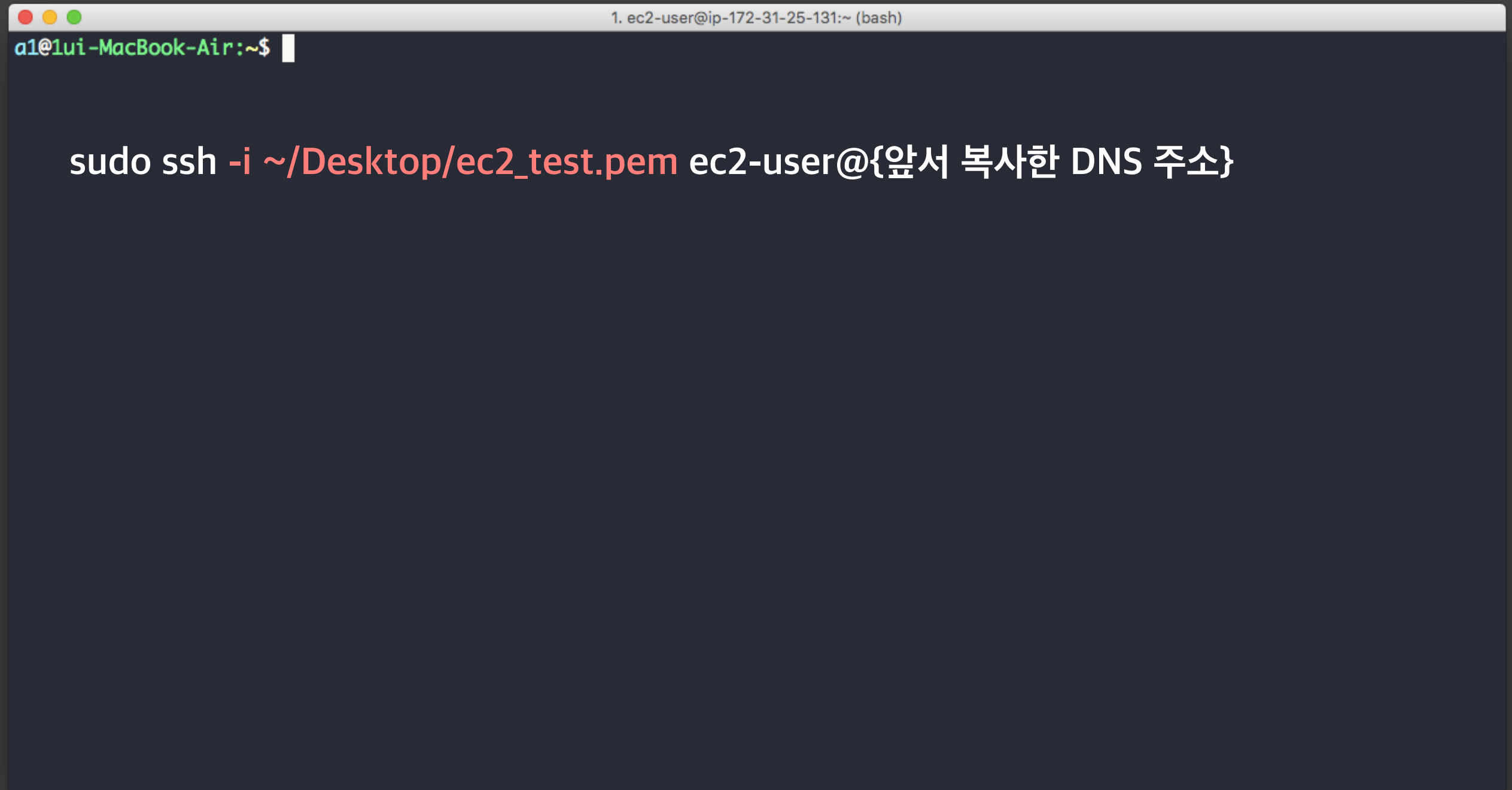
A terminal window with a dark blue background. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and the text "1. ec2-user@ip-172-31-25-131:~ (bash)" on the right. The terminal content shows a prompt "a1@1ui-MacBook-Air:~\$" followed by a command: "sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}".

```
a1@1ui-MacBook-Air:~$ sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}
```

SSH

- SSH(Secure shell)은 다른 컴퓨터에 접속하거나 원격 시스템에서 명령을 실행할 수 있는 클라이언트 / 프로토콜을 의미
- 대부분의 리눅스 배포판에선 기본적으로 설치되어있음
- 22번 포트 사용
- -i 옵션으로 비밀번호가 아닌 SSH key로 접속

SSH Key



A terminal window with a dark blue background. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and the text "1. ec2-user@ip-172-31-25-131:~ (bash)" on the right. The terminal content shows a prompt "a1@1ui-MacBook-Air:~\$" followed by a command: "sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}" where the file path is highlighted in red.

```
a1@1ui-MacBook-Air:~$ sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}
```


SSH Key

https://aws.amazon.com

인스턴스 유형 선택3. 인스턴스 구성4. 스토리지 추가5. 태그 추가6. 보안 그룹 구성7. 검토

인스턴스 시작 검토

on Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.2, Glibc 2.26, Binutils 2.29.1, and the software packages through e

바이스 유형: ebs가상화 유형: hvm

ECU	
Variable	

launch-wiza
launch-wiza

정보

인스턴스

네트워크 성능
Low to Moderate

설명 ⓘ

인스턴스

취소

기존 키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

새 키 페어 생성

키 페어 이름

owen

키 페어 다운로드

...

