

EC2 · EBS · ENI

서버 한 대는 무엇으로 이루어져 있고, 어떻게 연결되는가



서버 한 대는 무엇인가

EC2로 빌리고, AMI로 찍어낸다.

무엇이 남고 무엇이 사라지나

정지와 종료, 그리고 EBS와 스냅샷.

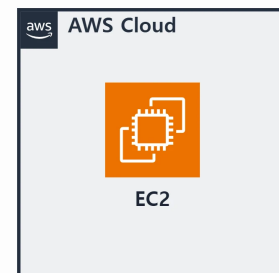
어떻게 연결되나

ENI 한 장에 IP도 보안그룹도 달린다.

01 EC2 — AWS가 빌려주는 서버

Elastic Compute Cloud

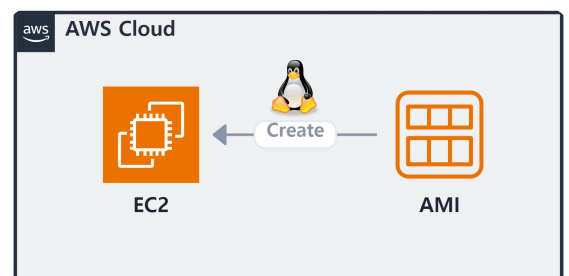
- AWS에서 서버를 빌려 쓰는 것
- 필요할 때 켜고, 필요 없으면 끄는 **가상 서버**.
- 늘리고 줄이기 쉬워서 **Elastic**이 붙었다. C가 2개라 **EC2**



02 AMI — 서버 템플릿

Amazon Machine Image

- 어떤 OS로, 어떤 설정으로 켤지 고르는 **템플릿**.
- 설정을 끝낸 서버를 다시 AMI로 구워 둘 수 있다.
- 그러면 똑같은 서버를 몇 대든 만들 수 있다.



🔧 만들 때 고르는 것 일곱 가지

콘솔 마법사의 순서 그대로

OS	Linux · Windows · macOS — AMI가 정한다	CPU	vCPU 몇 개. 타입 이름의 계열과 크기가 정한다
메모리	RAM. 같은 크기라도 계열마다 다르다	스토리지	네트워크로 붙는 EBS · EFS냐, 물리 서버에 직접 붙은 인스턴스 스토어나
네트워크	랜카드 속도, 그리고 공인 IP를 줄지 말지	방화벽	보안그룹 — 어디서 어느 포트로 들어오게 할지
첫 부팅 스크립트	User data — 처음 켤 때 딱 한 번 실행되는 스크립트		

03 인스턴스 타입 읽는 법

짧은 이름 하나가 **스펙표**

t3.micro

- t** 용도 — T·M 범용, C 컴퓨팅, R 메모리
- 3** 세대 — 숫자가 클수록 최신이고, 대개 더 싸고 빠르다
- micro** 크기 — 한 단계마다 메모리 두 배

용도부터 고르고 세대와 크기를 정한다.

타입 계열

시험 포인트

계열	용도	고르는 때	예
T	범용 · 몰아 쓰기(버스트)	평소 한가한 웹 / 개발 서버	t3.micro
M	범용	CPU와 메모리가 균형 잡힌 기본값	m7g.large
C	컴퓨팅 최적화	연산이 무거운 배치 · 게임 서버	c7i.xlarge
R / X	메모리 최적화	캐시 · 인메모리 DB	r7g.large
I / D	스토리지 최적화	고 IOPS · 대용량 로컬 디스크	i4i.large
G / P	가속 컴퓨팅	GPU 학습 · 추론 · 렌더링	g5.xlarge

