

| EC2 | | | |
|-----|-----------------|-----|-----|
| 일시 | 2023.03.19. (일) | 작성자 | 김민경 |

- **EC2** (Elastic Compute Cloud) = **IaaS**(Infrastructure as a Service)

- os (어떤 운영체제: Linux, Windows, Mac OS)
- 컴퓨터 성능, CPU 개수
- RAM의 양
- 네트워크 카드
- sg (security group)
- 용량(network-attached: EBS, EFS, hardware(EC2 Instance Store) 등

-다양한 인스턴스 유형이 있음(각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공)

[문의하기](#)
[지원](#)
[고객지원](#)
[한국어](#)
[내 계정](#)
[로그인](#)

[AWS 계정 생성](#)

[제품](#)
[솔루션](#)
[요금](#)
[설명서](#)
[학습하기](#)
[파트너 네트워크](#)
[AWS Marketplace](#)
[고객 지원](#)
[이벤트](#)
[자](#)
[>](#)
[Q](#)

[Amazon EC2](#)
[개요](#)
[기능](#)
[요금](#)
[인스턴스 유형](#)
[FAQ](#)
[시작하기](#)
[리소스](#)

←

이러지 콘텐츠

범용

컴퓨팅 최적화

메모리 최적화

가속화된 컴퓨팅

스토리지 최적화

HPC 최적화

인스턴스 기능

인스턴스 성능 측정

- 최대 2.9GHz의 인텔 제온 스케일러블 프로세서(하스웰 E5-2666 v3)
- EC2에 특별히 최적화된 고주파수 인텔 E5-2666 v3(Haswell) 프로세서
- 추가 비용 없이 스토리지 향상을 위해 기본적으로 EBS에 최적화됨
- 인텔 82599 VF를 지원하는 향상된 네트워킹을 통해 더 뛰어난 네트워킹 성능
- Amazon VPC, Amazon EBS 및 64비트 HVM AMI 필요

| 인스턴스 | vCPU* | 메모리 (GiB) | 스토리지 | 전용 EBS 대역폭(Mbps) | 네트워크 성능 |
|------------|-------|-----------|--------|------------------|---------|
| c4.large | 2 | 3.75 | EBS 전용 | 500 | 중간 |
| c4.xlarge | 4 | 7.5 | EBS 전용 | 750 | 높음 |
| c4.2xlarge | 8 | 15 | EBS 전용 | 1,000 | 높음 |
| c4.4xlarge | 16 | 30 | EBS 전용 | 2,000 | 높음 |
| c4.8xlarge | 36 | 60 | EBS 전용 | 4,000 | 10기가비트 |

2) 종류

| General Purpose | Compute Optimized | Memory Optimized | Storage Optimized |
|-----------------------------|--|--|---|
| 다양한 작업에 적합 (웹서버, 코드 저장소) | 훌륭한 cpu & 컴퓨팅을 요구하는 작업 | 메모리에서 대규모 데이터셋을 처리하는 유형 작업에 빠른 성능 제공 | 로컬 스토리지에서 대규모 데이터셋에 액세스 시 |
| | -일부 데이터의 일괄처리 -고성능 웹서버가 필요시 -HPC 필요시 (High performance computing) | -고성능, 관계형/비관계형 데이터베이스 -분산 웹스케일 캐시 저장소 (ex) Elastic cache) | -OLTP(온라인 트랜잭션 처리) 시스템 -관계형, NoSQL 데이터베이스 -Data warehousing 애플리케이션 |

3. EC2 Instances Purchasing Options

1) On-Demand Instances

- 비용: 가장 비용 다(사용한 만큼 지불)
- 장기 약정x
- 추천: 단기적 워크로드, 중단x는 워크로드 필요 시, app 거동 예측x 때

***On-Demand**
:수요에 반응하는

2) Reserved Instances

- On-Demand보다 더 할인
- Instance Type, 리전, OS 등 예약
- 약정기간: 1 & 3년
- 지불옵션: 전체 선결제, 부분 선결제, 매달 비용 지불
- 추천: 사용량이 일정한 app, 장기 워크로드(ex) DB)



* Convertible Reserved Instance (전환형 예약 인스턴스)

