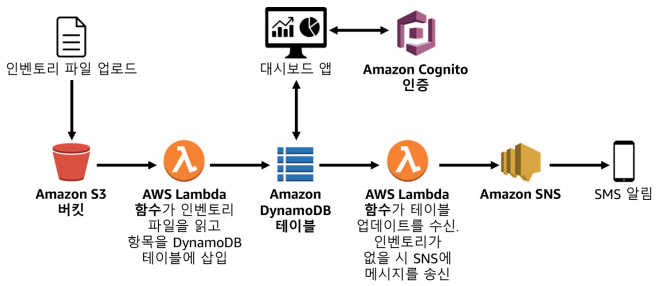
2026.02.19.목

생성일	@2026년 2월 19일 오전 9:11
태그	AWS Docker

목차

1. AWS Lambda 실습
2. Docker 기초
3. AWS EC2에 Docker 설치
4. Docker 기본 명령어

▼ 1) AWS Lambda 실습



1. Lambda 함수를 생성하고 데이터 로드
2. S3 이벤트 구성
3. 로딩 프로세스 테스트
4. 알림 구성(SNS)
5. 알림을 전송하는 Lambda 함수 생성
6. 시스템 테스트

▼ 2) Docker 기초

- 리눅스 컨테이너
  - chroot: root디렉토리 변경하는 시스템 콜, 특정 디렉토리를 해당 프로세스의 루트처럼 보이도록 격리
  - cgroup: 프로세스 그룹 별로 리소스 사용을 제한 및 모니터링
    - CPU
    - Memory
    - Disk I/O
    - Network
    - PIDs 수
  - namespace: 프로세스에 격리된 시스템 뷰를 제공하는 커널 기능

PID	프로세스 ID 격리 (컨테이너 내부는 PID 1부터 시작)
NET	네트워크 인터페이스 격리
MNT	마운트 포인트 격리
IPC	프로세스 간 통신 격리
UTS	hostname 격리
USER	UID/GID 격리

▼ 3) aws ec2에 Docker 설치

1️⃣ 기존 잔여 패키지 제거 (선택 but 권장)

```
sudo apt remove docker docker-engine docker.io containerd runc -y
```

2️⃣ 패키지 인덱스 업데이트

```
sudo apt update
```

3️⃣ 필수 패키지 설치

```
sudo apt install ca-certificates curl gnupg -y
```

4️⃣ Docker GPG 키 등록

```
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

5️⃣ Docker 공식 저장소 추가

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

6️⃣ 저장소 반영

```
sudo apt update
```

7️⃣ Docker Engine 설치

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

8️⃣ Docker 서비스 상태 확인

```
sudo systemctl status docker
```

9️⃣ 정상 동작 테스트

```
sudo docker run hello-world
```

🔟 sudo 없이 Docker 사용 (선택)

```
sudo usermod -aG docker $USER
newgrp docker
```

1️⃣1️⃣ Docker Compose 버전 확인

```
docker compose version
```

🎯 설치 완료 후 정상 상태

```
docker --version
docker compose version
```

```
sudo systemctl status docker
```

▼ 4) Docker 기본 명령어

이미지

- `docker pull [이미지]` — 이미지 다운로드
- `docker images` — 이미지 목록
- `docker rmi [이미지]` — 이미지 삭제
- `docker build -t [이름] .` — Dockerfile로 빌드

컨테이너 실행

- `docker run [옵션] [이미지]`
  - `d` 백그라운드 실행
  - `it` 인터랙티브 터미널
  - `p 호스트:컨테이너` 포트 매핑
  - `v 호스트:컨테이너` 볼륨 마운트
  - `-name` 컨테이너 이름 지정
  - `-rm` 종료 시 자동 삭제
  - `e` 환경변수 설정
  - `-network` 네트워크 지정

컨테이너 관리

- `docker ps` — 실행 중인 컨테이너 (`a` 전체)
- `docker start/stop/restart [컨테이너]`
- `docker rm [컨테이너]` — 컨테이너 삭제 (`f` 강제)
- `docker exec -it [컨테이너] bash` — 컨테이너 접속
- `docker logs [컨테이너]` — 로그 확인 (`f` 실시간)

시스템

- `docker inspect [대상]` — 상세 정보
- `docker stats` — 리소스 사용량
- `docker system prune` — 미사용 리소스 일괄 삭제

네트워크 / 볼륨

- `docker network ls / create / rm`
- `docker volume ls / create / rm`

Docker Compose

- `docker compose up -d` — 실행
- `docker compose down` — 중지 및 삭제
- `docker compose logs -f` — 로그