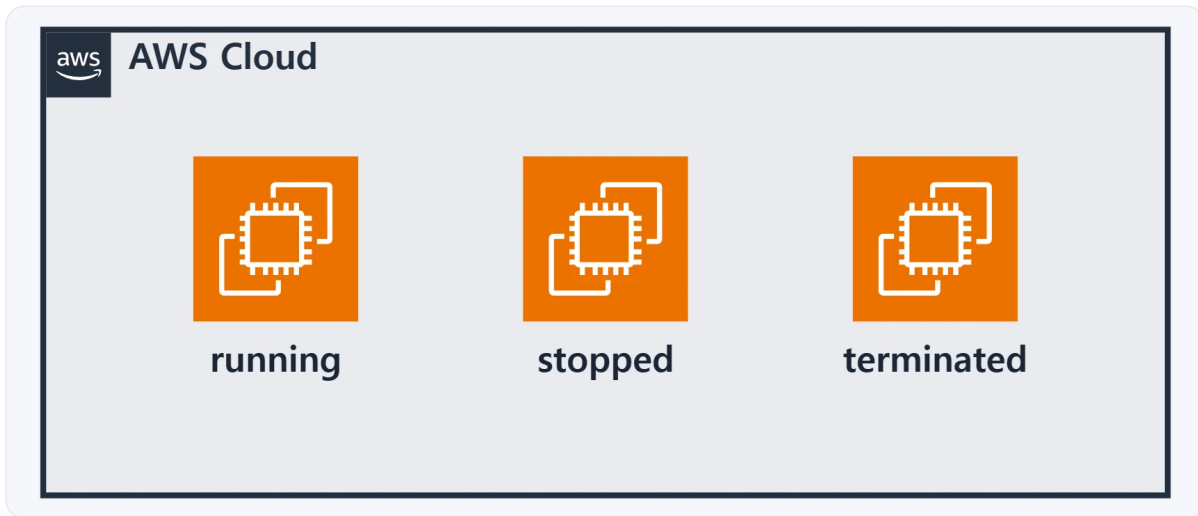
타입 크기

nano → micro → small	가장 작은 셋. 개발·테스트용
xlarge → 2xlarge → 4xlarge...	메모리 두 배, vCPU도 함께
.metal — 베어메탈	가상화 계층 없이 물리 서버를 통째로 쓴다. (물리 서버를 요구하는 라이선스일 때)

medium → large	실서비스의 시작점
a · g · i · n 접미사	AMD · Graviton · 인텔 · 네트워크 강화

04 상태 세 가지 — 요금이 걸리는 지점

켜져 있는 동안만 인스턴스 요금이 나간다.



running

켜져 있음. 요금이 나가는 중.

stopped

꺼져 있음. 인스턴스 요금은 멈춤.

terminated

완전 삭제. 되돌릴 수 없음.

상태 전이

시험 포인트

pending

켜지는 중

running

요금 발생

stopping

꺼지는 중

stopped

인스턴스 요금 0

terminated

복구 불가

reboot은 상태가 바뀌지 않는다. 재시작이라서, 공인 IP도 인스턴스 스토어도 그대로다. 최대 절전은 메모리 내용을 루트 볼륨에 저장해 두고 켜다가 그대로 되살린다.

배치 그룹 — 물리적으로 어떻게 놓나

클러스터	한 랙에 모은다. 지연이 가장 낮고, 랙이 죽으면 같이 죽는다
분산	서로 다른 하드웨어에 흩어 놓는다. AZ당 인스턴스 수에 제한이 있다
파티션	랙 묶음 단위로 나눈다. HDFS·Cassandra 같은 분산 저장소용

💰 요금 모델

On-Demand

쓰는 만큼 초 단위로. 기본값이자 가장 비싼 선택.

Savings Plans

1~3년, 시간당 일정 금액(\$/h)을 약정하고 할인.

Reserved

1~3년 동안 특정 타입을 쓰겠다고 예약한다. 금액이 아니라 인스턴스를 약정하는 것.

Spot

남는 용량을 최대 90% 싸게. 2분 통보 후 회수당한다.

Dedicated Host

물리 서버를 통째로. 라이선스가 코어에 묶일 때.

Capacity Reservation

용량만 미리 잡아 둔다. 싸지는 게 아니라 자리를 확보하는 것.

🔍 어느 것을 고르나

24시간 도는 서버	Savings Plans — 시간당 금액을 약정하고 깎는다
중간에 끊겨도 되는 작업	Spot — 최대 90% 싸고, 2분 통보 뒤 회수당한다
잠깐 켜 보는 것	On-Demand — 약정이 없다. 가장 비싸고 가장 자유롭다
코어 단위 라이선스	Dedicated Host — 물리 서버를 통째로 빌린다

📋 요금 항목

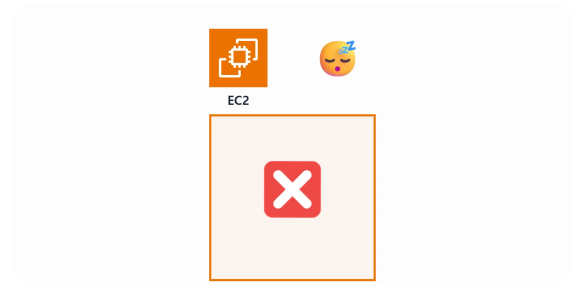
정지가 멈추는 것은 첫 줄뿐

—	언제 나가나	멈추려면
인스턴스 시간	running일 때만	정지하면 멈춘다
EBS 볼륨	만들어 둔 내내 (GB-월)	볼륨을 지워야 멈춘다. 정지로는 안 멈춘다
스냅샷	보관하는 내내 (GB-월)	삭제한다
공인 IPv4	할당 받고 계속 (시간당)	안 쓰는 EIP 는 반납 한다. 정지 중에도 나간다
데이터 전송	밖으로 나갈 때 (GB)	들어오는 트래픽은 무료, 같은 AZ 안의 사설 IP 통신은 무료

05 Instance Store

정지하면 사라지는 것

- EC2에 붙은 **인스턴스 스토어**는 정지하면 데이터가 사라진다.
- 물리 서버에 직접 붙은 임시 디스크라 빠르다
- **재부팅**에는 그대로지만, 정지·종료에는 사라진다.