계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.

취소

인스턴스 시작

SSH Key



SSH Client

Private Key

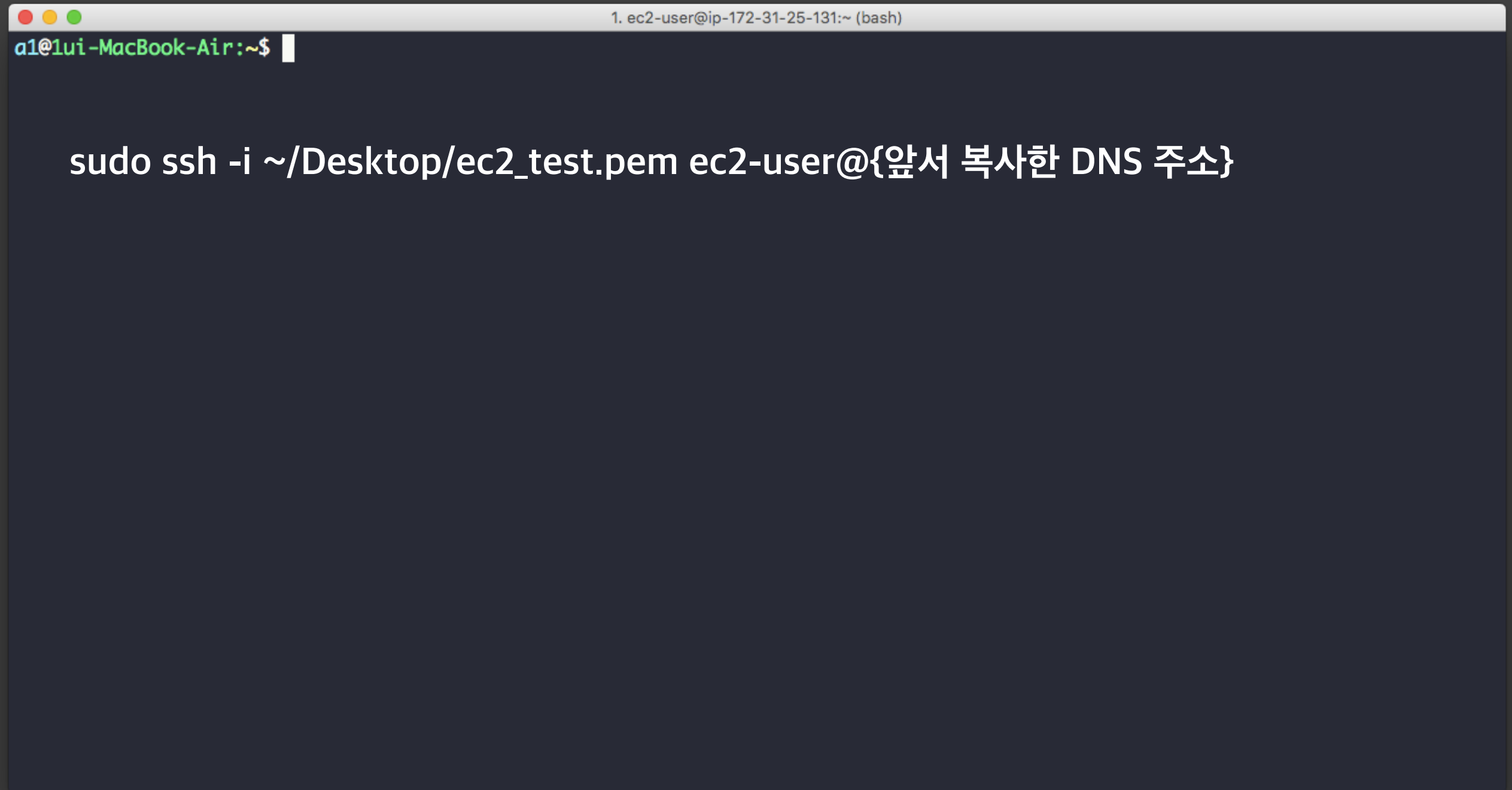


SSH Server

Public Key

- 비밀번호보다 높은 수준의 보안을 필요할 때 사용
- 서버에 저장된 Public Key와 사용자의 Private Key를 비교하여 확인됐을 때만 접속을 허용

SSH

A terminal window with a dark blue background. The title bar shows three colored window control buttons (red, yellow, green) on the left and the text "1. ec2-user@ip-172-31-25-131:~ (bash)" on the right. The terminal content shows a prompt "a1@1ui-MacBook-Air:~\$" followed by the command "sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}" with a white cursor at the end of the command.

```
a1@1ui-MacBook-Air:~$ sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}
```

SSH

```
1. ec2-user@ip-172-31-25-131:~ (bash)
a1@1ui-MacBook-Air:~$
```

sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}

- sudo
 - root(관리자) 권한으로 실행

SSH

1. ec2-user@ip-172-31-25-131:~ (bash)

a1@1ui-MacBook-Air:~\$

```
sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{앞서 복사한 DNS 주소}
```

- id@{ip/dns}
 - ssh로 접속할 계정 ID
 - 접속할 인스턴스의 ip 혹은 dns

EC2를 생성하고 SSH 클라이언트로 접속하여 해당 VM에
웹 서버를 설치 후 외부에서 확인하였습니다.

