aws FOR GAMES

GAMES ON AWS 2026

GS Neotek | MEGAZONE CLOUD

GAMES ON AWS 2026

aws FOR GAMES

Amazon SageMaker HyperPod 기반 오픈소스 모델 서빙 최적화하기

송정우

솔루션즈 아키텍트

AWS

정현진

Manager, DATA&AI Data Engineering

베이글코드

GS Neotek | MEGAZONE CLOUD

목차

- 오픈소스 모델 서빙의 필요성과 어려움
- 추론 on AWS
- 베이글 코드가 오픈소스 LLM 서빙을 고려하게 된 계기
- LLM 서빙의 어려움과 최적화
- Lessons Learned

1

오픈소스 모델 서빙의 필요성과 어려움

오픈소스 기반 오픈 웨이트 모델 서빙, 왜 필요한가?

데이터 경계

외부로 보낼 수 없는 데이터가 있다

비용 구조

토큰 과금에서 GPU 고정 비용으로

통제권

모델·성능·가용성을 우리가 정한다

그래서 직접 서빙을 시작하지만, 생각보다 어렵습니다

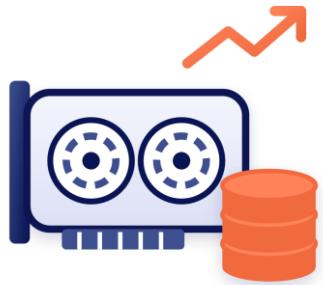
오픈소스 기반 오픈웨이트 모델 서빙의 어려움

01



오픈웨이트 모델의
빠른 릴리즈 주기

02



GPU 수급/비용의
어려움

03



기존과 다른
워크로드 유형

추론 플랫폼의 필요성

전략이 없는 모델 서빙

=

플랫폼화되지 않은 오픈소스 기반 오픈웨이트 모델 서빙

우리는 모델을 어디서 **테스트**하고,
이 모델을 어떻게 **운영**할 것인가?



2

추론 on AWS

Kubernetes 기반 추론, Why?

성숙한 커뮤니티

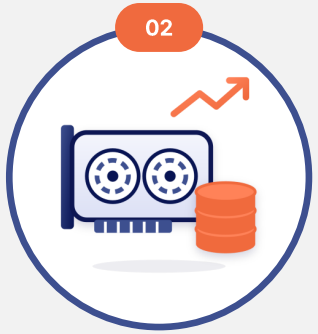
오케스트레이션, 플랫폼에 대한 철학

기존 운영 인력의 지식 활용





오픈웨이트 모델의
빠른 릴리즈 주기



GPU 수급/비용의
어려움



기존과 다른
워크로드 유형

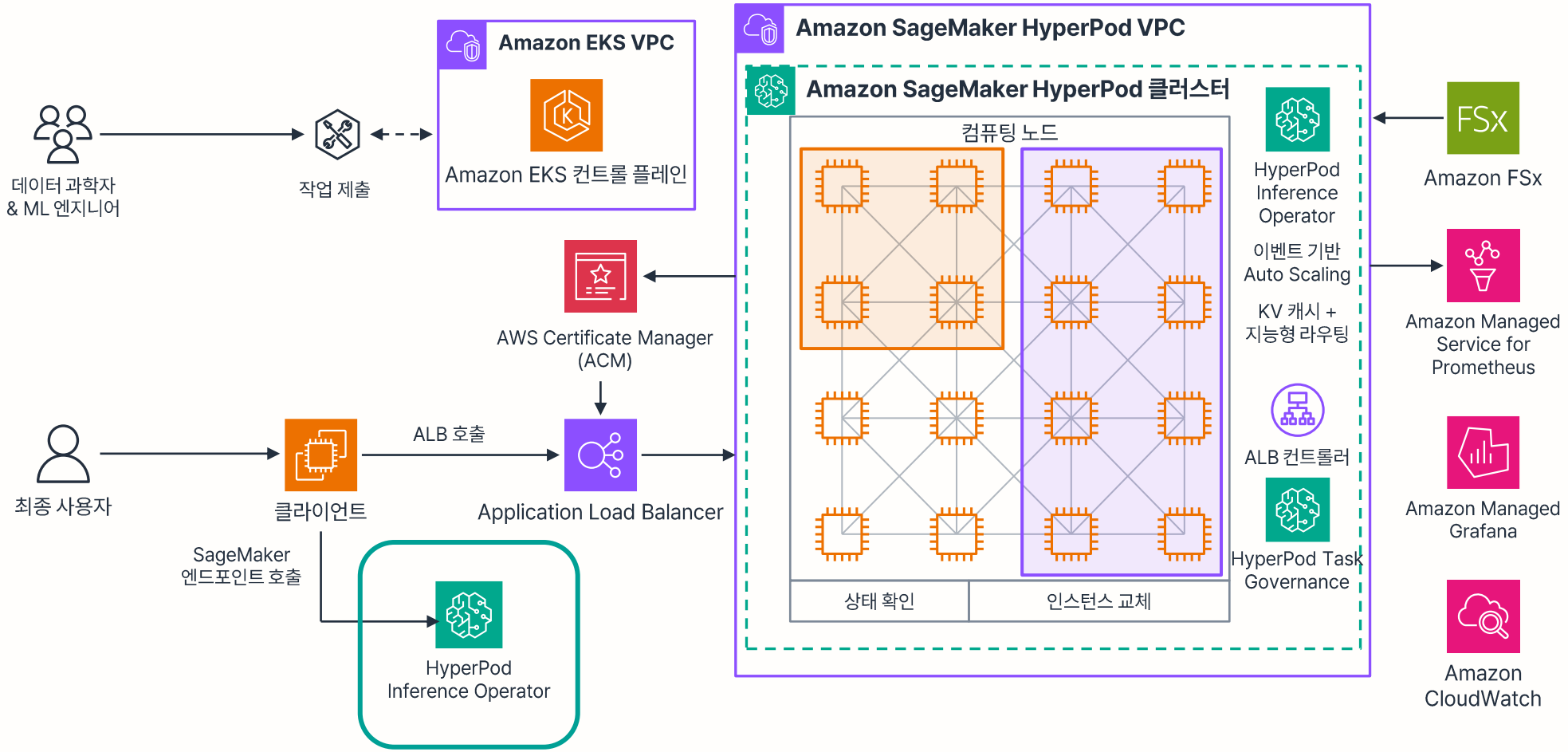
추론 on Amazon EKS



- 운영 비용이 줄어듭니다
- Amazon EKS Hybrid Nodes를 통해 여전히 기존 자원을 활용할 수 있습니다
- 오픈소스 라이브러리들을 유연하게 사용할 수 있습니다

Amazon SageMaker HyperPod on Amazon EKS?

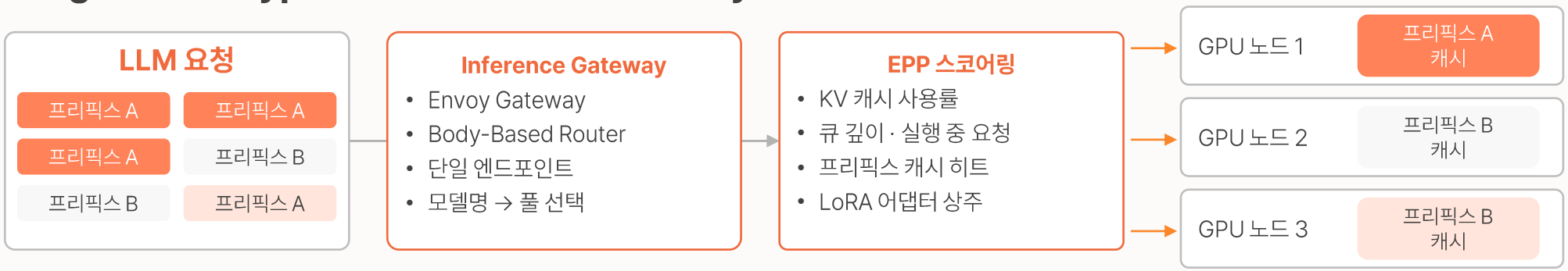
훈련, 추론용 GPU 클러스터의 운영(프로비저닝 / 복구 / 배포)은 AWS가,
팀은 모델과 서비스에 집중