06 EBS — 네트워크로 연결

Elastic Block Store

- 인스턴스가 사라져도 남는다.
- EBS는 떼어서 **다른 인스턴스에 붙일 수 있다**.
- 수명이 다르므로 요금도 따로 나간다 (인스턴스를 꺼도 볼륨은 청구)



👉 정지(stopped) vs 종료(terminated)

시험 포인트

—	정지	종료
인스턴스 요금	멈춤	멈춤
EBS 볼륨	그대로 청구	루트 볼륨은 삭제
안에 있던 데이터	그대로	사라진다
공인 IP	바뀐다 (EIP는 예외)	회수된다
되돌리기	재시작 가능	불가

🗄️ 디스크 세 가지, 헷갈리는 지점

EBS

네트워크로 붙는 디스크. 인스턴스를 지워도 남고, 떼서 다른 인스턴스에 붙일 수 있다.

인스턴스 스토어

물리 서버에 직접 붙은 임시 디스크. 빠르지만 정지·종료하면 사라진다. 재부팅은 그대로.

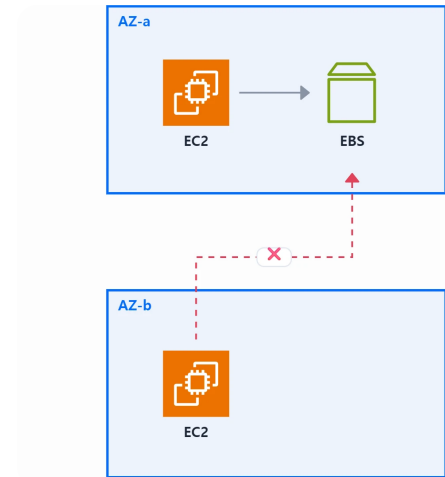
스냅샷

EBS를 S3에 증분 백업한 것. 다른 AZ·리전으로 옮기거나 AMI로 굽는 출발점.

07 볼륨은 하나의 가용영역에 속한다

다른 AZ의 인스턴스에는 **붙일 수 없다**.

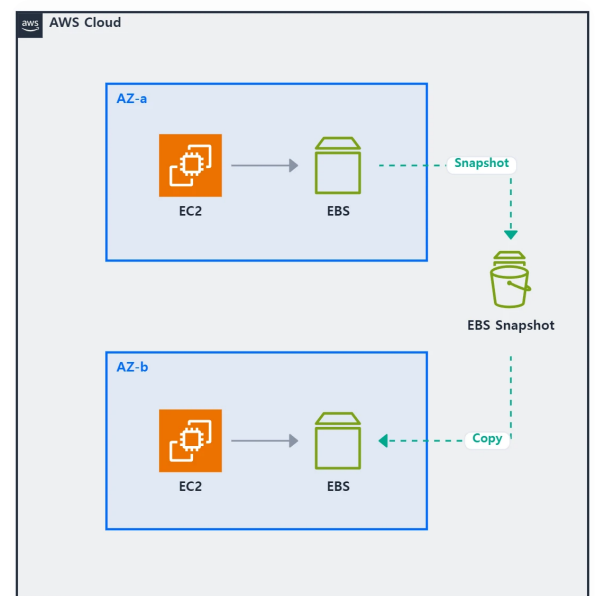
- 볼륨과 인스턴스가 같은 AZ에 있어야 연결된다.
- 그래서 AZ를 옮기는 일은 **복사**이지 이동이 아니다.
- 리전 단위로 쓰고 싶다면 EFS처럼 다른 서비스를 써야 한다.



08 스냅샷 — AZ와 리전을 건너는 법

스냅샷을 떼서 **다른 AZ·리전에 볼륨을 새로 만든다**.

- S3에 **증분**으로 저장된다. 처음 한 번만 전체를 저장
- 스냅샷에서 만든 볼륨은 원본보다 크게 만들 수 있다.
- AMI를 굽는 출발점도 결국 이 스냅샷이다.



스냅샷 — 뭐 하는 물건인가

시험 포인트

증분 백업

S3에 저장되고 처음 한 번만 전체를 저장.

AZ·리전 이동

복사해서 다른 AZ·리전에서 볼륨을 만든다. (볼륨이 AZ를 넘는 유일한 방법)

AMI의 출발점

인스턴스를 AMI로 굽는 일은 결국 루트 볼륨의 스냅샷을 쓰는 일이다.

자동화와 보관

DLM으로 일정을 걸어 자동으로 뜨고 지운다. 오래된 것은 아카이브 보관으로 내린다.

