

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



사용자 추가



사용자 세부 정보 설정

동일한 액세스 유형 및 권한을 사용하여 한 번에 여러 사용자를 추가할 수 있습니다. [자세히 알아보기](#)

사용자 이름*

aws_cli_eks

[+ 다른 사용자 추가](#)

AWS 액세스 유형 선택

해당 사용자가 AWS에 액세스하는 방법을 선택합니다. 마지막 단계에서는 액세스 키와 자동 생성된 비밀번호가 제공됩니다. [자세히 알아보기](#)

액세스 유형*



프로그래밍 방식 액세스

AWS API, CLI, SDK 및 기타 개발 도구에 대해 **액세스 키 ID** 및 **비밀 액세스 키** 을(를) 활성화합니다.



AWS Management Console 액세스

사용자가 AWS Management Console에 로그인할 수 있도록 허용하는 **비밀번호** 을(를) 활성화합니다.


환경 설정 - AWS IAM 설정


사용자 추가 - 권한 설정


사용자 추가

- 1
- 2
- 3
- 4
- 5

▼ 권한 설정

 그룹에 사용자 추가

 기존 사용자에서 권한 복사

 기존 정책 직접 연결


정책 생성

↻

정책 필터 ▼

🔍 검색

517 결과 표시

	정책 이름 ▼	유형	사용 용도
<input checked="" type="checkbox"/>	▶  AdministratorAccess	직무 기반	Permissions policy (3)

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 증명을 생성할 수 있습니다.

AWS Management Console 액세스 권한이 있는 사용자가 <https://697559359741.signin.aws.amazon.com/console>에 로그인할 수 있습니다.

↓ .csv 다운로드

	사용자	액세스 키 ID	비밀 액세스 키
▶	✓ 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`

```
chaiwon@surface-Laptop:~/awscli$ ls -la
total 31936
drwxrwxrwx 1 chaiwon chaiwon    4096 Mar 22 23:25 .
drwxr-xr-x 1 chaiwon chaiwon    4096 Mar 22 23:23 ..
drwxr-xr-x 1 chaiwon chaiwon    4096 Mar 17 01:31 aws
-rw-rw-rw- 1 chaiwon chaiwon 32637959 Mar 22 23:23 awscliv2.zip
chaiwon@surface-Laptop:~/awscli$ sudo ./aws/install

You can now run: /usr/local/bin/aws --version
chaiwon@surface-Laptop:~/awscli$
chaiwon@surface-Laptop:~/awscli$ aws --version
aws-cli/2.0.4 Python/3.7.3 Linux/4.4.0-18362-Microsoft botocore/2.0.0dev8
```

환경 설정 - AWS IAM 설정

aws

서비스 ▾ 리소스 그룹 ▾

devops_test ▾ 글로벌 ▾ 지원 ▾

Identity and Access Management(IAM)

대시보드

▼ 액세스 관리

그룹

사용자

역할

정책

자격 증명 공급자

계정 설정

▼ 보고서 액세스

액세스 분석기

아카이브 규칙

분석기 세부 정보

자격 증명 보고서

조직 활동

SCP(서비스 제어 정책)

Q IAM 검색

AWS 계정 ID:

사용자 > aws_cli_eks

요약

사용자 삭제 ?

사용자 ARN arn:aws:iam::697559359741:user/aws_cli_eks

경로 /

생성 시간 2020-03-22 23:40 UTC+0900

권한 그룹 태그 보안 자격 증명 액세스 관리자

로그인 자격 증명

요약

• 사용자에게 콘솔 관리 액세스 권한이 없습니다.

