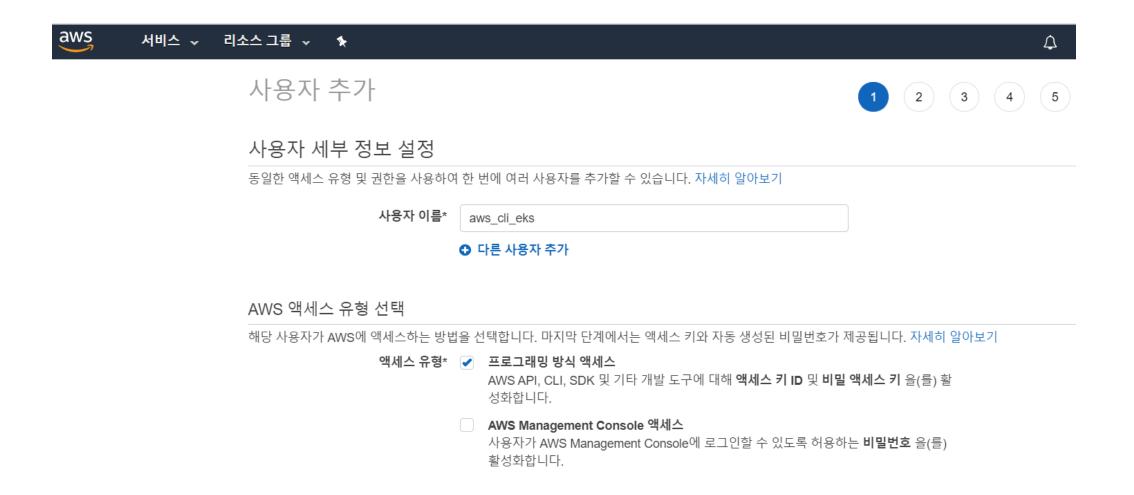
Multi-Cloud & Kubernetes

쿠버네티스를 활용한 멀티클라우드 도입과 운영전략 - AWS, Azure, GCP 비교와 실습

2020년 3월 아인인텔리전스 권재원, Ph.D. Lab: AWS CLI 환경설정

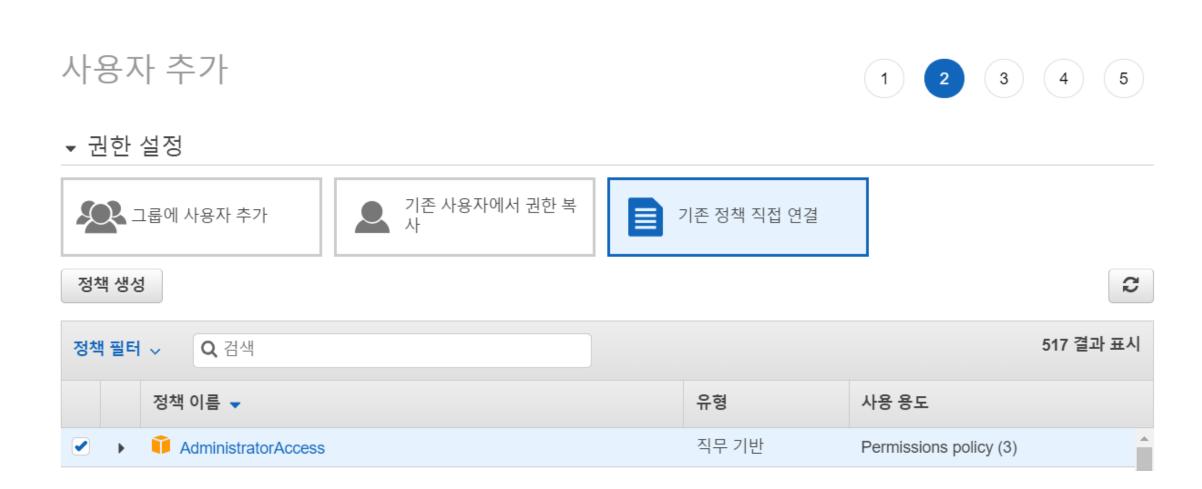
환경 설정 - AWS IAM 설정

사용자 추가 – 액세스 유형



환경 설정 - AWS IAM 설정

사용자 추가 – 권한 설정



환경 설정 – AWS IAM 설정

사용자 추가 – Access Key / Access Secret Key

사용자 추가













성공

아래에 표시된 사용자를 생성했습니다. 사용자 보안 자격 증명을 보고 다운로드할 수 있습니다. AWS Management Console 로그인을 위한 사용자 지침을 이메일로 보낼 수도 있습니다. 지금이 이 자격 증명을 다운로드할 수 있는 마지막 기회입니다. 하지만 언제든지 새 자격 증명 을 생성할 수 있습니다.

