Intro to Wheel Seminar

nenw

@ SPARCS 20' Summer Wheel Seminar

Wheel이 되기 위한 세미나에 참여하는 여러분을 환영합니다!

휠

> 휠 그룹은 일부 유닉스 시스템에서 su 또는 sudo 명령에 대한 액세스를 제어하기 위해 사용되는 특수 사용자 그룹입니다.

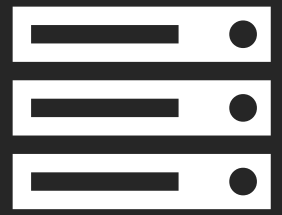
– Wikipedia, Wheel (computing)

휠

스팍스 권력의 한 축

휠 없이 플젝 못 띄운다

휠 없으면 서버 못 건드려~



물리서버를 만질 수 있다

마 니 물리서버 만져봤나

루트권한을 휘두를 수 있다

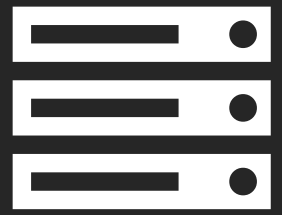
휠 밖에 못 써 ^^

휠

스팍스 권력의 한 축 (실제)

휠 없이 플젝 못 띄운다

플젝을 위해서 일해야한다



물리서버를 만질 수 있다

서버는 시험기간과 플젝 듀에만 터진다

루트권한을 휘두를 수 있다

그 권한으로 문제를 고쳐야한다

그래도...

휠

- 리눅스 시스템에 대해 기본적인 이해를 갖출 수 있다.

  → 밥먹고 사는데 도움이 될 수도 있다

- **장병규 대표님:** "스팍스 시절 서버 관리를 할 수 있었던 게 실력 향상에 큰 도움"
- **남세동 대표님:** "서버 관리 기술은 한번 배우면 수십년 쓸 수 있다"

휠 세미나

그 휠이 되기 위해 서버관리의 기본적인 것을 가르치는 세미나

휠 세미나 / 일정

1주차

Intro	nenw (김요한)
AWS	nenw (김요한)
Linux	jessie (윤지수), ball (최진호)
Shell Script, Cron	nunu (황현빈), kalogon (이건호)

3주차

백업, 비상상황 대비	victory (김주연)
Security	nunu (황현빈)
LDAP, pGina	victory (김주연), kalogon (이건호)
Network	nenw (김요한)
마무리	nenw (김요한)

2주차

File System, Memory Swap	jessie(윤지수)
NFS, FTP	ball (최진호)
Database	hyuk (장승혁), panya (김지연)
DNS	hyuk (장승혁)
가상화, Docker	panya (김지연)

주의사항

- **발표 준비**: 1개 실습, #wheel-seminar에 하루 전에 자료 업로드
- **실습 제출 기한**: 발표 다음 날 오후 23시 59분까지
- **지각 규칙**: 5분까지는 해당 없음, 이후 지각 (0.5 경고)
- **결석 규칙**: 불참 (1 경고), 실습 미제출 (1 경고)
- **이수 기준**: 3 경고 미만

아마존 웹 서비스

nenw

@ SPARCS 20' Summer Wheel Seminar

클라우드

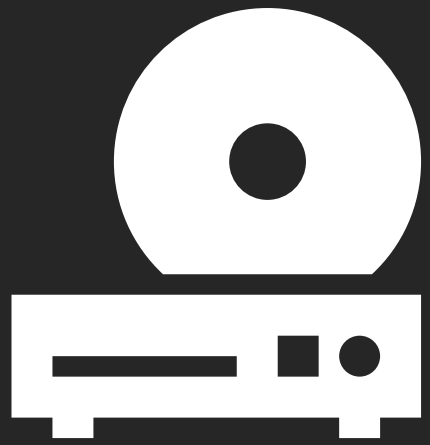
클라우드

클라우드

- 왜 많고 많은 단어 중 구름일까?

- 구름
  - 어디서든 보인다.
  - 나와는 멀리 떨어져있다.

