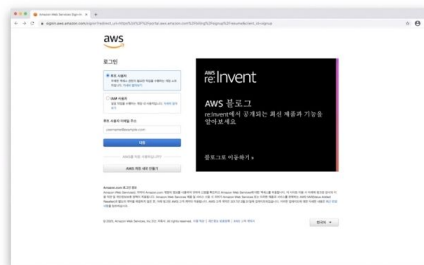
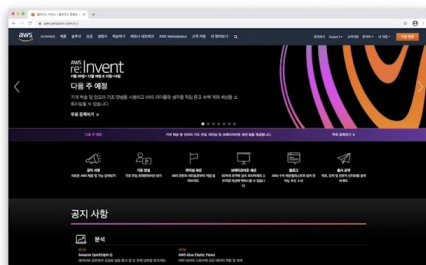


3. Practice: AWS Environment

AWS 계정 가입
EC2 생성 & Security group 설정
탄력적 IP 설정
VS Code로 환경 테스트

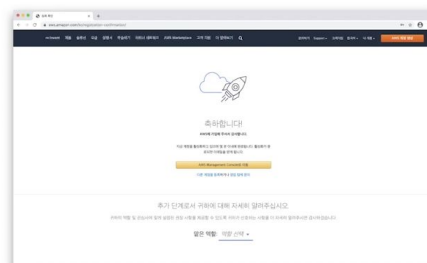
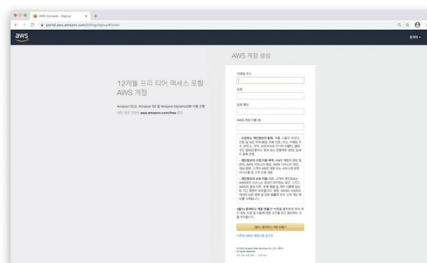
AWS 계정 가입 (1/2)

<https://aws.amazon.com/ko/> 접속



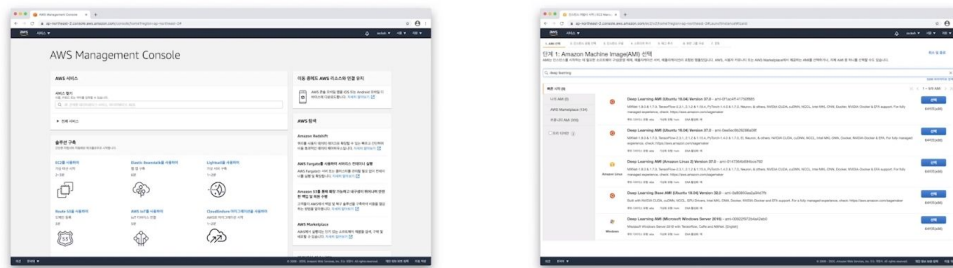
AWS 계정 가입 (2/2)

해외결제 가능한 카드 결제를 등록해야 회원가입이 가능
신규 회원은 프리티어 서비스에 대해 12개월 무료로 사용 가능



EC2 생성 (1/4) – AMI 선택

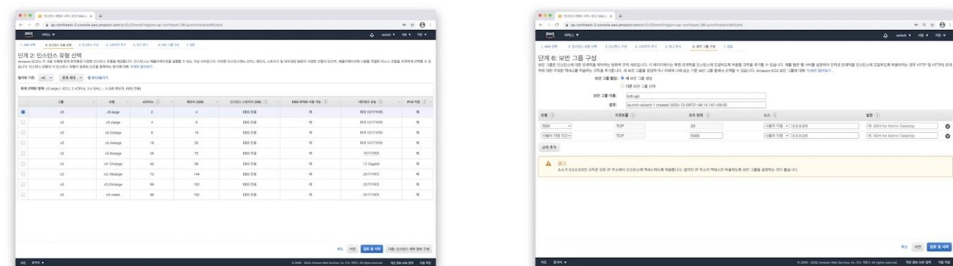
딥러닝 AMI이 설치된 EC2를 생성하여 필요 개발 환경 사전 세팅
검색 부분에 Deep learning 입력



