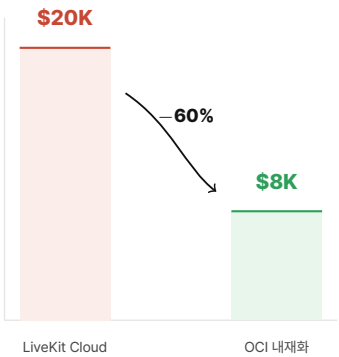
ZEP · LIVEKIT SELF-HOSTING ON OCI

오라클 클라우드로 400만 MAU를 받아내는 WebRTC 클러스터 만들기



LiveKit Cloud 인보이스의 63%는 bandwidth.
bandwidth를 직접 고를 수 있는 곳으로 이전.

이재규 (JQ Lee)

ZEP Tech Lead · Ouroboros maintainer
github.com/Q00

SPEAKER

이재규 JQ Lee

ZEP Tech Lead · Ouroboros maintainer

10년
소프트웨어 개발

6년
RTC 개발과 운영

400만
MAU 서비스 운영

linkedin.com/in/q00

github.com/Q00

TAKEAWAY

RTC 비용을 정하는 세 가지

1

인보이스의 절반은 bandwidth

값이 붙는 대상은 CPU가 아니라 나가는 바이트

2

인스턴스 말고 bandwidth선택

OCPU 하나당 1 Gbps. 필요한 만큼만 구매

3

batch job 분리

녹화와 실시간 서빙은 다른 node pool

SCALE · 4M MAU

ZEP에서 WebRTC는 옵션이 아니라 기본값

아바타가 가까워지면 음성 자동 연결. 사용자가 켜는 기능이 아니라 접속해 있는 내내 켜져 있는 기능. 트래픽이 사용 시간에 그대로 비례.

400만

월간 활성 사용자 MAU

45.9M

한 달 participant minutes. 평균 동시 참가자 약 1,060명

128 TB

한 달 downstream 전송량

N + 1

녹화 방이 늘면 노드도 같이 증가. room composite 구조

THE BILL · LIVEKIT CLOUD

LiveKit Cloud 월 인보이스

Additional WebRTC participant minutes ($0.15 per 1000 minutes)	45,917,313 분
Additional Downstream data transfer ($0.09 per GB)	128.44 TB
Additional Egress transcode minutes - video ($0.015 per minute)	608 분
합계	**$ 18,456 / month**

— 자잘한 항목까지 더하면 월 2만 달러 선. 연 24만 달러

ZEP custom plan 실제 사용량 · 금액은 인보이스에 적힌 단가로 계산

BREAKDOWN · $18,456 / MONTH

63% 가 트래픽 값

128.44 TB × $0.09/GB

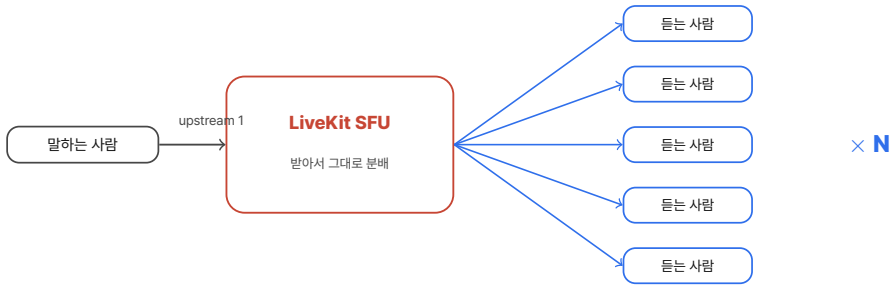
45.9M min × $0.15/1000 min

— participant minutes: 서버를 켜둔 시간. downstream: 그 서버에서 나간 바이트

— 자체 운영으로 옮기면 participant minutes 는 소멸. 이미 산 CPU 시간. 남는 판단 대상은 63% 의 트래픽 값

SFU · FANOUT

upstream 하나, downstream 사람 수만큼



8인 방에서 한 명이 말하면 7배 송출

— 녹화도 구독자 한 명. 방마다 녹화를 켜면 참가자 한 명분 트래픽 추가

문제 재정의

"서버비 절감"이 아니라 "나가는 바이트의 단가"

컴퓨트가 싼 클라우드는 많음. 나가는 바이트가 싼 클라우드는 소수.