암호화 — 만들 때 정해야 하는 것

KMS 키로

볼륨을 만들 때 켜다. 데이터도, 스냅샷도, 그 스냅샷에서 만든 볼륨도 함께 암호화된다.

나중에는 못 켜다

기존 볼륨은 스냅샷 → 암호화 복사 → 새 볼륨을 거쳐야 한다.

성능은 그대로

체감할 만한 손해가 없으므로 기본으로 켜 두는 편이 낫다.

볼륨의 규칙 — 어디에 붙고 언제 사라지나

시험 포인트

가용영역	볼륨은 한 AZ 에 속한다. 같은 AZ의 인스턴스에만 붙는다	연결	한 볼륨은 한 인스턴스에만 붙는다 (io1·io2 Multi-Attach 는 예외)
수명	인스턴스와 따로 산다 — 떼어서 다른 인스턴스에 붙일 수 있다	루트 볼륨	종료 시 삭제가 기본 꺼짐 . 데이터 볼륨은 기본 꺼짐
정지 중	볼륨은 그대로 남고 그대로 과금 된다		

EBS 볼륨 타입 — 무엇을 고르나

루트는 SSD 계열만

타입	무엇	고르는 때
gp3	범용 SSD (기본값)	기본 3,000 IOPS · 125 MB/s. 용량과 성능이 따로
gp2	범용 SSD (구형)	1 GiB당 3 IOPS — 크기가 성능을 정한다
io2	프로비저닝 IOPS SSD	IOPS를 사서 쓴다. 지연에 민감한 DB
st1	처리량 최적화 HDD	큰 파일을 순차로. 로그 · 빅데이터
sc1	콜드 HDD	가장 싸다. 어쩌다 한 번 읽는 데이터

↗ 만든 다음 바꿀 수 있는 것

늘리는 쪽으로만

크기 — 늘리기만

켜 둔 채로 키울 수 있고 **줄일 수는 없다**. 줄이려면 새 볼륨을 만들어 옮겨 담는다.

타입 — 바꿀 수 있다

gp2 → gp3처럼 볼륨을 살려 둔 채 바꾼다. 바꾼 뒤 성능이 자리 잡을 때까지 시간이 걸린다.

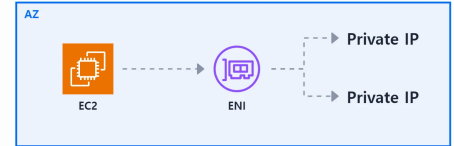
한도

볼륨 1 GiB ~ 16 TiB, gp3 최대 16,000 IOPS · 1,000 MB/s. 그보다 위의 인스턴스 자체의 EBS 대역폭이 막는다.

09 ENI — EC2가 통신할 때 쓰는 가상 랜카드

Elastic Network Interface

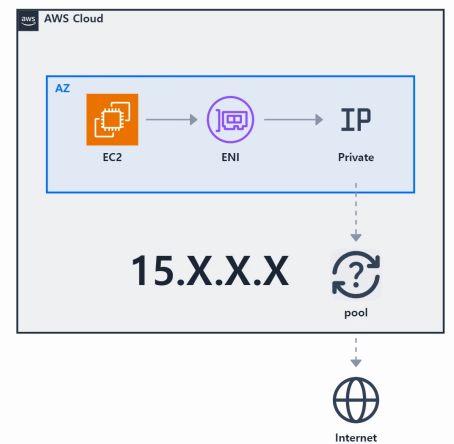
- ENI도 **한 가용영역**에 속한다 — 볼륨과 같은 규칙.
- 인스턴스마다 기본 ENI가 하나, 인스턴스 타입에 따라 더 붙일 수 있다.
- 떼어서 다른 인스턴스에 옮겨 붙일 수 있다 — 주소를 통째로 넘기는 셈.
- IP도 MAC도 보안그룹도 인스턴스가 아니라 **이 카드**에 달린다.



10 퍼블릭 IP는 어디에 붙나

인스턴스가 아니라 **ENI의 프라이빗 IP**에 붙는다.

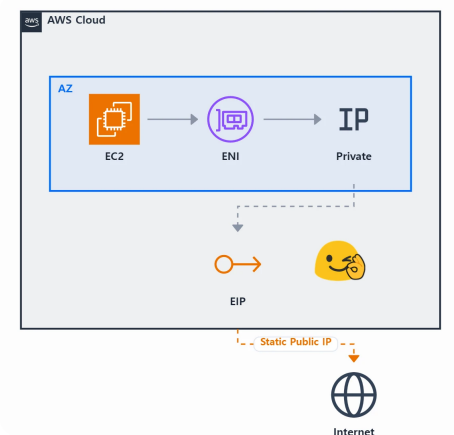
- 켜고 꺼질 때 AWS 풀에서 **랜덤으로** 하나 받는다 — 고를 수 없다.
- 프라이빗 IP는 살아 있는 동안 고정
- 2024-02부터 퍼블릭 IPv4는 계속 **시간당 과금**된다.
- 인스턴스를 중지했다 켜면 퍼블릭 IP는 **바뀐다**.