클라우드

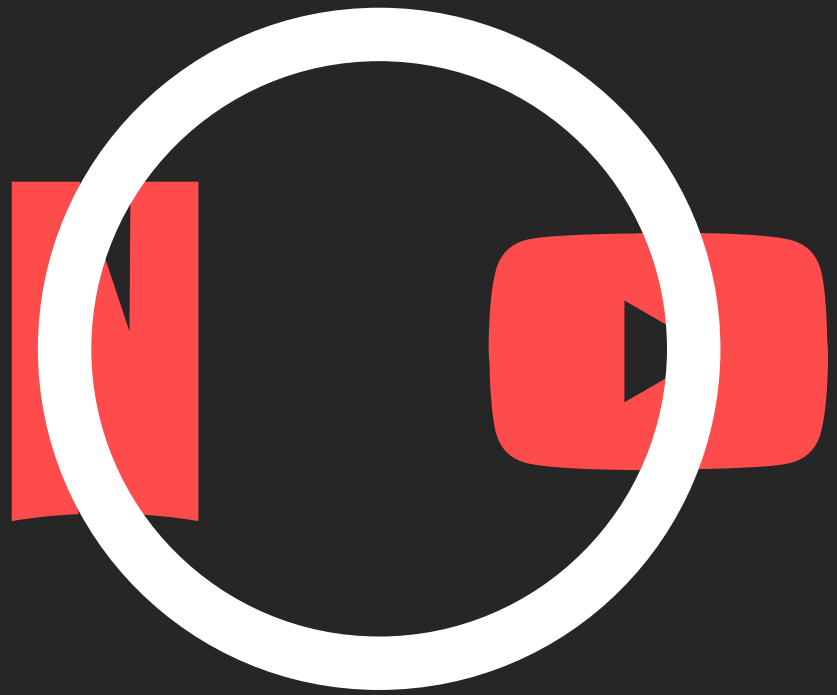


클라우드

구름은 들고 다닐 수 없다

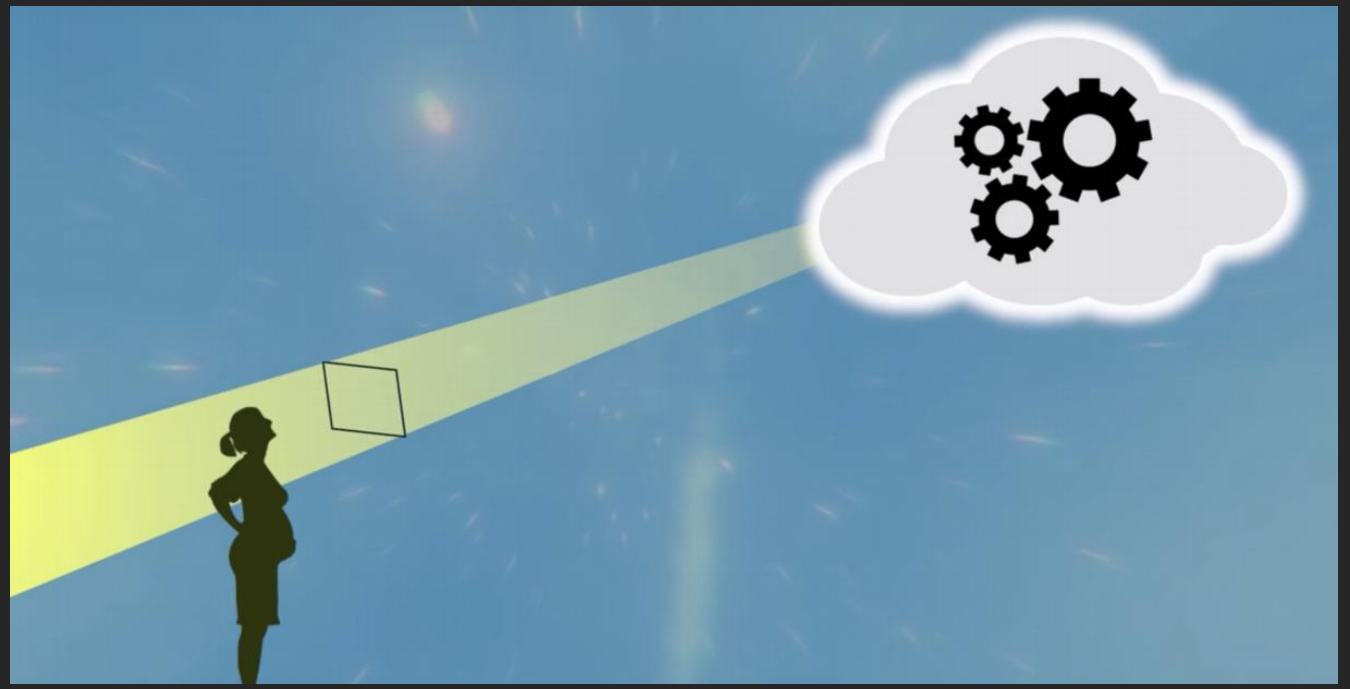
클라우드

클라우드



어디서든 사용할 수 있다

클라우드



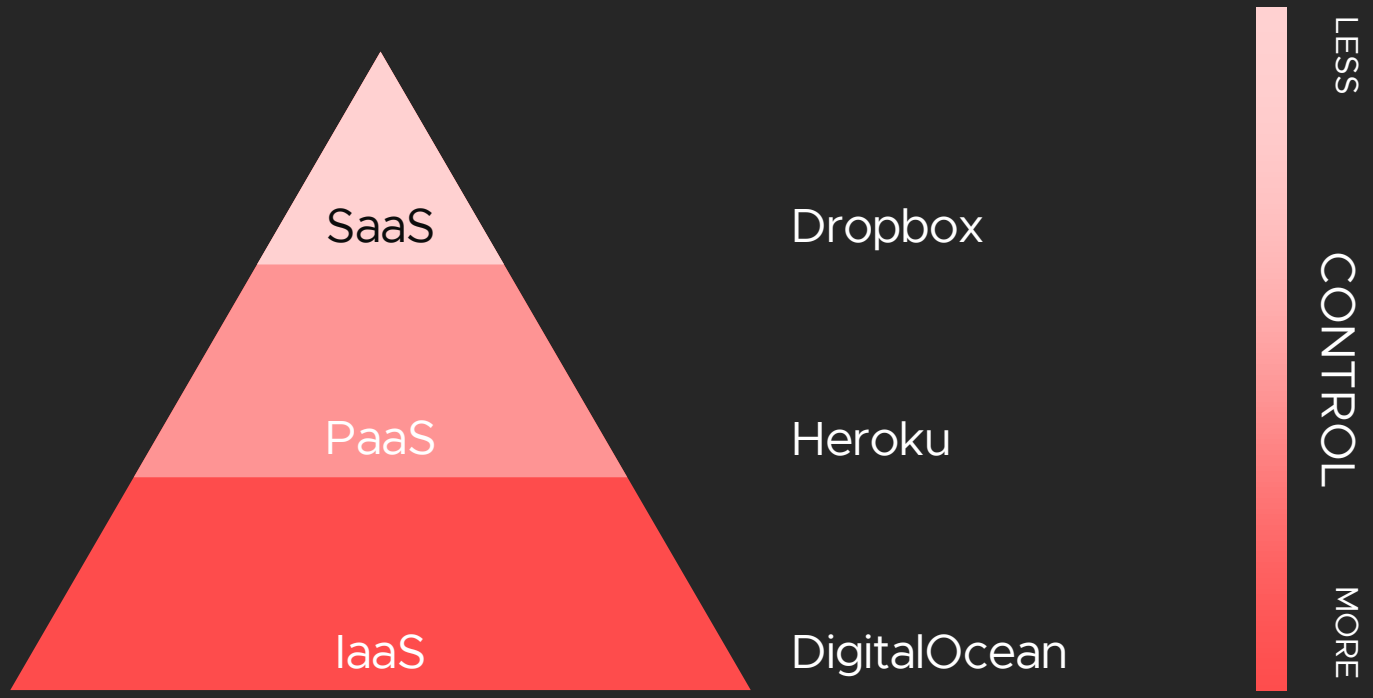
처리는 구름 속에서
사용자에게는 완성된 그림만

클라우드 / 장점

- 직접 무언가를 관리할 필요가 없다.
  - 서버실에 있는 비상전력발생기도 필요가 없고, 맨날 고장나는 하드를 교체해줄 필요도 없다.
- 상황에 유연하게 대처할 수 있다.
  - 사용자가 늘어난다고 새로 장비를 준비하고 할 필요가 없다.
- 장소의 제약이 없다.
  - 내가 인터넷이 불안정한 시골에 있어도 서버는 데이터센터에서 잘 돌아가고 있으니 상관이 없다.

네오위즈가 데이터센터를 폐쇄하고 클라우드로 전면 이전하는 데에는 이유가 있는 셈.

클라우드 / 종류


SaaS
PaaS
IaaS
Dropbox
Heroku
DigitalOcean
LESS
CONTROL
MORE

클라우드 / SaaS

- Software as a Service
- 응용 소프트웨어를 제공
- 사업자가 Dropbox나 Gmail과 같은 애플리케이션을 제공하고, 사용자는 이를 바로 사용

클라우드 / PaaS

- Platform as a Service
- 개발 및 배포 환경을 제공
- 빌드, 테스트, 배포, 업데이트 등의 수명 주기를 쉽게 관리할 수 있음

클라우드 / IaaS

- Infrastructure as a Service
- 하드웨어와 같은 컴퓨팅 자원을 제공
- 물리 서버를 구매 및 유지할 필요 없이 필요한 만큼, 즉시 사용 가능
- 네트워크 트래픽 변동, 하드웨어 장애 등 소규모 개발팀으로 해결하기 어려운 문제들은 대신 처리해줌

클라우드 / IaaS

- 즉시: 서버 컴퓨터를 구매하고, 기다리고 할 필요 없이 클릭 몇 번으로 바로 서버 생성 가능
- 필요한 만큼: 사용자가 많아지면 인스턴스 크기를 키우면 됨 (새로 서버 컴퓨터를 구매할 필요 없음)