EGRESS PRICING · $ PER GB

웬만한 클라우드는 인보이스가 그대로 따라옴

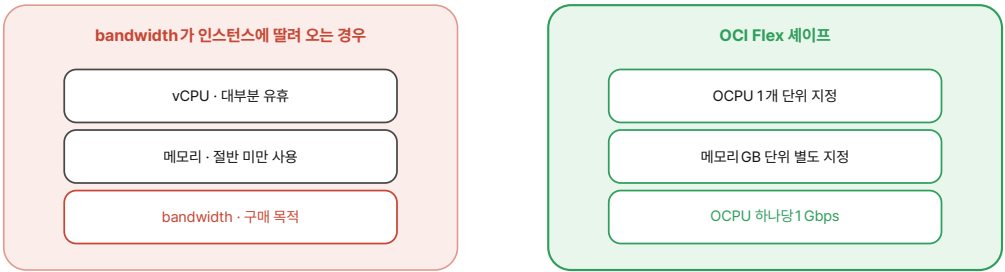
outbound 요금	무료 구간	이후 단가	128 TB 환산
LiveKit Cloud (downstream)	없음	$0.09 / GB	**$11,560**
AWS 인터넷 outbound	100 GB / 월	$0.09 / GB	**$11,551**
OCI 인터넷 outbound (서울)	**10 TB / 월**	**$0.025 / GB**	**$2,961**

정가 기준 단순 환산 · OCI outbound는 APAC 리전 $0.025/GB, 북미·유럽은 $0.0085/GB

SHAPE · BANDWIDTH BUNDLING

bandwidth를 사려고 코어를 사는 구조

RTC 노드에 필요한 건 네트워크. 인스턴스 타입이 bandwidth를 묶어 팔면 안 쓰는 vCPU까지 함께 구매.



VM.Standard.E5.Flex · 1 Gbps per OCPU, 최대 40 Gbps · docs.oracle.com Compute Shapes

SIZING · VM.STANDARD.E5.FLEX

노드당 4 Gbps, pod 하나에 노드 하나

VM.Standard.E5.Flex

4 OCPU / 16 GB

1 Gbps per OCPU = 4 Gbps

×

18 pods

노드당 1개 · podAntiAffinity

public subnet, 공인 IP 보유

=

72 Gbps

현재 서빙 상한 · 72 OCPU

node pool 60 대까지 확보 = 240 Gbps

— 서빙 대수는 helm replicaCount 기준 18. 30 → 25 → 23 → 18 로 조정된 상태. terraform node pool 은 60 대

RESULT

월 $20,000 이

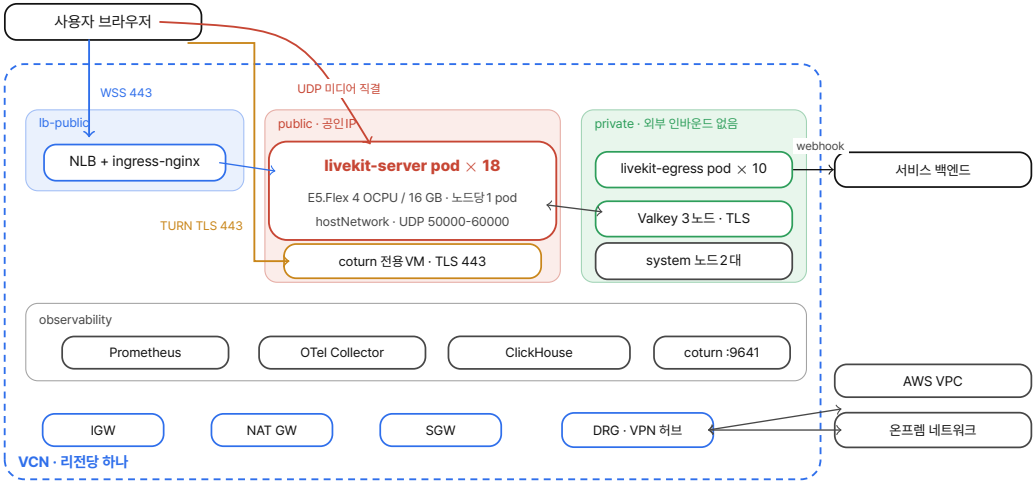
월 $8,000

−60%

트래픽 동일. 사용자 동일. 바뀐 것은 구매 형태.

