

IT 인프라 아키텍처 설계

TA가 알아야 할 온프레미스 IT 인프라

OT — TA의 일과 및 IT 인프라 큰 그림

3일 · 15세션 · TA(Technical Architect) 양성 과정

강사 박수현 ·  젠아이랩스(GenAI Labs)

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림



Session 0 · OT — TA의 일과 IT 인프라 큰 그림

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

최근 5년 한국 IT 3대 사건

🔥 ① 국가정보자원관리원 화재

2025.09.26 · 대전 본원

배터리 이설 작업 중 일부 전원 미차단 → 화재 → **정부 시스템 709개 마비** (4주+ 복구).

- 위기경보 「심각」 중대본 가동
- 모바일 신분증-국민신문고 중단
- 96개 시스템 대규 센터로 긴급 이전
- 단일 사이트 의존 + DR 미흡 노출

🎯 **본 강의 매핑** — Session 1 (DC) · 6 (백업·DR) · 8 (HA·운영)

🔒 ② SK텔레콤 USIM 해킹

2025.04.18 · 국내 통신 사상 최대

BPFdoor 악성코드로 SKT 핵심 시스템 침투 → **2,695만건 + IMEI 29만건** 유출 (9.82GB).

- 가입자식별키(IMS)-USIM 처리 정보
- 신고 지연 → 기업 책임론
- 집단소송 3만명+ 참여
- 「통신사 보안 인프라 점검」 의무화

본 강의 매핑 — Session 7 (보안 5대 도메인 다층 방어)

☀️ ③ 카카오 5일 장애

2022.10.15 · 역대 최장 메신저 장애

SK C&C 판교 DC 화재 → 배터리·UPS·운영도구 도미노 → **카카오 톡 127시간 33분 장애**.

- 다음·페이·뱅크·인증 연쇄
- DR 미흡·세션 동기화 부재
- 「카카오 먹통 보상법」 (방발법) 입법
- 부가통신사 안정성 의무 신설

본 강의 매핑 — Session 1·5·6·8 (전 영역에 교훈)

📌 **이 3건의 공통점** — 모두 「**TA가 다루는 5축(전력·NW·스토리지·보안·HA)**에서 한 점이 무너졌을 때」 사회 전체가 멈춤. 본 강의 3일은 이 5축을 한 번에 다룬다.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

3 / 45

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

OT — 회차 메타데이터

📋 세션 정보

- **회차:** 0 / 15 — OT (Orientation)
- **소속:** Day 1 — IT 인프라 큰 그림 + 서버·NW 기초
- **카테고리:** is-ethics (개요·철학)
- **예상 분량:** 본문 45 슬라이드 + 이미지 4장
- **강의 시간:** 약 45~60분
- **강사:** 박수현 — 젠아이랩스(GenAI Labs) **CEO / CTO**

🎯 핵심 메시지

TA는 응용·데이터·보안·운영을 떠받치는 인프라 책임자. **큰 그림 · 13개 도메인 · 8대 산출물 · 10대 의사결정 포인트**를 한 장에 켜다.

🎯 학습 목표 (Learning Objectives)

1. **TA의 정체** — 5대 아키텍트(EA·BA·AA·DA·TA) 매트릭스에서 본인의 책임 영역을 그린다
2. **시스템 8계층** — Application부터 보안·관제까지 각 계층의 책임·운영 주체를 짚는다
3. **인프라 4종 운영 모델** — 온프레미스·클라우드·하이브리드·멀티클라우드를 비교한다
4. **13개 도메인** — TA가 다루는 핵심 도메인 전체를 머릿속 지도로 묶는다
5. **8대 산출물** — 구성도·산정서·정책표·SOP·BOM·DR·매뉴얼·위험등록부의 목적과 작성 시점
6. **10대 의사결정 포인트** — TA가 현장에서 마주치는 핵심 판단 카탈로그

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

4 / 45

PART 1

TA가 누구인가 — 5대 아키텍트 매트릭스

응용·데이터·보안 위에 인프라가 있고, 그 인프라를 책임지는 사람이 **TA**.

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

5대 아키텍트 — 각자의 책임 영역



EA (Enterprise Architect)

기업 전체 표준·정합 - 전사 IT 전략·로드맵 - 표준 기술 스택 정의 - 거버넌스·컴플라이언스 - 산업 표준 (TOGAF·Zachman) - 정부 표준 (전자정부프레임워크)



BA (Business Architect)

업무·이해관계자 - 업무 프로세스 분석 - 요구사항 도출·정리 - KPI·MOE 정의 - 변경관리·이해관계자 합의 - 비즈니스 물



AA (Application Architect)

응용·플랫폼·통합 - 응용 아키텍처 - 마이크로서비스·API 설계 - 미들웨어·메시지큐 - 통합·인터페이스 설계 - 빌드·배포 파이프라인



DA (Data Architect)

데이터 모델·표준·메타 - 논리·물리 데이터 모델 - 마스터·메타 데이터 - 표준 코드·분류 체계 - ETL·데이터 흐름 - 데이터 거버넌스·품질



TA (Technical Architect) ★

인프라·서버·NW·HA·보안·운영 - 서버·OS·DB·미들웨어 사이징 - 네트워크 토폴로지 - 스토리지·백업·DR - 보안 아키텍처 - HA·용량·관찰성·자동화



5인의 협업

- 산출물은 5명의 합의·검토를 거쳐 확정
- TA는 모든 아키텍트의 결정을 물리화·정량화
- EA(표준) → AA(응용) → DA(모델) → BA(시나리오) → TA(인프라)

5대 아키텍트 관계도 — 추상 → 설계 → 물리화

 추상 · 정책·요구

- EA — 전사 표준·로드맵
- BA — 업무 분석·KPI

 설계 · 응용·데이터

- AA — 응용·MSA·API
- DA — 데이터 모델·ETL

 물리화 · 시스템

- TA — 서버·NW·스토리지·HA·보안·운영·BOM

입력 → TA 흐름

- EA → AA·DA (기술·메타 표준) · EA … TA (컴플라이언스)
- BA → AA·DA (기능·데이터 요구) · BA … TA (SLA)
- AA → TA (런타임 요구) · DA → TA (용량·I/O 요구)

핵심: EA·BA가 추상으로 정의 → AA·DA가 설계로 구체화 → TA가 물리 시스템으로 떨어뜨림. TA가 멈추면 다른 4명의 결정은 「문서」로만 남는다.

5대 아키텍트 매트릭스 — 입력 → 출력 흐름

역할	핵심 질문	주 산출물	TA에게 주는 입력값
EA	"전사 표준·로드맵은?"	IT 전략서·표준 가이드·아키텍처 원칙	표준 기술 스택·인증 요구사항·금기 솔루션
BA	"업무·사용자는 누구?"	업무 흐름도·요구사항 명세·KPI	동접·트래픽·피크 시간대·SLA 기대값
AA	"응용·통합 구조는?"	응용 아키텍처도·API 명세·시퀀스	API TPS·세션 크기·메시지 큐 처리량
DA	"데이터·저장은?"	논리/물리 모델·DDL·ETL 설계	저장량·증가율·보관기간·압축률
TA ★	"이 모든 것을 어떤 인프라에?"	구성도·BOM·산정서·정책·SOP·DR	(4명의 모든 산출물을 물리화)

TA의 정체성: TA는 단독으로 결정하지 않는다. EA·BA·AA·DA의 논리적 결정을 받아 물리 BOM·서버 SKU·NW 토폴로지·산정서로 변환하는 것이 본업.