Amazon SageMaker HyperPod Inference Operator

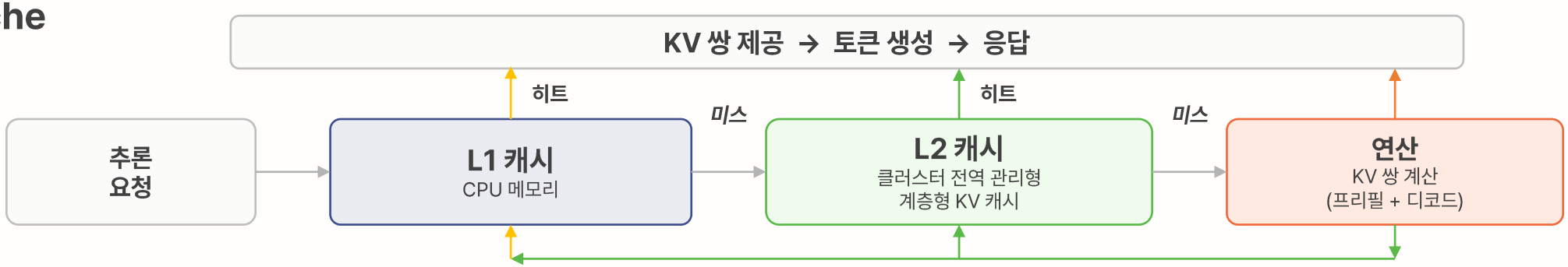
추론 테크닉의 가장 손쉬운 적용

Amazon SageMaker HyperPod Inference Gateway & 지능형 라우팅



Amazon EKS 관리형 애드온 · OpenAI 호환 엔드포인트 · 애플리케이션 변경 없음

KV Cache



새 KV 쌍은 두 계층에 모두 저장 · 히트 시 L2 → L1 승격

3

직접 서빙의 배경

Bagelcode



2012

창업 첫해부터 해외 시장을 향해

200+ 명

서울 하노이 드니프로

Casual

라이브 서비스 역량을 캐주얼 개발·퍼블리싱으로

Data-Driven → Agent-Driven

데이터와 코드로 일하던 회사가,
어떻게 에이전트와 일하는 회사가 됐는가?