EC2를 생성하고 SSH 클라이언트로 접속하여 해당 VM에
웹 서버(NginX)를 설치 후 외부에서 확인하였습니다.

Web Server

- HTTP 요청을 받고 응답하는 역할을 함
- HTTP transaction을 처리하고 정적인 파일(static file - HTML, CSS, Image 등)을 요청한 대상에게 보내준다
- Application logic을 직접 처리할 수 없어 WAS(Web application server)와 같이 사용한다
- 대표적으로 Apache, NginX와 같은 제품이 있다

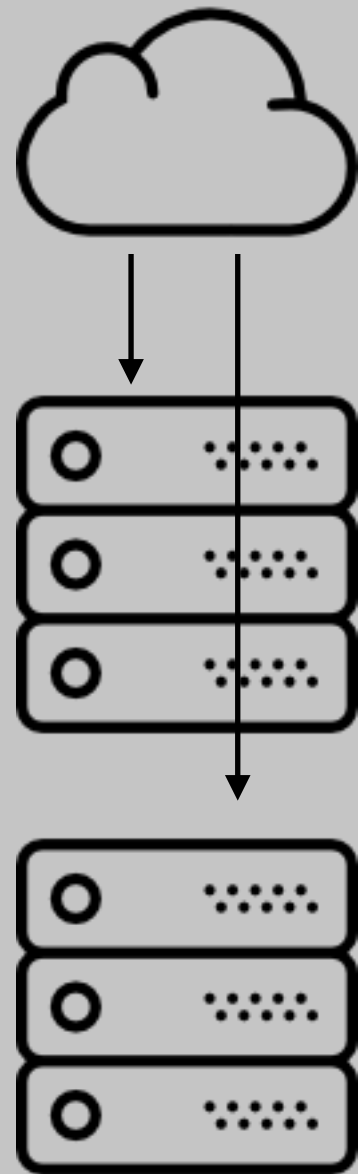
NginX

- Web server software
- Apache HTTPd의 한계를 극복하기 위해 나옴
- 동시 접속 10,000개 당 평균 memory 점유율 2.5mb
- 웬만하면 쓰세요 ~~윈도우 서버면 HS를 써야겠지만~~

WAS(Web Application Server)

- 동적으로 데이터를 생성하여 응답하는 서버
- Application(code) logic을 직접 실행하는 서버(JSP, Java, etc...)
- Tomcat, jBoss, WSGI 등
- 개발하고 있는 언어와 환경에 맞는 WAS를 구축합시다

Web Server / WAS

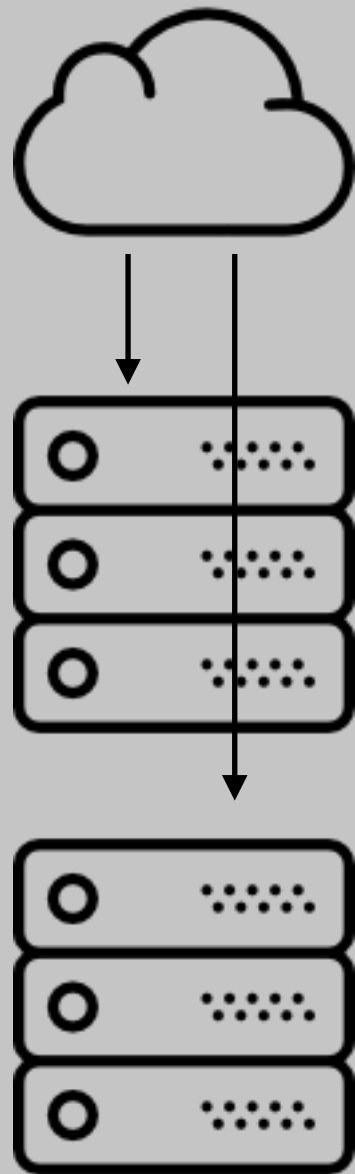


Client

Web Server

WAS

Web Server / WAS

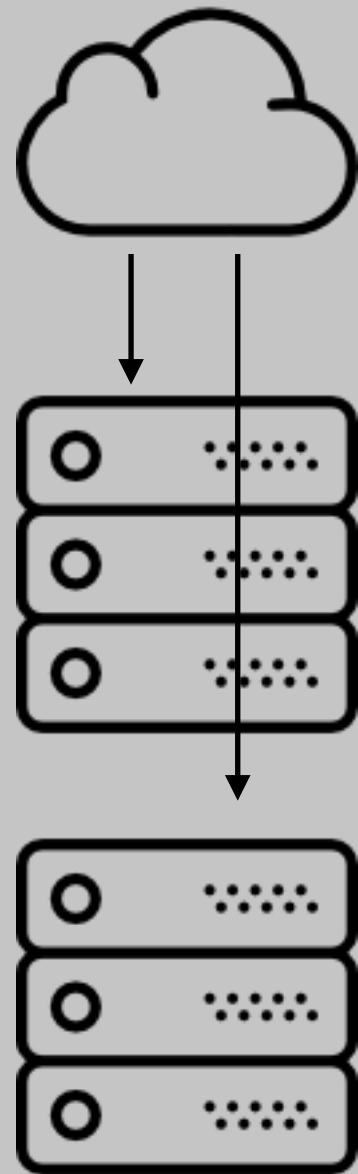


Web browser, Mobile, Etc...