- 안정성: 관리를 다 해주니 하드가 망가지거나 시스템이 과열로 죽거나 하는 걸 걱정할 필요 없음

아마존 웹 서비스

AWS

- Amazon Web Service
- 아마존이 제공하는 세계 1위 점유율을 가진 클라우드
- 아마존의 돈줄 (아마존 전체 순익의 50% 이상)

AWS 제품 살펴보기

분석 | 애플리케이션 통합 | 증강현실 및 가상현실 | AWS 비용 관리 | 블록체인

비즈니스 애플리케이션 | 컴퓨팅 | 컨테이너 | 고객 인게이지먼트 | 데이터베이스

Amazon Aurora
관리형의 고성능 관계형 데이터베이스

Amazon DynamoDB
관리형 NoSQL 데이터베이스

Amazon DocumentDB(MongoDB 호환)
완전관리형 문서 데이터베이스

Amazon ElastiCache
인 메모리 캐싱 시스템

Amazon Keyspaces(Apache Cassandra용)
관리형 Cassandra 호환 데이터베이스

Amazon Neptune
완전관리형 그래프 데이터 서비스

Amazon Quantum Ledger Database(QLDB)
완전관리형 원장 데이터베이스

Amazon RDS
MySQL, PostgreSQL, Oracle, SQL Server 및 MariaDB를 위한 관리형의 관계형 데이터베이스 서비스

Amazon RDS on VMware
온프레미스 데이터베이스 관리 자동화

Amazon Redshift
빠르고 간단하며 비용 효과적인 데이터 웨어하우징

Amazon Timestream
완전관리형 시계열 데이터베이스

AWS Database Migration Service
최소한의 다운타임으로 데이터베이스 마이그레이션

개발자 도구 | 최종 사용자 컴퓨팅 | 게임 기술 | 사물 인터넷 | Machine Learning

관리 및 거버넌스 | 미디어 서비스 | 마이그레이션 및 전송 | 모바일 | 네트워킹 및 콘텐츠 전송

양자 기술 | 로보틱스 | 인공위성 | 보안, 자격 증명 및 규정 준수 | 스토리지

(160+)

저희가 쓸 것 몇개만 배워봅시다...