01

데이터

데이터로 일하던 시기
전부 사람의 몫

자체 데이터 플랫폼 유저 행동 분석

02

AI

03

에이전트

01
데이터

데이터로 일하던 시기

02
AI

AI를 쓰던 시기
AI 격차 발생

게임잼 AI잼 · AI 페어

03
에이전트

AI 격차를 줄이기 위한 노력



게임잼



AI잼 · AI 페어

2023년부터 쌓은 경험 → 에이전트와 일하는 방식

01

데이터

데이터로 일하던 시기

02

AI

AI를 쓰던 시기

03

에이전트

에이전트와 일하는 시기
업무에 상주

조사 정리 초안

개인에서 팀, 조직의 기록으로

01

개인

전 직원 맥미니와
사내 에이전트 배포 완료

02

팀 에이전트

팀 채널에서
팀 문서 기반 답변

03

기록 에이전트

회의 기록과
참석자별 언어 통역

직접 모델 서빙의 이유는 저희도 같았다

01

데이터 레지던시

인사·재무·법무는 국내 리전·사내에서만

02

통제권

외부 API로는 모델 버전·설정 고정 불가

03

비용

에이전트 증가로 토큰 사용량 급증

04

외부 의존 위험

LLM 장애나 수출통제로 모델 접근 불가

직접 서빙까지의 타임라인

PHASE 0

Pre-Production

Demand Assessment

2026.03

→

PHASE 1

Amazon EKS

Hybrid Nodes

2026.06

→

PHASE 2

Tier 2 llm-d

2026.07

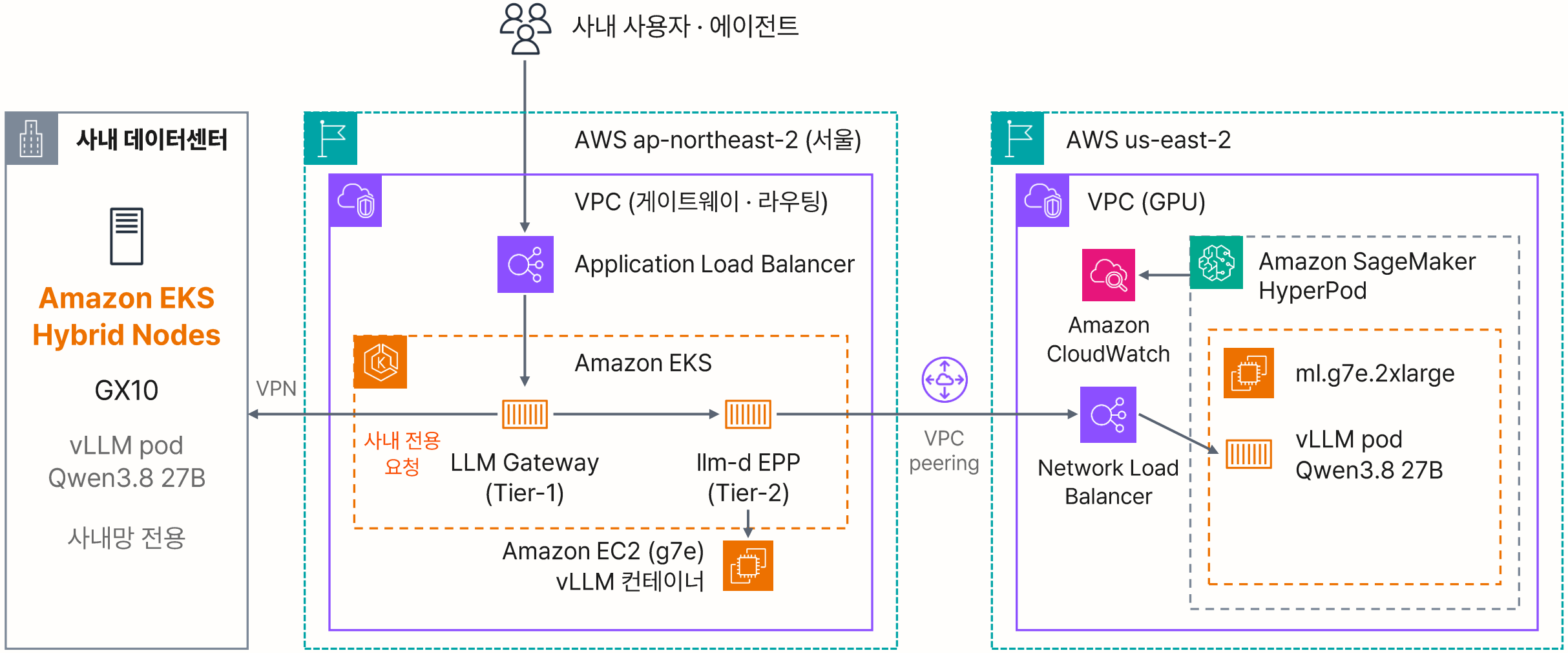
→

PHASE 3

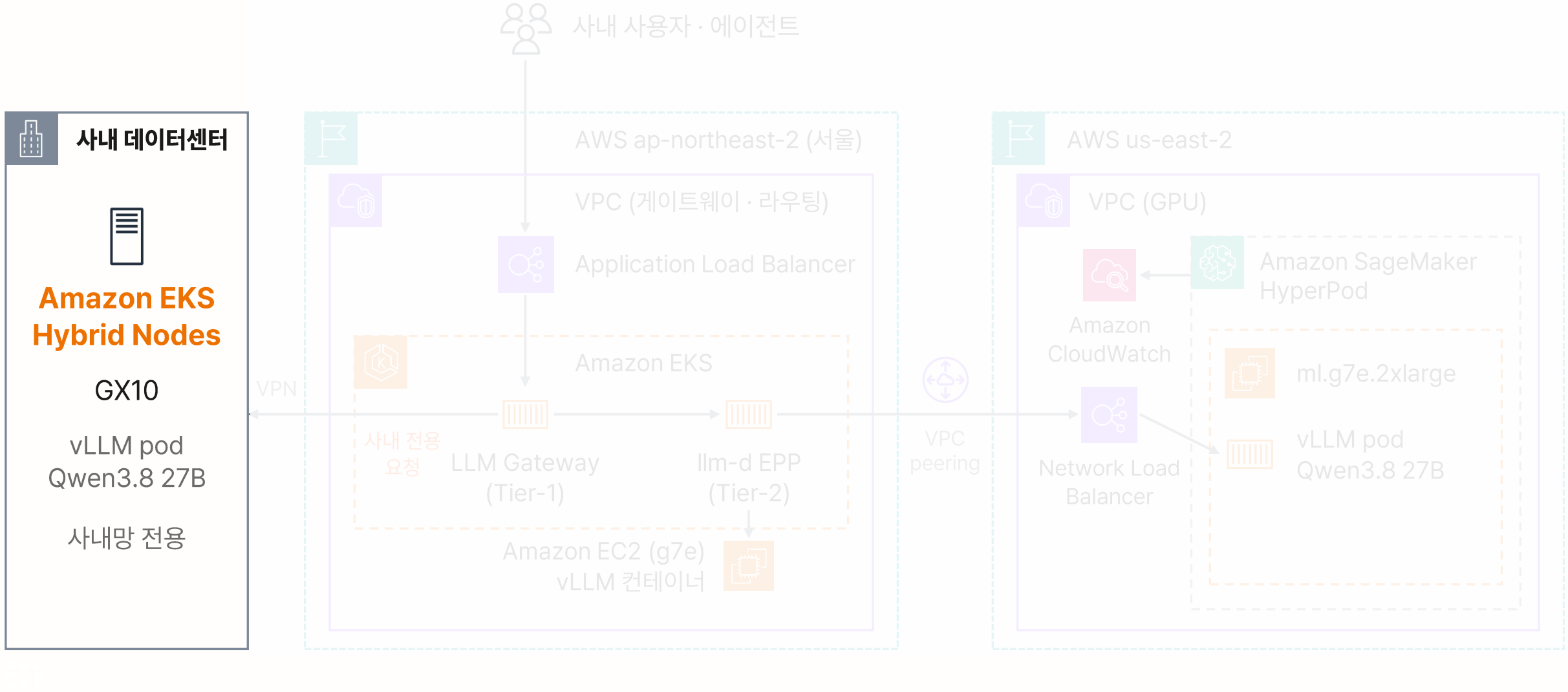
Amazon SageMaker HyperPod

2026.09

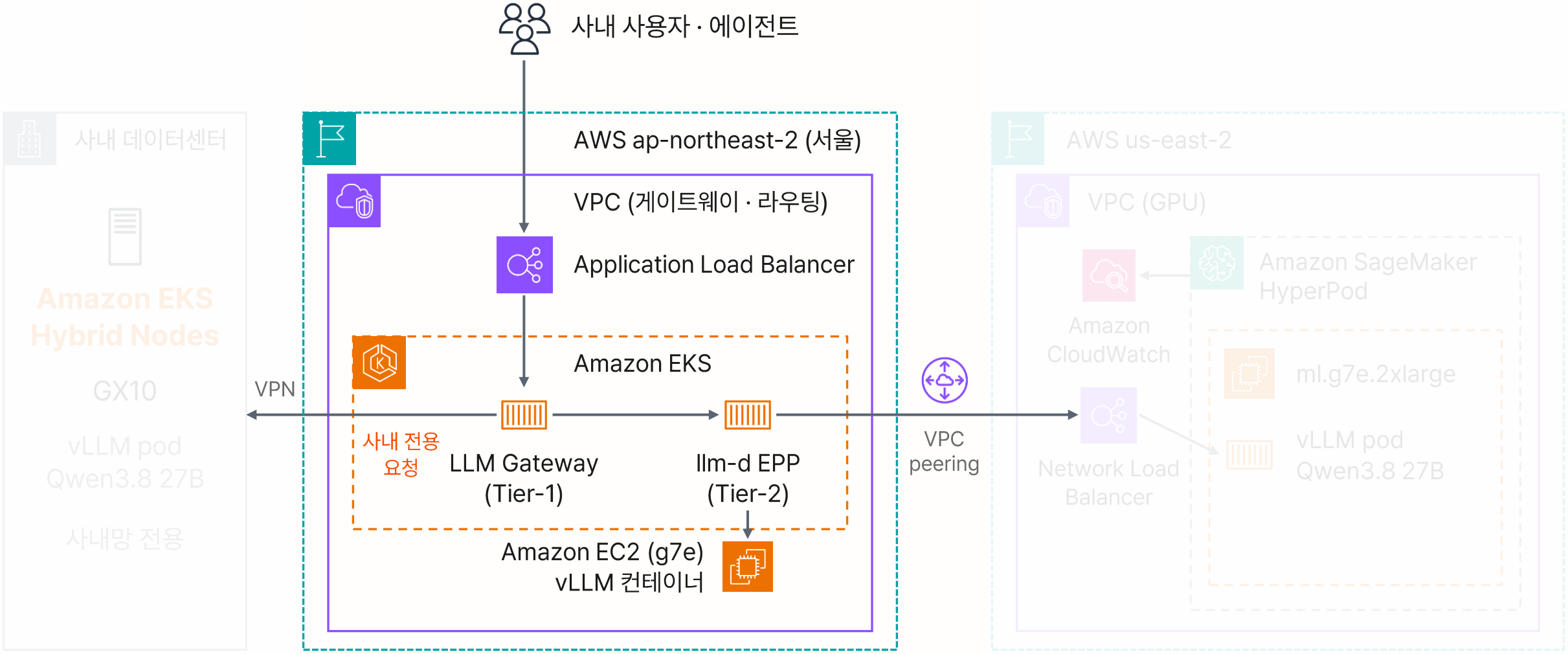
현재 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
사내망 전용
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
Amazon EC2 (g7e)
vLLM 컨테이너
VPC peering
AWS us-east-2
VPC (GPU)
Amazon CloudWatch
Amazon SageMaker HyperPod
ml.g7e.2xlarge
vLLM pod
Qwen3.8 27B
Network Load Balancer

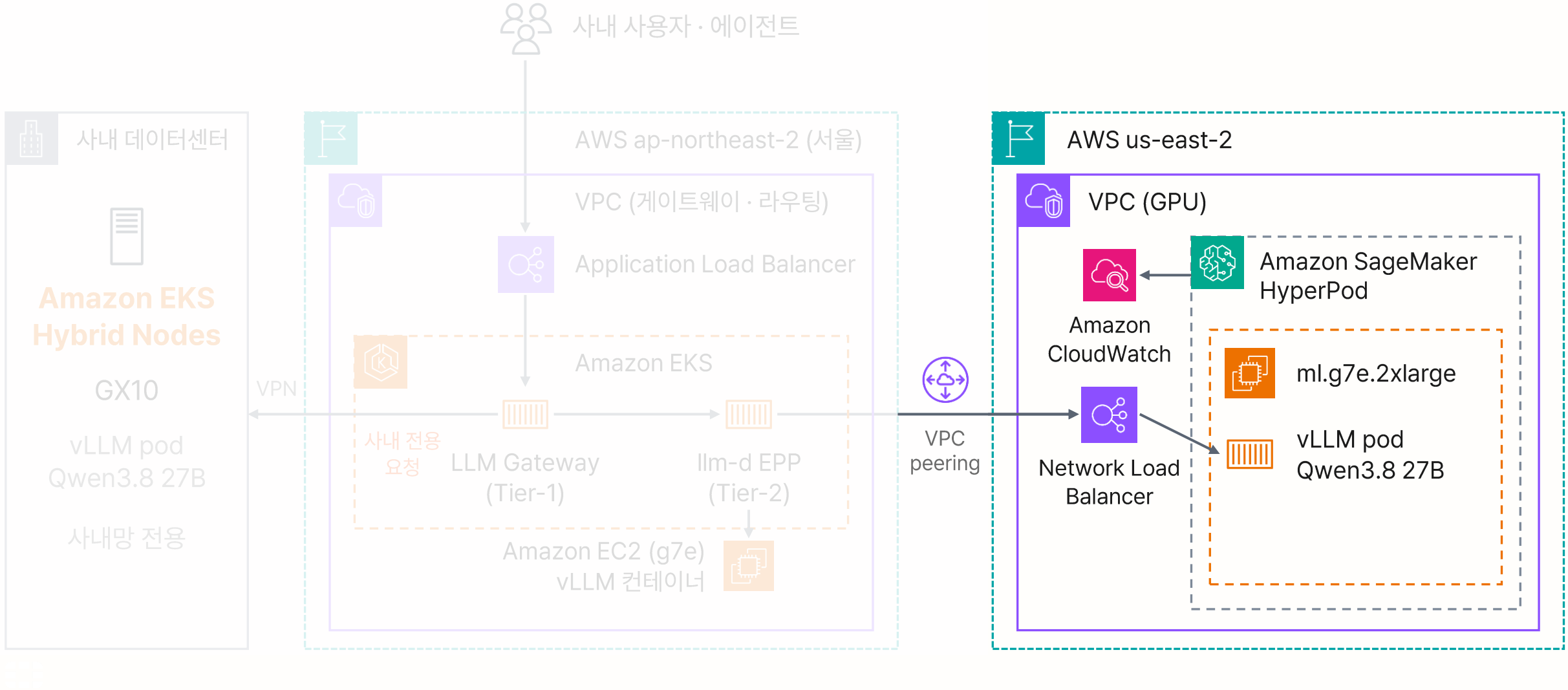
현재 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
사내망 전용
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
Amazon EC2 (g7e) vLLM 컨테이너
VPC peering
AWS us-east-2
VPC (GPU)
Amazon SageMaker HyperPod
Amazon CloudWatch
ml.g7e.2xlarge
Network Load Balancer
vLLM pod
Qwen3.8 27B