AWS Management Console 액세스 권한이 있는 사용자가 https://697559359741.signin.aws.amazon.com/console에 로그인할 수 있습니다.



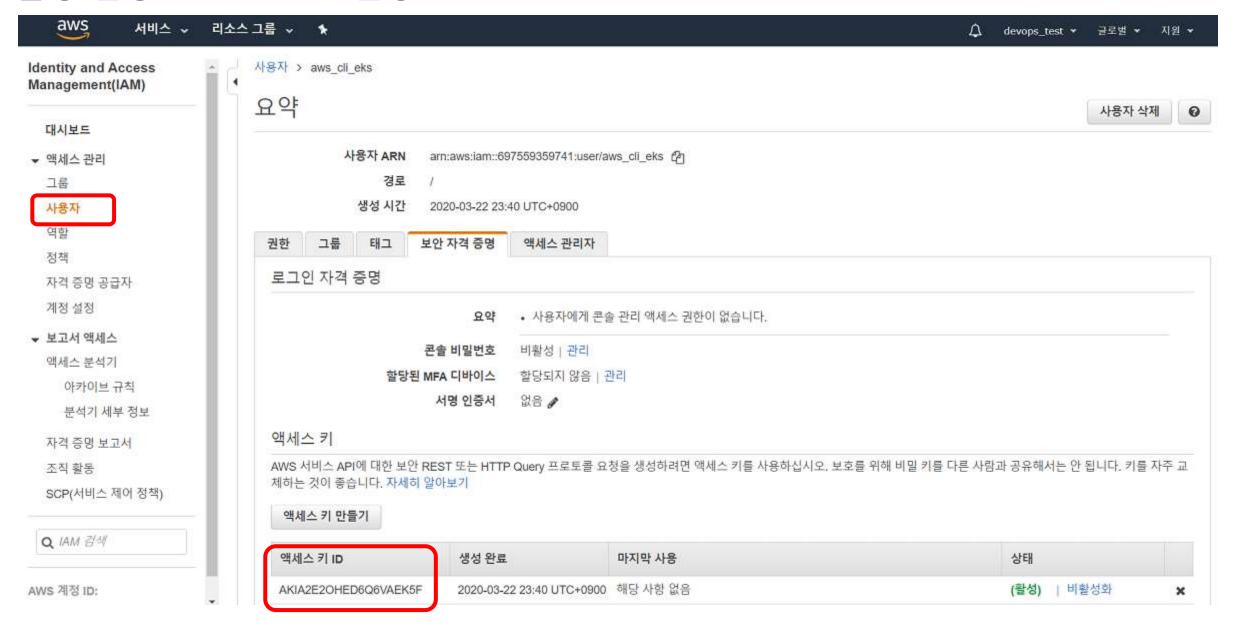
		사용자	액세스 키 ID	비밀액	세스 키	
)	9	aws_cli_eks	AKIA2E2OHED6Q6VAEK5F	****	표시	

환경 설정 - AWS CLI 설치

https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/install-cliv2-linux.html

- curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
- unzip awscliv2.zip
- sudo ./aws/install
- aws --version

환경 설정 - AWS IAM 설정



환경 설정 - AWS CLI 구성

빠르게 AWS CLI 구성

https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/cli-chap-configure.html#cli-quick-configuration

```
chaiwon@surface-Laptop:~/awscli$ aws configure
AWS Access Key ID [None]: A
AWS Secret Access Key [None]: A
Default region name [None]: ap-northeast-2
Default output format [None]: json
```

환경 설정 - Python

Python version과 pip 의 installation 여부 확인

- 최신 리눅스 distro에는 python2는 안 깔려있고, python3만 깔려 있음
- Pip도 python version에 따라 distro 가 제공하는 버전을 설치하는 것이 좋음

