

※위에서부터 아래로 차례로 읽어야 ECS의 용도를 알 수 있습니다.

※SAA의 특성상, ECS의 특징보단 사용 상황(컨테이너)에 대한 질문이 많으리라 예상됩니다.(초중급시험이므로)

* Virtual Machine, VM (가상 머신)

- 가상의 컴퓨터, 하나의 호스트에 안에 또다른 호스트를 만들어 사용하는 것
- CPU, Memory와 같은 주요 하드웨어 부품을 소프트웨어로 완전 재현해내어 기능을 흉내내게 하고(에뮬레이션), 격리된 실행 환경(OS)를 만듦
 - 즉 하드웨어를 직접 가상화함
- 호스트의 OS(Operating System) 내에 또다른 실행환경의 OS가 존재함
 - 윈도우 OS의 호스트 내에 리눅스, 우분투 등의 다양한 OS를 올릴 수 있음
- 다만 OS를 포함하기 때문에 용량을 많이 차지할 뿐더러, 사용자가 필요치 않는 기능까지 포함할 수 있으며 느림

* (Linux) Container (화물선에 싣는 그 컨테이너와 의미가 같음)

- ECS를 사용하는 목적이자 관리 대상
- 하드웨어가 아닌 OS를 가상화하여 커널을 공유하며 프로세스(컨테이너와 비슷)를 격리된 환경에서 실행하는 것
- VM와 달리 호스트의 OS에서 가상화를 실시하여, 이 OS 위에 프로세스들이 '컨테이너'로서 격리된 환경에서 실행됨
- 호스트의 입장에선 컨테이너는 프로세스에 불과하지만, 컨테이너 입장에서는 독립된 실행환경임
- OS를 포함하지 않는만큼 가볍고, 하드웨어를 가상화하지 않기 때문에 빠름

* Kernel

- Operaintg System에서 가장 중요한 역할을 맡고 있는 핵심(核心)
- 커널이 각 프로세스(실행환경)에 하드웨어 자원(CPU 등)을 할당하고, 작업 스케줄링(처리순서)를 담당하며, 프로세스 간 접근과 보안을 책임짐
- 과거에 커널이 없던 시절에도 컴퓨터는 존재할 수 있었으나 메모리를 초기화하기 위해서는 컴퓨터를 리부팅해야 하는 등, 자원관리/제어 주체의 필요성에 의해 탄생

* Docker(Container Orchestration)

- 앞서 설명한 Linux Container 기술에 근간을 두는 컨테이너 관리 솔루션
- 'Docker'라는 단어 자체가 '부두에서 일하는 노동자', 즉 컨테이너를 관리하는 존재임을 뜻함
- Linux Container 기술을 사용하는 솔루션이므로 별도의 OS를 설치하지 않고 컨테이너를 생성하여 애플리케이션을 실행함
- 컨테이너를 생성할 이미지(서비스에 필요한 리소스를 모아둔 최소한의 단위)를 기반으로 운영함
- 이미지만 가지고 있다면 어느 시점에서든 동일한 리소스의 컨테이너를 생성할 수 있음
- 그 밖에 컨테이너간의 연결, 다양한 API 제공 등의 기능을 보유함

* ECS(Elastic Container Service)

- 클러스터에서 도커 컨테이너를 손쉽게 관리하기 위한 컨테이너 관리 서비스
- 클러스터는 Task(작업) 또는 서비스로 일컬어지는 컨테이너들의 집합
- 2가지 구성 요소로 시작 가능
 - EC2(Container Instance) : EC2를 생성하여 EC2 내에 Task(컨테이너가 수행할 작업)
 - Fargate : EC2를 생성하거나 컨테이너를 실행하기 위한 Orchestration을 AWS가 맡아 하는 서비스로, 관리가 용이함
- 하나의 클러스터 내에 다수의 Task 혹은 컨테이너 인스턴스로 구성됨
- 또한 ELB, EBS 볼륨, VPC, IAM과 같은 기능을 사용하여 구성 가능
- 즉 ECS 각 작업의 권한, ECS 액세스를 IAM으로 조절하거나, EC2 유형의 컨테이너 인스턴스만이 OS에 액세스 가능한 특징 등을 갖게 됨

**참고자료

- <https://12bme.tistory.com/288>
- https://www.oss.kr/info_techtip/show/f1c6db27-7caf-44b9-97bd-b4a021e0e5f2
- https://docs.aws.amazon.com/ko_kr/AmazonECS/latest/developerguide/Welcome.html
- <https://mintnlatte.tistory.com/114>
- <https://ko.wikipedia.org/>

- <http://pyrasis.com/docker.html>
- <https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html>
- <https://likefree.tistory.com/18>
- https://docs.aws.amazon.com/ko_kr/AmazonECS/latest/developerguide/docker-basics.html
- <https://www.slideshare.net/pyrasis/docker-fordummies-44424016>
- <https://aws.amazon.com/ko/docker/>
- <https://www.44bits.io/ko/keyword/linux-container>
- <http://blog.drakejin.me/Docker-araboza-1/>
- <https://jason-lim.tistory.com/7>
- <http://www.itworld.co.kr/insight/110748>
- <https://www.sangkon.com/hands-on-docker-part1/>

hwanyoung oh