콘솔 비밀번호

비활성 | 관리

할당된 MFA 디바이스

할당되지 않음 | 관리

서명 인증서

없음

액세스 키

AWS 서비스 API에 대한 보안 REST 또는 HTTP Query 프로토콜 요청을 생성하려면 액세스 키를 사용하십시오. 보호를 위해 비밀 키를 다른 사람과 공유해서는 안 됩니다. 키를 자주 교체하는 것이 좋습니다. 자세히 알아보기

액세스 키 만들기

액세스 키 ID	생성 완료	마지막 사용	상태
AKIA2E2OHED6Q6VAEK5F	2020-03-22 23:40 UTC+0900	해당 사항 없음	(활성) 비활성화 ✕

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]: 
AWS Secret Access Key [None]: 
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

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links, with '보안 그룹 New' (Security Groups New) highlighted by a red rectangle. The main content area displays the 'Security Groups (1/4)' page. A table lists several security groups, with the 'launch-wizard-1' group selected (checkbox checked) and highlighted by a red rectangle.

	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	sg-087c3384f774f8166	launch-wizard-2	vpc-c84e86a3	launch-wizard-2 create...
<input type="checkbox"/>	sg-09b47d316c6e04594	DB Security Group	vpc-c84e86a3	DB Security Group
<input checked="" type="checkbox"/>	sg-0b226bc43639ef981	launch-wizard-1	vpc-c84e86a3	launch-wizard-1 create...
<input type="checkbox"/>	sg-5f645e3e	default	vpc-c84e86a3	default VPC security gr...

EC2 생성 - 보안 그룹 설정

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

Inbound rules

Outbound rules

Tags

Inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-

Outbound rules

Type	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

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar includes buttons for '인스턴스 시작' (Start Instance), '연결' (Connect), and '작업' (Actions). Below the navigation bar is a search bar and a table of instances. The table has columns for Name, 인스턴스 ID (Instance ID), 인스턴스 유형 (Instance Type), 가용 영역 (Availability Zone), 인스턴스 상태 (Instance State), 상태 검사 (Status Checks), 경고 상태 (Alerts), 퍼블릭 DNS(IPv4) (Public DNS (IPv4)), IPv4 퍼블릭 IP (IPv4 Public IP), and IPv6 IP (IPv6 IP). The instance i-0ae02709a579d1fdd is shown in the 'running' state with a public DNS of ec2-13-125-221-48.ap-northeast-2.compute.amazonaws.com and a public IPv4 address of 13.125.221.48.

The detailed view of the instance i-0ae02709a579d1fdd is shown below. The instance is in the 'running' state. The public DNS (IPv4) is ec2-13-125-221-48.ap-northeast-2.compute.amazonaws.com. The public IPv4 address is 13.125.221.48. The instance is connected to the launch-wizard-1 security group, which allows traffic on port 22 (SSH) from 0.0.0.0/0.

포트	프로토콜	소스	launch-wizard-1
22	tcp	0.0.0.0/0	✓

https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/connection-prereqs.html#connection-prereqs-get-info-about-instance

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.





← 설정

선택적 기능

선택적 기능

[선택적 기능 기록 보기](#)

+ 기능 추가

	Internet Explorer 11	3.20MB
	Korean handwriting	54.4MB
	Microsoft WebDriver	1.81MB
	Microsoft 빠른 지원	2.89MB
	OpenSSH 서버	9.43MB
	OpenSSH 클라이언트	10.1MB

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

EC2 실습 - EC2 연결 화면

The screenshot shows the AWS Management Console interface with a modal dialog titled "인스턴스에 연결" (Connect to Instance). The dialog lists three connection methods: "독립 실행형 SSH 클라이언트" (Selected), "Session Manager", and "EC2 인스턴스 연결(브라우저 기반 SSH 연결)". It then provides a 4-step guide for using an SSH client, including finding the private key, setting permissions, and using the public DNS to connect. A terminal command for setting permissions is shown: `chmod 400 aws_t2_1.pem`. Another command for connecting is shown: `ssh -i "aws_t2_1.pem" ubuntu@ec2-13-125-221-48.ap-northeast-2.compute.amazonaws.com`. The background shows the EC2 console with a list of instances and a table of public DNS and IP addresses.