ARCHITECTURE · LIVEKIT CLUSTER @ 서울 리전

OKE v1.35 · OCI_VCN_IP_NATIVE · private control plane



TRAFFIC · THREE PATHS

시그널링, 미디어, 우회로를 각각 다른 경로로

시그널링 · WSS 443

NLB → ingress-nginx
→ svc :7880 → pod

제어 신호 전용. 양은 적고 단절 불가. proxy timeout 3600초, client IP 기준 sticky.

미디어 · UDP 50000-60000

노드 공인IP로 직결
(load balancer 없음)

128 TB 전량이 이 경로. load balancer를 끼우면 bandwidth 요금과 병목이 한 점에 집중. 그래서 무경유.

우회로 · TURN TLS 443

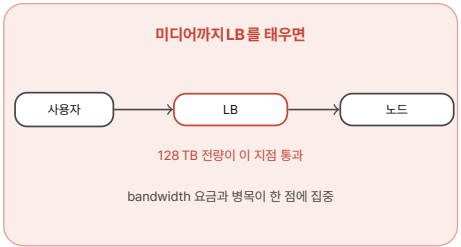
nginx stream (ALPN)
→ coturn → UDP relay

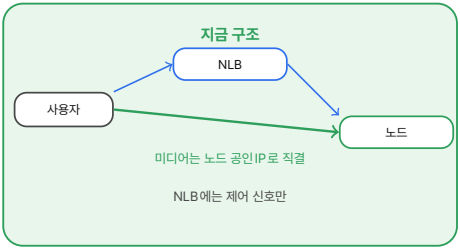
UDP 차단 환경용 단일 443 경로. 학교와 관공서 트래픽이 여기로.

MEDIA PATH · NO LOAD BALANCER

미디어는 로드밸런서를 타지 않음

시그널링만 NLB 경유. 미디어는 노드 공인 IP로 직결. hostNetwork와 use_external_ip를 켜면 SDP에 노드 공인 IP가 그대로 탑재.





— 전제 조건은 노드마다 공인 IP. livekit-server node pool만 public subnet, NSG는 UDP 50000-60000과 TCP 7881만 개방. 나머지 전부 private

TURN · STANDALONE COTURN VM

TURN 과 livekit 은 배포 주기가 다름

내장 TURN 사용 시 livekit 배포마다 TURN 동반 재시작. UDP 차단 환경 사용자에게 TURN 은 유일한 경로.

coturn 전용 인스턴스

E4.Flex 4 OCPU / 16 GB · public subnet

nginx stream 이 TLS 443 수신

coturn 이 UDP 49152-65535 로 relay

⇒

얻은 것

livekit 배포와 TURN 재시작 분리

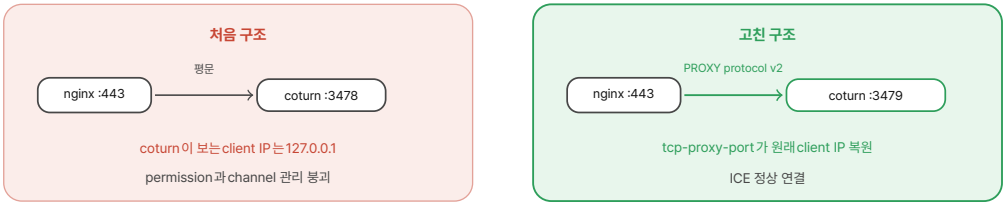
인증서 갱신은 nginx reload 로 종료

진행 중 세션 단절 0

WAR STORY · PROXY PROTOCOL

"연결은 되는데 바로 끊김"