TA의 업무 사이클 — 일일 · 주간 · 월간

 일일 (Daily)

- 08:30 NMS-APM-SIEM 대시보드 확인
- 09:00 야간 백업-배치 결과 점검
- 09:30 전일 장애-경보 리뷰
- 10:00 장애 티켓 처리-변경 작업
- 13:00 현장-발주처 회의
- 15:00 산출물 작성-도면 갱신
- 17:00 익일 변경 작업 사전 검토
- 18:00 일일 운영 일지 마감
- 상사: 발주처-AA-DA-BA 문의 응대

 주간 (Weekly)

- 주간 운영 보고서 작성
- 용량-성능 추이 리뷰 (CPU-Mem-디스크-NW)
- 변경관리 회의 (CAB)
- 보안 패치-취약점 점검 결과
- 백업 무결성 검증 (랜덤 복구 시험)
- 라이선스 만료-갱신 일정 점검
- 신규 사업-RFP 분석 (PM-영업과 협업)
- 협력사-벤더 미팅

 월간 (Monthly)

- 월간 가용성-SLA 보고서
- 위험 등록부 갱신 (월간 회의 안건)
- DR 훈련 (Tabletop-Walkthrough)
- 보안 진단 결과 보고
- TCO-예산 집행률 리뷰
- 분기 1회: 모의 침투-취약점 정밀 점검
- 반기 1회: DR Live 훈련
- 연 1회: 인프라 로드맵-표준 갱신

주니어 TA의 흔한 오해: "TA = 설계만 하는 사람" — 실제로는 설계 30% + 운영-점검 50% + 보고-협업 20%.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현 9 / 45

#10

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

5대 아키텍트 — 각자의 책임 영역

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현 10 / 45

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 5 / 155

TA가 다루는 IT 인프라 자산 — 6대 분류

**HW (하드웨어)**

- 서버 (Rack-Blade-HCI-GPU)
- 스토리지 (DAS-NAS-SAN-Object)
- 네트워크 장비 (SW-라우터-FW-LB)
- 백업 장비 (Tape-VTL-D2D)
- UPS-발전기-PDU-랙-KVM
- 부속품 (SFP-케이블-HBA)

**SW (소프트웨어)**

- OS (Linux-Windows-Unix)
- 가상화 (VMware-KVM-Hyper-V)
- 컨테이너 (Docker-K8s-OpenShift)
- DBMS (Oracle-MSSQL-MySQL-PostgreSQL)
- 미들웨어 (WAS-MQ-DNS-LDAP)

**라이선스**

- OS 코어-소켓-사용자 단위
- DB CPU-코어-EE/SE 라이선스
- 가상화 (Per-Socket-Per-Core-Per-VM)
- 백업 SW 용량-노드 라이선스
- 모니터링-관제 SW 호스트 단위
- 보안 SW 채널-세션-노드

**인증서·키**

- SSL/TLS 인증서 (CA-도메인)
- 코드 서명 인증서
- VPN 인증서·키
- HSM-KMS 마스터 키
- KCMVP 검증필 암호모듈 키

**네트워크 자원**

- 회선 (전용-인터넷-MPLS)
- 공인 IP-사설 IP-서브넷
- 도메인-DNS-VPN-VLAN ID

**서비스·계약**

- 유지보수 (HW-SW-NW)
- DC 코로케이션-회선 통신사
- 보안 관제 (MSSP) · 클라우드
- 사외 백업-DR

**인적 자원**

- NOC 3교대 · On-call 명단
- 협력사-벤더 엔지니어 연락처

자산 관리(CMDB)는 모든 TA 업무의 출발점 — **자산이 안 보이면 설계-운영-예산-보안 모두 무너진다.**

PART 2

시스템 8계층 — Application부터 관제까지

TA는 8개 계층 전체를 머릿속에 그리고 있어야 한다.

시스템 8계층 — 전체 스택 조망



보안·관제(⑧)는 모든 계층을 **수직으로 관통**한다 — 각 계층의 보안 정책·이벤트 로그·성능 메트릭을 통합 수집.

① Application (응용) — 사용자가 보는 모든 것

 **책임 영역**

- **사용자 인터페이스**: 웹·모바일·키오스크
- **업무 로직**: 비즈니스 룰·워크플로
- **API**: REST·GraphQL·gRPC
- **배치**: 일·월·연 정기 작업
- **외부 연계**: 타 시스템 인터페이스

 **대표 기술**

- Java (Spring Boot) · .NET · Node.js · Python · Go
- React · Vue · Angular · Next.js
- Spring Batch · Quartz · Airflow

 **TA의 관심사**

- **응용 인스턴스 수** → 컨테이너 수·VM 수 산정
- **TPS·동접·세션** → CPU·메모리·NW 대역
- **상태 저장 여부 (Stateful)** → 세션 클러스터·sticky LB
- **무중단 배포** → Blue/Green·Canary
- ****외부 연계** → 망분리·DMZ 설계
- **응답시간 SLA** → APM 모니터링 임계

응용 설계는 AA의 영역, TA는 **인프라 NFR**(가용성·성능·확장성)을 책임.

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

③ Database (데이터베이스) — 모든 상태의 종착점

DBMS 분류

- **RDBMS** — Oracle · MSSQL · MySQL · PostgreSQL · Tiberio · MariaDB · DB2
- **NoSQL**
- Document: MongoDB · CouchDB
- Column: Cassandra · HBase · ScyllaDB
- Key-Value: Redis · DynamoDB
- Graph: Neo4j
- **NewSQL** — TiDB · CockroachDB · YugabyteDB · Spanner
- **시계열** — InfluxDB · TimescaleDB
- **검색·벡터** — Elasticsearch · OpenSearch · Milvus · Weaviate · pgvector

TA의 관심사

- **IOPS-Latency** — DB는 항상 디스크 병목
- **Connection Pool** — 응용 ↔ DB 풀 크기
- **HA 모드** — Active-Standby-Active-Active-Sharding
- **복제 방식** — Sync-Async-Quorum
- **백업 RPO** — 풀+증분, 트랜잭션 로그
- **라이선스** — Oracle Core-SE/EE, MSSQL Per-Core
- **DBA와의 협업** — TA가 DBA를 검직하지 않지만 **인프라 사이징은 TA 책임**

DB는 인프라 비용의 30~50% — 라이선스·서버·스토리지·HA 전부 비쌈.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

15 / 45

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

⑥ Hardware (하드웨어) — CPU·메모리·디스크·NIC의 물리 계층

서버 폼팩터

- **Tower** — 지점·소규모
- **Rack 1U/2U/4U** — 표준 데이터센터
- **Blade** — 고밀도 (HPE Synergy·Cisco UCS)
- **HCI** — 컴퓨트+스토리지 일체 (Nutanix·VxRail·SimpliVity)
- **Composable** — 자원 풀링 (HPE Synergy·Dell PowerEdge MX)
- **OCP** — 오픈 컴퓨트 (하이퍼스케일)

CPU 아키텍처

- **x86** — Intel Xeon Scalable Gen 5/6 · AMD EPYC Genoa/Turin
- **ARM** — Ampere Altra · NVIDIA Grace · AWS Graviton
- **Power** — IBM Power9/10 (AIX·Linux)
- **Mainframe** — IBM Z16 (z/OS)
- **SPARC** — Oracle (Solaris)

메모리·스토리지

- **DRAM** — DDR4 · DDR5 · ECC · RDIMM · LRDIMM
- **CXL Memory** — 메모리 풀링 (2025+)
- **NVDIMM** — Persistent Memory
- **SSD** — SATA · SAS · NVMe (Gen 4/5)
- **HDD** — 7.2K SATA-SAS · 10K/15K SAS
- **폼팩터** — U.2/U.3 · E1.S · E3.S

NIC·HBA

- 1G·10G·25G·40G·100G·200G·400G·800G
- **SR-IOV** · **SmartNIC/DPU** (BlueField · Pensando)
- **RDMA** — RoCE · iWARP · IB
- **HBA** — FC 16/32/64G · FCoE · NVMe-oF

GPU·AI 가속기

- **NVIDIA** — L4 · L40S · A100 · H100 · H200 · B200/B300 · HGX/DGX
- **AMD** — MI300X · MI325X · MI350
- **Intel** — Gaudi 3
- **국산** — 사피온 X220/X330 · 리벨리온 아틀
- **GPU 가상화** — vGPU · MIG · SR-IOV
- **GPU 클러스터** — NVLink · InfiniBand HDR/NDR

TA의 관심사

- 단가 · 발열 · 전력
- EOL · 보증
- 호환 라이선스 (vGPU 라이선스 별도)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

16 / 45

⑦ Network (네트워크) — L1~L7 통신 계층

**L1~L2**

- 케이블 (Cat 6a · 광 SM/MM)
- SFP-QSFP-OSFP 모듈
- 이더넷 1G~800G
- WiFi 6/6E/7 (802.11ax/be)
- MAC-ARP-VLAN-LACP-STP

**L3~L4**

- IP · CIDR · 라우팅 (Static-OSPF-BGP-IS-IS)
- VRF · MPLS L3VPN
- TCP · UDP · QUIC
- VRRP · HSRP · BFD · ECMP

**L5~L7**

- DNS · DHCP · NTP · LDAP
- HTTP/1.1 · 2 · 3 · TLS 1.2/1.3
- SMTP · FTP · SSH
- gRPC · WebSocket
- 로드밸런서 (L4~L7) · GSLB

**보안 NW**

- Firewall · NGFW · UTM
- IPS · IDS · WAF
- Anti-DDoS · VPN

**TA의 관심사**

- 토폴로지** — 3-Tier vs Spine-Leaf
- L3 ↔ L2 분리** — 장애 격리
- VLAN 설계** — 운영·관리·DMZ
- HA 절체 시간** — VRRP <1s, MSTP <5s
- MTU-Jumbo** — 9000 Bytes (성능)
- NW 자동화** — Ansible · Netmiko · NetBox
- 관찰성** — NetFlow · sFlow · IPFIX
- 벤더** — Cisco · Juniper · Arista · HPE Aruba · 다산네트웍스

NW는 TA의 가장 큰 비중. 3·4세션 통째로 다룸.