인스턴스에 연결

연결 방법

- 독립 실행형 SSH 클라이언트 ⓘ
- Session Manager ⓘ
- EC2 인스턴스 연결(브라우저 기반 SSH 연결) ⓘ

인스턴스 액세스 방법:

1. SSH 클라이언트를 엽니다(PuTTY를 사용하여 연결 방법 알아보기).
2. 프라이빗 키 파일(aws_t2_1.pem)을 찾습니다. 마법사가 인스턴스를 시작하는 데 사용되는 키를 자동으로 검색합니다.
3. SSH가 작동하려면 키가 공개적으로 표시되지 않아야 합니다. 필요할 경우 이 명령을 사용합니다.

```
chmod 400 aws_t2_1.pem
```

4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:

```
ec2-13-125-221-48.ap-northeast-2.compute.amazonaws.com
```

예:

```
ssh -i "aws_t2_1.pem" ubuntu@ec2-13-125-221-48.ap-northeast-2.compute.amazonaws.com
```

대부분의 경우 위의 사용자 이름이 맞지만, AMI 사용 지침을 숙지하여 AMI 소유자가 기본 AMI 사용자 이름을 변경하지 않도록 하십시오.

인스턴스에 연결하는 데 도움이 필요한 경우 [연결 설명서](#)을(를) 참조하십시오.

닫기

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 찾을



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 : ubuntu@ec2-xxxxxx.amazonaws.com
- 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 해결](#)을 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-user`입니다. 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 AAAAB3NzaC1yc2EAAAADAQABAAQACdVDChMFD45s3U1YqoCstk7avheOtkSNP8W0sFtrFjzSowSm
oVdAPdVfuQt6B0oPPZ7aB9qrfb3A2X1zRtczTnaRHQnEF3vNwvncpVVOKippfXDRnN+m/1jwENsqOZ+LNtpE4n
zyho5wDaSEWy20k3GrFE1SAroetgTefTJYaL1Vc98UqUBPJNDHWMqj9Z698VjQK3F10DBu/uH1bKn5LqJvE51
iCxQNruqRV0kv+ySv7AHw4oMKoBnVsGwGIgA8V3f6aHC+MjG0WlQ94wHxFoAb9BBM+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.

The screenshot shows the AWS Management Console interface for a Network ACL. The left sidebar contains navigation links for VPC resources. The main content area shows the 'Network ACL' configuration for 'acl-bd40c5d6'. The 'Inbound Rules' tab is selected, displaying a table of rules.

Name	네트워크 ACL ID	연결 대상	기본값	VPC	소유자
	acl-bd40c5d6	3개의 서브넷	예	vpc-c84e86a3	697559359741

규칙 #	유형	프로토콜	포트 범위	소스	허용/거부
102	모두 TCP	TCP (6)	0 - 65535	0.0.0.0/0	ALLOW
*	모두 트래픽	모두	모두	0.0.0.0/0	DENY

Lab: Docker 설치

<https://jeongchul.tistory.com/611>

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

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

aws-ecr-lab01/repo-test01 리포지토리 생성 완료

푸시 명령 보기

ECR > 리포지토리

리포지토리 (1)

리포지토리 찾기

리포지토리 생성

삭제

편집

푸시 명령 보기

리포지토리 생성

리포지토리 이름 ▲	URI	생성 날짜 ▼	태그 변경 불가능	푸시할 때 스캔
aws-ecr-lab01/repo-test01	[redacted].dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/repo-test01	20. 03. 24. 오후 10:02:31	비활성화됨	비활성화됨

AWS ECR - 리포지토리 생성

- 생성한 리포지토리를 선택하고 푸시 명령 보기를 선택하여 새 리포지토리에 푸시

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

aws-ecr-lab01/repo-test01 리포지토리 생성 완료

푸시 명령 보기

ECR > 리포지토리

리포지토리 (1)

리포지토리 찾기

aws-ecr-lab01/repo-test01

lab01/repo-test01

20. 03. 24. 오후 10:02:31

비활성화됨

비활성화됨

AWS ECR - 리포지토리 생성

aws-ecr-lab01/repo-test01에 대한 푸시 명령

macOS / Linux

Windows

최신 버전의 AWS CLI 및 Docker가 설치되어 있는지 확인합니다. 자세한 내용은 [Amazon ECR 시작하기](#) 을(를) 참조하십시오.

