

AWS SAA-C03 통합 가이드 - Part 2: 복원력 아키텍처 (26%)

고가용성 및 장애 복구 - 시험 빈출 영역!

복원력 도메인 핵심 개념

RTO vs RPO

- **RTO (Recovery Time Objective)**: 복구 시간 목표
- **RPO (Recovery Point Objective)**: 데이터 손실 허용 범위

고가용성 원칙

1. **Multi-AZ**: 최소 2개 AZ 사용
2. **자동 페일오버**: 수동 개입 최소화
3. **무상태(Stateless)**: 세션은 외부 저장
4. **느슨한 결합**: 서비스 간 비동기 통신

Auto Scaling + ELB

WHY

- 단일 장애점(SPOF) 제거가 고가용성의 핵심
- 트래픽 급증 시 자동 확장으로 비용 효율성 + 안정성 확보
- 장애 발생 시 자동 복구로 서비스 중단 최소화

HOW

Auto Scaling Group (ASG)

구성 요소



Launch Template/Configuration

↓

Auto Scaling Group

- └─ Desired Capacity (원하는 용량)
- └─ Min Size (최소 용량)
- └─ Max Size (최대 용량)
- └─ Scaling Policies (확장 정책)

Health Check 유형

유형	설명	권장
EC2 Health Check	인스턴스 상태만 확인	기본
ELB Health Check	애플리케이션 레벨 확인	<input checked="" type="checkbox"/> 권장
Custom Health Check	사용자 정의 조건	특수한 경우

Scaling Policy

정책 유형	설명	사용 사례	권장도
Target Tracking	목표 메트릭 유지 (예: CPU 70%)	대부분의 경우	★★★
Step Scaling	임계값별 단계적 확장	세밀한 제어	★★
Simple Scaling	단일 조건 (레거시)	단순한 경우	★
Scheduled Scaling	시간 기반	예측 가능한 트래픽	★★
Predictive Scaling	ML 기반 예측	반복적 패턴	★★★

Target Tracking 예시



json

```
{  
  "TargetValue": 70.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ASGAverageCPUUtilization"  
  }  
}
```

수명 주기 허크 (Lifecycle Hooks)

- **Pending**: 인스턴스 시작 전 작업 (예: 소프트웨어 설치)
- **Terminating**: 인스턴스 종료 전 작업 (예: 로그 백업)
- EventBridge + Lambda로 자동화 가능

Cool Down vs Warm Up

- **Cool Down**: 스케일링 후 대기 시간 (기본 300초)
- **Warm Up**: 새 인스턴스가 워밍업하는 시간

ELB (Elastic Load Balancer) 비교

항목	ALB	NLB	GLB
OSI Layer	Layer 7 (HTTP/HTTPS)	Layer 4 (TCP/UDP)	Layer 3 (IP)
성능	높음	초고성능 (100만 RPS)	초고성능
라우팅	경로/호스트/헤더/쿼리 IP/포트	IP	
고정 IP	✗	✓ (Anycast)	✓
WebSocket	✓	✓	✗
SSL 종료	✓	✓ (TLS)	✗
Target 유형	Instance, IP, Lambda	Instance, IP, ALB	GWLB Endpoint
사용 사례	웹 애플리케이션	DB, 게임, IoT	방화벽, IDS/IPS

ALB 고급 기능

- **Path-based Routing**: /api/* → Target Group A
- **Host-based Routing**: api.example.com → Target Group B
- **HTTP Header Routing**: X-Custom-Header 기반
- **Query String Routing**: ?version=v2 기반
- **Lambda as Target**: 서비스 통합
- **Weighted Target Groups**: A/B 테스팅

NLB 특징

- **고정 IP**: Elastic IP 할당 가능
- **초저지연**: 마이크로초 단위
- **Connection Draining**: Deregistration Delay
- **Preserve Source IP**: 클라이언트 IP 유지