PART 3

온프레미스 · 클라우드 · 하이브리드 · 멀티클라우드

어디에 인프라를 둘 것인가 — 4가지 운영 모델의 장단점.

인프라 운영 4종 — 비교 매트릭스

모델	위치	자본 비용	운영 비용	확장성	통제력	적합
온프레미스 (On-Premises)	자사 DC 또는 코로케이션	높음 (CapEx 일시)	중간 (인력·전력·유지보수)	낮음 (장비 발주 필요)	최고	보안 민감·공공·금융 핵심
퍼블릭 클라우드	AWS · Azure · GCP · NCP · KT Cloud	낮음	높음 (OpEx 월정)	최고 (분 단위)	낮음	신규·스타트업·글로벌·변동 부하
하이브리드	온프레미스 + 클라우드 연결	중간	중간~높음	중간~높음	중간	대다수 기업 (현실적 표준)
멀티클라우드	2개 이상 클라우드 동시	낮음	높음 (운영 복잡)	최고	중간	벤더 종속 회피·DR·법규 분산

한국 2026년 현실: 공공 92%·금융 85% 이상이 온프레미스 + 일부 하이브리드. 순수 퍼블릭 클라우드는 신규 서비스·스타트업·B2C 중심.

온프레미스 — 왜 여전히 표준인가

✔ 장점

- 물리적 통제 — 데이터·HW 위치 확정
- 법규 대응 용이 — 개인정보·금융·공공
- 장기 TCO 유리 — 5년+ 사용 시 클라우드보다 저렴
- 성능 예측 가능 — Noisy Neighbor 없음
- 외부망 의존 없음 — 인터넷 단절도 운영 지속
- 기존 라이선스 재활용 — Oracle·VMware Bring-Your-Own

✖ 단점

- 초기 자본 (CapEx) 큰 부담 — 수억~수십억
- 확장 시 발주·구축 lead time — 8~12주
- 인력 확보 — TA·운영·DBA·NW·보안 풀세트
- 재해 대응 비용 — DR 사이트 추가 부담
- 노후화·EOL 압박 — 4~5년 단위 교체
- 기술 발전 속도 추적 어려움

TA의 결정 포인트: 단순 CapEx vs OpEx가 아니라 **법규·통제·인력·기간 4축**으로 평가.

퍼블릭 클라우드 — 변동·신규·글로벌의 표준

✓ 장점

- 즉시 자원 확보 — 분 단위 프로비저닝
- CapEx 최소화 — 초기 투자 거의 0
- 글로벌 확장 — 다중 리전 즉시
- PaaS-SaaS 풍부 — DB·MQ·ML 매니저드
- 자동 스케일 — 부하 따라 +/-
- 벤더 운영 책임 — HW·OS·MW까지 위임

✗ 단점

- 장기 비용 — OpEx 누적 부담
- 벤더 종속 (Lock-in) — 이전 어려움
- 데이터 주권 — 해외 리전 우려
- 네트워크 비용 — Egress 비싸짐
- 컴플라이언스 — 일부 산업 제한
- 인터넷 장애 = 서비스 장애

한국 클라우드 공급자

분류	사업자
글로벌	AWS · Azure · GCP · Oracle Cloud · IBM Cloud
국내 (CSAP)	네이버 클라우드(NCP) · KT Cloud · NHN Cloud · 삼성SDS · 카카오엔터프라이즈
공공 클라우드	CSAP 인증 사업자만 공공 사업 가능 (상·중·하 등급)

하이브리드 · 멀티클라우드 — 현실의 표준



하이브리드 (Hybrid)

정의: 온프레미스 + 클라우드를 하나의 시스템처럼 운영

대표 사례 - DR: 운영 = 온프레미스, DR = 클라우드 - Burst: 평소 = 온프레미스, 피크 = 클라우드 확장 - 개발/운영 분리: 개발·테스트 = 클라우드, 운영 = 온프레미스 - 데이터 분리: 민감 데이터 = 온프레미스, 분석 = 클라우드

연결 기술 - VPN · 전용선 (Direct Connect·ExpressRoute·Cloud Connect)
- Anthos · Azure Arc · AWS Outposts



멀티클라우드 (Multi-Cloud)

정의: 2개 이상의 클라우드를 동시 사용

도입 동기 - 벤더 종속 회피 (Lock-in 분산) - DR 사이트 분산 - 법규 (국내 + 해외 데이터) - 각 클라우드의 강점 활용 (AI = GCP/Azure, 게임 = AWS) - 계약·가격 협상력 확보

난이도 - 거버넌스·비용·운영 복잡도 ↑↑↑ - IAM 통합·네트워크 연동·인증서 관리 부담 - CSPM·CNAPP 같은 멀티클라우드 보안 도구 필수

클라우드 서비스 계층 — *IaaS · PaaS · SaaS*



모델	사용자 책임	공급자 책임	예시
On-Prem	App-MW-DB-OS-VM-HW-NW-시설	없음	자가 DC
IaaS	App-MW-DB-OS	VM-HW-NW-시설	EC2-NCP
PaaS	App-데이터	DB-MW-OS-VM-HW-NW-시설	RDS-App Svc
SaaS	데이터-사용자 관리	모두	M365-Salesforce

PART 4

TA가 다뤄야 할 13개 핵심 도메인

큰 그림을 본 후, 이제 깊이 들어갈 13개의 영역.

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

TA 13 도메인 — 카탈로그·미리보기

본 OT에서는 13 도메인의 **이름·범위·세션 매핑**만 확인. 깊이는 Session 1~9 에서.

① 데이터센터·전력·냉각

Tier · UPS · CRAC · PUE · 상면·랙
→ **Session 1**

② 서버·OS

폼팩터 · CPU · RHEL · Windows · Unix
→ **Session 3** (A-D PART)

③ DB·미들웨어

RDBMS · NoSQL · WAS · MQ · 캐시
→ **Session 3** (E PART)

④ GPU·AI 인프라

H100 · MIG · NVLink · InfiniBand
→ **Session 3** (F PART)

⑤ 가상화·컨테이너

VMware · KVM · K8s · OCI
→ **Session 3** (G-H PART)

⑥ 네트워크 L1~L7

OSI · TCP/IP · VLAN · LB · GSLB · SDN
→ **Session 4-5**

⑦ 스토리지·백업·DR

SAN · NAS · RAID · Tape · DR
→ **Session 6**

⑧ 보안 5대 도메인

NW · EP · Data · 트렌드 · 거버넌스
→ **Session 7**

⑨ HA·이중화

N+1 · 2N · Active-Active · 클러스터
→ **Session 8** (A-B)

⑩ 용량·성능·관찰성

TPS · IOPS · Prometheus · Grafana
→ **Session 8** (C-D-E)

⑪ 운영·자동화·SRE

IaC · Ansible · Terraform · GitOps
→ **Session 8** (G)

⑫ 거버넌스·컴플라이언스

ISMS-P · KCMVP · CC 인증 · 망분리
→ **Session 7** (E PART)

⑬ 비용·TCO·라이선스

CAPEX vs OPEX · VMware Per-Core
→ 전 세션 결정 포인트

TA의 13 도메인 사용법

13개 도메인 각각의 깊이는 본 강의 흐름(Session 1~9)에서 다룬다. 본 OT에서는 「내가 책임지는 영역의 전체 지도」를 머릿속에 한 번 새기는 것만으로 충분.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

25 / 45

PART 5

TA의 8대 산출물 — 언제 · 무엇을 · 어떻게

TA의 일은 결국 **산출물로 평가**받는다.

TA 8대 산출물 — 카탈로그·미리보기

본 OT에서는 8 산출물의 이름·작성 시점·핵심 항목만 확인. 다음 페이지에서 핵심 2개(구성도·BOM)를 상세히.