AMI를 통해서 프로비저닝 과정을 빠르게 단축 가능, AMI는 사전 설치된 소프트웨어라고 볼 수 있다. 내가 필요로 하는 AMI를 지정해서 생성된 인스턴스를 사용할때 많은 시간이 절약된다. 특히 딥러닝 같은 경우에는 GPU가속기를 연결하는데 별도의 CUDA나 CUDNN같은 소프트웨어를 가속기와 운영체제에 설치해야 하는데 버전 호환성을 따지는데 상당히 많은 시간과 노력을 기울여야 한다. 하지만 이렇게 AMI를 통해서 인스턴스를 생성하게 되면 이미 아마존에서 어느정도 테스트가 끝난 안정적인 환경으로 세팅이 되기 대며 Pytorch나 tensorflow같은 필수 라이브러리가 자동으로 설치가 되어있다.

EC2 생성 (2/4) – 인스턴스 유형 선택 & 보안 그룹 설정

프리티어인 t2.micro 또는 computing에 최적화된 가장 저렴한 c5.large 인스턴스 유형 선택한 후 “다음: 인스턴스 세부정보 구성” 클릭
“단계6: 보안그룹 구성” 이 나올 때까지 계속 다음 클릭
원격으로 API 서버에 접근/호출할 수 있도록 사용자 설정 포트를 새로 생성 (ex. 포트범위: 5000, 소스 0.0.0.0/0) 후 검토 및 시작



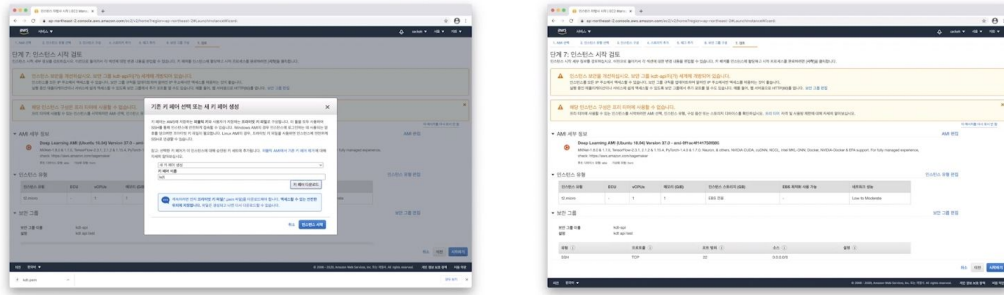
주의: 인스턴스 타입에 따라 과금이 발생할 수 있음, 사용하지 않을 때는 “중지” 또는 “종료”

딥러닝 AMI를 선택했지만 인스턴스를 GPU가속기가 없는 인스턴스를 사용하게 된다면 CPU로만 돌아가는 인스턴스를 사용하게 되는것이고 만약 가속기를 사용하게 된다면 P2, P3같은 가속기가 있는 인스턴스를 선택해야 한다.

보안그룹이라는 것은 보안레벨을 사용자들마다 접근제한을 설정할 때 사용한다. 그래서 보안그룹을 통해서 모든 IP와 포트를 통해서 내 서비스에 인스턴스에 접근 할 수있도록 모든 길을 열어 놓으면 굉장히 보안에 취약 할 수있어 실제 서비스를 할때는 특정 IP들만 또는 특정 포트에서 특정 프로토콜만 접근 가능하도록 사전에 규칙을 정해서 보통 만드는데 일반적이다. 실제 실습에서는 용이하게 실습을 진행하기위해 모든 포트와 프로토콜을 열어 놓고 진행하도록 한다.