Cross-Zone Load Balancing

ELB 기본 설정	비용
ALB 활성화	무료
NLB 비활성화	AZ 간 데이터 전송 비용
CLB 비활성화	AZ 간 데이터 전송 비용

효과

- AZ 간 균등 분산 (노드 수 차이 보정)
- 예: AZ1 (10 인스턴스), AZ2 (2 인스턴스) → 각 인스턴스 동일 부하

Sticky Session (Session Affinity)

- **ALB**: 쿠키 기반
 - Application Cookie
 - Duration Cookie
- **NLB**: 클라이언트 IP 기반
- **주의**: 부하 불균형 가능 (ElastiCache 권장)

Connection Draining / Deregistration Delay

- 인스턴스 제거 시 기존 연결 유지

- 기본: 300초
- 0-3600초 설정 가능

✳️ SCENARIO

상황	정답	이유
CPU 70% 이상 시 자동 확장	Target Tracking Policy	가장 간단하고 효과적
평일 오전 9시 트래픽 급증	Scheduled Scaling	예측 가능한 패턴
EC2 장애 자동 교체	ASG + ELB Health Check	애플리케이션 레벨 확인
HTTPS 경로 기반 라우팅	ALB	Layer 7 기능
고성능 TCP, 고정 IP 필요	NLB	Layer 4 + 고정 IP
방화벽/IDS 배포	GLB	Layer 3 트래픽 검사
세션 유지 필요	Sticky Session (단기), ElastiCache (권장)	상태 관리
A/B 테스트 (70:30)	ALB Weighted Target Groups	트래픽 비율 제어
인스턴스 종료 전 로그 백업	Lifecycle Hook (Terminating)	자동화

🎯 시험 키워드 매칭

- "자동 확장" → ASG + Target Tracking
- "고정 IP" → NLB
- "경로 기반 라우팅" → ALB
- "초고성능" → NLB
- "예측 가능한 트래픽" → Scheduled Scaling
- "반복 패턴" → Predictive Scaling

🚫 TRAP

- ✗ ELB만으로 고가용성 보장 (ASG 병행 필수)
- ✗ NLB에 HTTPS 인증서 직접 연결 (TLS 종료는 지원, ALB 권장)
- ✗ Sticky Session으로 모든 상태 관리 (ElastiCache/DynamoDB 권장)
- ✗ Cool Down 너무 짧으면 불필요한 확장 발생
- ✗ ALB는 고정 IP 지원 안 함 (NLB 또는 Global Accelerator 사용)

💬 SQS & SNS (비동기 메시징)

🧠 WHY

- 동기 호출 → 비동기 전환으로 시스템 결합도 감소
- 장애 전파 차단 + 메시지 손실 방지
- 트래픽 스파이크 완화 (버퍼 역할)

⚙️ HOW

SQS (Simple Queue Service)

Queue 유형 비교

항 목	Standard Queue	FIFO Queue
순서 보장	✗ (Best-effort)	✓ (정확한 순서)
중복 메시지	가능 (최소 1회 전달)	✗ (정확히 1회)
처리량	무제한	3,000 TPS (배치 30,000)
메시지 그룹	✗	✓ (Message Group ID)
Content-based Deduplication	✗	✓
사용 사례	대부분의 경우	주문 처리, 금융 거래

핵심 개념

Visibility Timeout



1. Consumer가 메시지 읽기
↓
2. 메시지가 Visibility Timeout 동안 숨김 (기본 30초)
↓
- 3-a. 처리 완료 → 메시지 삭제
- 3-b. 처리 실패 → 타임아웃 후 다시 표시

- 기본: 30초
- 최대: 12시간
- ChangeMessageVisibility API로 연장 가능

Dead Letter Queue (DLQ)

- 처리 실패 메시지 격리
- maxReceiveCount 설정 (예: 3회 시도 후 DLQ 이동)
- 실패 원인 분석 및 수동 재처리

Long Polling

- 메시지가 도착할 때까지 대기 (최대 20초)
- 빈 응답 감소 → 비용 절감
- 권장: WaitTimeSeconds = 20

Delay Queue

- 메시지 전달 지연 (0-15분)
- 전체 큐 또는 개별 메시지에 설정 가능

Message Retention

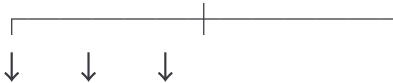
- 기본: 4일
- 범위: 1분 ~ 14일

SNS (Simple Notification Service)