현재 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod Qwen3.8 27B
사내망 전용
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
VPN
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
Amazon EC2 (g7e) vLLM 컨테이너
VPC peering
AWS us-east-2
VPC (GPU)
Amazon SageMaker HyperPod
Amazon CloudWatch
ml.g7e.2xlarge
Network Load Balancer
vLLM pod Qwen3.8 27B

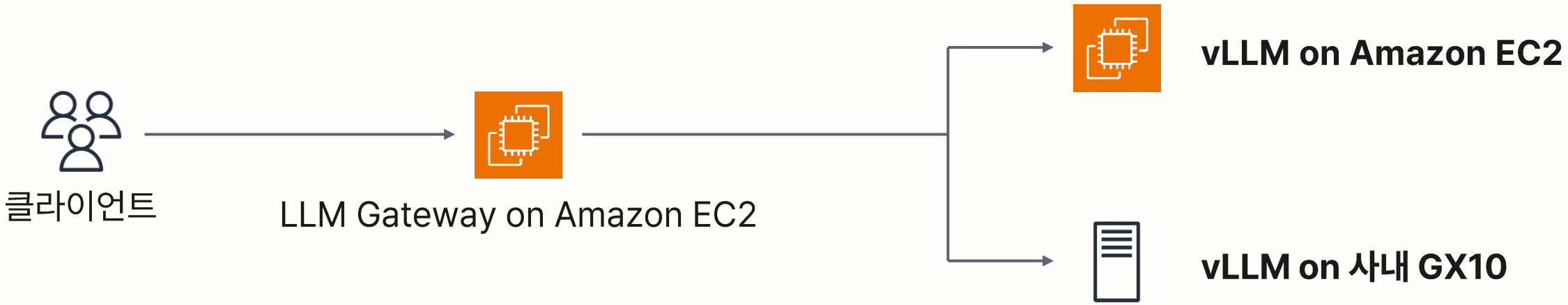
현재 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
사내망 전용
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
Amazon EC2 (g7e)
vLLM 컨테이너
VPC peering
AWS us-east-2
VPC (GPU)
Amazon CloudWatch
Amazon SageMaker HyperPod
ml.g7e.2xlarge
vLLM pod
Qwen3.8 27B
Network Load Balancer

3.1

Phase 0와 실측

PHASE 0. Pre-production



AS-IS

머신이 바뀌면 접속 주소도 바뀌어 클라이언트 수정 필요

TO-BE

게이트웨이가 논리 모델 이름으로 백엔드 탐색

사내 Agent 토큰 사용량 실측

사용자 약 120명 중

10명 → 80%

에이전트는 24시간 요청, 인당 사용량 증가

입력 토큰

230억 → 410억

월간 호출

47,000 → 86,000

전사로 퍼지면 20배

워크로드 실측

01

입력 대비 출력 토큰

1.3%

긴 입력, 짧은 답

02

최대 PREFILL 처리량

20K tok/s

28일 사용 로그 기준

워크로드 특징

Prefill-bound

우선순위

Prefill 처리량, KV Cache 용량

모델 선정

사내 업무 골든셋 5,400문항으로 업무 기준 평가

GLM-5.2

구조

744B MoE, FP8

가중치 크기

740 GB

통과율

89.0%

Qwen3.8

구조

27B dense, NVFP4

가중치 크기

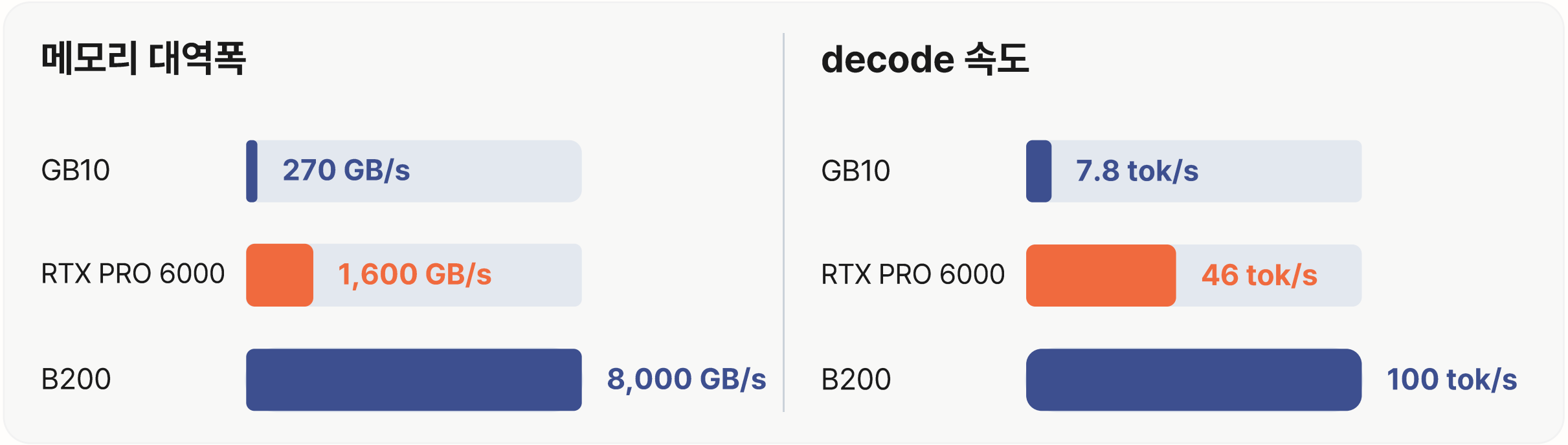
22 GB

통과율

88.7%

GPU 선택 기준

Qwen3.8 27B FP8, 단일 스트림



GPU 확보 방안

구매 | 외부 플랫폼 임대 | 클라우드



전력·네트워크·정전

온프레미스가 안고 가는 운영 이슈

- 베어메탈 계약, IDC 입주, 전원·회선 계약
- 세대가 바뀌면 다시 구매

GPU 확보 방안

구매 | 외부 플랫폼 임대 | 클라우드



저렴하지만 보안 위험

개인용은 가능, 회사에는 안정성·보안 문제

- 가중치와 데이터가 외부 호스트에 올라간다
- 호스트 사정으로 예고 없이 중단

GPU 확보 방안

구매 | 외부 플랫폼 임대 | **클라우드**



필요한 만큼만

구매와 임대에서 겪은 문제를 함께 해결

- 세대가 바뀌면 인스턴스 타입만 변경
- 데이터가 VPC 밖으로 나가지 않음
- 리전 선택으로 레지던시 충족

3.2

추론 최적화

Prefill 최적화

01

VLLM PREFIX CACHE

90%+

캐시 적중률

02

PREFIX-CACHE AWARE ROUTING

2배+

TTFT 단축

Quantization

25 → 22 GB

FP8 → NVFP4 가중치

46 → 64 tok/s

단일 스트림 decode, 1.4배

NVFP4

- Blackwell 세대에서 네이티브 지원
- 4비트 양자화된 가중치
- 민감한 층은 높은 정밀도로 유지
- KV 캐시는 FP8로 양자화

Multi-Token Prediction

MTP off

Output token throughput (tok/s)	**64.12**
Mean TPOT (ms)	**15.48**

MTP k=2

Output token throughput (tok/s)	**109.40**
Mean TPOT (ms)	**9.00**
Acceptance length	2.39