① 구성도 (Architecture Diagram)

시점: 제안·설계 초기 · 변경 시
핵심: 논리·물리·네트워크·보안망 분리도



② BOM (Bill of Materials)

시점: 발주 직전
핵심: 모델·수량·라이센스·EOL·유지보수



③ 용량 산정서 (Capacity Plan)

시점: 설계 단계
핵심: TPS·동접·CPU·메모리·IOPS·NW



④ 정책표 (Policy Matrix)

시점: 설계·운영 표준화
핵심: 보안·백업·접근·로그·인증서 정책



⑤ SOP (Standard Operating Procedure)

시점: 운영 표준화
핵심: 백업·복구·변경·장애·침해 4종 표준



⑥ DR 설계서 (Disaster Recovery Plan)

시점: BCP 수립 시
핵심: RTO·RPO·DR 사이트·절체 절차



⑦ 운영 매뉴얼 (Operations Manual)

시점: 인수·운영 이관 시
핵심: 일·주·월 점검 항목·NOC 절차



⑧ 위험 등록부 (Risk Register)

시점: 분가·연간 갱신
핵심: 위험·영향·완화 방안·잔여 위험

TA의 산출물 8종 = "TA가 책임지는 결과물" — 한 권의 책처럼 묶을 수 있어야 인수·재구축이 가능.

산출물 ① — 논리/물리 구성도

무엇인가

- 논리 구성도 — 기능·역할·흐름 중심
- 물리 구성도 — 실제 장비·랙·케이블 중심
- 네트워크 토폴로지 — L2/L3 별도
- 랙 외형도 (Front/Rear)
- 케이블링 도면 — 색·라벨·길이



도구

- Visio · draw.io · Lucidchart
- Mermaid · PlantUML (코드 기반)
- 전문 CAD (AutoCAD·EPLAN — 전기)
- NetBox·DCIM (자산·도면 통합)



작성 시점

- 사전 영업 — 개념 구성도 (1~2장)
- 확정 설계 — 논리 + 물리 (10~20장)
- 시공 — 시공 도면 (CAD 표준)
- 인수 — 최종 As-Built 도면
- 운영 — 변경 시마다 갱신



체크리스트

- 모든 장비에 호스트명·IP·역할 라벨
- 케이블에 양 끝 라벨 (FROM-TO)
- 색상 — 운영 / 관리 / DMZ 구분
- VLAN ID·서브넷·라우팅 표시
- 이중화 경로는 두 줄로

산출물 ② — BOM (Bill of Materials)



무엇인가

- **HW BOM** — 서버·NW·스토리지·랙·UPS
- **SW BOM** — OS·DB·MW·라이선스
- **케이블·부속 BOM** — SFP·HBA·케이블
- **시공 BOM** — 토목·전기·통신 자재
- **단가 + 수량 + 합계 + 운영 단가**



양식 표준

컬럼	예시
분류	서버 / NW / SW
품명	Dell PowerEdge R760
사양	Xeon 6430·256GB SSD 1.92TB×4
수량	4
단가	₩12,000,000
합계	₩48,000,000
비고	무상보증 3년



작성 시점

- **확정 설계** — Draft BOM
- **발주** — 확정 BOM
- **납품·검수** — 검수 BOM (실 시리얼)
- **인수** — 최종 BOM + 시리얼 + 라이선스 키



자주 누락

- **케이블·SFP·HBA 부속** — 본체 견적에 미포함
- **랙 · KVM · 콘솔**
- **유지보수·확장 라이선스**
- **운반비·설치비·시운전**
- **예비품 5%**
- **CCTV·물리 보안**

BOM은 ±10% 정확도가 표준 — 5%를 누락하면 견적 손실 직결.

PART 6

TA가 마주치는 10대 의사결정 포인트

현장에서 가장 자주 나오는 10가지 의사결정 주제.

TA 10대 의사결정 포인트 — 카탈로그·미리보기

본 OT에서는 10개의 **의사결정 주제**만 한 화면에. 다음 3장에서 핵심 3개(온프VS클라우드·가용성·라이센스)만 상세.

① 온프레미스 vs 클라우드

선택 기준: 데이터 민감도·예측 가능성·운영 인력

② 가용성 목표 (SLA)

선택 기준: 업무 중단 비용 vs 이중화 투자

③ 라이선스 모델 (소켓·코어·구독)

선택 기준: TCO 5년 + EOL · 종속성

④ 스토리지 유형 (SAN·NAS·HCI·Object)

선택 기준: IOPS·용량·관리 인력 · 비용

⑤ 보안 인증 (ISMS·P·KCMVP·CC)

선택 기준: 사업 영역(공공·금융·일반) · RFP 의무

⑥ 네트워크 토폴로지 (3-Tier·Spine-Leaf)

선택 기준: East-West 비중 · 확장성 · 비용

⑦ 백업·DR 전략 (3-2-1·DR 사이트 등급)

선택 기준: RTO·RPO · 데이터 크기 · 예산

⑧ 가상화 vs 컨테이너 (VMware·K8s)

선택 기준: 워크로드 특성 · 운영 인력 · 라이선스

⑨ 운영 자체 vs 위탁 (Co-Lo·MSP·MSSP)

선택 기준: 핵심 역량 · 24/7 부담 · 보안 책임

⑩ 신기술 도입 시점 (Early·Mainstream·Late)

선택 기준: 리스크 허용도 · 인력 학습 · 표준화 단계

TA의 의사결정 = 10개 축의 다차원 매트릭스. 한 결정이 다른 결정에 영향을 줌 (예: 클라우드 선택 → 라이선스·운영 모델 모두 바뀜).

결정 ① — 온프레미스 vs 클라우드 vs 하이브리드



상황

- 신규 시스템 구축
- 노후 시스템 교체 (5년 차)
- 사업 확장·글로벌 진출
- 비용 절감 요구



결정 축

축	온프레미스 유리	클라우드 유리
법규·통제	공공·금융·의료	일반 B2C
부하 변동	안정·예측 가능	변동·계절성
기간	5년+ 장기	1~3년 단기·시범
인력	자체 운영 가능	인력 부족
신기술	표준 기술	AI·신기술 빠른 채택



TA의 답변 패턴

1. **5년 TCO 비교표 제시** — 양쪽 다 계산
2. **법규·인증 요건 확인** — KCMVP·CSAP 등
3. **하이브리드 우선 검토** — 단일화는 위험
4. **마이그레이션 비용 + 리스크 별도 산정**
5. **벤더 종속 위험 정량화**

2026 현실: 신규 = 하이브리드 우선, 노후 교체 = 기존 자산 활용으로 온프레미스 갱신이 다수.

결정 ② — 가용성 목표 (SLA)

SLA 등급별 다운타임

가용성	연간 다운	월간 다운	패턴
99.0%	87.6h	7.3h	단순 단일
99.9%	8.8h	43.8분	Active-Standby
99.99%	52.6분	4.4분	Active-Active
99.999%	5.3분	26.3초	Multi-Site AA
99.9999%	31.5초	2.6초	사실상 무중단

비용 곱

- 99.9% → 1×
- 99.99% → **3~5×**
- 99.999% → **10~20×**

TA의 답변 패턴

- 시스템별 영향도 매트릭스
- 무조건 **99.999%** 거부 — 비용 폭증
- 합리적 목표**: - 핵심 결제·실시간 = 99.99% - 업무 시스템 = 99.9% - 통계·BI = 99.5%
- 계획 다운 (계획 점검) 별도 정의
- SLA 위반 시 보상 조항**

흔한 오해

- "장애 = 다운타임" — 부분 장애 산정 별도
- "측정 기준 = 핑" — 응답 시간·기능별 측정

결정 ③ — 라이선스 모델

주요 라이선스 모델

- Oracle DB**
 - EE Per-Core (가장 비싸고 흔함)
 - SE2 Per-Socket
 - NUP (Named User Plus)
 - ULA (무제한, 3년 후 정산)
- VMware (2024+)**
 - Per-Core (기존 Per-Socket 폐지)
 - vSphere Foundation / Cloud Foundation 묶음
- MSSQL**
 - Per-Core EE / Standard
 - CAL (사용자 기준)
- RHEL** — 물리 소켓 + 가상 게스트 수

TA의 답변 패턴

- 모든 라이선스 모델을 표로 정리
- 3~5년 TCO 비교**
- 사용자 수 vs 코어 수 산정 정확도 ↑
- 라이선스 감사 대비 — 자체 인벤토리 매년
- 오픈소스 대안 검토 — PostgreSQL · MariaDB
- Bring-Your-Own vs 클라우드 매니지드 위험**
 - Oracle Java SE 라이선스 (2023+ 변경)
 - VMware Per-Core 전환 (2024+)
 - MSSQL 가상화 시 모든 호스트 Core 산정 필요

PART 7

학습 평가 · 실습 · 과제 안내

어떻게 평가하고 무엇을 제출하는가.