Pub/Sub 구조



Publisher → SNS Topic



Subscriber1 Subscriber2 Subscriber3
(Email) (SQS) (Lambda)

프로토콜

- HTTP/HTTPS
- Email/Email-JSON
- SMS
- SQS
- Lambda
- Kinesis Data Firehose
- Mobile Push (APNS, GCM 등)

FIFO Topic

- **SQS FIFO와만 연동**
- 순서 보장
- 정확히 1회 전달
- Message Group ID 지원

Message Filtering

- Subscription Filter Policy로 메시지 필터링
- JSON 기반 필터



json

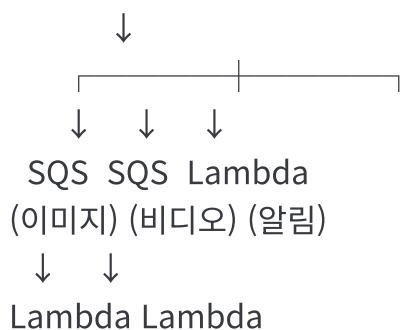
```
{  
  "order_type": ["premium"],  
  "price": [{"numeric": [">=", 100]}]  
}
```

SNS + SQS 조합 패턴 (Fan-out)

패턴 1: 병렬 처리



S3 Event → SNS Topic



장점

- 메시지 손실 방지 (SQS 버퍼)
- 독립적 처리 (하나 실패해도 다른 것 영향 없음)
- 확장성 (구독자 쉽게 추가)

SCENARIO

상황	정답	이유
이미지 업로드 시 썸네일 생성	S3 → SQS → Lambda	비동기, 재시도 보장
이벤트를 여러 대상에 전달	SNS (Fan-out)	Pub/Sub
처리 실패 메시지 추적	DLQ	실패 원인 분석
FIFO 순서 처리 필요	FIFO Queue	주문/결제 시스템
Lambda 동시성 제한 방지	SQS → Lambda	버퍼 역할
여러 서비스 동시 알림	SNS	Email, SMS, Lambda 동시
주문 처리 후 재고 업데이트	FIFO Queue	순서 보장 필수
메시지 15분 후 전달	Delay Queue	지연 전달

👉 시험 키워드 매칭

- "비동기 처리" → SQS
- "순서 보장" → FIFO Queue
- "여러 대상 전달" → SNS
- "처리 실패 추적" → DLQ
- "분산 처리" → Fan-out (SNS + SQS)
- "Lambda 동시성 제한" → SQS 버퍼

🚫 TRAP

- ✗ Lambda를 S3에 직접 연결 (대량 이벤트 시 Lambda 제한 초과)
- ✗ SNS를 큐처럼 사용 (메시지 저장 안 됨)
- ✗ FIFO Queue의 처리량 제한 간과 (3,000 TPS)
- ✗ Visibility Timeout 짧으면 중복 처리 발생
- ✗ Standard Queue는 순서 보장 안 됨

RDS 고가용성

WHY

- 데이터베이스는 **SPOF** 가능성이 가장 높음
- Multi-AZ로 장애 대응, Read Replica로 읽기 확장
- 자동 백업으로 데이터 손실 방지

HOW

Multi-AZ vs Read Replica

항목	Multi-AZ	Read Replica
목적	고가용성 (Failover)	읽기 확장 (Scale-out)
복제	동기 복제	비동기 복제
읽기 가능	X (Standby 전용)	✓
장애 조치 자동	(DNS 전환 1-2분)	수동 승격 가능
엔진	모든 RDS 엔진	모든 RDS 엔진
다중 리전	X (동일 리전만)	✓ (Cross-Region 가능)
비용	추가 인스턴스 비용	추가 인스턴스 + 복제 비용
쓰기	Primary만	Read-Only