Web Server

WAS

Web Server / WAS

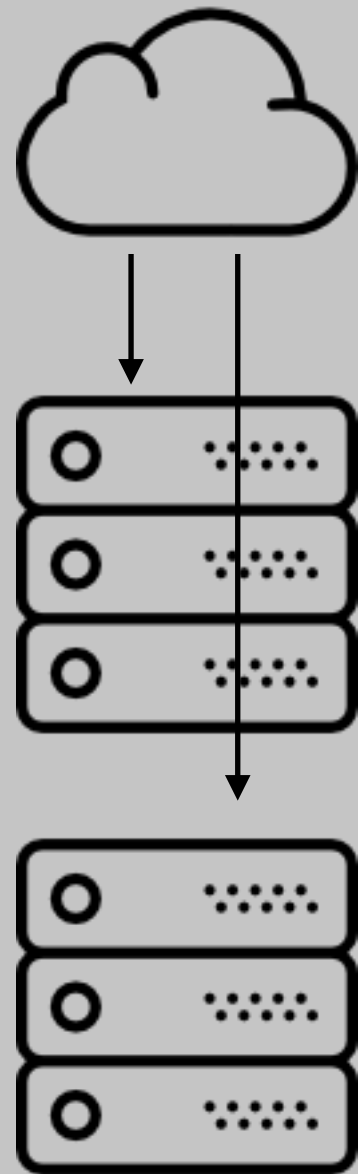


Client

NginX, Apache HTTPd

WAS

Web Server / WAS



Client

Web Server

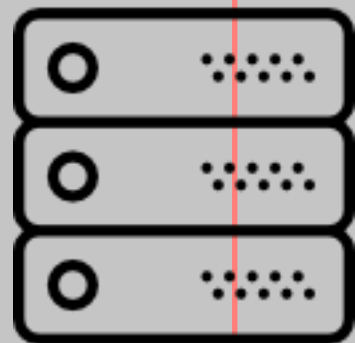
Tomcat, jBoss, Etc...

Web Server / WAS

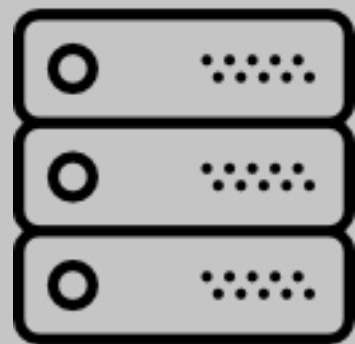


Client

HTTP 요청 / 응답



Web Server



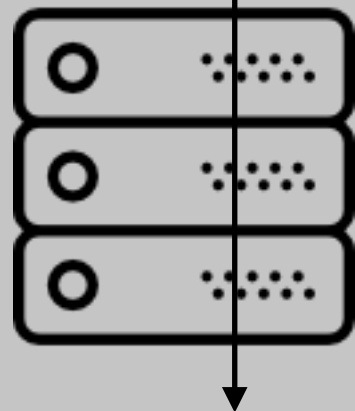
WAS

Web Server / WAS

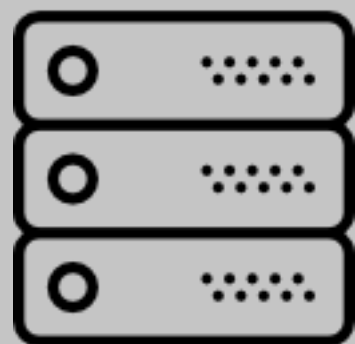


Client

Static file(HTML, Image, etc...), reverse proxy
Traffic balancing(Load balancing), Caching, ...

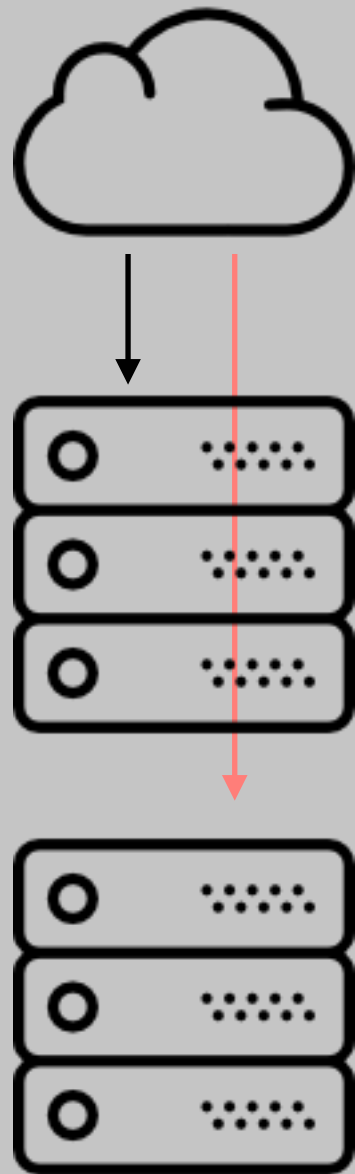


Web Server



WAS

Web Server / WAS



Client

Web Server

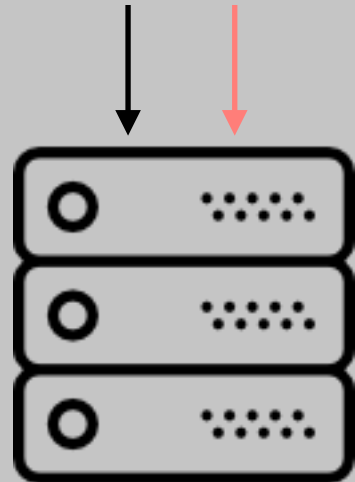
동적 데이터(DB 정보, Session, etc...)