11 EIP — 주소를 고정한다

Elastic IP

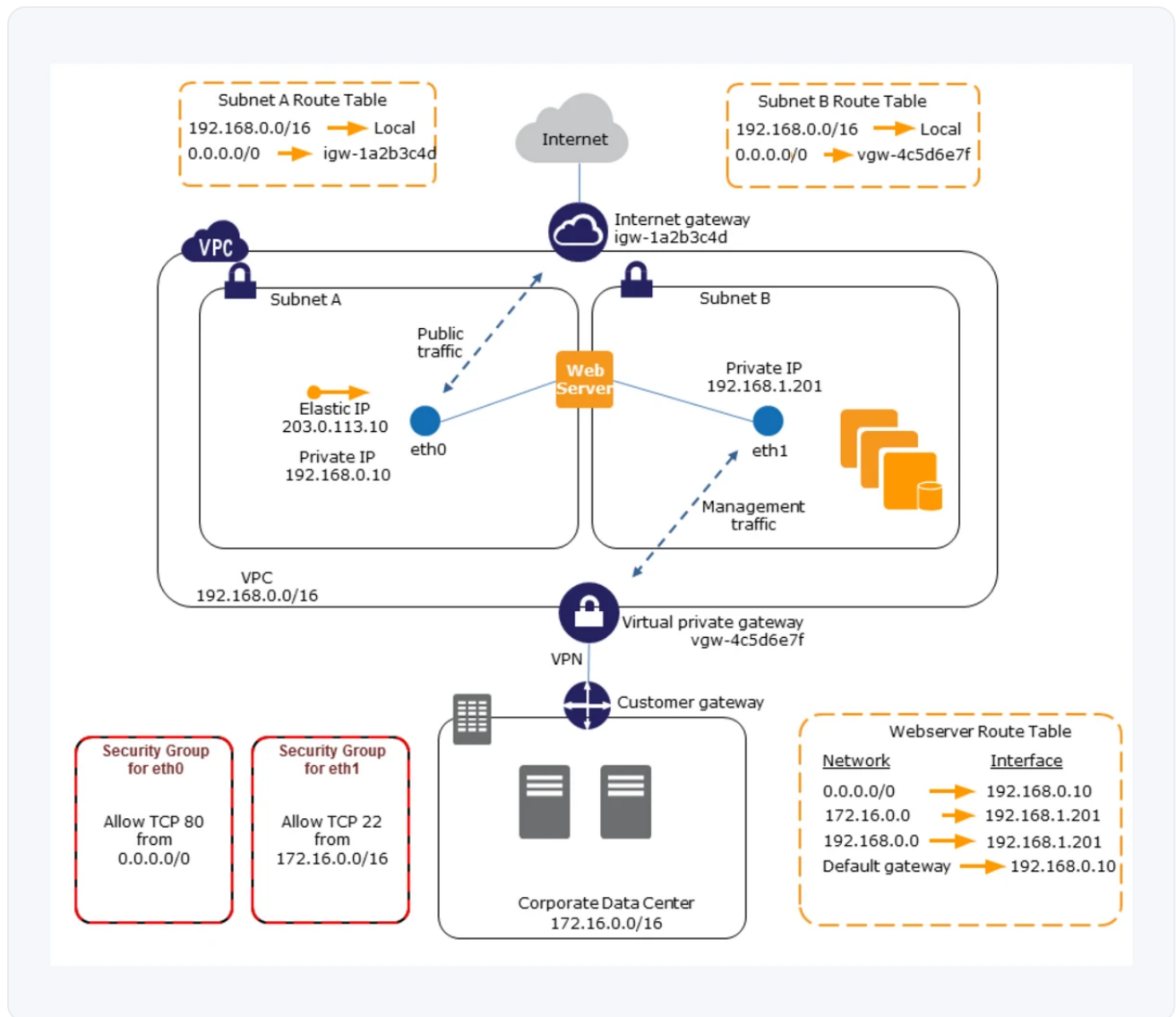
- 계정에 할당받아 ENI에 붙이는 **고정 공인 IP**.
- 붙여 두든 아니든 시간당 과금. 안 쓰면 반납하자.
- 리전에 속한다. 계정당 기본 5개
- 주소를 고정하는 다른 방법은 도메인(Route 53)이다.