컴퓨팅

- EC2
  - 가상화된 컴퓨터를 빌려줌
- Lambda
  - 코드를 업로드해서 어떤 일이 있으면 (트리거) 그 코드를 실행하게 함

컴퓨팅 / EC2

- Elastic Computing Cloud
- 새 서버 인스턴스 만들기
- 시간단위 가격 (+ 트래픽)

컴퓨팅 / EC2 / 가격정책

- 온 디맨드
  - 사용한 만큼 과금
- 스팟 인스턴스
  - 현재 남는 서버를 싸게, 단기간 빌리기
- 예약 인스턴스
  - 1년(또는 3년)치의 서버를 한번에 싸게 빌리기

스토리지

- EBS
  - EC2에 달 수 있는 드라이브
- S3
  - 공개적으로 접근하게도 할 수 있는 파일 저장소
- S3 Glacier
  - 데이터를 가져올 때는 오래 걸리지만, 훨씬 싼 파일 저장소

스토리지 / S3

- 객체 스토리지 서비스
  - 객체: 데이터(파일) + 메타데이터
- 액세스 제어 가능
  - 이 객체는 누구나 접근 가능, 이 객체는 저 계정으로는 삭제 불가 등
- EBS보다 싸다

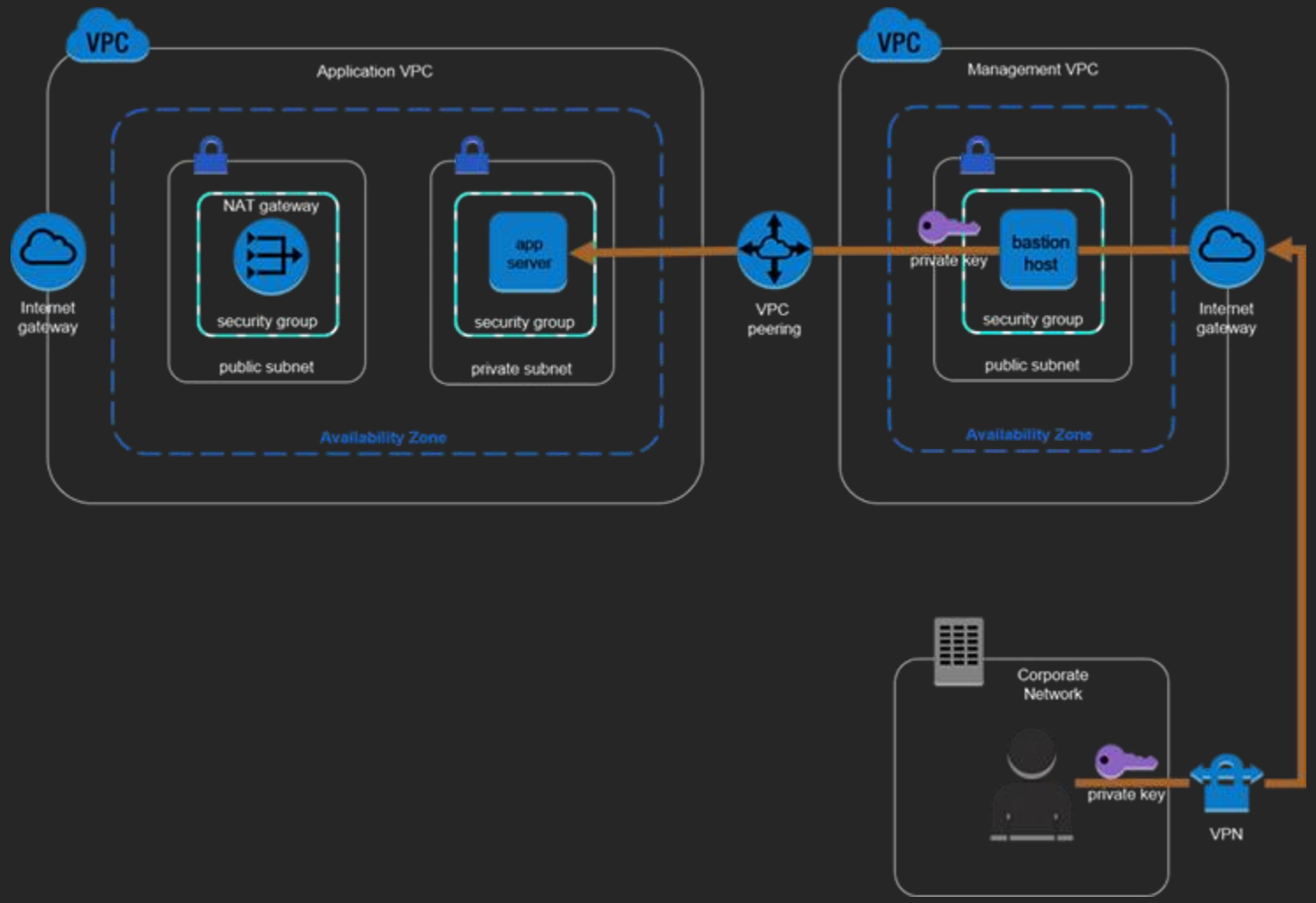
데이터베이스

- RDS
  - MySQL, PostgreSQL 등 관계형 DB
- Document DB
  - MongoDB와 같은 도큐먼트 DB
- DynamoDB
  - Key-Value DB
- ElasticCache
  - Redis, Memcached 등 In-Memory DB