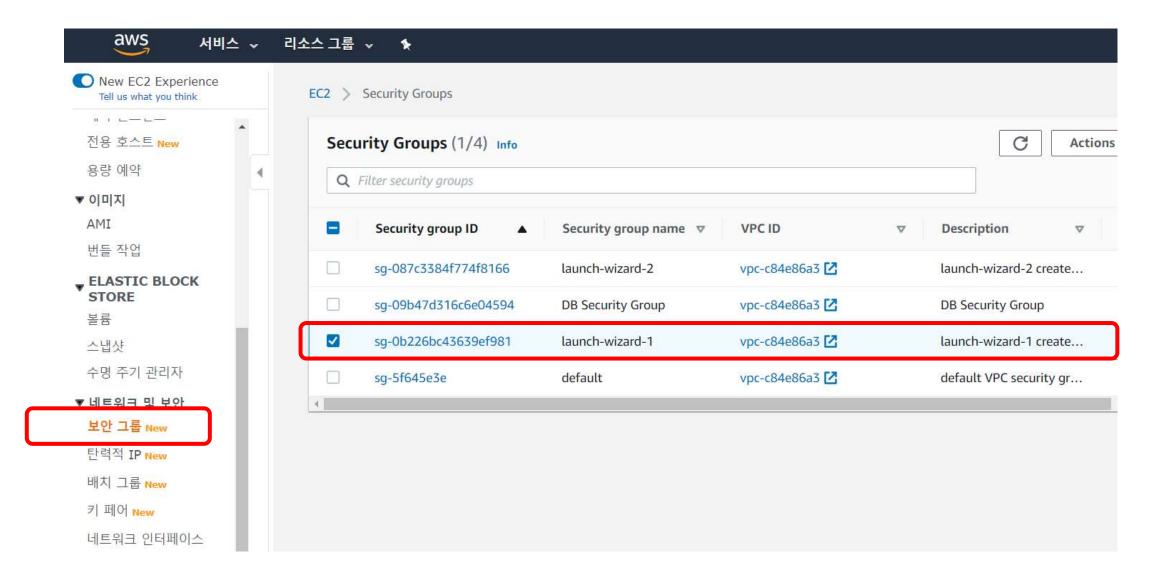
```
$ python --version
$ python3 --version
$ which python3
$ pip --version
$ pip3 --version
```

Pip3 설치 후 이를 이용하여 가상화 환경을 제공하는 virtualenv을 제공하는 것이 좋음

```
$ sudo apt-get update
$ sudo apt-get install python3-pip
$ sudo apt-get install virtualenv
```

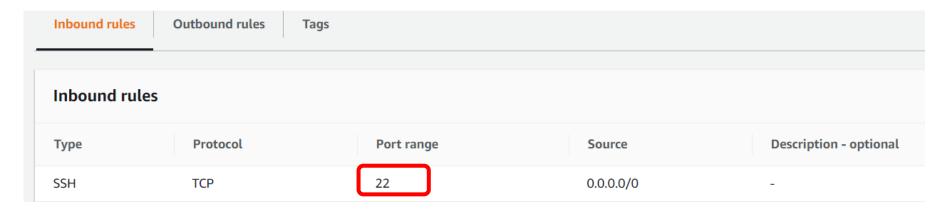
Lab: AWS EC2 생성

EC2 생성 – 보안 그룹 설정



EC2 생성 – 보안 그룹 설정

- 보안 그룹 설정
 - 보안 그룹의 인바운드는 SSH 포트만 열고 아웃바운드는 모두 열음



Outbound rules	Outbound rules								
Туре	Protocol	Port range	Destination	Description - optional					
All traffic	All	All	0.0.0.0/0	-					

EC2 생성

• Key Pair 생성

\$ aws ec2 create-key-pair --key-name TestKeyPair --query 'KeyMaterial' --output text > TestKeyPair.pem

• EC2 Instance 생성

\$ aws ec2 run-instances --image-id ami-082bdb3b2d54d5a19 --count 1 --instance-type t2.micro --key-name TestKeyPair --security-group-ids [YOUR SECURITY GROUP ID]

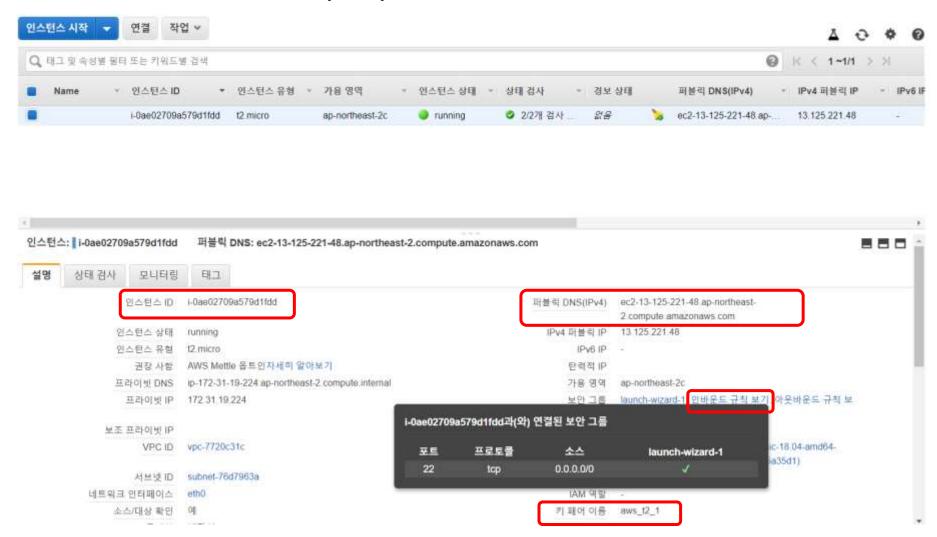
EC2 실습 - Linux 인스턴스에 연결

연결 옵션

- Linux 및 macOS X에 대한 옵션
 - SSH 클라이언트
 - EC2 Instance Connect
 - AWS 시스템 관리자 Session Manager
- Windows에 대한 옵션
 - PuTTY
 - SSH 클라이언트
 - AWS 시스템 관리자 Session Manager
 - Windows Subsystem for Linux