워크로드별 이득

- 코드와 JSON 2.0배, 산문 1.4배
- 동시성 1~2에서 최대, 높으면 이득 없음

Decode 최적화

Decode 속도 ≈ 메모리 대역폭 ÷ 가중치 크기 × 읽기당 토큰 수

양자화

읽는 양을 줄인다

25GB → 22GB

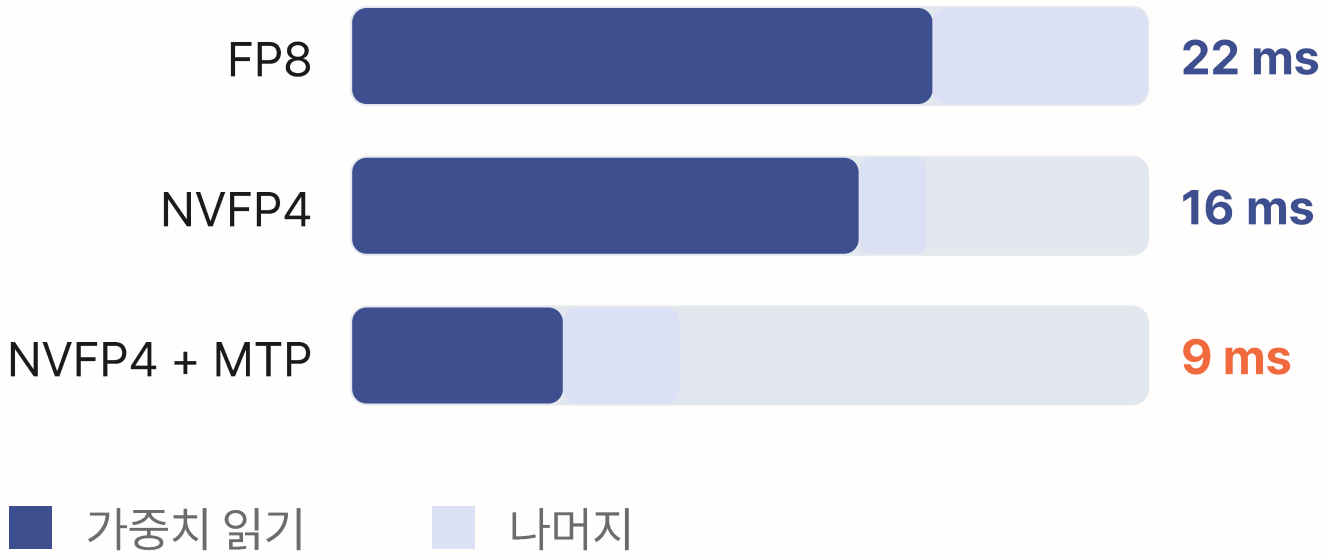
MTP

한 번에 더 뽑는다

평균 2.4토큰

Decode 최적화

토큰 하나를 만드는 시간



46 → 109

초당 생성 토큰

가중치 읽기가 시간의 대부분

양자화와 MTP 적용 후
22ms → 9ms

플랫폼화의 필요성

최적화는 한 번으로 끝나지 않는다

어려움

- 모델 교체마다 최적화를 다시
- GPU 수급: 확보가 먼저
- 배포, 네트워크, 노드 복구 직접 운영

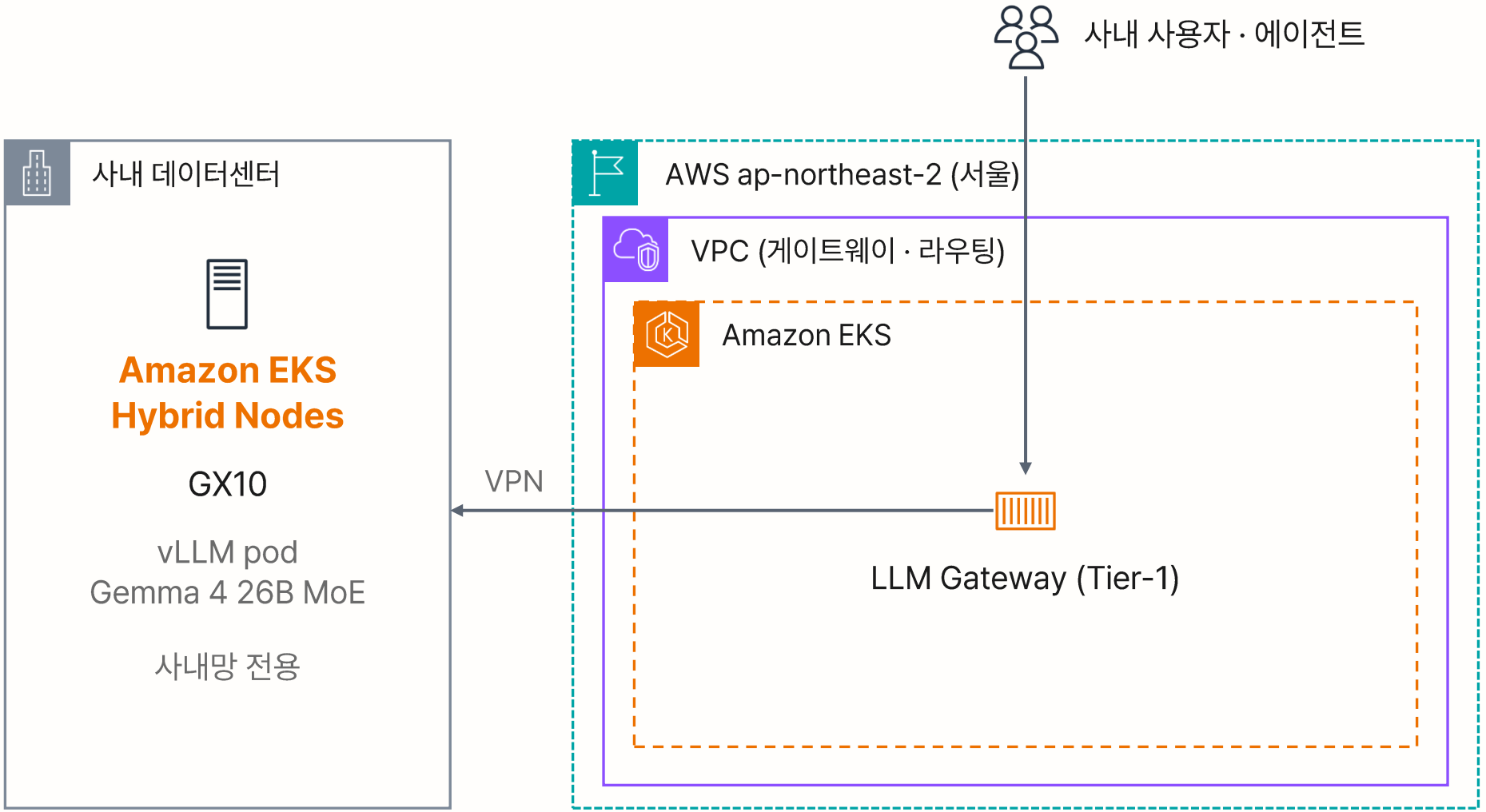
필요했던 것

- **클라이언트 호출 이름 고정**
- **GPU, 리전, 모델은 백엔드에서 교체**
- **배포와 운영 자동화**

3.3

아키텍처

Phase 1


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS
Hybrid Nodes
GX10
vLLM pod
Gemma 4 26B MoE
사내망 전용
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Amazon EKS
LLM Gateway (Tier-1)