ENI의 규칙

[시험 포인트](#)

가용영역	ENI도 한 AZ 에 속한다	기본 ENI	인스턴스마다 하나(eth0). 지울 수 없고 인스턴스와 함께 사라진다
추가 ENI	인스턴스 타입이 허용하는 만큼 추가 가능.	프라이빗 IP	ENI마다 기본 1개, 타입에 따라 보조 IP를 더 가진다
보안그룹	인스턴스가 아니라 ENI 에 붙는 stateful 허용 규칙	MAC 주소	카드에 붙어 다닌다 (MAC에 묶인 라이선스라면 ENI째로 옮긴다)



eth0은 Subnet A에서, eth1은 Subnet B에서 각각 그 서브넷의 대역으로 사설 IP를 받고, 보안그룹도 카드마다 따로 붙는다.

[AWS 「Multiple network interfaces」 문서 ↗](#)

🔗 그림이 말하는 것 네 가지

시험 포인트

ENI는 서브넷에 속한다

사설 IP를 그 서브넷의 대역에서 받는다.

보안그룹은 카드마다

인스턴스가 아니라 ENI에 붙는다 — 그림의 eth0은 80·443을, eth1은 22만 연다.

EIP는 사설 IP 위에 얹힌다

카드의 사설 IP 하나에 공인 IP 하나. 카드를 떼어 옮기면 주소가 통째로 따라간다.

한 인스턴스가 서브넷 둘에

같은 가용영역이라면 카드를 서로 다른 서브넷에 걸칠 수 있다. 공개 서브넷, 관리용 사설 서브넷으로 구성

IP 주소 세 가지

시험 포인트

—	어떻게 받아	정지 후 다시 켜면	요금
프라이빗 IP	서브넷에서 자동	그대로	무료
퍼블릭 IP	컬 때 풀에서 랜덤	바뀐다	시간당 \$0.005
Elastic IP	계정에 할당해 붙인다	그대로	시간당 \$0.005 (안 붙여도)

🔒 접속과 권한 — 사고가 나는 곳

키페어

프라이빗 키(.pem)는 만들 때 **한 번만** 받는다. 잃으면 재발급이 없다.

보안그룹

허용 규칙만 있다 — 어디서 어느 포트로 들어오게 할지. 22·3389를 0.0.0.0/0으로 열지 말 것.

IAM 역할

인스턴스 프로파일로 붙인다. 액세스 키를 서버에 두지 말 것.

☑ 실전 체크 리스트 — 인스턴스 띄우기 전 확인

- ☐ 키페어를 받아 두었는가 — 다시 내려받을 수 없다
- ☐ 루트 볼륨의 **종료 시 삭제** 옵션을 확인했는가
- ☐ 종료 방지(termination protection)를 켜는가
- ☐ 태그(Name, Env, Owner)를 붙였는가 — 없으면 청구서에서 못 찾는다
- ☐ 보안그룹이 22/3389를 0.0.0.0/0으로 열고 있지 않은가
- ☐ User data는 **첫 부팅에 한 번만** 실행된다는 걸 아는가
- ☐ IAM 역할을 붙였는가 — 액세스 키를 넣지 말 것
- ☐ 볼륨 암호화를 켜는가 — **만들 때만** 켤 수 있다

범위 — 무엇이 리전에 속하고 무엇이 AZ에 속하나

시험 포인트

대상	범위	알아 둘 것
AMI	리전	만든 리전에서만 보인다 — 다른 리전에서 쓰려면 복사 한다
스냅샷	리전	리전 간 복사가 되고, 그것이 볼륨이 리전을 넘는 유일한 길
EBS 볼륨	가용영역	같은 AZ의 인스턴스에만 붙는다
인스턴스	가용영역	컬 때 고른 AZ에 산다 — 옮기려면 AMI로 다시 띄운다
EIP	리전	같은 리전 안이면 AZ를 넘어 옮겨 붙는다. 계정당 리전 기본 5개

❗ 꺼야 하나 — 컨 채로 되는 일, 정지해야 하는 일

스냅샷은 안 꺼도 된다

하려는 일	인스턴스	알아 둘 것
스냅샷 쓰기	컨 채로	디스크에 쓰인 것 까지만 담긴다 — 걱정되면 잠깐 멈추고 뜬다
볼륨 크기·타입 바꾸기	컨 채로	키운 뒤 OS에서 파일시스템을 늘려야 실제로 커진다
데이터 볼륨 붙이기·떼기	컨 채로	떨 때는 마운트를 먼저 푼다
루트 볼륨 떼기	정지	실행 중에는 뗄 수 없다
인스턴스 타입 바꾸기	정지	갓다 켜는 사이에 바꾼다 — 퍼블릭 IP도 함께 바뀐다
User data 고치기	정지	정지 상태에서에서만 바꾼다. 그래도 기본은 첫 부팅에 한 번
AMI 굽기	컨 채로	기본은 재부팅하고 굽는다 — 재부팅 없이 구우면 데이터 정합성은 내 책임

🔄 되돌리기 — 무엇을 잃었을 때 무엇으로

잃은 크기가 방법을 정한다

파일 하나	볼륨을 그대로 두고, 스냅샷에서 새 볼륨을 떼서 붙인 뒤 꺼낸다
볼륨 통째	스냅샷에서 볼륨을 복원한다. 같은 AZ면 그대로, 다른 AZ면 복사부터
부팅 불가	루트 볼륨을 떼어 다른 인스턴스에 데이터 디스크로 붙여 고친다
서버 통째	AMI를 떠 뒀다면 그것으로 새로 띄운다
종료해 버림	루트 볼륨은 사라진다. 스냅샷이나 AMI가 없으면 되돌릴 길이 없다

☁ 풀어 보자!