네트워크

- CloudFront
  - CDN 서비스
- VPC
  - 내부 네트워크 구성
- Route53
  - 도메인 관리

네트워크 / VPC


VPC
Application VPC
NAT gateway
app server
security group
security group
public subnet
private subnet
Availability Zone
Internet gateway
VPC peering
VPC
Management VPC
private key
bastion host
security group
public subnet
Availability Zone
Internet gateway
Corporate Network
private key
VPN

그 외

- CloudWatch
  - AWS에 올린 어플리케이션 상태 모니터링
- ECS
  - 도커 컨테이너 관리
- IAM
  - 접근 권한 및 계정 관리

아마존 웹 서비스 실습

실습 순서

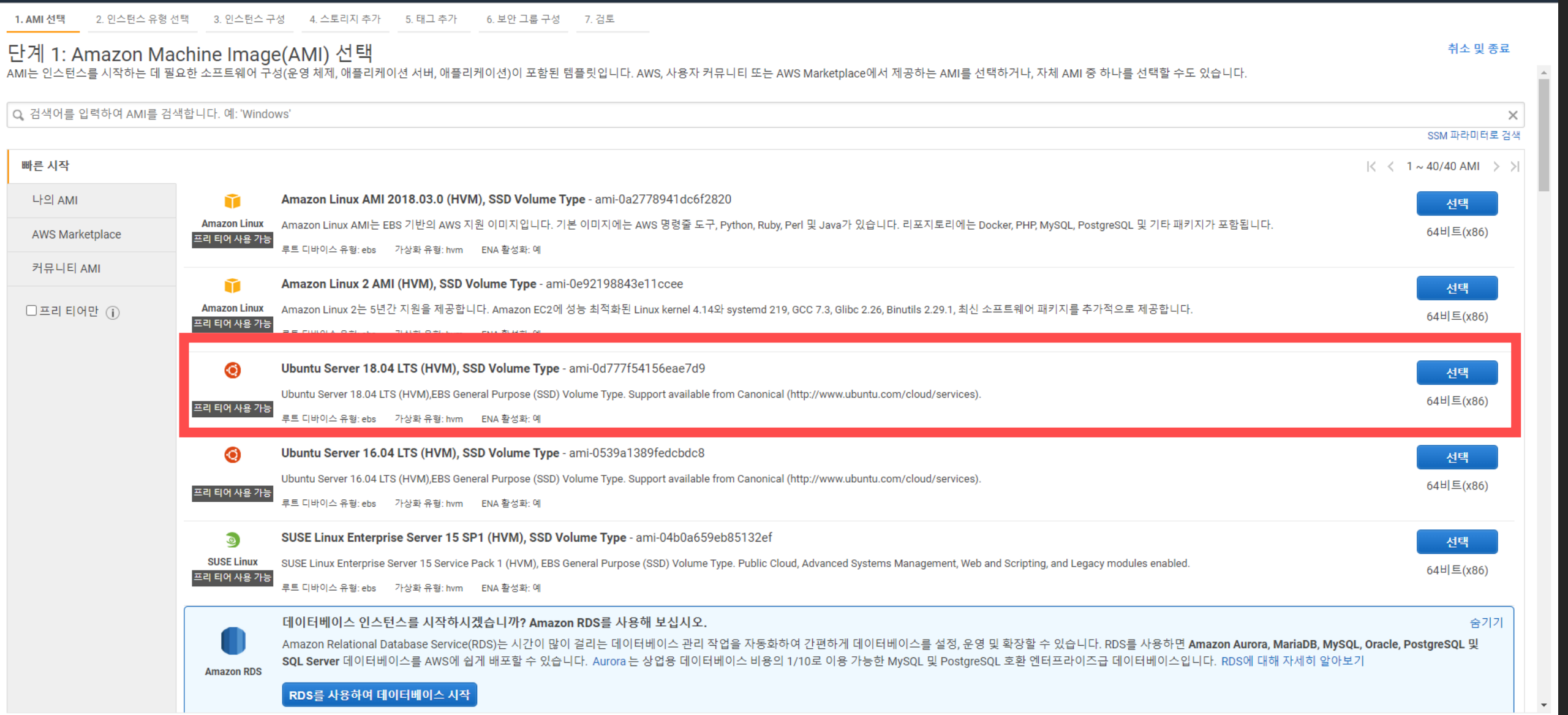
1. IAM 유저 만들기
2. EC2 + S3로 간단한 애플리케이션 배포해보기

실습 / IAM

1. 서비스 > IAM > IAM 사용자 로그인 링크 사용자 지정 후 마음에 드는 주소로 바꾸기
2. IAM 그룹 생성하기
    1. 이름: ConsoleAdmin
    2. 권한: AmazonEC2FullAccess, AmazonDocDBFullAccess, AmazonS3FullAccess
3. 그룹 하나 더 생성하기
    1. 이름: S3Admin
    2. 권한: AmazonS3FullAccess
4. IAM 계정 생성하기
    1. 이름: (원하는 이름)
    2. 액세스 유형: AWS Management Console
5. 계정 하나 더 생성하기
    1. 이름: S3Admin
    2. 액세스 유형: 프로그래밍 방식 (csv 파일 저장)

실습 / EC2

서비스 > EC2 > 인스턴스 시작



서버에서 사용할 이미지 선택

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

단계 2: 인스턴스 유형 선택

Amazon EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스는 애플리케이션을 실행할 수 있는 가상 서버입니다. 이러한 인스턴스에는 CPU, 메모리, 스토리지 및 네트워킹 용량의 다양한 조합이 있으며, 애플리케이션에 사용할 적절한 리소스 조합을 유연하게 선택할 수 있습니다. 인스턴스 유형과 이 인스턴스 유형이 컴퓨팅 요건을 충족하는 방식에 대해 자세히 알아보기.