Phase 2

request



Tier-1 베이글코드 게이트웨이 인증 / 권한 / 논리 모델 → 백엔드 그룹 선택

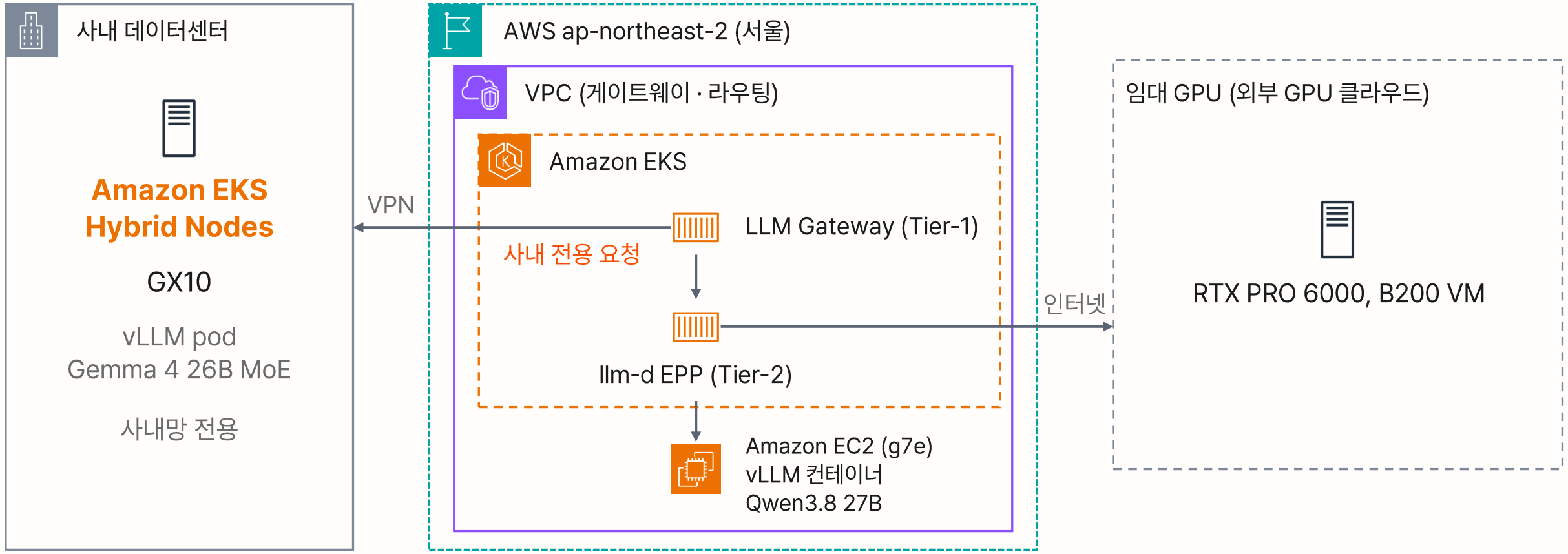


Tier-2 llm-d EPP (Endpoint Picker) Prefix-cache aware routing



vLLM 실제 추론과 KV 캐시

Phase 2


사내 데이터센터
Amazon EKS
Hybrid Nodes
GX10
vLLM pod
Gemma 4 26B MoE
사내망 전용
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Amazon EKS
LLM Gateway (Tier-1)
사내 전용 요청
llm-d EPP (Tier-2)
Amazon EC2 (g7e)
vLLM 컨테이너
Qwen3.8 27B
인터넷
임대 GPU (외부 GPU 클라우드)
RTX PRO 6000, B200 VM

3.4

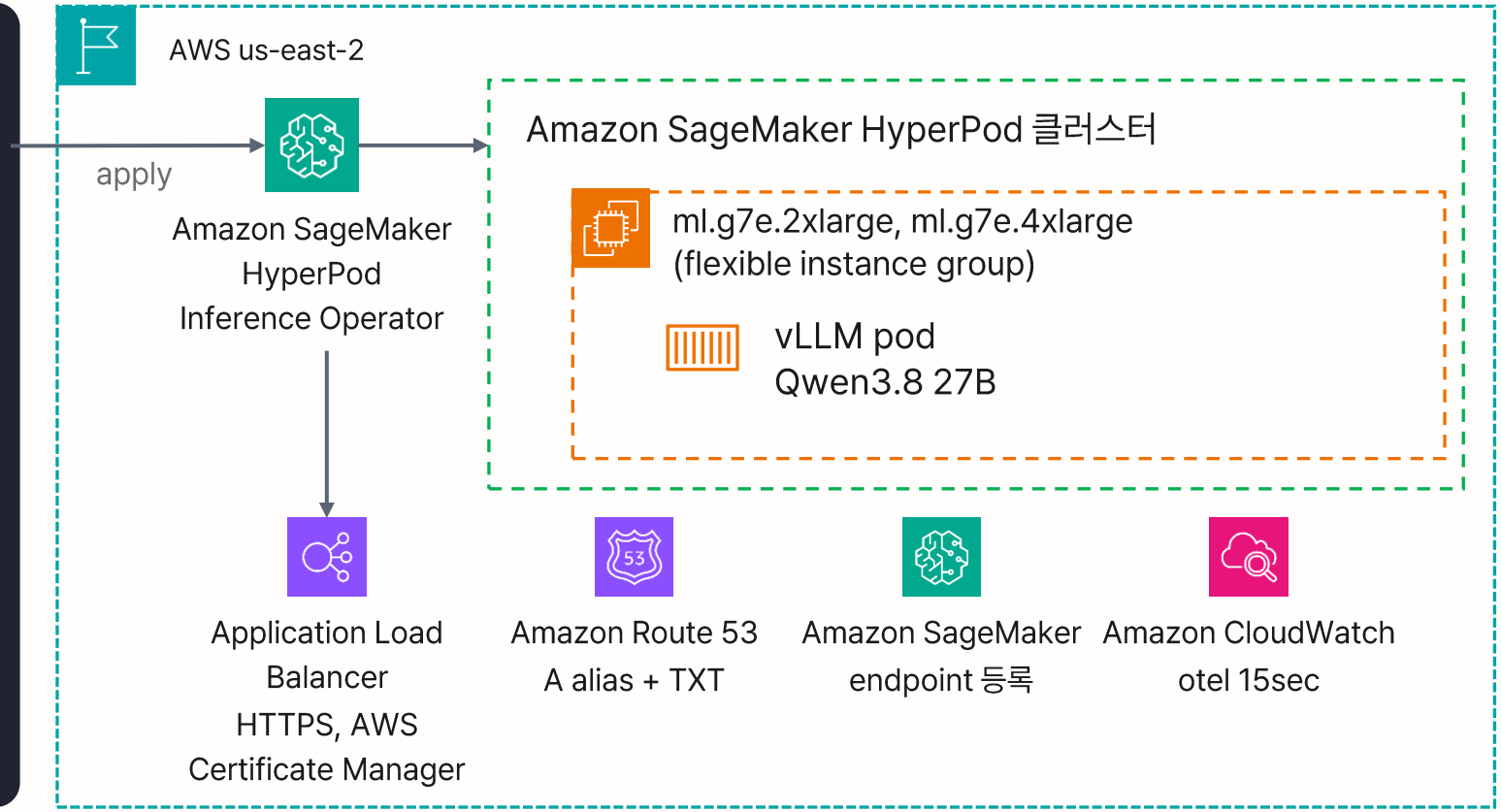
Amazon SageMaker HyperPod 도입

Phase 3. Amazon SageMaker HyperPod

Inference Operator

InferenceEndpointConfig

```
kind: InferenceEndpointConfig
spec:
  endpointName: qwen38-poc
  instanceTypes: [ml.g7e.2xlarge, ml.g7e.4xlarge]
  invocationEndpoint: v1/chat/completions
  modelSourceConfig:
    huggingFaceModel:
      modelId: nvidia/Qwen3.8-27B-NVFP4
  tlsConfig:
    customCertificateConfig:
      acmArn: arn:aws:acm:us-east-2:...
      domainName: qwen38.hyperpod-use2....
  metrics: {enabled: true, modelMetrics: {port: 8080}}
  worker:
    image: .../vllm:0.28-gpu-py312-cu130-...-sagemaker
    resources: {nvidia.com/gpu: "1", memory: 48Gi}
    environmentVariables:
      - name: SM_VLLM_SPECULATIVE_CONFIG
        value: '{"method":"mtp", ...}'
```



Amazon SageMaker HyperPod

Amazon SageMaker HyperPod

- 노드 프로비저닝과 복구
- 모델 배포 (설정 파일)
- TLS 연결과 DNS 레코드
- Amazon SageMaker endpoint 등록

베이글코드

- 인증과 권한
- 논리 모델 이름과 라우팅
- 폴백, 사용량, 예산, 모니터링
- AWS Certificate Manager

Phase 2와 Phase 3 비교

Phase 2. g7e, 임대 GPU

배포	**Script 기반**
TLS, DNS	**HTTP, IP 직접 지정**
캐퍼시티	**단일 타입, 수동 대응**
재현	**수작업 절차**
모니터링	**vLLM /metrics 수동**