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

학습 평가 — 3축 평가 기준

① 개념 이해 (30%)

- 매 세션 종료 시 5문항 퀴즈
- 객관식 + 짧은 서술
- 출제 범위 = 강의 슬라이드
- 합격선 70%
- 누적 평균으로 평가

② 설계 산출물 (40%)

- 설계 워크숍 산출물
- 논리·물리 구성도
- BOM + 용량 산정서
- 정책표 + SOP
- 팀별 제출·발표
- 평가 = 정확성·완결성·실현 가능성

③ 케이스 분석 (30%)

- 케이스 1개 선정 → Deep-dive 보고서 (10~20p)
- 개인 제출 (워드 또는 PDF)
- 평가 = 실전성·근거·설계 깊이

합격선: 3개 평균 **70점 이상** + 출석 80% 이상

사전 준비

 도구·환경

- 노트북 (회사 또는 개인)
- 다이어그램 도구
- draw.io (무료)
- Lucidchart / Visio (유료)
- 표·산정 도구
- Excel / Google Sheets
- 메모 도구
- Confluence / Notion / Obsidian
- 인터넷 접속 (참고 자료 검색)

 사전 학습 (선택)

- OSI 7 Layer 기본
- VM·컨테이너 기본 개념
- 기본 리눅스 명령어 (top·df·free·systemctl)

 마인드셋

- 암기 X, 큰 그림과 결정 패턴 위주
- 질문 환영 — 어떤 질문도 가치 있음
- 메모 우선 — 본인의 언어로 기록
- 실전 사례 공유 — 본인 운영 경험을 강의에 녹임
- 팀 협업 — 워크숍 + 상호 리뷰

 강사 권장

3일 동안 **A4 한 장으로 본인만의 TA 도메인 지도**를 그려본다. 종강일에 동료에게 5분 간 그 지도를 설명해본다.

PART 8

TA 관련 자격증 + 실습 환경

자격증은 학습 로드맵·실습 환경은 강의 첫날 셋업.

TA 자격증 — 국내 (한국)

🏆 기술사 (최상위)

- 정보관리기술사 (한국산업인력공단)
- 컴퓨터시스템응용기술사
- 정보통신기술사
- 경력 9년 이상·서술형 + 면접
- TA·아키텍트의 최종 목표

📖 기사·산업기사

- 정보처리기사
- 정보보안기사
- 정보처리산업기사
- 경력 4년 또는 학사 졸업

🌐 네트워크·시스템

- 네트워크관리사 1·2급 (한국정보통신자격협회)
- 리눅스마스터 1·2급
- SQLD / SQLP (한국데이터산업진흥원)
- DAP / DAsP — 데이터아키텍처 전문가/준전문가

🔒 보안·감사

- CPPG — 개인정보관리사
- 정보시스템감리원 (한국정보통신산업진흥원)
- CISA-K 등 국내 보안 인증

🔧 벤더 국내 인증

- OCP-Korea (Oracle Korea)
- 삼성/LG CNS 자체 인증
- SK C&C TA 트랙 인증

📊 권장 학습 로드맵 (신입 → 시니어)

- 0~3년: 정보처리기사 + 리눅스마스터 2급 + 네트워크관리사 2급
- 3~6년: 정보보안기사 + SQLD + 네트워크관리사 1급
- 6~10년: 정보시스템감리원 + DAP
- 10년+: 정보관리기술사 또는 정보통신기술사

🏢 자격증 발급기관 공식 - 한국산업인력공단 — Q-Net — 기술사·기사 - 한국정보통신자격협회 — ICQA — 네트워크관리사·리눅스마스터 - 한국데이터산업진흥원 — K-DATA — SQLD·SQLP·DAP - 한국인터넷진흥원 — KISA — 정보보안기사·CPPG - 한국정보통신산업진흥원 — NIPA — 정보시스템감리원



TA 자격증 — 국내 (한국)

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

TA 자격증 — 국제 (Global)



아키텍처·관리

- **TOGAF 9/10** — The Open Group
- **PMP** — PMI
- **PRINCE2** — Axelos
- **ITIL 4 Foundation** — Axelos



가상화·HCI

- **VCP-DCV / VCAP-DCV** — VMware
- **NCP-MCI / NCP-DS** — Nutanix
- **Citrix CC-VAD** — Citrix



보안

- **CISSP** — ISC²
- **CISA / CISM** — ISACA
- **CEH** — EC-Council
- **GIAC GSEC / GCIH** — SANS
- **OSCP** — Offensive Security



네트워크

- **CCNA · CCNP · CCIE** — Cisco
- **JNCIA · JNCIP** — Juniper
- **HCIA · HCIP** — Huawei
- **PCNSE** — Palo Alto Networks
- **컨테이너·SRE**
- **CKA · CKAD · CKS** — CNCF (K8s)
- **Docker Certified** — Mirantis
- **HashiCorp** — Terraform-Vault-Consul
- **Red Hat OpenShift** — DO180/DO280



OS·시스템·DB

- **RHCSA · RHCE · RHCA** — Red Hat
- **LFCS · LFCE** — Linux Foundation
- **OCP · OCM** — Oracle (DB)
- **MCSA / Microsoft Certified** — Microsoft
- **클라우드 (참고)**
- **AWS SAA · SAP · DOP** — Amazon
- **Azure AZ-104 · AZ-305** — Microsoft
- **GCP Associate · Professional** — Google

TA 권장 조합 (1순위): RHCE + CCNA/CCNP + VCP + CISSP → 온프레미스 TA 폴스택 인증

국제 자격증 공식 - [Red Hat Certification](#) — RHCSA·RHCE·RHCA - [Cisco Learning Network](#) — CCNA·CCNP·CCIE - [ISC² CISSP](#) — CISSP - [ISACA CISA](#) — CISA·CISM - [The Open Group TOGAF](#) — TOGAF 9/10 - [PMI PMP](#) — PMP - [CNCF Kubernetes Certifications](#) — CKA·CKAD·CKS - [VMware Certification \(Broadcom\)](#) — VCP·VCAP - [Linux Foundation Training](#) — LFCS·LFCE

IT 인프라 아키텍처 설계 · Session 0 · OT · TA의 일과 IT 인프라 큰 그림

실습 환경 — 가상화 도구 비교·선택



VirtualBox 7.x

Oracle · 개인 무료 · 본 강의 표준

- 호스트: Win·macOS·Linux
- 가벼움·스냅샷·내장 NAT/Host-only
- 성능은 VMware 대비 약간 낮음



VMware Workstation Pro 17

Broadcom · 2024.5~ 개인 무료

- 호스트: Win·Linux (Mac은 Fusion)
- 성능·3D 가속·vSphere 호환
- Broadcom 인수 후 정책 변경 잦음



Hyper-V

Microsoft · Win Pro/Ent 내장

- 호스트: Windows 10/11 Pro 이상
- OS 통합·PowerShell 자동화
- VirtualBox와 동시 사용 불가



호스트 PC 권장 — VT-x / AMD-V 활성화 필수

- **최소**: 4코어 · 16GB RAM · 100GB SSD
- **권장**: 8코어+ · 32GB+ RAM · 256GB+ NVMe
- **OS**: Windows 10/11 · macOS 13+ · Linux 최신 LTS

본 강의 표준은 **VirtualBox 7.x** — VMware Workstation 17(개인 무료) 사용자는 그대로 따라와도 무방.