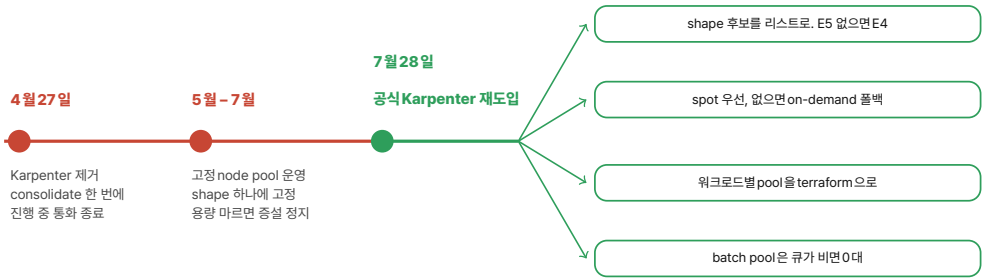
nginx TLS termination 후 coturn에 평문 전달. livekit tester는 연결 성공, 브라우저는 allocation 직후 세션 폐기.



— 원인은 client IP. TURN은 client IP를 permission과 channel의 키로 사용. 127.0.0.1로 보이면 프로토콜이 깨짐

NODE POOL · KARPENTER OFF → ON

OCI 카펜터 도입기



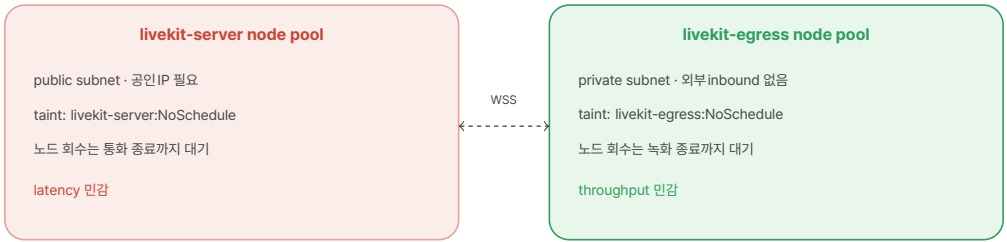
— 회수 기능은 잠근 채로 사용. consolidation은 빈 노드만, expireAfter Never, drift 자동 교체 금지, terminationGracePeriod 5일

— 5일은 강제 종료 상한. 없으면 노드 삭제가 무한 대기, 짧으면 그 노드 참가자 전원 종료

BATCH JOB · EGRESS ISOLATION

실시간 서빙과 batch job은 별도 node pool

egress는 방에 참가자로 입장해 화면 합성과 인코딩 수행. CPU 점유가 크고 최대 1시간 지속.



— egress는 방에 참가자로 입장. 이 트래픽은 VCN 내부라 인터넷 outbound 과금 대상 아님

RECORDING · ROOM COMPOSITE → TRACK EGRESS

합성을 실시간에서 떼어냄

room composite는 방마다 Chrome이 세션 내내 상주하며 합성과 인코딩. track egress는 raw track만 기록하고 종료.

room composite

Chrome 상주 · 실시간 합성과 인코딩

`room_composite_cpu_cost  3.0`

녹화 방 N개 = 노드 N대

중간에 죽으면 녹화 유실

⇒

track egress

Chrome 없음. raw track만 기록

`track_cpu_cost  0.3  (× 1/10)`

노드 수가 방 수에서 분리

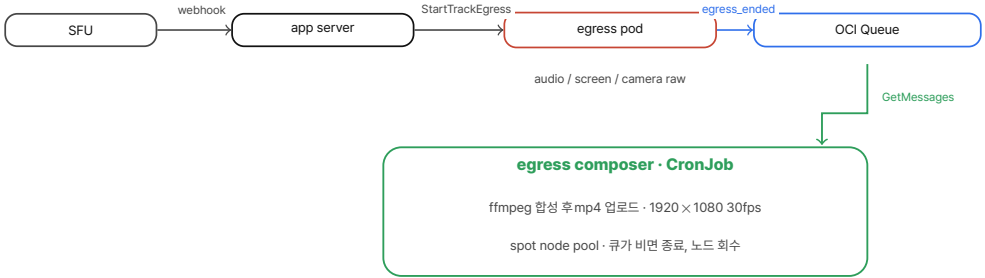
합성은 큐 뒤 batch로 이동

— 기본값 track 1.0으로는 track 3개가 room composite 하나와 동일 비용. 그대로 두면 track 요청도 똑같이 거절. 실측 기준 0.3으로 재산정