EC2 연결 준비 - 인스턴스에 대한 정보

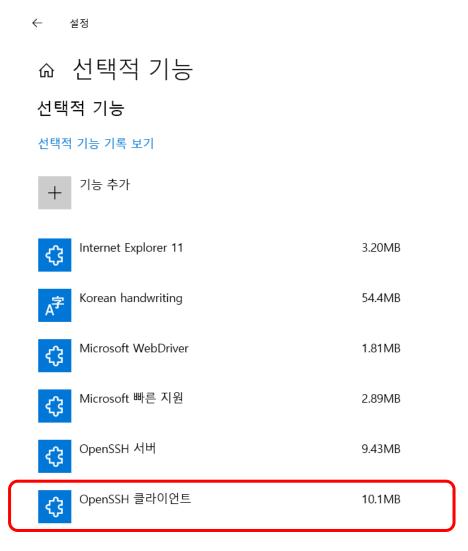
• 인스턴스 ID 와 퍼블릭 DNS(IPv4) 정보 파악



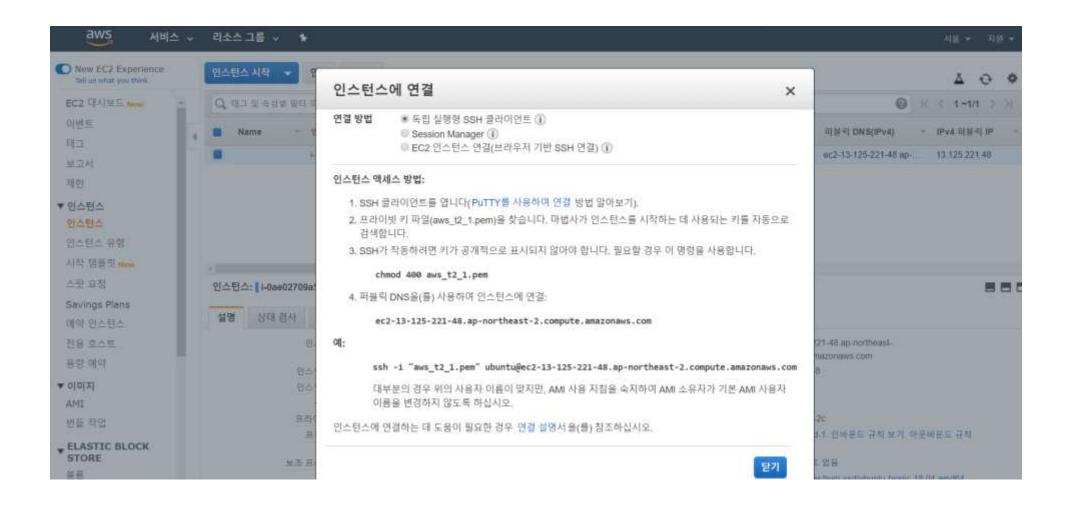
참고 - OpenSSH

Windows Server 2019 및 Windows 10 (Build > 1809)용 OpenSSH 설치

- 윈도우 설정 > 앱 및 기능 > 선택적 기능 관리로 이동
- 기능추가 : OpenSSH 클라이언트
- OpenSSH 서버를 설치하면 "OpenSSH-Server-In-TCP"라는 방화벽 규칙이 생성되고 활성화. 이 규칙은 포트 22에서의 인바운드 SSH 트래픽을 허용.



EC2 실습 - EC2 연결 화면



EC2 연결 준비 – 기본사용자 및 pem 위치 확인

인스턴스를 시작하는 데 사용한 AMI의 기본 사용자 이름

- Amazon Linux 2 또는 Amazon Linux AMI의 경우 사용자 이름은 ec2-user
- CentOS AMI의 경우 사용자 이름은 centos
- Debian AMI의 경우 사용자 이름은 admin 또는 root
- Fedora AMI의 경우 사용자 이름은 ec2-user 또는 fedora
- RHEL AMI의 경우 사용자 이름은 ec2-user 또는 root
- SUSE AMI의 경우 사용자 이름은 ec2-user 또는 root
- Ubuntu AMI의 경우 사용자 이름은 ubuntu

Pem 파일이 다운로드된 위치를 브라우저 설정 > 고급 > 다운로드 에서 찾음



WSL을 사용한 EC2 Instance 연결

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

- Amazon linux : ec2-user @ec2-xxxxxx.amazonaws.com
- Ubuntu: <u>ubuntu@ec2-xxxxxx.amazonaws.com</u>
- 접속 후 sudo passwd root 명령으로 root 의 암호를 설정한다.