EC2 생성 (3/4) – 키 페어 생성 & 인스턴스 시작 검토

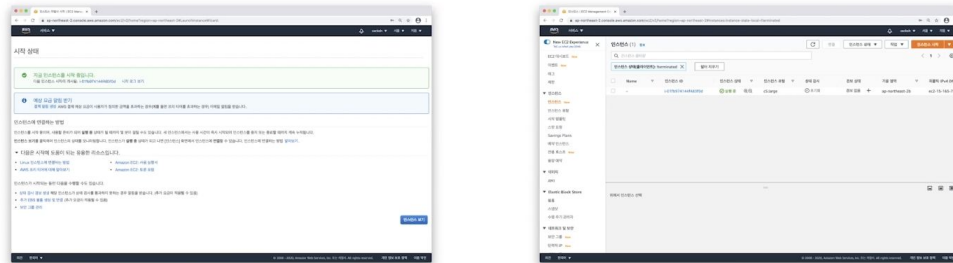
보안을 위한 키 페어 생성을 위해 키 페어 이름을 입력하고 (ex. kdt) “키 페어 다운로드”를 클릭하여 키 페어 다운로드
다운로드한 키 페어는 이후 인스턴스에 접속하기 위해 필요
인스턴스 시작



보안키를 생성해서 인스턴스에 접속할 수 있도록 보통 장치를 마련한다. 따라서 다운로드한 키 페어는 이후 이후 인스턴스에 접속하기 위해 필요하니 안전한 저장소에 저장해 두어야 한다.

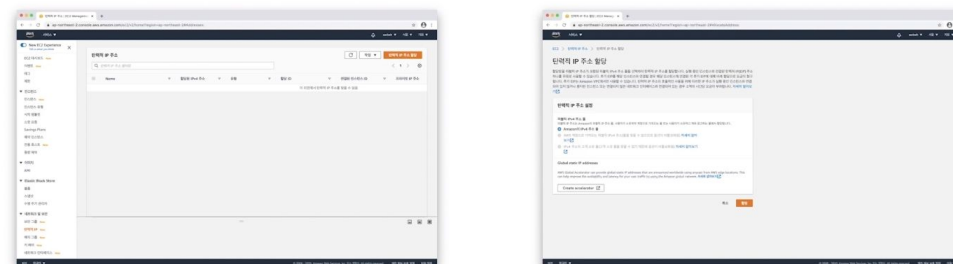
EC2 생성 (4/4) – 인스턴스 생성 확인

실행 중인 인스턴스 확인



탄력적 IP 설정 (1/3) – 탄력적 IP 생성

인스턴스를 중지 또는 종료 후 다시 시작하거나 생성하게 되면 기존 퍼블릭 IP가 변경됨
퍼블릭 IP를 고정으로 사용하고 싶을 때 탄력적 IP 주소를 할당할 수 있으나 추가 과금 발생
탄력적 IP 생성을 위해 “탄력적 IP 주소 할당” 클릭



탄력적 IP를 종종 필요에 따라 사용하게 되는데 퍼블릭 IP가 보통 인스턴스가 생성이 되면 들어와지는데 퍼블릭IP는 인스턴스를 종료하거나 중지해서 다시시작하게 되면 변경이 된다. 그래서 특정IP로 되도록 세팅을 해놓았는데 인스턴스가 갑자기 재시작을 하게 되면서 바뀌어 버리면 다른

서비스에서 고정을 해 놓았던 부분이 접근이 안된다. 이런 부분에서 IP를 고정으로 사용하고 싶다!
그럼 IP를 구매하는 방식이다. 탄력적 IP 설정을 통해 어떤 인스턴스에 어떤 IP를 지정하고 싶을때 사용하는 서비스이다.