Multi-AZ 작동 방식



Primary (AZ-A)
↓ (동기 복제)
Standby (AZ-B)
↓ (장애 발생)
Standby → Primary (자동 DNS 전환)

Read Replica 작동 방식



Primary
↓ (비동기 복제)
Read Replica 1, 2, 3, ... (최대 5개)
↓
READ 요청만 처리

RDS Proxy

- **연결 풀링**: 데이터베이스 연결 재사용
- **장애조치 시간 단축**: 66% 감소
- **Lambda 최적화**: 연결 수 관리
- **IAM 인증**: 암호 없이 접근
- **지원 엔진**: MySQL, PostgreSQL, MariaDB, SQL Server

백업 옵션

유형	설명	보관 기간	RPO	비용
자동 백업	매일 스냅샷 + 트랜잭션 로그	1~35일	5분	무료 (DB 크기까지)
수동 스냅샷	사용자 생성	무제한	스냅샷 시점	유료
PITR	트랜잭션 로그 기반 복구	자동 백업 기간	5분	포함

Point-in-Time Recovery (PITR)

- 백업 기간 내 모든 시점으로 복구 가능
- 트랜잭션 로그 기반
- 5분 간격 복구 가능

Aurora 특수 기능

Aurora Serverless

- **간헐적/예측 불가 워크로드**
- **자동 스케일링** (ACU 단위)
- **비용**: 사용량 기반
- **v2**: 즉시 확장, 프로비저닝 모드와 혼용 가능

Aurora Global Database

- **1초 미만 교차 리전 복제**
- **최대 5개 보조 리전**
- **RPO < 1초, RTO < 1분**
- **재해 복구 + 글로벌 읽기 확장**

Aurora Cloning

- Copy-on-Write 방식
- 스냅샷 없이 빠른 복제
- 테스트 환경 생성

Backtrack

- 특정 시점으로 즉시 되돌리기 (최대 72시간)
- 스냅샷 복원보다 빠름 (분 단위)

Reader Endpoint

- 읽기 부하 자동 분산

- 최대 15개 Read Replica
- 연결 로드 밸런싱

Custom Endpoint

- 특정 인스턴스 그룹 지정
- 예: 분석 쿼리용 그룹

SCENARIO

상황	정답	이유
DB 장애 시 자동 복구 (1-2분)	Multi-AZ	DNS 자동 전환
읽기 부하 분산	Read Replica	비동기 복제
다른 리전 백업	Cross-Region Read Replica	DR 구성
특정 시점 복구 (5분 전)	PITR	트랜잭션 로그
간헐적 트래픽 (주말만)	Aurora Serverless	사용량 기반 과금
1초 미만 글로벌 복제	Aurora Global Database	재해 복구
Lambda 연결 관리	RDS Proxy	연결 풀링
분석 쿼리 분리	Aurora Custom Endpoint	특정 그룹 지정
테스트 DB 빠른 생성	Aurora Cloning	Copy-on-Write

시험 키워드 매칭

- "자동 장애 조치" → Multi-AZ
- "읽기 확장" → Read Replica
- "글로벌 복제" → Aurora Global Database
- "간헐적 워크로드" → Aurora Serverless
- "연결 관리" → RDS Proxy
- "5분 전 복구" → PITR

TRAP

- ❌ Read Replica가 자동 페일오버 (수동 승격만)
- ❌ Multi-AZ는 읽기 부하 분산 불가 (Standby 읽기 불가)
- ❌ Aurora Reader가 Primary로 자동 전환 (Multi-AZ 필요)
- ❌ RDS Snapshot이 자동으로 다른 리전 복제 (명시적 복사 필요)
- ❌ Aurora Serverless는 Multi-AZ 지원 안 함 (v2는 지원)

S3 복원력

WHY

- S3는 11개의 9 (99.99999999%) 내구성 제공
- 연간 10,000,000개 객체 중 1개 손실 확률
- 리전 간 복제로 재해 복구 구현

HOW

내구성 vs 가용성

개념	설명	S3 Standard
내구성 (Durability)	액체 손실 확률	99.999999999% (11 9's)
가용성 (Availability)	서비스 이용 가능 시간	99.99% (연간 52분 다운)