WSL을 사용하여 Linux 인스턴스에 연결

Windows Subsystem for Linux(WSL)를 사용하여 Linux 인스턴스에 연결하려면 다음 절차를 사용하십시오. 인스턴스에 연결을 시도하는 동안 오류가 발생한 경우 인스턴스 연결 문제 해결을 참조하십시오.

SSH를 사용하여 인스턴스에 연결하려면

1. 터미널 창에서 ssh 명령을 사용하여 인스턴스에 연결합니다. 프라이빗 키(.pem) 파일, AMI의 사용자 이름 및 인스턴스의 퍼블릭 DNS 이름을 지정합니다. 예를 들어 Amazon Linux 2 또는 Amazon Linux AMI를 사용한 경우 사용자 이름은 ec2-user입니다. 인스턴스의 AMI 및 DNS 이름에 대한 사용자 이름을 찾는 방법에 대한 자세한 내용은 인스턴스에 대한 정보 가져오기 단원을 참조하십시오.

sudo ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com

다음과 같은 응답이 표시됩니다:

The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.

RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.

Are you sure you want to continue connecting (yes/no)?

참고 - SSH 접속시 RSA 보안 키 위치

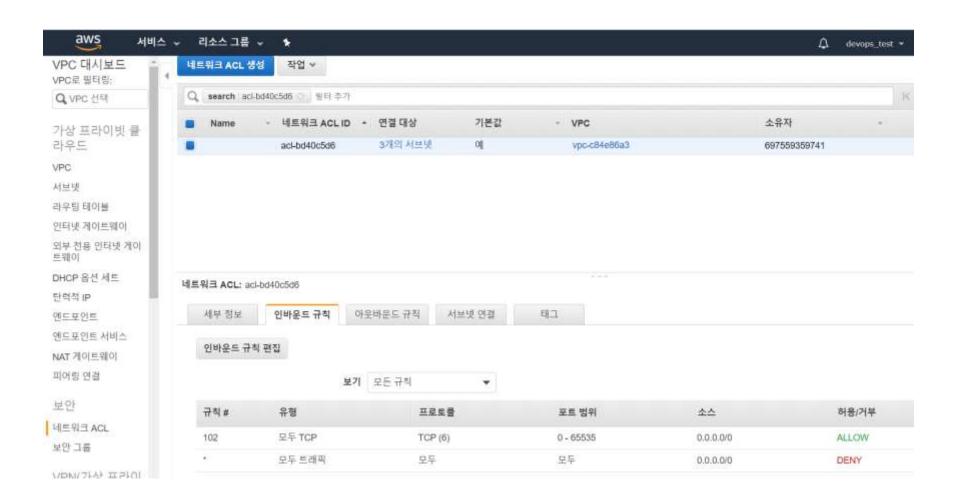
- Ubuntu 계정으로 접속했을 때, 개인 home directory는 /home/ubuntu 이며
- 보안 파일은 ~/.ssh/authorized_keys 에 들어있다.

```
ubuntu@ip-172-31-44-139:~$ cd .ssh
ubuntu@ip-172-31-44-139:~/.ssh$ ls -la
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 24 15:30 .
drwxr-xr-x 4 ubuntu ubuntu 4096 Mar 24 16:29 ...
-rw----- 1 ubuntu ubuntu 393 Mar 24 15:30 authorized_keys
ubuntu@ip-172-31-44-139:~/.ssh$ cat authorized keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCdVDChMFD45s3U1YqoCstk7avheOtksNP8W0sFtrFjzSowSm
oVdAPdVfuQt6B0oPPZ7aB9qrfb3A2XlzRtczTnaRHQnEF3vNwvncpVVOKippfXDRnN+m/1jwENsqOZ+LNtpE4n
zyho5wDaSEWy2Ok3GrFEd1SAroetgTeftJYaL1Vc98UqUBPJNDHWMqj9Z698VjQK3F10DBu/uHlbKn5LqJvE5l
iCxQNruqRV0kv+ySv7AHw4oMKoBnVsGWgIgA8V3f6aHC+MjG0WIQ94wHxFoAb9BBM+pVsTXsAvqS/OYOXs7H+y
btPNFRfpW/nev+cwEMnLo4S67CsQN4TdMXA/ TestKeyPair
ubuntu@ip-172-31-44-139:~/.ssh$ cd ..
```

참고 - EC2 SSH 접속이 되지 않을 때

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

- 보안그룹의 인바운드 traffic (tcp port 22)
- VPC ID, 서브넷 ID, 라우팅 테이블, 네트워크 ACL 등을 살펴봐야 한다.