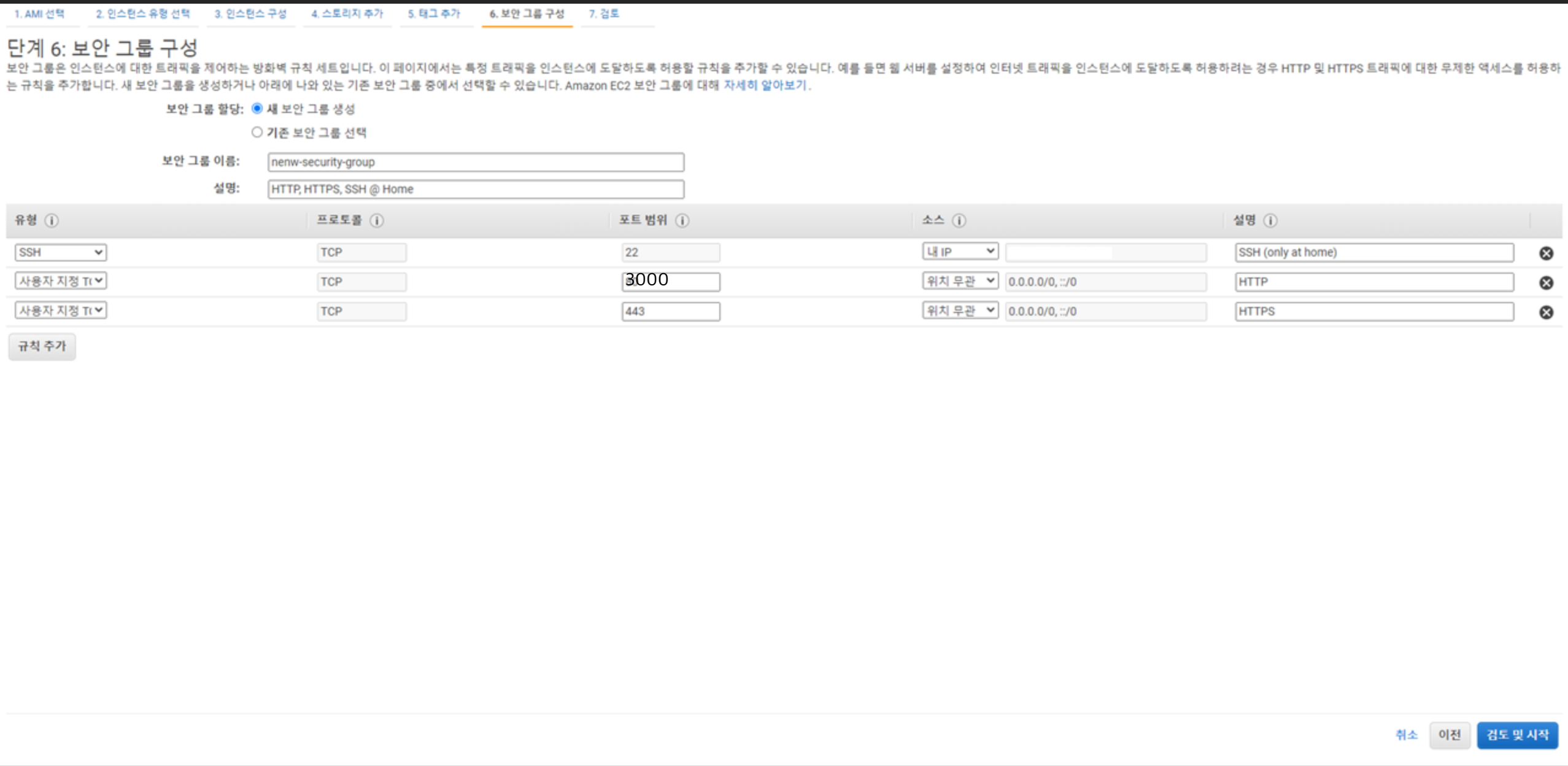
필터링 기준: 모든 인스턴스 유형 | 현재 세대 | 열 표시/숨기기

현재 선택된 항목: t2.micro (Variable ECU, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB 메모리, EBS 전용)