- 유연한 인스턴스(Instance Type, 리전, OS 등 바꿀 수 0)
- 유연성이 더 크기 때문에 할인은 Reserved Instance보단 적음
- 장기 워크로드

3) Saving Plans

- 장기간 사용 시 할인받을 수 0 (사용량이 한도를 넘어서면 On-Demand 가격으로 청구)
- 약정기간: 1 & 3년
- 특정한 instance family & AWS region에 고정됨
- 유연하게 전환 가능(Instance Type, OS 등)
- 장기 워크로드

4) Spot Instances

- 할인 폭이 제일 큼(On-Demand에 비해, 가장 비용 효율적인 인스턴스)
- 지불할 수 있는 최대 스팟 가격 정의를 하고, 이 최대 스팟 가격보다 현재 인스턴스 가격이 낮을 때 사용
- 만약 spot 가격이 그 가격을 넘게 되면 인스턴스 손실 가능성0 (신뢰성↓)
- 추천: 서비스 중단에도 복구가 쉬운 워크로드
- 비추천: 중요한 작업 or DB 작업

5) Dedicated Hosts

- 가장 비쌈
- AWS 데이터 센터 내 하나의 서버 전체 임대(물리적 서버 전체를 예약하고 인스턴스 배치를 제어)
- 규정 준수 요건의 처리가 쉽고, 기존의 서버 결합 소프트웨어 라이선스 사용이 가능
- 구매옵션
 - ① On-Demand: 초당 비용 지불 or 1년/3년 예약
 - ② Reserved: 1년/3년 예약
- 추천: 라이선싱 모델과 함께 제공되는 소프트웨어인 경우, 규정이나 법규를 반드시 준수해야 하는 회사

6) Dedicated Instances

- 우리의 전용 하드웨어에서 실행
- 같은 계정에서 다른 인스턴스와 함께 하드웨어 공유

-인스턴스 배치 제어x

*Dedicated Hosts VS Dedicated Instances

- Dedicated Hosts: 우리가 물리적 서버 자체에 대한 접근권 갖고, 낮은 수준의 하드웨어에 대한 가시성 제공
- Dedicated Instances: 자신의 인스턴스를 자신만의 하드웨어에 가짐

7) Capacity Reservations

- 원하는 기간 동안 특정 AZ에서 On-Demand 인스턴스 예약 0
- 약정기간: x
- 인스턴스를 실행하는 것과는 무관하게 On-Demand 요금 부과
- 추천: 특정 AZ에 있어야 하는 단기적이고 중단x는 워크로드

*정리

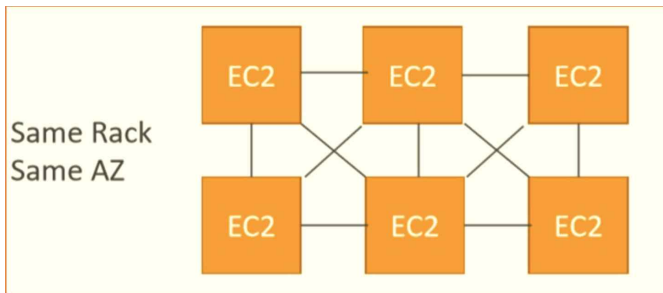
| | |
|-----------------------|--|
| On-Demand Instances | 리조트가 있고, 원할 때 언제나 리조트에 오고 전체 가격을 지불 |
| Reserved Instances | 미리 계획을 하고 그 리조트에 아주 오래 체류할 것이라는 걸 알고 있음(1-3년), 오래 체류할 것이기 때문에 많은 할인 받음 |
| Saving Plans | 내 리조트에서 일정한 금액을 지출할 것임을 알고 있음, 12개월 동안 300달러 지출 예정, 시간 지나면 객실 타입 변경0 |
| Spot Instances | 많은 할인 받을 수 0, 하지만 이 리조트에 나보다 객실 요금을 더 많이 낼 손님이 생긴다면 언제든지 쫓겨날 수 있음 |
| Dedicated Hosts | 자신만의 리조트 받음 |
| Capacity Reservations | 객실을 예약하기는 하는데 나의 체류 여부는 확실하지 않은 것, 체류하지 않아도 그 객실을 예약하는 전체 비용 지불 |