Lab: Docker 설치

Docker 설치

https://docs.docker.com/install/linux/docker-ce/ubuntu/

• 오래된 버전의 docker 제거

\$ sudo apt-get remove docker docker-engine docker.io containerd runc

• 설치에 필요한 패키지들을 사전 설치

```
sudo apt-get update && sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common
```

• 도커의 공식 GPG 키와 저장소를 추가

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

Docker 설치

• docker 패키지가 검색되는지 확인

\$ sudo apt-get update && sudo apt-cache search docker-ce

```
docker-ce-cli - Docker CLI: the open-source application container engine docker-ce - Docker: the open-source application container engine
```

• 도커 CE 설치

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

\$ docker --version

```
ubuntu@ip-172-31-44-139:~$ docker --version
Docker version 19.03.8, build afacb8b7f0
```

Docker 실행

• docker 실행

\$ sudo service docker start

• docker 그룹에 사용자 id 추가

\$ sudo usermod -a -G docker <user_id>

• 도커 정보 확인

\$ sudo docker-info

* 비록 사용자id 가 sudo 로 들어갔지만 ec2 재실행 후 작동됨

참고: https://docs.aws.amazon.com/ko_kr/AmazonECS/latest/developerguide/docker-basics.html

Docker 컨테이너 Build

Dockerfile

```
FROM node:6.9.2

EXPOSE 8080

COPY server.js .

CMD node server.js
```

server.js

```
var http = require('http');
var handleRequest = function(request, response) {
    response.writeHead(200);
    response.end("Hello World!");
}
var www = http.createServer(handleRequest);
www.listen(8080);
```

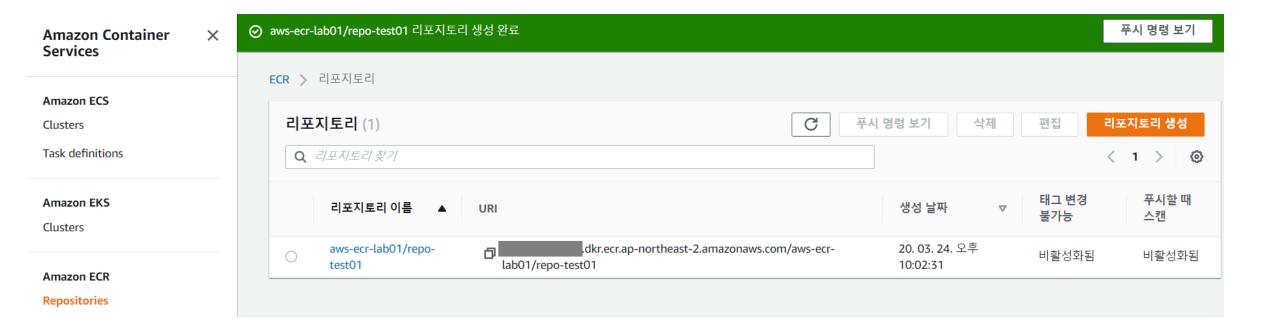
docker build

```
$ sudo docker build -t <aws_id>.dkr.ecr.ap-northeast-2.amazonaws.com/<aws-ecr-repo-name> .
```

Lab: Amazon ECR (Elastic Container Registry)

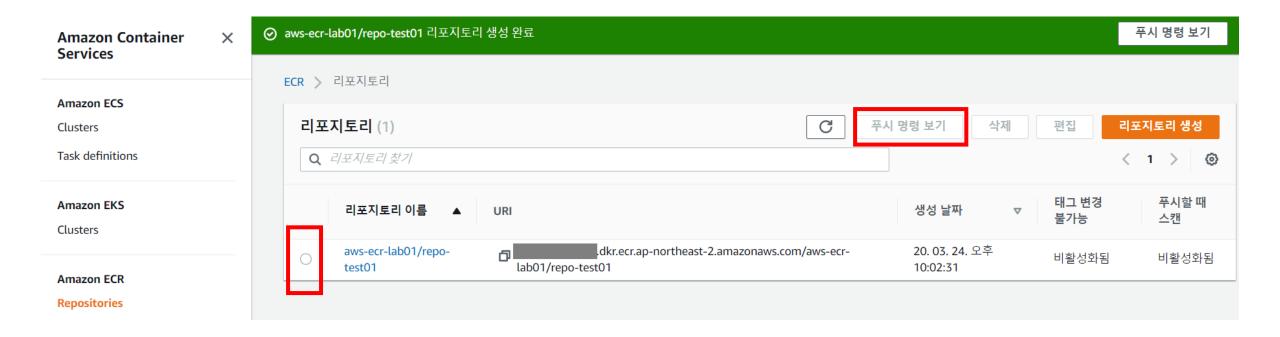
AWS ECR - 레지스트리

- https://docs.aws.amazon.com/ko_kr/AmazonECR/latest/userguide/repository-create.html
- 기본 레지스트리의 URL: https://<aws_account_id>.dkr.ecr.<region>.amazonaws.com