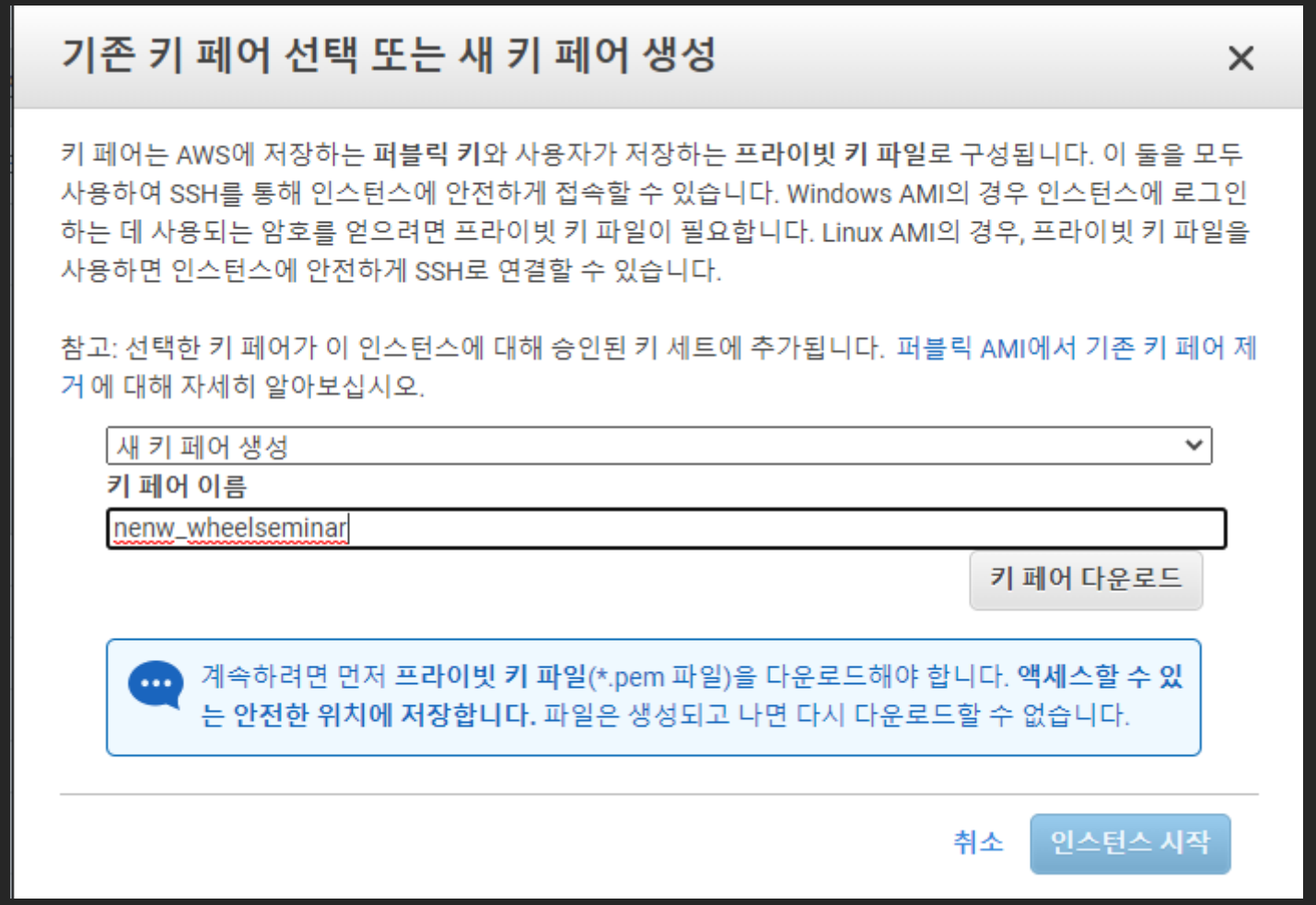
	그룹	유형	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GB)	EBS 최적화 사용 가능	네트워크 성능	IPv6 지원
☑	General purpose	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음에서 중간	예
☐	General purpose	t2.small	1	2	EBS 전용	-	낮음에서 중간	예
☐	General purpose	t2.medium	2	4	EBS 전용	-	낮음에서 중간	예
☐	General purpose	t2.large	2	8	EBS 전용	-	낮음에서 중간	예
☐	General purpose	t2.xlarge	4	16	EBS 전용	-	보통	예
☐	General purpose	t2.2xlarge	8	32	EBS 전용	-	보통	예
☐	General purpose	t3a.nano	2	0.5	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.micro	2	1	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.small	2	2	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.medium	2	4	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.large	2	8	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.xlarge	4	16	EBS 전용	예	최대 5기가비트	예
☐	General purpose	t3a.2xlarge	8	32	EBS 전용	예	최대 5기가비트	예

취소 | 이전 | 검토 및 시작 | 다음: 인스턴스 세부 정보 구성

무료로 쓸 수 있는 t2.micro 씁시다.