다음 단계를 사용하여 이미지를 인증하고 리포지토리에 푸시합니다. Amazon ECR 자격 증명 헬퍼를 비롯한 추가 레지스트리 인증 방법은 [레지스트리 인증](#) 을(를) 참조하십시오.

- 인증 토큰을 검색하고 레지스트리에 대해 Docker 클라이언트를 인증합니다.
AWS CLI 사용:

```
aws ecr get-login-password --region ap-northeast-2 | docker login --username AWS --password-
```

참고: AWS CLI를 사용하는 중 오류가 발생하면 최신 버전의 AWS CLI 및 Docker가 설치되어 있는지 확인하십시오.
- 다음 명령을 사용하여 도커 이미지를 빌드합니다. 도커 파일을 처음부터 새로 빌드하는 방법에 대한 자세한 내용은 [여기](#) 지침을 참조하십시오. 이미지를 이미 빌드한 경우에는 이 단계를 건너뛸 수 있습니다.

```
docker build -t aws-ecr-lab01/repo-test01 .
```
- 빌드가 완료되면 이미지에 태그를 지정하여 이 리포지토리에 푸시할 수 있습니다.

```
docker tag aws-ecr-lab01/repo-test01:latest 697559359741.dkr.ecr.ap-northeast-2.amazonaws.com
```