실습 환경 — VM 4대 + 가상 네트워크 5종

 강의 표준 VM 4대

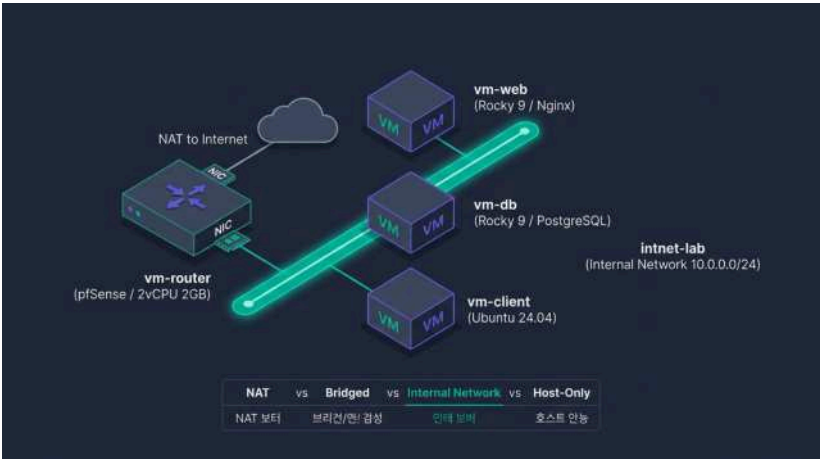
VM	게스트 OS	역할	vCPU-RAM-디스크
vm-router	pfSense CE	라우터·방화벽	2-2GB-20GB
vm-web	Ubuntu Server 24.04 LTS	Web/WAS (Nginx-Tomcat)	2-2GB-30GB
vm-db	Ubuntu Server 24.04 LTS	DB (PostgreSQL-MySQL)	2-4GB-30GB
vm-client	Ubuntu Desktop 24.04	클라이언트·테스트	2-2GB-20GB

- 호스트 4 VM 동시 가동: 약 10~12 vCPU + 10GB RAM + 100GB 디스크
- 다운로드:
 - Ubuntu Server / Desktop 24.04: ubuntu.com/download
 - pfSense CE: pfsense.org/download

 VirtualBox 가상 네트워크 5종

모드	외부 통신	호스트와 통신	VM 간 통신	실습 용도
NAT	✓ (호스트 경유)	✗		인터넷 접근만
NAT Network	✓		✓	외부 + VM끼리
Bridged	(DHCP 직접)			호스트 LAN 직접
Internal Network	✗		✓	VM끼리만 (실험실)
Host-Only		✓	✓	관리·격리 실험

- 본 강의 기본: `vm-router` = NAT + Internal Network 2개, 나머지 VM = Internal Network만
- 실습용 LAN: `intnet-lab` (내부망), `intnet-dmz` (DMZ)



실습 환경 — VM 4대 + 가상 네트워크 5종

IT 인프라 아키텍처 설계

Session 0 · 끝 TA의 큰 그림

5대 아키텍트 · 8계층 · 13 도메인 · 8 산출물 · 10 의사결정 포인트

OT 종료 — TA의 큰 그림 정리

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계

데이터센터 전력 · 냉각 · 물리 인프라 · BCP

Session 1 · Day 1 — TA의 데이터센터 전권

Tier 표준 · 전력 토폴로지 · UPS · 디젤발전기 · CRAC · Liquid Cooling · 랙 · 케이블링 · 물리 보안 · 화재진압 · BCP

강사 박수현 ·  젠아이랩스(GenAI Labs)

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

데이터센터가 멈추면 무엇이 멈추는가 — 국내 3대 사례

🔥 ① SK C&C 판교 DC 화재

2022.10.15 (토)

배터리실 화재 → 카카오톡·다음·카카오페이 약 127시간 장애 (역대 최장 메신저 장애).

- 화재 원인: 리튬이온 UPS 배터리
- 영향: 카카오 자회사 다수
- 사회 파장: 「카카오 먹통 보상」 국정감사·법안

💡 **교훈** — DR 사이트 분라·BCP 시험·UPS 배터리실 별도 구획

참고: 「2022 판교 데이터센터 화재」 검색

② KT 아현 통신구 화재

2018.11.24 (토)

서울 서대문구 아현지사 통신구 화재 → 강북·은평·마포·중구 KT 유·무선 마비 (최대 1주일).

- 119·112 전화·결제·POS·ATM 중단
 - 카드 결제 불가 → 자영업자 피해
 - 통신 3사 의무 「우회 회선」 강화
- 교훈 — 통신구 이중화·우회 경로·BCP 통신망 다중화

참고: 「KT 아현화재」 검색

🔥 ③ 국가정보자원관리원 대전 화재

2025.09.26 (금)

대전 본원 전산실 화재 → 정부 시스템 709개 마비 (당초 647개 → 정정). 모바일 신분증·국민신문고·우체국 금융 등.

- 배터리 이설 작업 중 일부 전원 미차단
- 노후 인프라 (2005년 건립·구 KT 건물)
- 배터리·서버 미분리 공간
- 4주+ 복구 (대구 센터로 96개 이전)

💡 **교훈** — 노후 DC 점검·배터리실 분리·DR 사이트 사전 준비

참고: 위키백과 「2025년 국가정보자원관리원 화재」

📌 이번 세션의 시각 — 위 3건은 모두 「Tier 등급 미달」「BCP·DR 시험 누락」「우회 경로 부재」가 공통 원인. TA가 결정해야 할 5축(전력·냉각·상면·보안·BCP)을 한 번에 다룬다.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

2 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



Session 1 · Day 1 — TA의 데이터센터 전권

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

3 / 56

Session 1 — 회차 메타데이터

📋 세션 정보

- **회차:** 1 / 15 — 데이터센터 전권
- **소속:** Day 1 — IT 인프라 큰 그림 + 서버·NW 기초
- **카테고리:** 인프라 도메인 ① (데이터센터·전력·냉각·물리)
- **예상 분량:** 본문 56 슬라이드 + 이미지 12장
- **강의 시간:** 약 90~120분
- **강사:** 박수현 — 젠아이랩스(GenAI Labs) **CEO / CTO**

🎯 핵심 메시지

데이터센터는 건물이다 — **전력·냉각·상면·보안·BCP** 5축이 무너지면 어떤 서
버도 살릴 수 없다.

🎯 학습 목표 (Learning Objectives)

1. **DC 분류·Tier** — Enterprise·Colo·Edge·Hyperscale + Uptime Tier
I~IV의 차이와 가용성 수치를 설명한다
2. **전력 토폴로지** — 한전 2회선·ATS·UPS·디젤발전기·PDU·A/B 이중화의 전체
경로를 그릴 수 있다
3. **냉각 설계** — CRAC/CRAH·In-Row·Containment·Free Cooling·Liquid
Cooling의 장단점을 비교한다
4. **상면·랙·케이블링** — 면적·하중·랙·ToR vs EoR·구조화 배선을 산정·표준화할
수 있다
5. **물리 보안·화재** — 출입통제·CCTV·맨트랩·FM-200·VESDA의 결정 포인트를
안다
6. **현장 환경·BCP** — 풍하중·낙뢰·접지·KS/IEC 표준·환경시험성적서·정전 BCP
를 점검할 수 있다

본 세션의 지도 — 8개 PART · 56 슬라이드

🗺️ Part A~D (전반부 · 약 60분)

- **A. 데이터센터 큰 그림** — 분류·Tier·5대 영역
- **B. 전력 인프라** — 한전·UPS·발전기·ATS·PDU·이중화
- **C. 냉각 인프라** — CRAC·In-Row·Containment·Free·Liquid
- **D. 상면·랙·케이블링** — 면적·하중·42U·ToR/EoR·배선

🗺️ Part E~H (후반부 · 약 70분)

- **E. 물리 보안** — 출입통제·CCTV·맨트랩·무인경비
- **F. 화재 진압·VESDA** — FM-200·Inergen·Novec-1230·CO2
- **G. 현장 환경·표준** — 풍하중·낙뢰·접지·KS/IEC·KOLAS
- **H. 운영·BCP·산출물** — DCIM·PUE·체크리스트·TA 결정

PART A

데이터센터 큰 그림 — 분류 · Tier · 5대 영역

"데이터센터는 서버를 두는 방이 아니라 전력·냉각·보안의 통합 시설이다."

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

데이터센터란 — 일반 서버실과 무엇이 다른가 (5대 영역)

① 전력 인프라

- 일반 서버실 — 상용 전원 1계통. 정전 시 서비스 중단.
- 데이터센터 — 이중 전력 인입(Dual Feed) + UPS + 비상 발전기 + ATS(자동 절체 장치). 정전이 발생해도 서비스가 계속 유지되도록 설계.

② 냉각 인프라

- 일반 서버실 — 일반 에어컨.
- 데이터센터 — CRAC·CRAH·수냉식·외기 냉방(Free Cooling). 수백~수천 대 서버의 열을 지속적으로 제거.

③ 네트워크 인프라

- 일반 서버실 — 인터넷 회선 1~2개.
- 데이터센터 — 다수 통신사 회선 + 백본 연결 + MMR(Meet-Me Room). 회선 장애가 발생해도 통신 유지.

④ 물리 보안

- 일반 서버실 — 출입문 잠금 수준.
- 데이터센터 — 출입카드 + 생체인증 + CCTV + 24시간 관제. 일반 사무실보다 훨씬 강한 보안 수준.

⑤ 재해 대응

- 일반 서버실 — 소화기 수준.
- 데이터센터 — 가스 소화설비 + 누수 감지 + 지진 대응 + 재난 복구 체계. 시설 자체가 재난 상황을 고려하여 설계.

핵심 — 5축 (Power · Cooling · Space · Network · Security/BCP) 이 중 하나라도 결함이 있으면 데이터센터의 가치를 상실한다.