어디에서 어디로 접근 할 수 있게 할지 선택

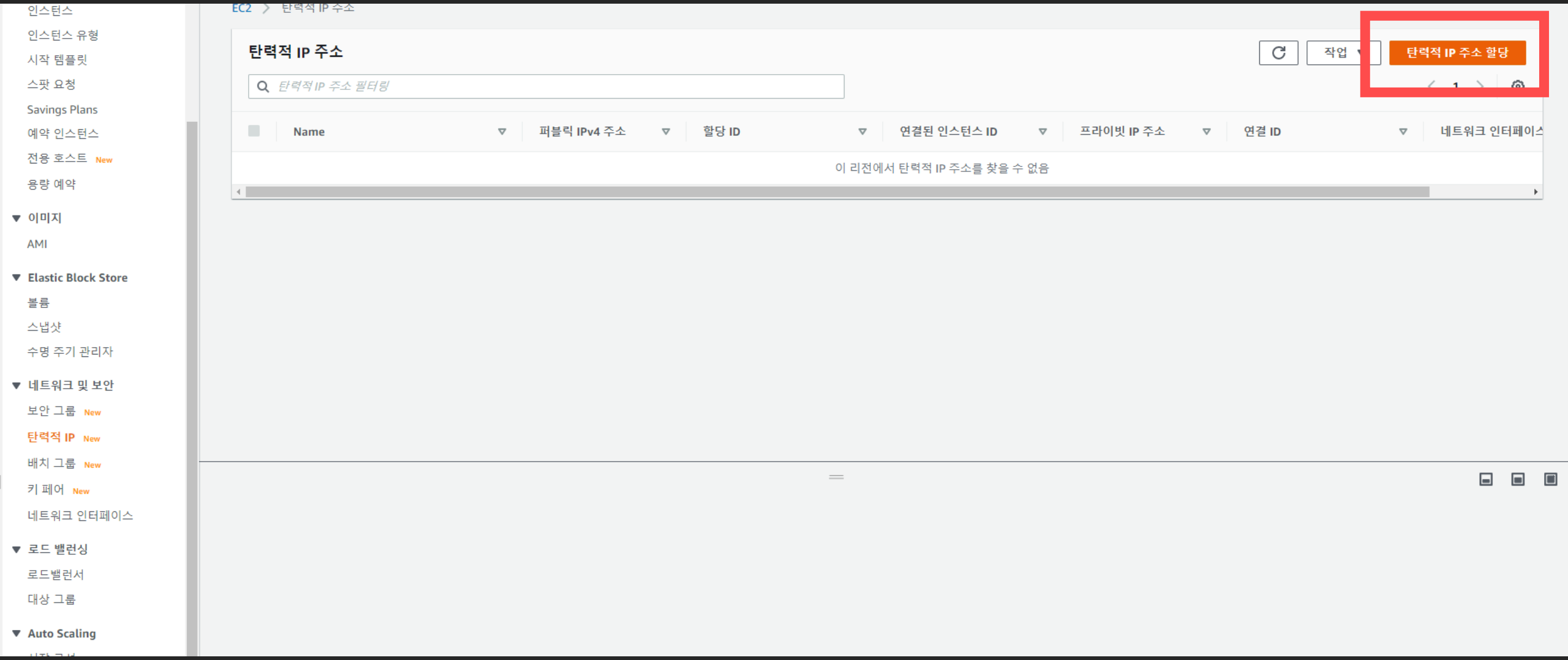


저 파일이 있어야 SSH로 연결할 수 있습니다.

이 IP로 접근하면 되긴 한데...
인스턴스의 상태가 바뀔 때마다 IP가 바뀌니 정적인 IP를 물려줍시다.

퍼블릭 DNS(IPv4)	ec2-13-209-8-197.ap-northeast-2.compute.amazonaws.com
IPv4 퍼블릭 IP	13.209.8.197

서비스 > 탄력적 IP



이후 작업 > 탄력적 IP 연결 > 인스턴스 > (내 인스턴스 선택) > 연결

실습 / 설치 및 백엔드

- SSH 연결하기

```
$ ssh -i (내 SSH 파일) ubuntu@(아까 할당한 IP 주소)
WARNING: UNPROTECTED PRIVATE KEY FILE -> 퍼미션 수정하기
(윈도우: 우클릭 -> 속성 -> 보안, Linux / macOS: 0600)
```

- Node 설치

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
$ source ~/.bashrc
$ nvm install --lts
```

- MongoDB 설치

```
$ wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list
$ sudo apt update
$ sudo apt install mongodb-org
$ sudo systemctl start mongod
```

실습 / 백엔드

- 백엔드 설치

```
$ git clone https://github.com/gothinkster/node-express-realworld-example-app.git my-app
$ cd my-app
$ npm i
$ npm i pm2 --global
$ pm2 start
$ cd ..
```

실습 / 프론트엔드

- 프론트엔드 설치

```
$ git clone https://github.com/gothinkster/vue-realworld-example-app.git my-app-front
$ cd my-app-front
$ npm i
$ npm run build
```

실습 / 프론트엔드

- S3로 배포

```
$ sudo apt update
$ sudo apt install awscli
$ cd ~
$ mkdir .aws
$ vim .aws/credentials

      [default]
      aws_access_key_id=xxx
      aws_secret_access_key=xxx

$ vim .aws/config

      [default]
      region=ap-northeast-2
```

실습 / 프론트엔드

- S3 만들기
  - 서비스 > S3 > 버킷 만들기 > (Unique한 이름) > ... > 퍼블릭 액세스 차단 선택 해제
  - 속성 > 정적 웹 사이트 호스팅 > 인덱스 문서: index.html, 오류 문서: index.html, 리다이렉트 규칙
  - 권한 > 버킷 정책 > 정책 생성기
- 리다이렉트 규칙

```
<RoutingRules>
    <RoutingRule>
        <Condition>
            <KeyPrefixEquals>api/</KeyPrefixEquals>
        </Condition>
        <Redirect>
            <HostName>(내 퍼블릭 DNS):3000</HostName>
        </Redirect>
    </RoutingRule>
</RoutingRules>
```

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect (•) Allow ( ) Deny

Principal *

Use a comma to separate multiple values.

AWS Service Amazon S3 [ ] All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected [ ] All Actions ('*')

Amazon Resource Name (ARN) arn:aws:s3:::nenw-s3/*

ARN should follow the following format: arn:aws:s3:::<bucket_name>/<key_name>.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

GetObject

arn:aws:s3:::(버킷 이름)/*

실습 / 디플로이

- S3로 배포

```
$ cd ~/my-app-front/dist
$ aws s3 sync . s3://nenw-s3
```

- https://(버킷 이름).s3-website.ap-northeast-2.amazonaws.com/

감사합니다