실전 문제

① 개발 서버 한 대를 매일 저녁 정지하고 아침에 시작한다. 그런데 꺼 둔 시간의 요금이 0원이 되지 않는다. 정지 중에도 계속 청구되는 것을 모두 고르면? **정답 2개**

- Ⓐ 인스턴스 시간 요금
- Ⓑ 붙어 있는 루트 EBS 볼륨
- Ⓒ 인스턴스 스토어에 남아 있던 데이터
- Ⓓ 인스턴스에 붙여 둔 EIP
- Ⓔ 인스턴스가 주고받던 데이터 전송

② 부하에 따라 10~50대까지 늘었다 줄었다 하는 인스턴스들이, 여러 가용영역에 걸쳐, 같은 50GB 볼륨을 함께 읽고 써야 한다. 무엇을 붙이나?

- Ⓐ 볼륨 하나를 만들어 모든 인스턴스에 연결한다
- Ⓑ 인스턴스마다 볼륨을 하나씩 만들고 서로 동기화한다
- Ⓒ EFS 파일 시스템을 만들어 전부 마운트한다
- Ⓓ 인스턴스 스토어에 두고 필요할 때 복사한다

③ 랜덤 읽기·쓰기가 아주 많은 DB. 성능이 일정해야 하고, 데이터는 인스턴스보다 오래 남아야 한다. 어느 디스크인가?

- Ⓐ 인스턴스 스토어 — 물리 서버에 직접 붙어 가장 빠르다
- Ⓑ gp2 — 볼륨을 크게 잡으면 IOPS도 따라 올라간다
- Ⓒ st1 — 처리량 최적화 HDD
- Ⓓ io2 — 프로비저닝 IOPS SSD

정답과 해설

- ① **B · D** 정지로 멈추는 것은 **인스턴스 시간 요금 하나뿐**이다. 볼륨은 인스턴스와 수명이 달라서 그대로 남고, 남아 있는 동안 계속 청구된다. 2024-02부터 공인 IPv4와 EIP는 인스턴스를 꺼도 비용은 그대로 나간다.
- ② **C** 볼륨은 한 AZ에 속하고, 한 볼륨은 한 인스턴스에 붙는다. io1·io2의 Multi-Attach가 예외지만 그것도 같은 AZ의 Nitro 인스턴스 최대 16대까지라, 여러 AZ에 걸친 50대에는 쓸 수 없다. 여럿이 같은 데이터를 동시에 쓰는 순간 필요한 것은 블록 장치가 아니라 **파일 시스템**이고, EFS는 리전 단위로 여러 인스턴스가 함께 마운트한다. 인스턴스 스토어는 물리 서버에 직접 붙은 디스크라 공유 자체가 되지 않는다.
- ③ **D** 조건이 두 개고 각각 보기를 하나씩 지운다. **오래 남아야 한다**가 인스턴스 스토어를 지운다 — 빠르지만 정지·종료에 사라진다. **일정해야 한다**가 gp2를 지운다: 1 GiB당 3 IOPS라 성능이 용량에 묶이고, 작은 볼륨은 버스트에 기댄다. st1은 큰 파일을 순차로 넘기는 타입이라 랜덤 IO가 아니다. io2는 만들 때 적어 둔 IOPS를 계속 내주는 타입이고, 그래서 "일관된 성능"이 조건일 때의 답이다.

⚠ 자주 걸리는 함정

시험 단골

정지하면 요금 0원이라고 생각하기

인스턴스 요금만 멈춘다. 붙어 있는 EBS도, 잡아 둔 공인 IPv4도 그대로 청구된다.

크기부터 고르기

용도(알파벳)를 먼저 정해야 한다. 같은 크기라도 C와 R은 다른 서버다.

User data가 매번 실행되는 줄 알기

최초 첫 부팅에만 한 번만 동작한다.

다른 AZ의 볼륨을 붙이려 하기

EBS는 한 AZ에 속한다. AZ를 넘는 길은 스냅샷을 떼서 복사하는 것뿐.

종료 버튼을 정지인 줄 알고 누르기

루트 볼륨까지 사라진다. 종료 방지를 켜 두는 것이 유일한 안전장치.

퍼블릭 IP를 고정으로 믿기

정지했다 켜면 바뀐다. 주소가 고정이어야 하면 Elastic IP나 도메인을 쓴다.