cpuCost · roomComposite 3.0 / trackComposite 2.0 / track 0.3

COMPOSE · QUEUE + SPOT CRONJOB

합성은 큐 뒤 batch로



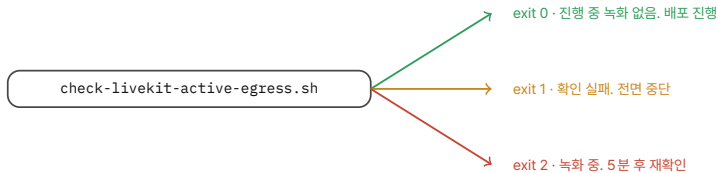
— at-least-once. 업로드 성공 후에만 메시지 삭제. SIGTERM에 in-flight 메시지 반납, 재시도는 큐가 담당

— 같은 Karpenter, pool마다 다른 disruption 정책. 실시간은 통화 종료까지 대기, batch는 큐가 비면 1분 뒤 회수

OPS · RESTART GUARD

녹화 진행 중 재시작 금지

1시간짜리 녹화 도중 배포가 나가면 해당 녹화 유실. 재시작 앞단에 guard 배치.



- ListEgress로 진행 중 job만 조회. 차단 시 출력은 egress id, room, status, started_at. secret은 미출력
- rollout restart wrapper가 guard를 선행 실행

STATE · VALKEY 7.2

방 상태는 Valkey 단일 지점

livekit은 Redis로 room과 participant 상태 공유. 여러 대가 한 클러스터로 보이는 근거.

OCI CACHE

Valkey 7.2 · 3노드 16GB

private subnet 배치, NSG는 6379만 개방. TLS 필수

WHY MANAGED

직접 운영 대상 아님

장애 시 서빙 전체 정지. 절감액보다 손실이 큰 자리

SIZING

2GB에서 16GB로

동시 방 증가에 따라 메모리 2회 증설

OBSERVABILITY · STACK

매니지드 이탈 시 대시보드도 함께 이탈

LiveKit Cloud 콘솔이 제공하던 화면을 직접 구성. 이관 공수의 절반.

Prometheus

livekit-server :6789, egress :9090, coturn :9641. 15초 간격 수집 후 원격 저장소로 remote write.

OTel Collector

DaemonSet으로 전 노드 trace와 로그 수집. OTLP로 전송.

ClickHouse

trace와 로그 저장소. room과 participant 단위 조회는 여기서.

Webhook

room과 participant 이벤트를 백엔드로. 제품 지표와 인프라 지표의 접점.

— TURN 분리 후 coturn 메트릭 별도 필요. 공식 RPM에 Prometheus 미포함, 도커 이미지로 운영

TIMELINE · 2026.3 – 2026.8

첫 커밋부터 지금까지

— live 전환까지 두 달. 그 뒤로도 계속 바뀌는 중

— Karpenter는 4월에 껐다가 7월에 다시. dev 한 달로는 판단이 안 서는 것들이 있었음

GLOBAL · PER-REGION MODULE

해외 리전은 단일 인스턴스로 시작

서울은 OKE 클러스터. 트래픽이 적은 리전은 livekit과 egress를 한 인스턴스에 도커로.

리전당 인스턴스 1대

VCN · IGW · NSG · 예약 공인 IP

Caddy가 인증서 발급과 WSS/TURN 수신

livekit + egress + node_exporter 도커 구동

리전마다 10 TB가 다시 무료

근접 리전 수용으로 latency 감소

무료 구간이 리전 수만큼 증가

— 동일 테라폼 모듈, 변수만 교체. 사용자 증가 시 해당 리전만 클러스터로 승격

WRAP UP

정리

01

인보이스 분해가 먼저

줄일 대상이 컴퓨트인지 트래픽인지. ZEP은 63%가 트래픽, 여기서 후보가 좁혀짐

02

bandwidth 선택 가능 여부

OCPU 하나당 1 Gbps, 리전당 10 TB 무료. 묶음으로 딸려 오는 bandwidth는 안 쓰는 코어 값

03

batch와 실시간 분리

node pool, taint, grace period, 재시작 guard. 여기까지가 녹화 중 배포의 전제

$ 20,000 → $ 8,000 · 두 달

감사합니다

이재규 (JQ Lee)

ZEP Tech Lead · Ouroboros maintainer · github.com/Q00