WAS

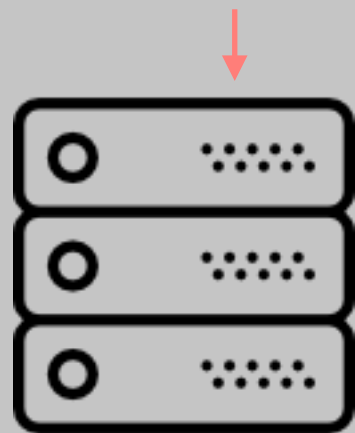
Web Server / WAS



Client



Web Server



WAS

Web server에서 캐싱되지 않은 데이터 요청 시

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

EC2를 생성하고 웹 서버를 설치 후 웹 브라우저를 통해 DNS:80포트에서 확인하였습니다.

DNS / Port

aws

서비스 ▾ 리소스 그룹 ▾

owen ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 [자세히 알아보기](#)

보안 그룹 할당: ☒ 새 보안 그룹 생성 ☐ 기존 보안 그룹 선택

보안 그룹 이름:

설명:

유형 ⓘ	프로토콜 ⓘ	포트 범위 ⓘ	소스 ⓘ	설명 ⓘ
SSH ▾	TCP	22	사용자 지정 ▾ 0.0.0.0/0	예: 관리자 데스크톱용 SSH ×
HTTP ▾	TCP	80	사용자 지정 ▾ 0.0.0.0/0, ::/0	예: 관리자 데스크톱용 SSH ×

⚠ 경고

소스가 0.0.0.0/0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다.

취소

이전

검토 및 시작

DNS

- 도메인 네임 서비스
- 모든 디바이스(인터넷에 연결이 가능한)는 IP(ex. 192.10.x.x)로 통신합니다
- IP를 외우기 귀찮으니 IP와 URL을 매핑하여 전화번호부같은 역할을 합니다

Port

- TCP/IP로 접속 시 마지막 종단점
- 클라이언트 실행 시 동적으로 Port 번호가 부여됨
- 해당 인스턴스에서 실행되는 클라이언트의 구분을 위해 사용
- 일반적으로 SSH(22번), HTTP(80번), WAS(8080번)을 부여
- 도메인(URL)로 접속 시 기본적으로 80번 포트를 사용(생략 가능)

EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

Amazon Linux2를 기반으로 하는 AWS AMI를 활용하여
EC2를 생성하고 웹 서버를 설치 후 외부에서 확인하였습니다.

Amazon Linux2를 기반으로 하는 AWS AMI를 활용하여 EC2를 생성하고 SSH 클라이언트로 접속하여 해당 VM에 웹 서버(NginX)를 설치 후 외부에서 확인하였습니다.

Amazon Linux2를 기반으로 하는 AWS AMI를 활용하여 EC2를 생성하고 SSH 클라이언트로 접속하여 해당 VM에 웹 서버(NginX)를 설치 후 웹 브라우저를 통해 DNS:80포트에서 확인하였습니다.

Docker 설치, 실행

1. ec2-user@ip-172-31-25-131:~ (bash)

a1@1ui-MacBook-Air:~\$

```
sudo ssh -i ~/Desktop/ec2_test.pem ec2-user@{접속할 인스턴스 DNS 주소}
```

```
sudo su ↵
```

```
amazon-linux-extras install docker ↵
```

```
service docker start ↵
```

```
docker pull nginx:latest ↵
```

1. ec2 인스턴스에 ssh를 통해 접속
2. 관리자 권한으로 실행
3. amazon-linux-extras 레포지토리를 활용해 docker 설치
4. docker 데몬 실행
5. dockerhub에서 nginx 이미지 다운로드

Docker 설치, 실행

1. ec2-user@ip-172-31-25-131:~ (bash)