닫기

AWS ECR - 리포지토리 생성

- `aws ecr get-login` 명령으로 Docker 클라이언트를 인증하는 데 사용할 수 있는 `docker login`을 검색

```
$ aws ecr get-login
```

```
aws> ecr get-login
docker login -u AWS -p eyJwYXlsb2FkIjoiZGhqTG1lVks3R1laZi8xa0ZZbU12VnVYMWwrTTVkJ1ZqeVlhEkxRRFFhUWRvSVo4dmZWeExQSEtpa0hF
RTQ3SEFSUnpmRzNvT3FiY3NCVmJac0VMcXV1SWVWek1qZjRkYkRDcjBXd2FRWHF1b01mY1IxBmZSU1V2Vkv5Kzd
NeVU4MXVWMG1WQUhKYjB1bVJSdGFmeWlBcVNMOGphTVpwVzVCEcxMHFYdmdPbmVtEGJUYk1tYzZlVWZrYWhOWGpibVkb3OTVYVk13YV1kTXJSRTJlQWQ5a5
Evd0RpR004K0VmWU5ESE8yL2pvSEJ3TjBQSnJXUUhPTJpXk0aFZEVW1QaGxrUW1Pa3htd01MbjFncUdtU1VDSG5MRHRrMnpma1NsZlFESzBnbk8wMFVhV
0M4K0ZIL0VMSWE5VFJZVTk2K05TZGtse1ZVb0FTSG5aWm1nLzljdHQ3Mm96ZE1qSDVXaUvQYUhlcEo5M0Y0aVhOT2Jwd3JhV0tYSg4raFlEc2ZhbUp0SFg2
dTJTSEhnbTlEb3NqQU5Tc0paZHQyYXVOQmVNZk9ZVVVoYWNiZTQ5bXhlTzVTdDlNZTVGbzZvOVMrTzdFTlNGaXIRmZnNOL1NPd25NUlBjVmFLWmViWfK4dHh
yQjJvZDlxZzgrdm5uMTRtVUEzTDBBdGNYTUvXt2JFUXpETnFBRlBTbnU1WFNXdEgzNFVOM25zUlhYcUtsTEJEdTdjZXpsdDMwbE0wdzZoOVZIRTPVWZSaW
xnZkdhZURQOXBIWVlvMGZiM25USmdESSStKZGZ2NEtztUZZrnp6VzRSVThhUDI4V2l6NEZoanNURGtFdEE3MW5JYkhQNXJwL0FIYkdiTkZpZTJxeGdYUk5Rc
jV2enkvr1AweTU3NmXvMXhWRkQ3eDFSTkZpME1pQ2RvMjZCdKlydWJlWnJpZwtQWTYyL2pwaEdxdmwyQUMvcXVIQWk3MDlOQmpFVE1wenYzOG50SW14aEZT
d0h6cFplQ1ZoQVp6RnVDa25KWTayckloeGJVYisra1RTUnNWQWhlMW9HakNTUk5PejVsMURBTXNwU3haVXh4eJjQJzVtMiyZW1aeUZZc0xOMm50TmI1bHc
xUEdlbXkzc1MwOWx2MXM2ejFhBTlKRHVb3JuRTlNaDc2RUVLrZJwK1ZiZy8zUVhNN1FCbmw3REthNXYra25RUWMwUklvU0ZTU0ZlOMWhOclJlTdmRBdHducU
RVWFnML085S0FoYm5xVmsvZTVRRHNaMjBFN2ZodkdaeFAwSS9xQkc5WlVqZnJNbEhKb1Jja1pnYzZ4a1VhS05SY2RIN2tITVl2ZEhLVFl1S3JhYy93eEZBd
CtzQ1gzVzJJUVFSN1ZoOUUyd2dFK3RJZVJObm8rYTJJSnJHNUFWSjF1SEZjOWltQ2kxRF1TbHZBPT0iLCJkYXRha2V5IjoiQVFJQkFiIEFPc2FXMmdaTjA5
V050TkdrWWM4cXAxMXhTaFovZHJFRW95MUhrOExYV2dIM0JYVThPTUMxTElnOWZPYjdqMSsyQUFBQWZqQjhCZ2txaGtpRz13MEJCd2FnYnpCdEFnRUFNR2d
HQ1Nxr1NJYjNEUUVIQVRBZUJnbGdoa2dCWlFNRUFTNHdFUVFNSjE0YkdxTXIxNEwxcVRObUFnRVFnRHR1SDJobWo1RVluL3R6dThxTF15SFJZa1VSd0JvZk
x0aUV2cHJvak16NnQ3TUs5RkZyM0VlT0wzajlZUDIzMXQyc292Vkg4R0RVQk9hTjJBPT0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUiOiJEQVRBXC0tFWSIsImV4c
GlyYXRpb24iOiJE1ODUwOTkxNjN9 -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 login`을 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-stdin [REDACTED].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.amazonaws.com/aws-ecr-lab01/repo-test01
The push refers to repository [697559359741.dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/repo-test01]
24646920997b: Pushed
381c97ba7dc3: Pushed
604c78617f34: Pushed
fa18e5ffd316: Pushed
0a5e2b2ddeaa: Pushed
53c779688d06: Pushed
60a0858edcd5: Pushed
b6ca02dfe5e6: Pushed
latest: digest: sha256:bca8a3761edaa9595d5bd16811dd5b813daf69441157586a292f2b062683f338 size: 2002
```

AWS ECR - Docker 컨테이너 Push

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

Images

Permissions

Lifecycle Policy

Tags

ECR > 리포지토리 > aws-ecr-lab01/repo-test01

aws-ecr-lab01/repo-test01

이미지 (1)

이미지 찾기

<input type="checkbox"/>	이미지 태그	이미지 URI	푸시 위치 ▼	다이제스트	크기 (MB) ▼	스태tus
<input type="checkbox"/>	latest	697559359741.dkr.ecr.ap-northeast-2.amazonaws.com/aws-ecr-lab01/repo-test01:latest	20. 03. 25. 오전 03:50:34	sha256:bca8a3761...	260.92	-