볼륨을 줄일 수 있다고 생각하기

늘리는 쪽으로만 가능. 줄이려면 새 볼륨을 만들어 옮겨 담아야 한다.

보안그룹으로 막으려 하기

보안그룹에는 거부 규칙이 없다. 허용만 적는다 — 막는 것은 네트워크 ACL의 일이다.

📖 용어 — 한 줄로

AMI 서버를 찍어내는 템플릿. OS와 초기 설정이 들어 있다.

인스턴스 스토어 물리 서버에 직접 붙은 임시 디스크. 정지·종료하면 사라진다.

스냅샷 EBS를 S3에 증분 백업한 것.

종료 방지 종료 API를 막는 스위치. 정지는 막지 않는다.

Graviton AWS가 만든 ARM CPU. 이름에 g가 붙는다.

가용영역(AZ) 리전 안의 독립된 데이터센터 묶음. EBS도 ENI도 여기에 속한다.

EIP 계정에 할당받아 붙이는 고정 공인 IP. 붙여 두든 아니든 과금.

DLM 스냅샷을 일정대로 뜨고 지우는 수명주기 관리자.

인스턴스 타입 용도·세대·크기를 한 이름으로 적은 스펙표.

EBS 네트워크로 붙는 디스크. 인스턴스를 지워도 남고, 떼서 옮길 수 있다.

User data 첫 부팅에 한 번, 루트로 실행되는 스크립트.

전용 호스트 물리 서버 자체를 빌린다. 소켓·코어 단위 라이선스용.

IMDS 인스턴스가 자기 정보를 물어보는 내부 전용 주소 (169.254.169.254).

ENI EC2가 통신할 때 쓰는 가상 랜카드. IP·MAC·보안그룹을 들고 다닌다.

Multi-Attach 한 io1-io2 볼륨을 같은 AZ의 여러 인스턴스에 붙이는 기능.

한 줄 정리

서버 한 대는 EC2(계산) · EBS(디스크) · ENI(랜카드)다. 셋 다 한 가용영역에 묶이고, 그 밖으로 나가려면 스냅샷을 뜨거나 주소를 고정해야 한다.

EC2

AMI

인스턴스 타입

상태

EBS

스냅샷

ENI

EIP

공식 문서

AWS 설명서 · 영상 · 글

- 
EC2 사용 설명서 ↗ 1쪽
 인스턴스 · AMI · 수명주기의 원본
docs.aws.amazon.com/ec2
- 
인스턴스 수명주기 ↗ 3 · 5쪽
 상태마다 무엇이 남고 무엇이 사라지는가
docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html
- 
AMI ↗ 1쪽
 이미지가 담는 것과 담지 않는 것
docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html
- 
인스턴스 타입 ↗ 2쪽
 계열 이름을 읽는 법, 세대별 표
aws.amazon.com/ec2/instance-types
- 
EBS 사용 설명서 ↗ 5 · 7쪽
 볼륨이 인스턴스 밖에 있다는 것의 전부
docs.aws.amazon.com/ebs/latest/userguide/what-is-ebs.html
- 
EBS 볼륨 종류 ↗ 7쪽
 gp3 · io2 · st1의 IOPS와 처리량
docs.aws.amazon.com/ebs/latest/userguide/ebs-volume-types.html
- 
스냅샷 ↗ 6 · 11쪽
 볼륨을 가용영역 밖으로 옮기는 유일한 길
docs.aws.amazon.com/ebs/latest/userguide/ebs-snapshots.html
- 
ENI ↗ 8 · 9쪽
 카드를 떼어 다른 인스턴스에 붙이기
docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html
- 
탄력적 IP ↗ 8 · 10쪽
 주소를 고정한다는 것과 그 요금
docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html
- 
EC2 요금 ↗ 4쪽
 온디맨드 · 예약 · 스팟의 오늘 값
aws.amazon.com/ec2/pricing
- 
요금 계산기 ↗ 4쪽
 이 시트의 숫자를 내 구성으로 바꿔 보기
calculator.aws
- 
Amazon EC2로 첫 워크로드 시작하기 ↗ 1쪽
 인스턴스 하나를 처음부터 띄워 보는 한국어 세션. AWS Korea 공식 채널
www.youtube.com/watch?v=Mg6lLromiQM
- 
One to Many: Evolving VPC Design ↗ 11쪽
 한 대에서 여러 대로 갈 때 무엇이 먼저 부러지나
aws.amazon.com/blogs/architecture/one-to-many-evolving-vpc-design
- 
보안그룹 ↗ 9 · 10 · 13쪽
 허용만 적는 방화벽. 거부는 네트워크 ACL의 일 — 다음에 볼 것
docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
- 
EC2 Auto Scaling ↗ 4쪽
 한 대를 여러 대로, 수요에 맞춰 — 다음에 볼 것
docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html