a1@1ui-MacBook-Air:~\$

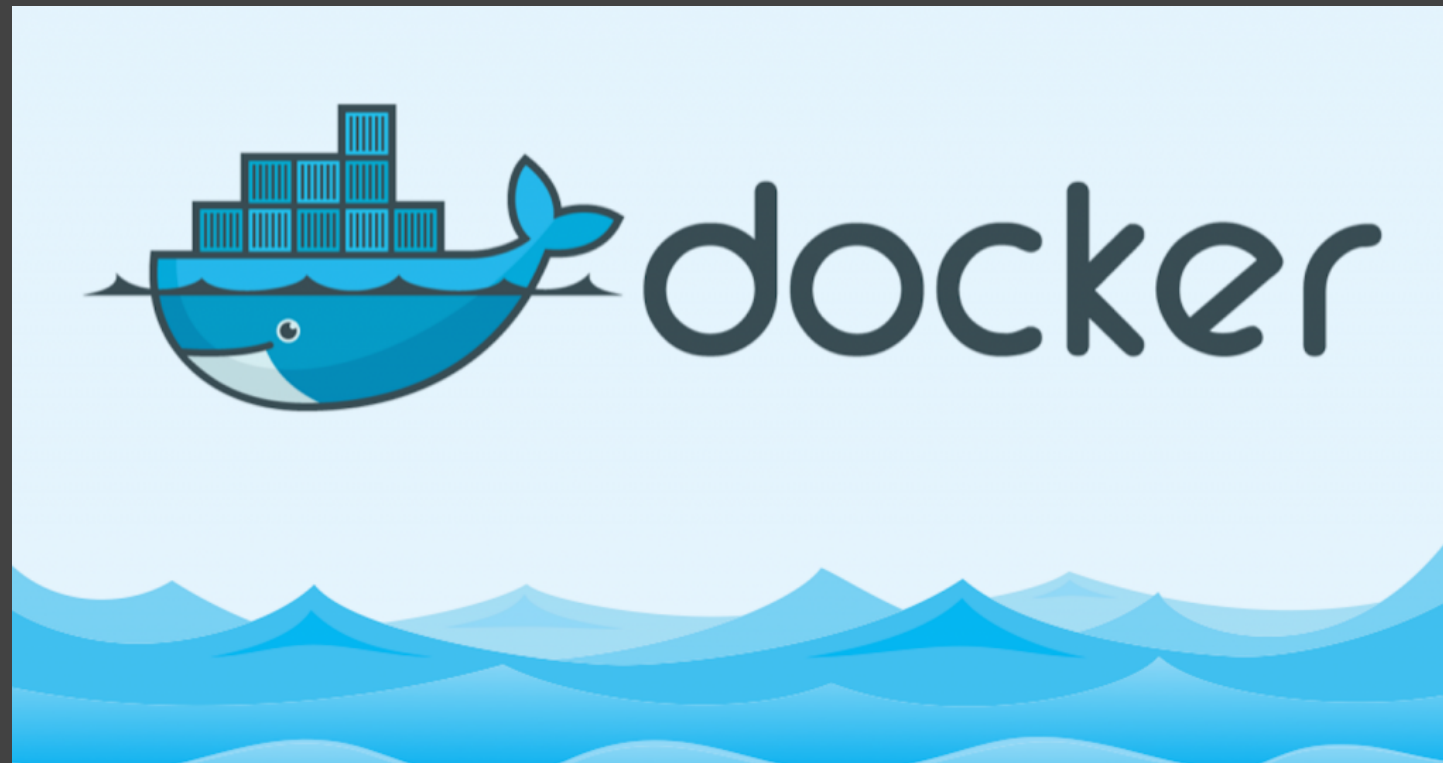
`docker images` ↵

`docker run -d -p 8080:80 --name NginX nginx:latest` ↵

`docker ps -a` ↵

1. 현재 로컬에 저장된 docker image 조회
2. nginx:latest 이미지를 기반으로 docker container 실행
3. 실행되고 있는 컨테이너 프로세스 확인