S3 Versioning (버전 관리)

기능

- 객체 변경/삭제 이력 보존
- 삭제 시 **Delete Marker** 추가 (실제 삭제 X)
- 복구 가능 (Delete Marker 제거)

주의사항

- 버전마다 스토리지 비용 발생
- Lifecycle Rule로 이전 버전 관리 권장
- Versioning 활성화 후 중지 불가 (일시 중지만 가능)

MFA Delete

- 버전 영구 삭제 시 MFA 인증 필수
- **루트 계정만** 활성화 가능
- 최고 보안 수준

S3 Replication (복제)

유형	설명	사용 사례
CRR (Cross-Region)	다른 리전 복제 재해 복구, 지역 감소	
SRR (Same-Region)	동일 리전 복제 로그 짐계, 테스트 환경	

Replication 요구사항

- Versioning 활성화 (소스 + 대상)
- IAM Role로 권한 부여
- KMS 암호화 시: CMK 권한 공유 필요

Replication 제약

- **기존 객체는 복제 안 됨** (활성화 이후만)
 - 해결: S3 Batch Replication
- Delete Marker는 선택적 복제
- SSE-C 암호화 객체는 복제 불가
- 체인 복제 불가 (A→B→C 시 A→C 직접 설정 필요)

Replication Time Control (RTC)

- 15분 내 복제 보장
- SLA 제공
- 추가 비용

S3 Lifecycle Rules

전환 규칙



Standard (30일)



Standard-IA (60일)



Glacier Flexible Retrieval (90일)



Glacier Deep Archive (365일)



삭제 (2555일 = 7년)

Versioning + Lifecycle



json

```
{
  "Rules": [
    {
      "Id": "Archive old versions",
      "Status": "Enabled",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 30,
          "StorageClass": "GLACIER"
        }
      ],
      "NoncurrentVersionExpiration": {
        "NoncurrentDays": 90
      }
    }
  ]
}
```

S3 Event Notifications

대상

- SNS Topic
- SQS Queue
- Lambda Function
- EventBridge (권장, 더 유연)

이벤트 유형

- s3:ObjectCreated:*
- s3:ObjectRemoved:*
- s3:ObjectRestore:*
- s3:Replication:*

SCENARIO

상황	정답	이유
객체 삭제 복구	Versioning + Delete Marker 제거	삭제 이력 보존
리전 간 자동 백업	CRR	재해 복구
동일 리전 테스트 환경	SRR	동일 리전 복제
30일 후 Glacier 이동	Lifecycle Rule	비용 절감
KMS 암호화 객체 복제	CMK 권한 공유 + CRR	교차 계정 키 공유
이전 버전 자동 삭제 (90일)	Lifecycle NoncurrentVersionExpiration	스토리지 비용 절감
15분 내 복제 보장	Replication Time Control	SLA 필요
기존 객체 복제	S3 Batch Replication	일괄 복제

🎯 시험 키워드 매칭

- "삭제 복구" → Versioning
- "재해 복구" → CRR
- "자동 아카이빙" → Lifecycle
- "11 9's" → S3 내구성
- "15분 복제" → RTC

🚫 TRAP

- ✗ CRR은 기존 객체 복제 안 함 (Batch Replication 사용)
- ✗ KMS 암호화 객체 복제 시 CMK 권한 누락
- ✗ Versioning 활성화 시 비용 증가 (Lifecycle로 관리 필요)
- ✗ Delete Marker는 기본 복제 안 됨 (선택적 활성화)

✳️ CloudFront + Route 53

🧠 WHY

- 엣지 로케이션 (200+) 활용으로 전 세계 응답속도 향상
- DNS 기반 장애 조치 + 헬스 체크로 고가용성 확보
- DDoS 방어 (AWS Shield 통합)

⚙️ HOW

CloudFront 핵심

Origin 유형

- S3 Bucket
- ALB / NLB
- EC2 Instance
- API Gateway
- MediaStore / MediaPackage
- Custom HTTP Server