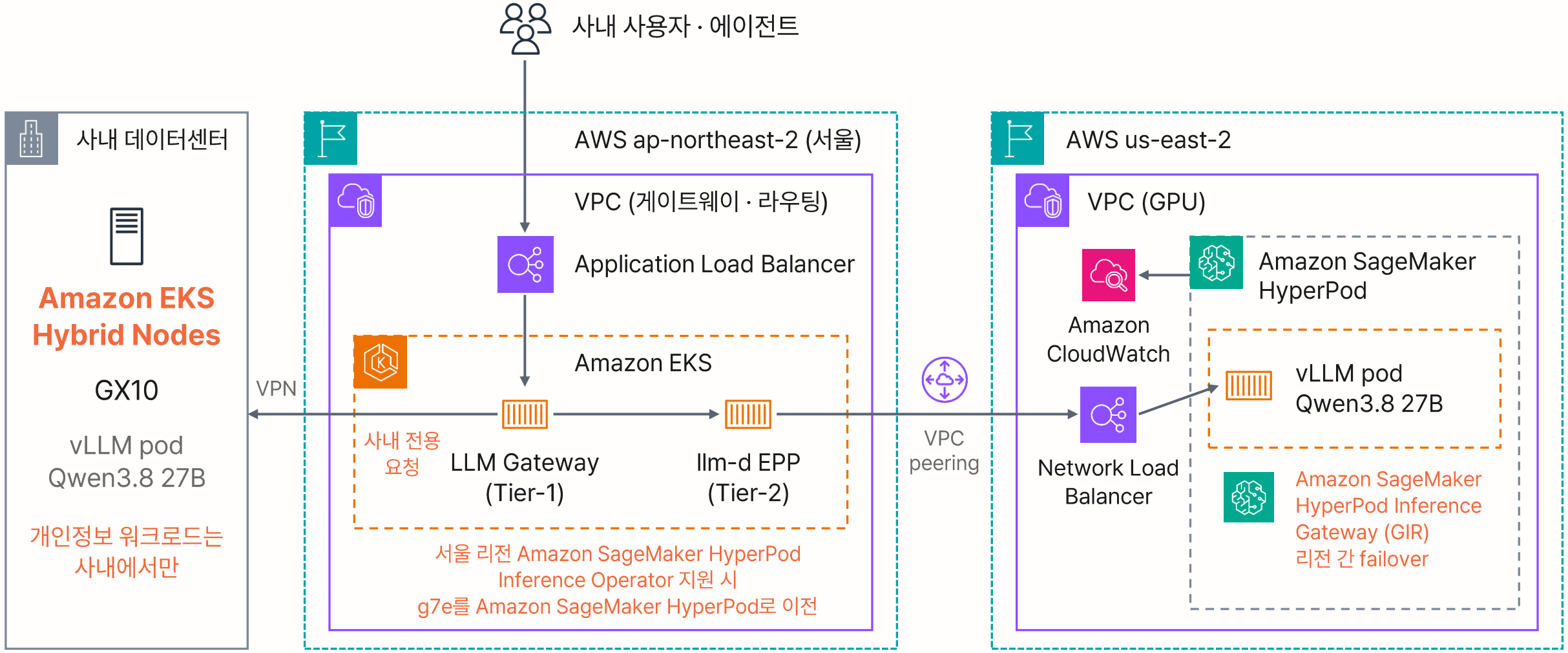
Phase 3. Amazon SageMaker HyperPod

배포	**설정 파일 기반**
TLS, DNS	**AWS Certificate Manager Amazon Route 53 자동화**
캐퍼시티	**flexible instance group**
재현	**Terraform 모듈**
모니터링	**otel → Amazon CloudWatch / Grafana**

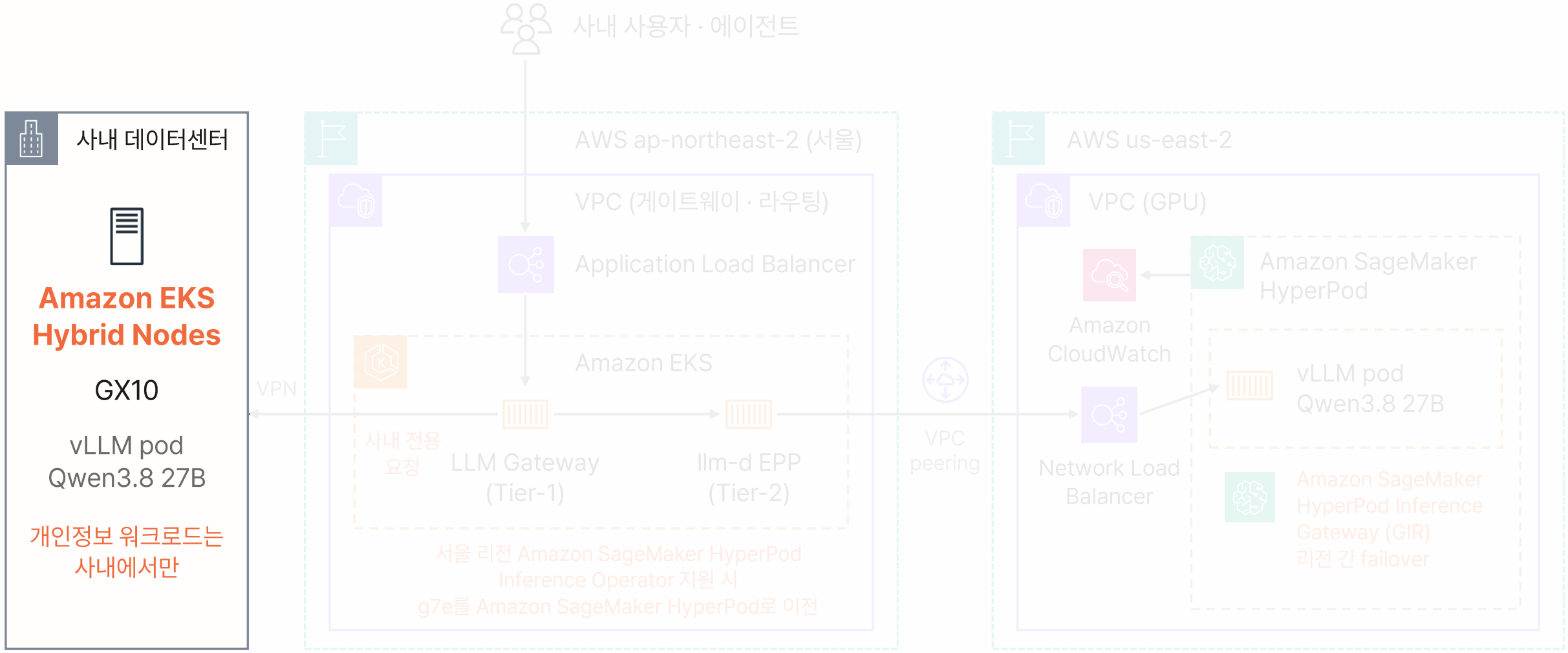
단계별 AWS 서비스의 확장

	AWS 서비스	우리가 얻은 것
Phase 0	Amazon EC2	GPU만, 나머지 직접 운영
Phase 1	Amazon EKS Amazon EKS Hybrid Nodes	사내와 클라우드 GPU를 한 제어면에
Phase 2	Amazon EKS 위 llm-d	모델 풀 안의 replica 라우팅
Phase 3	Amazon SageMaker HyperPod	배포, TLS, DNS, 노드 복구, 오토스케일링 자동화