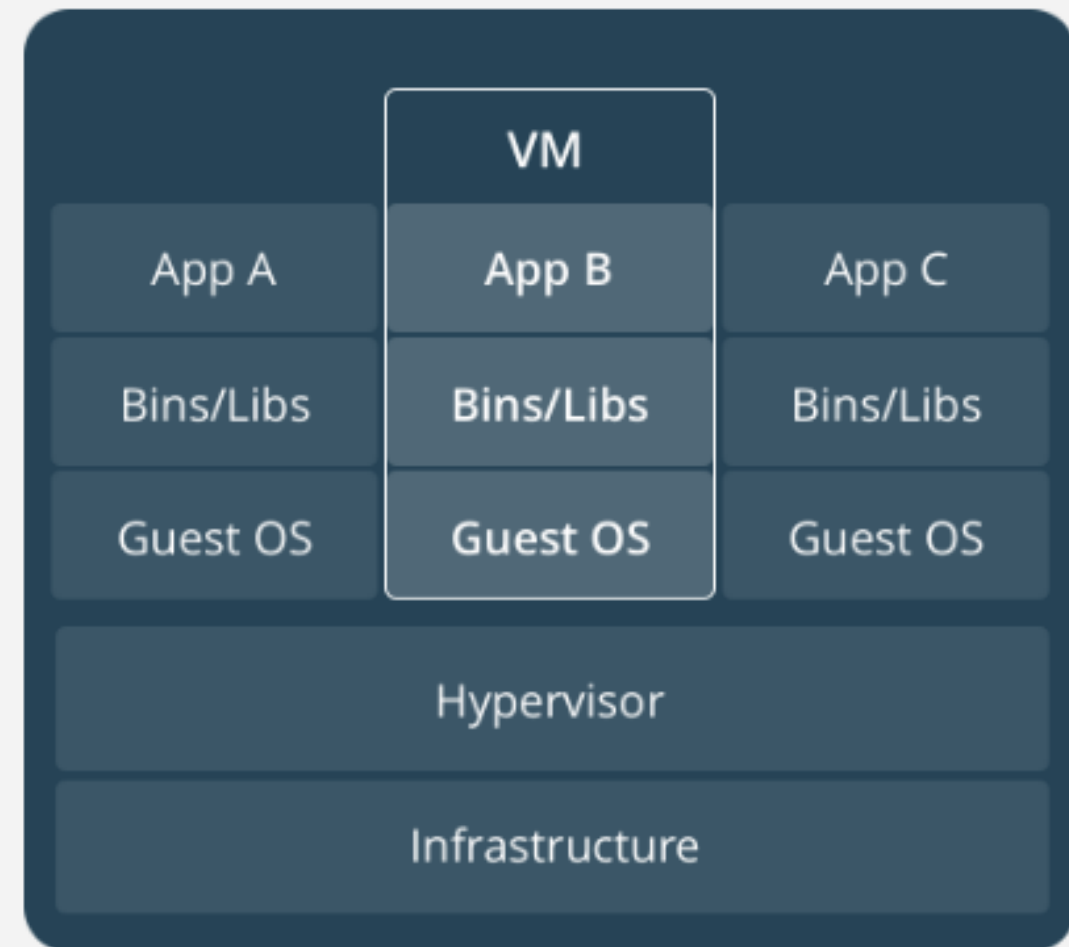
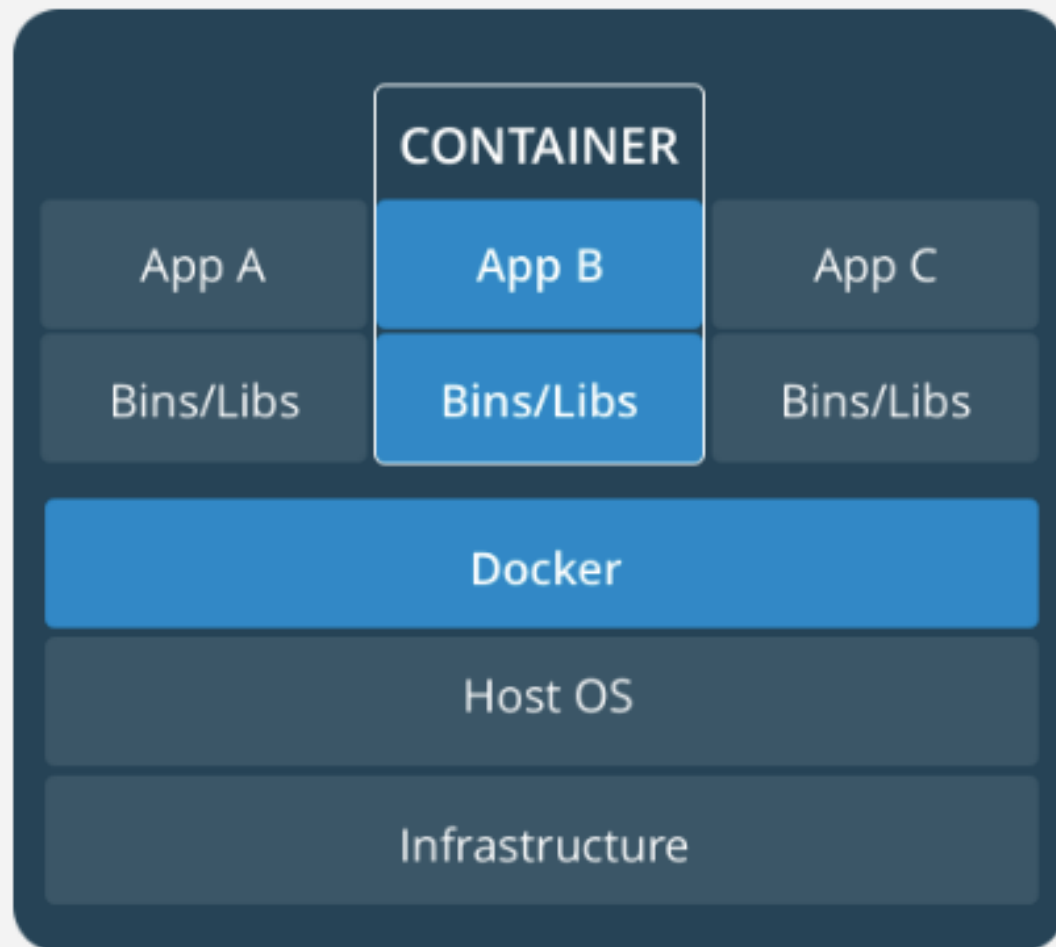
Docker 설치, 실행