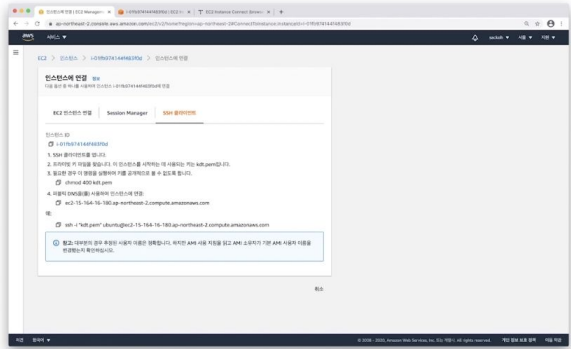
3. Practice: AWS Environment46

VS Code로 환경 테스트 (1/4) – 인스턴스 연결 초기화(initialization)

다운로드 받은 키 페어가 있는 위치에서 AWS 가이드에 따라 진행

Mac OS 또는 Linux는 자체 터미널로 바로 수행 가능

Windows는 OpenSSH 클라이언트 설치 후 CMD 또는 PowerShell에서 진행 가능



키는 권한 변경을 해야 한다. chmod 명령어를 통해서 권한 설정을 변경 해놔야 인스턴스 접속할때 키가 사용이 가능하다

3. Practice: AWS Environment47

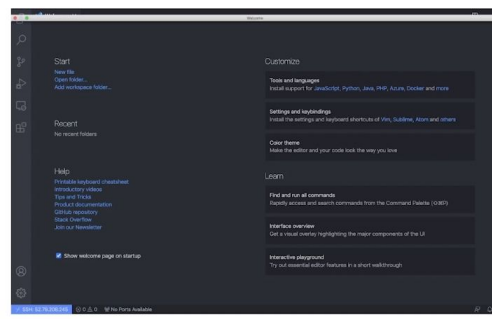
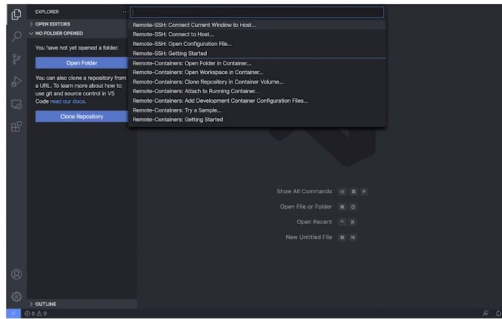
VS Code로 환경 테스트 (2/4) – 원격 접속 및 개발을 위한 VS Code 플러그인 설치



이 remote 확장프로그램을 설치하면 쉽게 접근이 가능하다

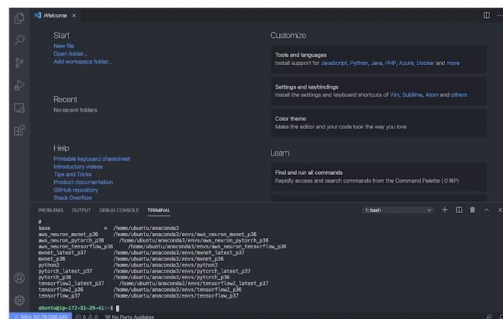
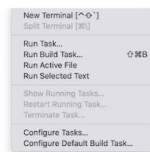
VS Code로 환경 테스트 (3/4) - 인스턴스에 접속

1. Remote-SSH: Connect to Host... 선택
2. `ssh -i "kdt.pem" ubuntu@public-ip-address` 입력
3. Select SSH configuration file



VS Code로 환경 테스트 (4/4) - 터미널 열고 환경 확인

터미널에서 "conda env list" 로 세팅되어 있는 가상환경 확인



설정하고 다운받은 pair key를 읽을수 있는 경로를 지정해서 사용자이름과 접속하고자 하는 주소를 입력하게 되면 접근이 가능해 진다. 지금 예시같은 경우에는 딥러닝 AMI를 사용했기 때문에 "conda env list"을 했을때 실제 conda가 설치 되어있어서 가상환경에 대한 list가 나오게 된다. 만약 AWS에서 제공하는 기본 AMI를 사용하게 된다면 이런 딥러닝 패키지라던가 anaconda같은 데이터 사이언스 패키지, 소프트 웨어가 따로 설치되어있지 않을 것이다. 따라서 필요한 수강생 분들은 AMI를 바꿔서 인스턴스를 만들어 사용한다던지 아니면 기존에 인스턴스에서 필요한 환경을 직접 구축해야 한다.