데이터센터의 정량 지표 — PUE · 바닥 하중 · 천장 높이



PUE (Power Usage Effectiveness)

- 공식 — $PUE = \frac{\text{전체 시설 전력}}{\text{IT 장비 전력}}$
- 이상 1.0, 하이퍼스케일 1.1~1.3, 신규 IDC 1.2~1.5, 기존 IDC 1.5~2.0
- 목표 표현 정정 — "1.4 이하 목표" 보다는 「최근 데이터센터는 PUE 1.2~1.5 수준을 목표」가 정확.
- 이후 PART C(냉각)에서 상세 다룸.



바닥 하중 (Floor Load)

- 일반 사무실 300 kg/m² → IDC 부적합
- 데이터센터 750~1,500 kg/m²
- 폴랙 1대 무게 500~1,500 kg
- 특히 GPU 서버 · 스토리지 · HCI 노드는 매우 무거움 → 신규 도입 시 반드시 사전 산정



천장 높이 — 3.5~4.5m

- 단순히 "Hot Aisle 환기" 때문이 아니라 여러 설비가 천장 위 공간을 공유하기 때문.
- 필요한 공간:
- 케이블 트레이 (강전·약전 분리)
- 공조 설비 (덕트·환기구)
- 소방 설비 (스프링클러·VESDA·가스 노즐)
- Hot/Cold Aisle Containment 격벽
- 유지보수 작업 공간



데이터센터의 5대 영역 (요약)

- 전력 (Power) — 한전 수전·변압기·UPS·배터리·발전기·PDU
- 냉각 (Cooling) — CRAC·CRAH·항온항습·수냉·외기 냉방
- 상면 (Space) — 랙·케이블링·면적·바닥 하중
- 네트워크 (Network) — 통신사 회선·MMR·코어 스위치·백본
- 보안 + BCP — 출입 통제·CCTV·화재 대응·DR·BCP

데이터센터 4대 분류 — 규모·운영주체별



Enterprise DC (자가)

- 운영주체: 기업 직접 운영
- 규모: 100~1,000랙
- 사례: 통신사·금융 본사·정부 통합전산센터
- TCO: 초기 CapEx 매우 큼, 장기 운영비 유리
- 통제력: 최고



Colocation (상면 임대)

- 운영주체: IDC 사업자 (KT·LGU+·SK)
- 규모: 1/4랙 ~ 폴랙 단위 임대
- 사례: 대다수 중견 기업 IT 인프라
- TCO: 월 임대료 + kWh 전기료
- 통제력: 중간 (서버까지만 입주사 권한)



Edge DC

- 운영주체: 통신사·CDN·5G MEC
- 규모: 1~20랙, 분산 배치
- 사례: 자율주행·실시간 영상·IoT 게이트웨이
- 목적: 저지연 (10ms 이내)
- 위치: 도시 거점·통신사 국사



Hyperscale DC

- 운영주체: AWS·Azure·GCP·NCP
- 규모: 10,000~100,000랙
- 사례: 클라우드 리전·AI 학습 클러스터
- 특이성: PUE 1.1대·OCP·수냉 도입 선도
- 건설: 부지·전력 계약 단위 GW

Tier 표준 — Uptime Institute Tier I~IV



Uptime Institute 표준

- 세계 표준 데이터센터 등급 인증
- TIA-942와 유사하지만 별도 체계
- 4단계: Tier I → II → III → IV
- 각 등급은 누적 요건 (상위는 하위 요건 모두 포함)
- 인증: Tier Certification of Design Documents (TCDD) + Constructed Facility (TCCF)



평가 축

1. 이중화 (N+1~2N)
2. 동시 유지보수 가능성 (Concurrently Maintainable)
3. 장애 허용성 (Fault Tolerance)
4. 인증 다운타임 (Annual Downtime)



한국 현황

- 상용 IDC 대부분 = Tier III 또는 동등 수준
- Tier IV = 금융·국방·통신 코어 일부
- 공공 통합전산센터 = Tier III 수준 (대전 본원·광주)
- 신축 IDC는 Tier III Constructed 인증 추진 추세



TA의 확인 항목

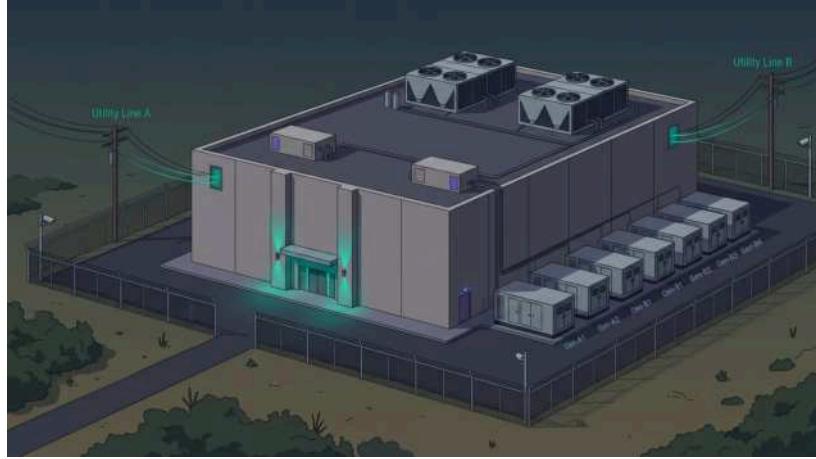
- 공식 인증서 vs "Tier III 동등"이라는 마케팅 문구
- 전원·냉각·NW 진입 경로 수
- 동시 유지보수 시 실제 가용 잔여 용량

Tier I·II·III·IV — 완전 매트릭스

등급	가용성	연간 다운	이중화	핵심 특성	일반 용도
Tier I	99.671%	약 28.8시간	단일 경로 (N)	점검 시 전체 중단	소형 사무실·실험실
Tier II	99.741%	약 22시간	부분 이중화 (N+1)	점검 시 일부 중단	중소 사이트·연구실
Tier III	99.982%	약 1.6시간	N+1 (모든 컴포넌트)	동시 유지보수 가능	대부분 기업 IDC (표준)
Tier IV	99.995%	약 26.3분	2N 또는 2(N+1)	완전 장애 허용 (FT)	금융·국방·통신 코어

TA의 의사결정 가이드: 기업·공공 표준은 Tier III. Tier IV는 비용이 1.5~2배이며, 99.995%가 정말 필요한 워크로드(증권 매매·결제망·국방 C4I)에만 정당화.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



Tier IV급 데이터센터 외관 (삽화)

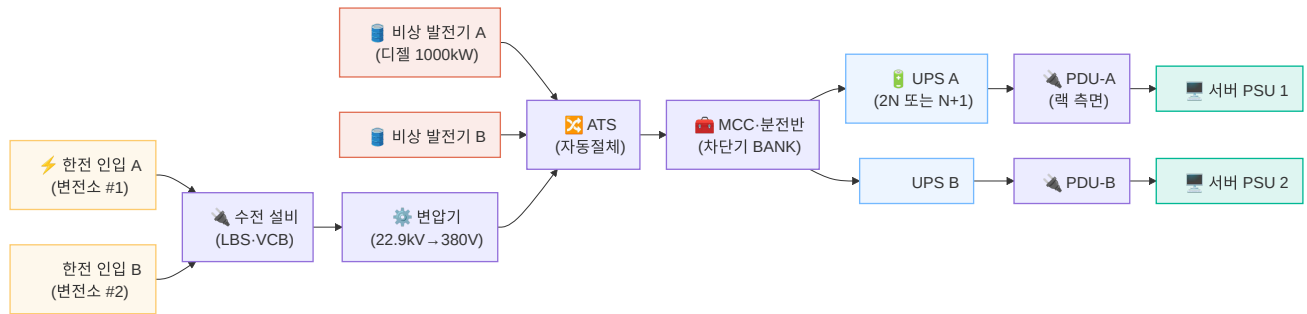
PART B

전력 인프라 — 무중단이 설계의 출발점

전력은 단일 장애점이 가장 많고, 장애 시 모든 IT 인프라가 즉시 정지하는 영역.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

