

# 멀티 리전 DR(재해복구) RUNBOOK 〈Seoul ↔ Osaka〉

버전	v1.0
최종 수정	2025-08-31 UTC
대상 시스템	Medical-DR (Primary: ap-northeast-2, DR: ap-northeast-3)

## 1) 개요

목적: 장애 발생 시 안전한 전환(Failover)을 표준 절차로 수행하여 목표 RTO/RPO 내 복구

적용 범위: 정적(CloudFront+리전별 S3 오리진 그룹), API(Global Accelerator→ALB), DB(RDS Read Replica→Promote)

제약/결정: S3 MRAP는 SigV4A 제약으로 직접 오리진으로 사용하지 않으며, 리전별 S3 오리진 그룹을 사용

역할: Incident Commander(IC) / Infra / App·QA / Comms

## 2) 사전 준비

- CloudFront OAC 활성화 & 버킷 정책 일치(퍼블릭 차단)
- S3 오리진 그룹 구성(Primary/DR) + Failover 코드 404,500,502,503,504
- 정적 캐시: HTML·config.js = no-cache, 해시 자산 = immutable
- 배포: aws s3 sync --delete + 부분 무효화(HTML/설정 경로)
- GA EndpointGroup: Primary 100 / DR 0, 헬스체크 Path /health
- RDS ReplicaLag 알람 정상, 백업/스냅샷 On
- VPC 엔드포인트: S3(Gateway), SSM/EC2Messages/SSMMessages(Interface)

## 3) 장애 유형별 절차

### 3.1 DB 장애 (Primary Writer)

감지: CloudWatch 알람(RDS 연결 불가/ReplicaLag 증가), 앱 DB 오류

# (1) 시각 기록

```
T0=$(date -u +%s); echo "[DB-1] T0=$T0"
```

# (2) 승격

```
aws rds promote-read-replica --db-instance-identifier <DR_REPLICA_ID> --region ap-northeast-3
```

# (3) 연결 확인(정상 응답까지 대기)

```
until mysql -h <DR_WRITER_ENDPOINT> -u <USER> -p<PASS> -e "SELECT 1;" >/dev/null 2>&1; do sleep 2; done
```

```
T1=$(date -u +%s); echo "DB RTO=$((T1-T0))s"
```

검증: 읽기/쓰기쿼리 정상, 최근 데이터 확인 → RTO/RPO 기록

### 3.2 API 레이어 장애 (ALB/EC2)

감지: /health 5xx/Timeout 증가, ALB Target Unhealthy

```
T0=$(date -u +%s)
aws globalaccelerator update-endpoint-group --endpoint-group-arn
<GA_ENDPOINT_GROUP_ARN> --endpoint-configurations
EndpointId=<PRIMARY_ALB_ARN>,Weight=0 EndpointId=<DR_ALB_ARN>,Weight=100

until [ "$(curl -sk -o /dev/null -w "%{http_code}" "https://<GA_DOMAIN>/health?t=$(date -u +%s)")" = "200" ]; do sleep 2; done
T1=$(date -u +%s); echo "API RTO=$((T1-T0))s"
```

검증: 핵심 API(로그인/목록/상세) 200, 오류율/지연 정상화

### 3.3 스토리지/정적(CloudFront+S3) 장애

감지: CloudFront 4xx/5xx, Primary S3 접근 오류

```
T0=$(date -u +%s)
aws s3 cp ./site/login.html s3://medical-dr-web-pr/login.html --cache-control "no-cache,
no-store, must-revalidate" --content-type "text/html" --metadata-directive REPLACE

aws cloudfront create-invalidation --distribution-id <CF_DIST_ID> --paths "/login.html"
curl -sl "https://<CF_DOMAIN>/login.html?t=$(date -u +%s)" | head -n1
T1=$(date -u +%s); echo "STATIC RTO=$((T1-T0))s"
```