4. EC2 Placement Groups

-모든 인스턴스가 기본 하드웨어의 전반에 분산되도록 해 상호 관련 오류 위험을 줄여줌

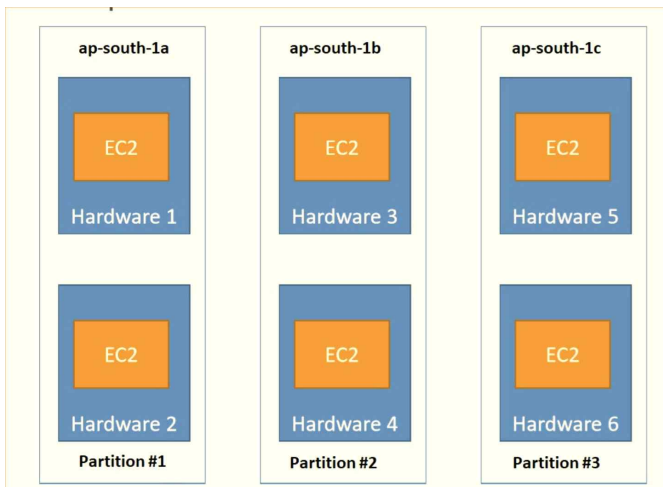
1) 종류

① Cluster (클러스터)



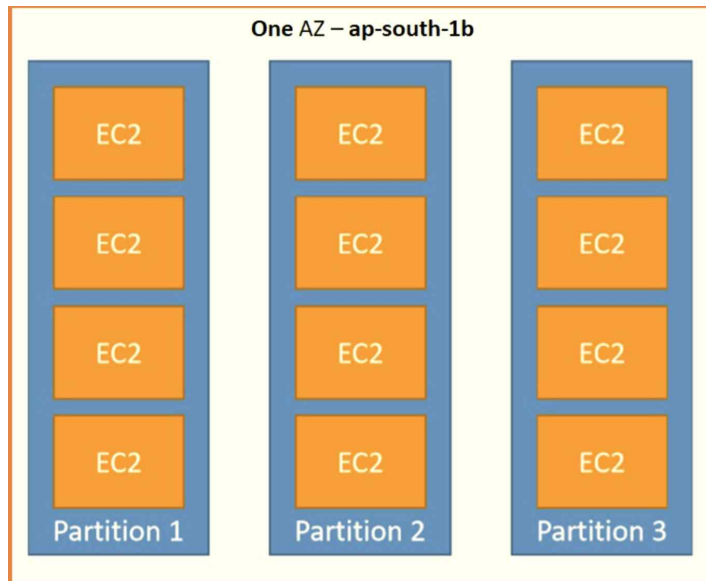
- 인스턴스를 같은 하드웨어 & AZ 안에 서로 근접하게 배치
- 단점: 하드웨어 실패 시 모든 EC2 인스턴스가 동시에 실패
- 추천: 짧은 net 지연시간, 높은 net 처리량

② Spread (분산)



- 모든 EC2 인스턴스가 각각 다른 하드웨어에 위치해 실패 위험을 최소화
- 장점: 여러 AZ에 걸쳐 있을 수 있음, 동시 실패 위험이 감소
- 단점: 배치그룹의 AZ당 7개 인스턴스로 제한됨
- 추천: 가용성을 극대화하고 위험을 줄여야 하는 app, 인스턴스 오류를 서로 격리해야 하는 경우

③ Partition (파티션)



- 한 파티션에 있는 인스턴스 그룹이 다른 파티션의 인스턴스 그룹과 하드웨어 공유하지 않은 것 (파티션 1이 다운돼도 파티션 2는 사용 가능)
- 각 파티션에는 많은 EC2 인스턴스가 있음, 파티션은 AWS의 랙을 나타냄
- 장점: 파티션은 여러 AZ에 걸쳐 있을 수 있음, 최대 수백 개 EC2 인스턴스 가질 수 있음(\Rightarrow Spread)
- 추천: HDFS, HBase, Cassandra

<https://docs.aws.amazon.com/AmazonECS/latest/pt1-architecture.html>