Cache Behavior

- 경로 패턴별 설정 (예: /api/*, /images/*)
- TTL 설정 (최소/최대/기본)
- 압축 (gzip, brotli)
- 허용된 HTTP 메서드
- Query String/Cookie/Header 전달

OAC vs OAI

- **OAC (Origin Access Control)**: 권장, S3 전용, IAM 기반
- **OAI (Origin Access Identity)**: 레거시

OAC 구성



1. CloudFront에서 OAC 생성
↓
2. S3 Bucket Policy에 OAC 허용
↓
3. 직접 S3 접근 차단 (Block Public Access)

Geo Restriction

- Whitelist: 특정 국가만 허용
- Blacklist: 특정 국가 차단
- IP 기반 아님, 국가 기반

Signed URL / Signed Cookie

- **Signed URL**: 개별 파일 접근 제어
- **Signed Cookie**: 여러 파일 접근 제어
- 만료 시간 설정
- 프리미엄 콘텐츠 보호

Lambda@Edge / CloudFront Functions

항 목	Lambda@Edge	CloudFront Functions
실행 위치	Regional Edge Cache	Edge Location
지연 시간	밀리초	마이크로초
런타임	Node.js, Python	JavaScript (ES5)
최대 실행 시간	5-30초	< 1ms
사용 사례	복잡한 로직	간단한 변환

Route 53 Routing Policy

정책	설명	사용 사례	헬스 체크
Simple	단일 리소스, 랜덤 선택 기본 설정		✗
Weighted	비율 기반 트래픽 분배 A/B 테스트 (70:30)		✓
Latency	최저 지연시간 리전 선택 글로벌 앱		✓
Failover	Active-Passive 구조 재해 복구		✓ 필수
Geolocation	사용자 위치 기반 지역별 콘텐츠		✓
Geoproximity	지리적 거리 + 바이어스 트래픽 이동		✓
Multi-Value	여러 IP 반환 (8개까지) 간단한 로드밸런싱		✓

Health Check 유형

- **Endpoint**: IP/도메인 모니터링 (30초 간격)
- **Calculated**: 여러 헬스 체크 조합 (AND/OR)
- **CloudWatch Alarm**: 메트릭 기반 헬스 체크

Failover 구성



Primary (us-east-1)
 ↓ (Health Check 실패)
 Secondary (eu-west-1)

Alias Record

- AWS 리소스 직접 연결
- 지원: CloudFront, ELB, S3 Website, API Gateway
- **무료** (DNS 쿼리 비용 없음)
- Zone Apex 지원 (example.com)

✳️ SCENARIO

상황	정답	이유
전 세계 응답속도 향상	CloudFront	엣지 캐싱
리전 장애 시 자동 전환	Route 53 Failover + Health Check	Active-Passive
두 리전 70:30 분배	Route 53 Weighted	A/B 테스트
최저 지연시간 리전	Route 53 Latency-based	글로벌 앱
유럽 사용자만 특정 리전	Route 53 Geolocation	지역별 콘텐츠
S3를 CloudFront로만 접근 OAC + S3 Bucket Policy		직접 접근 차단
프리미엄 콘텐츠 보호	Signed URL/Cookie	시간 제한 URL
요청 헤더 변환	CloudFront Functions	마이크로초 지연

🎯 시험 키워드 매칭

- "전 세계 속도" → CloudFront
- "장애 조치" → Route 53 Failover
- "A/B 테스트" → Route 53 Weighted
- "최저 지연" → Route 53 Latency
- "지역별" → Route 53 Geolocation

🚫 TRAP

- ❌ CloudFront만으로 글로벌 DB 동기화 불가
- ❌ Route 53은 실시간 부하분산 아님 (DNS 캐시 영향)
- ❌ OAI는 레거시, OAC 권장
- ❌ Alias Record는 모든 리소스 지원 X (AWS 리소스만)

💡 백업 & 재해 복구 (DR)

🧠 WHY

- RTO/RPO 기준으로 전략 설계
- 비용과 복구 시간의 트레이드오프

⚙️ HOW

DR 전략 (비용 ↔ 복구 시간)