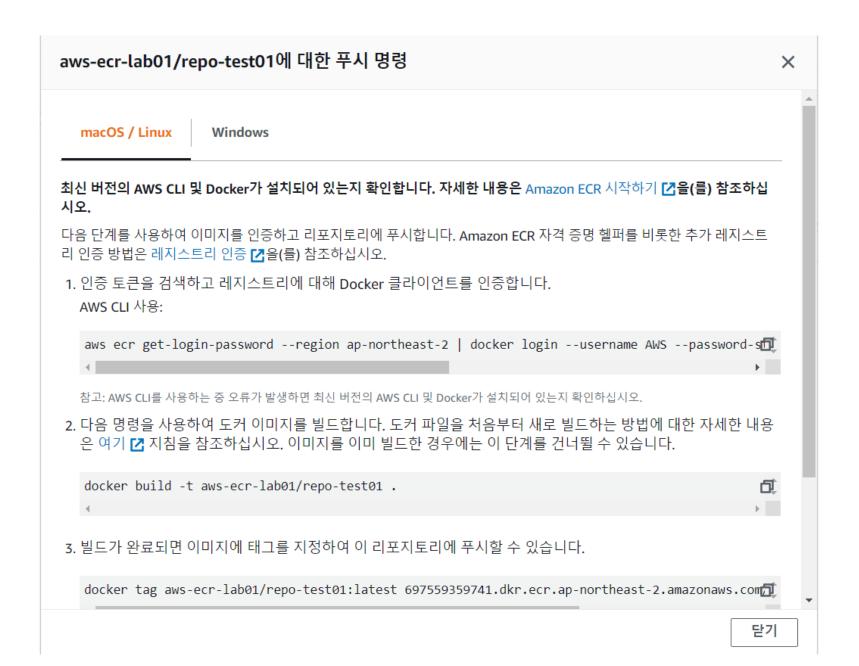


AWS ECR - 리포지토리 생성

• 생성한 리포지토리를 선택하고 푸시 명령 보기를 선택하여 새 리포지토리에 푸시



AWS ECR - 리포지토리 생성



AWS ECR - 리포지토리 생성

• aws ecr get-login 명령으로 Docker 클라이언트를 인증하는 데 사용할 수 있는 docker login을 검색

\$ aws ecr get-login

aws> ecr get-login

docker login -u AWS -p eyJwYXlsb2FkIjoiZGhqTG1lVks3R1laZi8xa0ZZbU12VnVYMWwrTTVkQ1ZqeVlhekxRRFFhUWRvSVo4dmZWeExQSEtpa0hF RTQ3SEFSUnpmRzNvT3FIb3NCVmJac0VMcXV1SWVWek1qZjRkYkRDcjBXd2FRWHF1b01mY1IxSVF2T2RxYWFUdzhpOHBOdms1VDRjVnlXbmZSUlV2VkV5Kzd NeVU4MXVWMG1WQUhKYjB1bVJSdGFmeWlBcVNMOGphTVpwVzVCeEcxMHFYdmdPbmVTeGJUYk1tYzMvWFZrYWhOWGpibVk3OTVYWkJ3YW1kTXJSRTJ1QWQ5a3 Evd0RpR004K0VmWU5ESE8yL2pvSEJ3TjBQSnJXUUhpWTJPdXk0aFZEVW1QaGxrUW1Pa3htd01MbjFncUdtU1VDSG5MRHRrMnpma1NsZ1FESzBnbk8wMFVhV 0M4K0ZIL0VMSWE5VFJZVTk2K05TZGtselZVb0FTSG5aWmlnLzljdHQ3Mm96ZE1qSDVXaUVQYUhlcEo5M0Y0aVhOT2Jwd3JhV0tYSG4raFlEc2ZhbUpOSFg2 dTJTSEhnbTlEb3NqQU5Tc0paZHQyYXVOQmVNZk9ZVVVoYWNiTzQ5bXhlTzVTdDlNZTVGbzZvOVMrTzdFTlNGaXIrMzNOL1NPd25NUlBjVmFLWmViWFk4dHh yQjJvZDlxZzgrdm5uMTRtVUEzTDBBdGNYTUVxT2JFUXpETnFBRlBTbnU1WFNXdEgzNFVOM25zUlhYcUtsTEJEdTdjZXpsdDMwbE0wdzZoOVZIRTVPVWZSaW xnZkdhZURQOXBIWVlvMGZiM25USmdESStKZGZ2NEtzTUZzRnp6VzRSVThhUDI4V2l6NEZoanNURGtFdEE3MW5JYkhQNXJwL0FIYkdiTkZpZTJxeGdYUk5Rc jV2enkvR1AweTU3NmxvMXhWRkQ3eDFSTkZpME1pQ2RvMjZCdklydWJlWnJpZWtQWTYyL2pwaEdxdmwyQUMvcXVIQWk3MDlOQmpFVE1wenYzOG50SWl4aEZT d0h6cFplQ1ZoQVp6RnVDa25KWTAyckloeGJVYisra1RTUnNWQWhlMW9HakNTUk5PejVsMURBTXNwU3haVXh4ejJqUjZvTmIyZW1aeUZZc0xOMm50TmI1bHc xUEdlbXkzclMwOWx2MXM2ejFHbTlKRHVWb3JuRTlnaDc2RUVLRzJwK1ZiZy8zUVhNN1FCbmw3REthNXYra25RUWMwUklvUUZTUDlOMWhOclJTdmRBdHducU RVWFNML085S0FoYm5xVmsvZTVRRHNaMjBFN2ZodkdaeFAwSS9xQkc5WlVqZnJNbEhKb1Jja1pnYzZ4alVhS05SY2RIN2tITV12ZEhLVFl1S3JhYy93eEZBd CtzQ1gzVzJJUVFSNlZoOUUyd2dFK3RJZVJObm8rYTJJSnJHNUFWSjF1SEZjOWltQ2kxRFlTbHZBPT0iLCJkYXRha2V5IjoiQVFJQkFIaEFPc2FXMmdaTjA5 V050TkdrWWM4cXAxMXhTaFovZHJFRW95MUhr0ExYV2dIM0JYVThPTUMxTElnOWZPYjdgMSsyQUFBQWZgQjhCZ2txaGtpRz13MEJCd2FnYnpCdEFnRUFNR2d