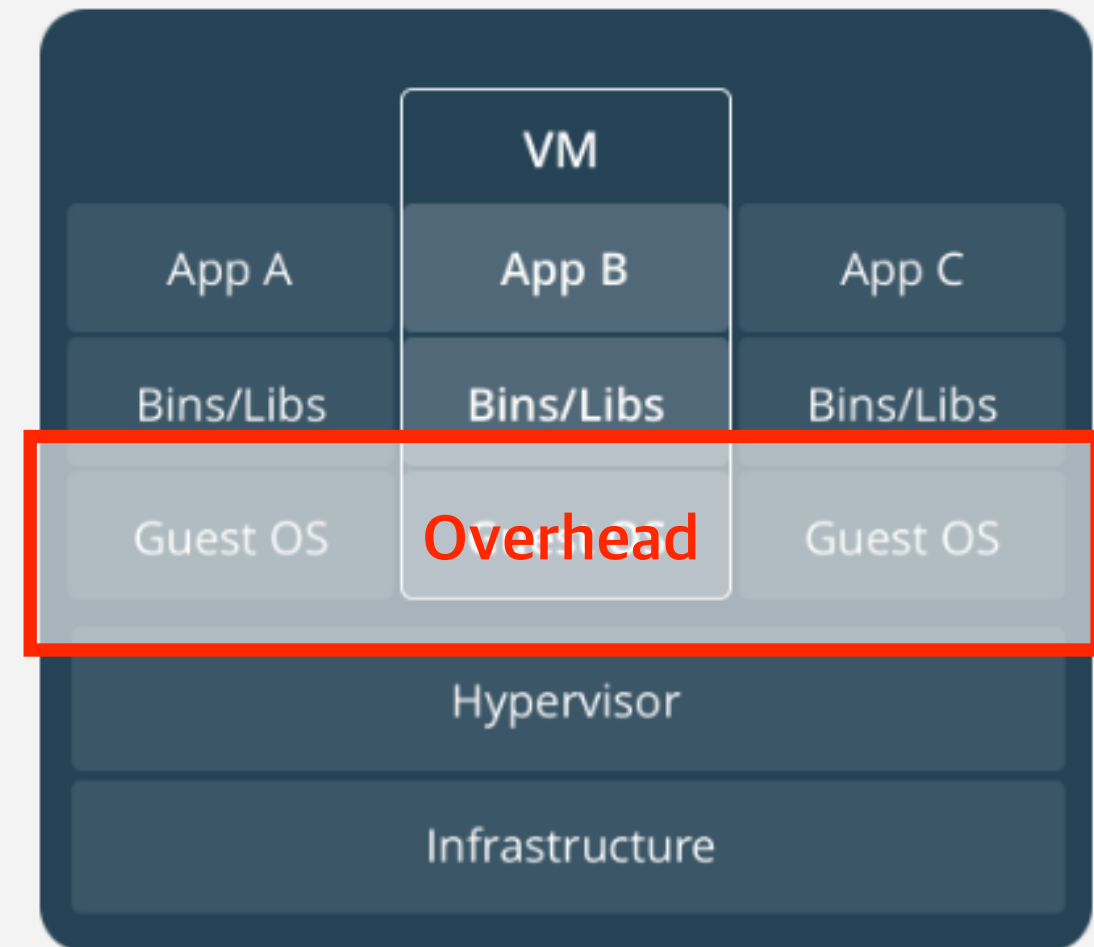
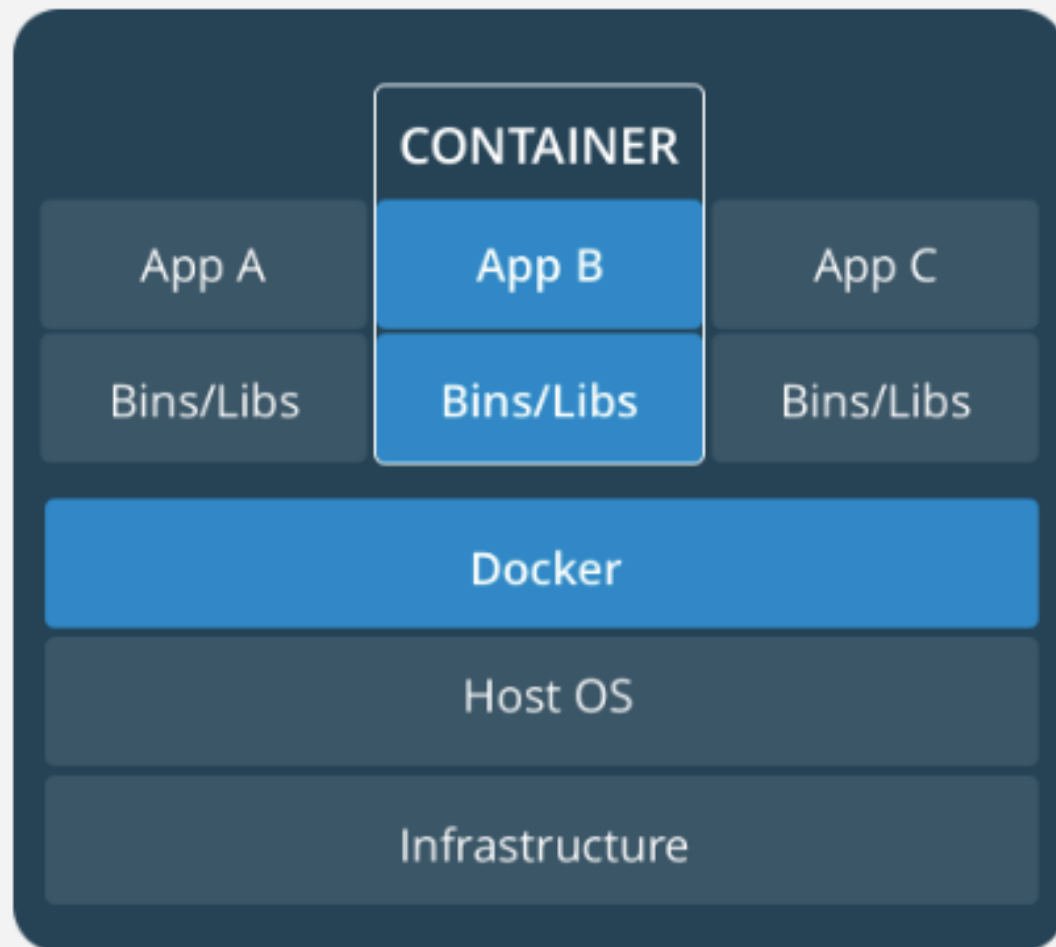
- 컨테이너 기반의 오픈소스 가상화 플랫폼
- 다양한 프로그램, 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 단순화 할 수 있음
- OS(Host)위에 또 다른 OS(Guest)를 올려야 하는 비효율적인 가상 머신의 단점을 극복하기 위해 나온 가상화 어플리케이션

참고 <https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html>

Container vs Virtual Machine



Container vs Virtual Machine



1주차 과제

1. Vagrant로 VM 환경 구성 - Cent OS, Ubuntu

2. Docker 환경에서 Cent OS, Ubuntu 설치해보기