검증: login.html 200, Cache-Control no-cache 확인 → Static RTO 기록

## 4) 공통 검증 절차

- RTO: T0(감지) → T1(정상화) 실측 시간 기록 (DB/API/정적 각각)
- RPO: 장애 직전 데이터 DR 존재 여부 확인 (DB: 쿼리 / 정적: 데이터 복제 시간)

## 5) Failback(원복) 절차

- 원 리전 복구/안정화 확인 → 신규 Writer 복구
- DR(현재 Writer) 스냅샷 → 원 리전 복원 후 동기화
- GA 가중치 복귀: Primary 100 / DR 0
- CloudFront 정상 확인, 필요 시 부분 무효화

## 6) 커뮤니케이션 템플릿

[DR Incident] <서비스명> / <날짜 UTC>

- T0 장애 감지: <시각>, 증상: <요약>
- 조치: <DB-1 / A-1 / S-1> 실행 (명령/파라미터 요약)
- T1 복구 완료: <시각>, RTO=<초>
- RPO: Static <초>, DB <초> (해당 시)
- 영향/고객 공지: <링크>
- 후속: 원인/재발 방지/Failback 일정

## 7) 자동화 계획

- SSM Document로 DB Promote, GA 전환, CF 무효화 원클릭 실행
- Synthetic Monitoring(헬스+사용자 시나리오) 도입
- Runbook 실행 로그/지표 자동 수집 및 대시보드화

## 8) 보안/권한 표준

- OAC 전용 버킷 정책 + Block Public Access 유지(퍼블릭 ACL/정책 금지)
- 배포 Role: S3(List/Get/Put), CloudFront(CreateInvalidation) 최소 권한
- Terraform: 모듈 단방향, alias(aws.p/aws.d/필요시 aws.us\_east\_1)
- SSM: 프라이빗 서브넷에 Interface 엔드포인트 3종 + AmazonSSMManagedInstanceCore

## 9) 부록A: 공통 변수(셸)

```
export REGION_P=ap-northeast-2
export REGION_D=ap-northeast-3
export CF_DIST_ID=<CF_DISTRIBUTION_ID>
export CF_DOMAIN=<CF_DOMAIN>
export S3_P=medical-dr-web-pr
export S3_D=medical-dr-web-dr
export GA_ENDPOINT_GROUP_ARN=<GA_ENDPOINT_GROUP_ARN>
export ALB_P_ARN=<PRIMARY_ALB_ARN>
export ALB_D_ARN=<DR_ALB_ARN>
export GA_HEALTH_URL="https://<GA_DNS_NAME>/health"
export RDS_DR_REPLICA_ID=<DR_REPLICA_ID>
export RDS_DR_WRITER_ENDPOINT=<DR_WRITER_ENDPOINT>
```

## 10) 부록B: 측정 스크립트(RTO/RPO)

### API RTO

```
t0=$(date -u +%s)
until [ "$(curl -sk -o /dev/null -w "%{http_code}" "$GA_HEALTH_URL?t=$(date -u +%s)")" =
"200" ]; do sleep 2; done
echo "API RTO=$(( $(date -u +%s) - t0 ))s"
```

## Static RPO (S3 CRR)

```
KEY=rpo_test.txt
T0=$(date -u +%s); echo "rpo_test $T0" > /tmp/$KEY
aws s3api put-object --bucket $S3_P --key $KEY --body /tmp/$KEY --metadata t0="$T0"
LM=$(aws s3api head-object --bucket $S3_P --key $KEY --query 'LastModified' --output text)
TOP=$(date -u -d "$LM" +%s)
until aws s3api head-object --bucket $S3_D --key $KEY --region $REGION_D >/dev/null 2>&1; do
sleep 1; done
echo "Static RPO=$(( $(date -u +%s) - TOP ))s"
```

## DB RPO

```
SELECT TIMESTAMPTDIFF(SECOND, created_at, NOW()) AS rpo_seconds
FROM dr_rpo_probe WHERE id = <writer_inserted_id>;
```