전략	설명	RTO	RPO	비용
Backup & Restore	백업 → 수동 복구	시간~일	시간	★ 최저
Pilot Light	핵심 서비스만 대기	10분~1시간	분~시간	★★
Warm Standby	축소 규모 운영	분 단위	분 단위	★★★
Multi-Site	양쪽 모두 활성화	실시간	실시간	★★★★★ 최고

Backup & Restore



프로덕션 리전

↓ (정기 백업)

S3 / Glacier

↓ (재해 발생)

백업에서 복원 (수동)

Pilot Light



프로덕션 리전 (Full)

↓ (데이터 복제)

DR 리전 (핵심만 실행: RDS)

↓ (재해 발생)

나머지 리소스 시작 (EC2, ELB 등)

Warm Standby



프로덕션 리전 (Full)

↓ (실시간 복제)

DR 리전 (축소 규모 운영)

↓ (재해 발생)

DR 리전 확장 (Auto Scaling)

Multi-Site



프로덕션 리전 (Active)

↑ (양방향 복제)

DR 리전 (Active)

↓ (재해 발생)

Route 53으로 트래픽 자동 전환

AWS Backup

지원 서비스

- EBS, EC2
- RDS, Aurora
- DynamoDB
- EFS, FSx
- Storage Gateway
- DocumentDB, Neptune

기능

- 중앙집중 백업 관리
- 백업 정책, 스케줄
- 교차 리전/계정 복사
- **Vault Lock**: WORM (규정 준수)
- 자동 보존 기간 관리

Vault Lock

- 백업 삭제 방지
- 규정 준수 (HIPAA, GDPR)
- 루트도 삭제 불가

Cross-Region 백업 전략

서비스	백업 방법	RPO
RDS	자동 백업 + 교차 리전 Read Replica 5분	
Aurora	Aurora Global Database	< 1초
S3	CRR	15분 (RTC 사용 시)
EBS	스냅샷 → 교차 리전 복사	스냅샷 주기
DynamoDB	PITR + 글로벌 테이블	초 단위

SCENARIO

상황	정답	이유
전체 리전 다운 대비	Multi-Site 또는 Aurora Global	RTO/RPO 최소
RPO < 1분	실시간 복제 (Aurora Global, DynamoDB Global)	동기/비동기 복제
정기 백업 (비용 절감)	AWS Backup	중앙 관리
규정 준수 (백업 보호)	AWS Backup Vault Lock	WORM
RTO < 1시간, 비용 절감	Pilot Light	핵심만 대기
재해 복구 자동화	Multi-Site + Route 53 Failover	자동 전환

시험 키워드 매칭

- "재해 복구" → Multi-Site 또는 Warm Standby
- "RTO < 1시간" → Pilot Light
- "비용 최소화" → Backup & Restore
- "규정 준수" → Vault Lock
- "RPO < 1초" → Aurora Global

TRAP

- ✗ 단일 AZ 백업으로 DR 완성
- ✗ RDS Snapshot이 자동 복제 (명시적 복사 필요)
- ✗ Pilot Light는 모든 서비스 대기 중 (핵심만)

복원력 도메인 최종 체크리스트

Auto Scaling + ELB

- Target Tracking Policy 이해
- ALB vs NLB 차이
- Cross-Zone Load Balancing
- Lifecycle Hooks

SQS + SNS

- Standard vs FIFO Queue
- DLQ 활용
- Fan-out 패턴
- Long Polling

RDS

- Multi-AZ vs Read Replica
- Aurora 특수 기능
- PITR
- RDS Proxy

S3

- Versioning + MFA Delete
- CRR vs SRR
- Lifecycle Rules
- Event Notifications

재해 복구

- DR 4가지 전략
- RTO/RPO 이해
- AWS Backup
- Cross-Region 백업

👉 복원력은 고가용성의 핵심입니다! Multi-AZ, Auto Scaling, Replication을 항상 고려하세요.

다음: [Part 3 - 고성능 아키텍처 \(24%\)](#).