HQ1NxR1NJYjNEUUVIQVRBZUJnbGdoa2dCWlFNRUFTNHdFUVFNSjE0YkdxTXIxNEwxcVRObUFnRVFnRHR1SDJobWo1RVluL3R6dThxTF15SFJZa1VSd0JvZk x0aUV2cHJvakl6NnQ3TUs5RkZyM0VlT0wzajlzUDIzMXQyc292Vkg4R0RVQk9hTjJBPT0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUiOiJEQVRBX0tFWSIsImV4c GlyYXRpb24i0jE10DUwOTkxNjN9 -e none https://697559359741.dkr.ecr.ap-northeast-2.amazonaws.com aws>

AWS ECR - 레지스트리 인증

https://docs.aws.amazon.com/ko_kr/AmazonECR/latest/userguide/Registries.html

• aws ecr get-login 명령으로 Docker 클라이언트를 인증하는 데 사용할 수 있는 docker login을 검색

\$ aws ecr get-login-password --region ap-northeast-2 | sudo docker login --username AWS --password-stdin 697559359741.dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/repo-test01

```
chaiwon@surface-Laptop:~$ aws ecr get-login-password --region ap-northeast-2 | docker login --username AWS --password-s
tdin .dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/repo-test01
WARNING! Your password will be stored unencrypted in /home/chaiwon/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

AWS ECR - Docker 컨테이너 Push

https://docs.aws.amazon.com/ko_kr/AmazonECR/latest/userguide/docker-push-ecr-image.html

```
$ sudo docker images
$ sudo docker tag <container_id> <aws_id>.dkr.ecr.ap-northeast-2.amazonaws.com/<aws-ecr-repo-name>
$ sudo docker push <aws_id>.dkr.ecr.ap-northeast-2.amazonaws.com/<aws-ecr-repo-name>
```

```
ubuntu@ip-172-31-44-139:~/ecr_docker$ sudo docker push 697559359741.dkr.ecr.ap-northeast-2.amazona
ws.com/aws-ecr-lab01/repo-test01
The push refers to repository [697559359741.dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/rep
o-test01]
24646920997b: Pushed
381c97ba7dc3: Pushed
604c78617f34: Pushed
fa18e5ffd316: Pushed
0a5e2b2ddeaa: Pushed
53c779688d06: Pushed
60a0858edcd5: Pushed
60a0858edcd5: Pushed
b6ca02dfe5e6: Pushed
latest: digest: sha256:bca8a3761edaa9595d5bd16811dd5b813daf69441157586a292f2b062683f338 size: 2002
```

AWS ECR - Docker 컨테이너 Push