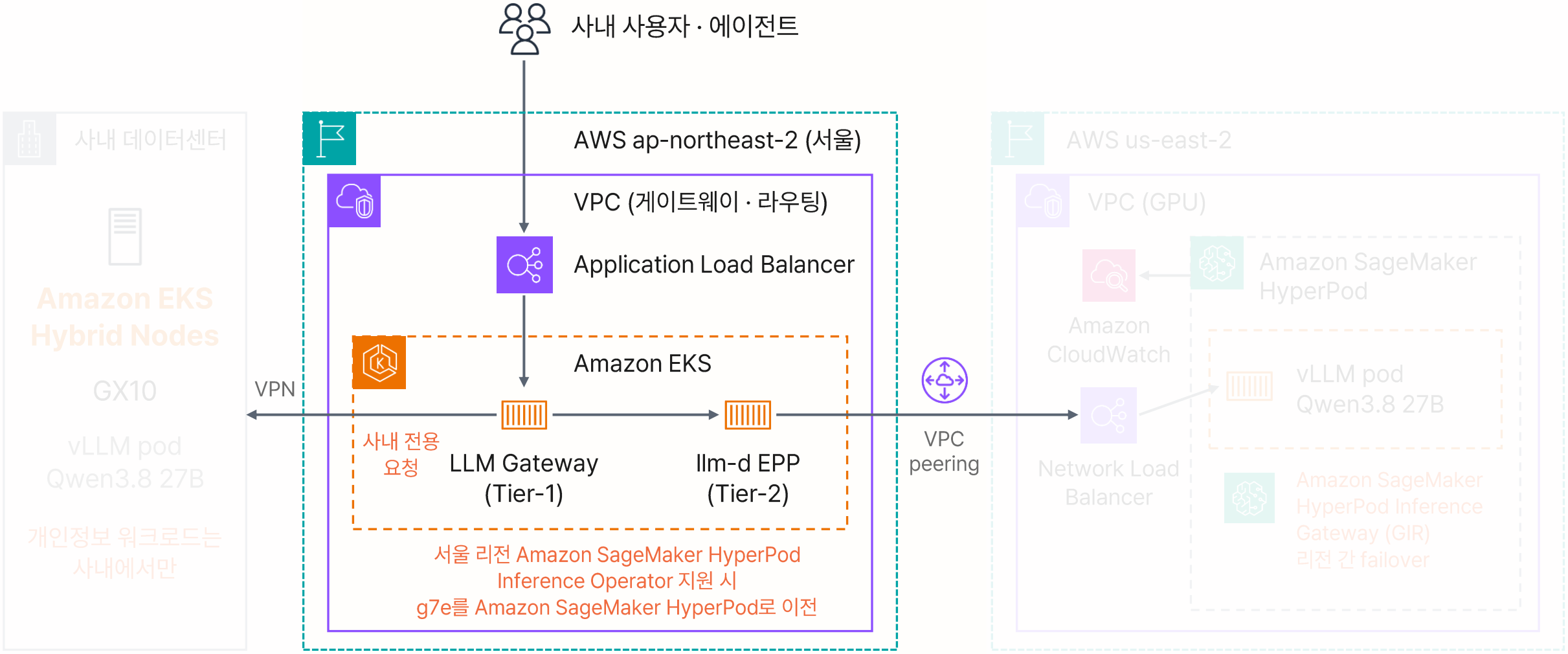
앞으로의 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
개인정보 워크로드는 사내에서만
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
서울 리전 Amazon SageMaker HyperPod Inference Operator 지원 시 g7e를 Amazon SageMaker HyperPod로 이전
VPC peering
AWS us-east-2
VPC (GPU)
Amazon CloudWatch
Amazon SageMaker HyperPod
vLLM pod
Qwen3.8 27B
Network Load Balancer
Amazon SageMaker HyperPod Inference Gateway (GIR)
리전 간 failover

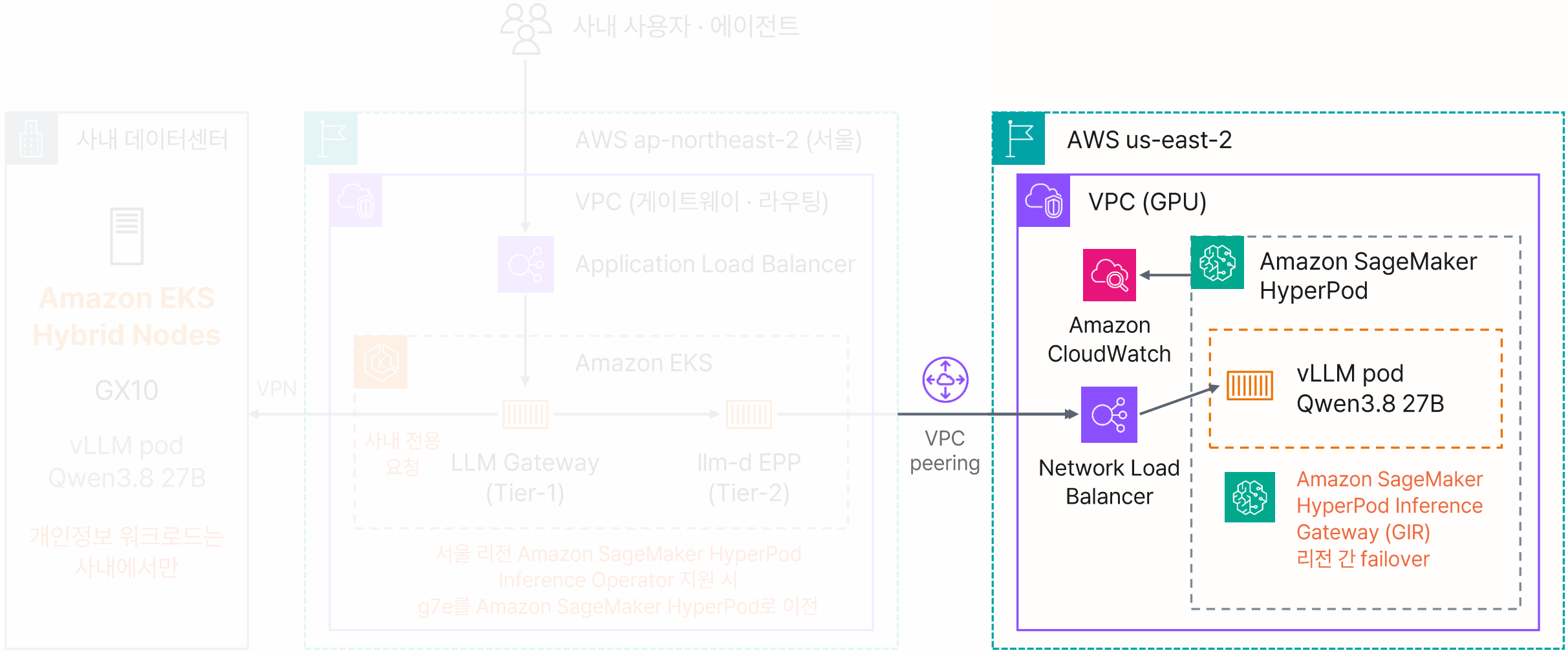
앞으로의 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
개인정보 워크로드는 사내에서만
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
서울 리전 Amazon SageMaker HyperPod Inference Operator 지원 시 g7e를 Amazon SageMaker HyperPod로 이전
VPC peering
AWS us-east-2
VPC (GPU)
Amazon CloudWatch
Amazon SageMaker HyperPod
vLLM pod
Qwen3.8 27B
Network Load Balancer
Amazon SageMaker HyperPod Inference Gateway (GIR)
리전 간 failover

앞으로의 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
개인정보 워크로드는 사내에서만
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
VPN
사내 전용 요청
LLM Gateway (Tier-1)
llm-d EPP (Tier-2)
서울 리전 Amazon SageMaker HyperPod Inference Operator 지원 시 g7e를 Amazon SageMaker HyperPod로 이전
VPC peering
AWS us-east-2
VPC (GPU)
Amazon CloudWatch
Amazon SageMaker HyperPod
vLLM pod
Qwen3.8 27B
Network Load Balancer
Amazon SageMaker HyperPod Inference Gateway (GIR)
리전 간 failover

앞으로의 아키텍처


사내 사용자 · 에이전트
사내 데이터센터
Amazon EKS
Hybrid Nodes
GX10
vLLM pod
Qwen3.8 27B
개인정보 워크로드는
사내에서만
VPN
AWS ap-northeast-2 (서울)
VPC (게이트웨이 · 라우팅)
Application Load Balancer
Amazon EKS
사내 전용
요청
LLM Gateway
(Tier-1)
llm-d EPP
(Tier-2)
서울 리전 Amazon SageMaker HyperPod
Inference Operator 지원 시
g7e를 Amazon SageMaker HyperPod로 이전
VPC
peering
AWS us-east-2
VPC (GPU)
Amazon
CloudWatch
Amazon SageMaker
HyperPod
vLLM pod
Qwen3.8 27B
Network Load
Balancer
Amazon SageMaker
HyperPod Inference
Gateway (GIR)
리전 간 failover

4

Lesson Learned

지금 시작하신다면

1

워크로드 실측

입력·출력 비율과 동시 요청 수부터

2

자체 평가셋으로 모델 선정

실제 업무 문항으로 비교

3

확보 가능한 GPU를 모델 구조에 맞추어 선택

MoE는 메모리 용량, dense는 대역폭

4

호출 규격과 정책은 게이트웨이에

모델 이름 · 인증 · 사용량 · 폴백을 한곳에

5

검증 절차를 처음부터

새 모델·옵션마다 평가셋과 스모크 테스트

정리

수요

에이전트가 수요를 만든다

플랫폼

새로운 모델을 빠르게
검증 · 최적화하는 전략과 플랫폼

관리형

운영은 HyperPod에 맡기고
라우팅 정책과 검증에 집중

직접 서빙을 고민한다면 지금이 시작하기 좋은 시점

aws FOR GAMES

Thank you

GS Neotek | MEGAZONE CLOUD