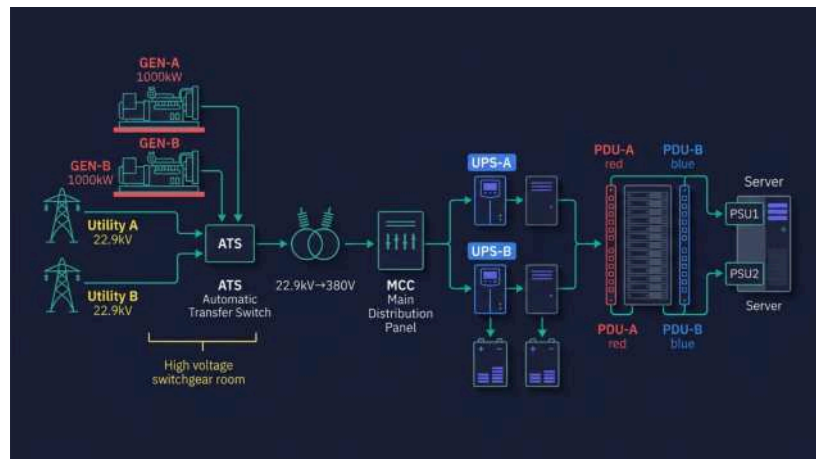
전력 전체 경로 — 한전 → 서버 PSU



© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

14 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



전력 토폴로지 (삼화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

15 / 56

UPS (Uninterruptible Power Supply) — 분 단위 전원 보전



정의·역할

- 정전 시 즉시 배터리로 전환 (0~10ms)
- 가동 지속: **5~30분** (통상 15분)
- 역할: **발전기 기동까지의 공백 보전**
- 또한 **전원 품질 안정화**:
- 전압 sag/swell 보정
- 고조파 (Harmonic) 필터링
- 노이즈·서지 흡수
- IT 부하의 **약 1.3배** 용량으로 산정 (효율 마진)



UPS 구성

- 정류기 (Rectifier) — AC → DC 변환
- 인버터 (Inverter) — DC → AC 재변환
- 배터리 — 정전 시 DC 공급
- 바이패스 (Bypass) — UPS 고장 시 상용 직결
- 모니터링 — SNMP·Modbus·BMS 연동

용량 단위: **kVA**(피상전력)·**kW**(유효전력). 역률 0.8~1.0 적용.

UPS 3종 — Online · Line-Interactive · Standby

종류	절체 시간	보호 수준	효율	가격	용도
Standby (Offline)	4~10ms	정전만	95~98%	저가	소형 PC·홈 NAS
Line-Interactive	2~4ms	정전 + 전압 sag/swell	90~95%	중가	소형 서버실·SOHO
Online (Double Conversion)	0ms	모든 전원 이상	88~94%	고가	DC·금융·의료 표준



Online UPS 동작

- 평시에도 정류 → 인버터 항상 동작
- 부하는 **항상 인버터 출력** 사용
- 정전 시 절체 없이 배터리 → 인버터
- **완전한 절연** — 모든 입력 이상 차단
- **에코 모드** 가능 — 평시 바이패스로 효율 ↑



DC급 표준

- **Online + N+1 또는 2N 이중화**
- 용량 100~500 kVA 모듈 다수 병렬
- 모듈러 UPS (Hot-swap) 추세
- 대표 벤더: Schneider (APC) · Vertiv (Liebert) · ABB · Eaton · Riello · Fuji
- 국산: LG·동아일렉콤·삼화EOC

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

디젤 발전기 — 시간 단위 장기 전원

디젤 발전기 (Diesel Generator)

- **연료:** 경유 (Diesel) — 가장 흔함
- **기동 시간:** 10~30초 (Cold Start)
- **운전 지속:** 연료 탱크에 따라 **24~72시간**
- 옥상·지하·옥외 별도 설치
- 1MW 디젤 발전기 = 시간당 약 **200~250 리터** 소모
- 연료 탱크: 옥외 매설 또는 옥상 day tank

환경 영향

- **배기·매연** — 도심 IDC에서 민감
- **소음** — 80~95 dB → 방음 격실 (Acoustic Enclosure)
- **진동** — 방진 패드·격실

대체·보조

- **가스 터빈 발전기** — 도심 IDC (LNG 연료, 소음 낮음)
- **연료전지 (Bloom Energy)** — Hyperscale 도입 사례
- **이중연료 (Bi-fuel)** — 경유 + LNG

TA의 운영 점검

- **월 1회 무부하 시험 (Auto-Start)**
- **분기 1회 부하 시험 (실제 전환)**
- **연료 잔량 점검** — 항상 70% 이상
- **윤활유·냉각수·배터리** 정기 교체
- **연료 장기 보관 (1년+) 부적합** — 산화·미생물

TA의 실수 사례: 정전 발생 → 발전기 자동 기동 실패 → 원인은 배터리 방전. 발전기 시동 배터리도 UPS 못지않게 중요.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

18 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

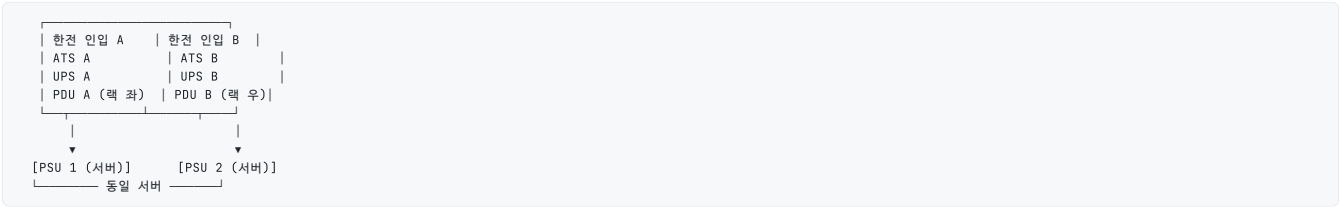


UPS·발전기 룸 (삽화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

19 / 56

A/B 이중 전원 — 모든 단일 장애점을 두 계통으로



✓ 올바른 A/B 구성

- 서버는 **2개의 PSU** 보유 (1+1 Redundant)
- PSU 1 → PDU A → UPS A → ATS A → 한전 A
- PSU 2 → PDU B → UPS B → ATS B → 한전 B
- **어느 한 쪽 계통이 완전 다운**해도 무중단
- 네트워크 장비도 동일 원칙 (이중 전원)

✗ 흔한 실수

- 서버 PSU 2개를 **동일 PDU**에 연결 → A/B 의미 상실
- 한전 A·B를 동일 변전소에서 인입 → 변전소 장애 시 동시 정전
- UPS 1대로 A/B 모두 공급 → UPS 장애 시 동시 정전
- 발전기 1대만 → 발전기 시동 실패 시 전면 정전
- **단일 전원** 서버 (1 PSU) 사용 시 외부 ATS·STS 필요

전력 산정 — 현장 공식과 단위

📐 기본 공식

- $W = V \times A \times \text{역률 (PF)}$
- $kVA \times PF = kW$ (역률 0.8~0.95)
- $kWh = kW \times \text{시간 (전기로 단위)}$
- $BTU/h = W \times 3.412$ (열량 변환)
- **단상 220V 30A** = 6.6 kVA (한국 일반)
- **3상 380V 30A** = 19.7 kVA
- 💡 **한국 전기 요금 (산업용 갑1·2025년 기준)**
 - 계약 전력 기본료 + 사용량 요금
 - 고압 A 선택 II: kWh 약 130~180원
 - 심야 vs 주간 요금 차이 큼 (3~5배)

🏢 산정 사례 (1랙)

1U 서버 400W × 20대	= 8,000 W = 8 kW
스위치 200W × 2대	= 400 W
2U 서버 600W × 5대	= 3,000 W
KVM 50W × 1대	= 50 W
랙 소비 전력	= 11.45 kW
전류 (3상 380V, PF 0.9):	
$I = 11,450 / (380 \times \sqrt{3} \times 0.9)$	
= 19.3 A	
PDU 정격 32A 선정 (80% 룰)	
연 전기료 = 11.45 × 24 × 365 × 150원	
= 1,500만원/년	

현장 감각: 11~12 kW 랙 1구 연 전기료 약 1,500만원. IDC 임대료(랙 200만/월)와 거의 동일 규모.

PART C

냉각 인프라 — 전력 = 열, 1:1 변환

서버가 소비한 1kW는 모두 열로 배출. 데이터센터는 본질적으로 열관리 시설.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

냉각의 본질 — 1W 전력 = 3.412 BTU/h 발열



열역학 1법칙

- 에너지 보존: 입력 전력 = 출력 (열)
- $1W = 3.412 \text{ BTU/h} = 0.86 \text{ kcal/h}$
- 11 kW 랙 → **37,500 BTU/h** 냉각 필요
- 1톤(RT) = 12,000 BTU/h $\approx$ 3.5 kW



항온항습 기준 (ASHRAE TC 9.9)

- 권장 (Recommended): 18~27°C / 40~60% RH
- 허용 (Allowable A1): 15~32°C / 20~80% RH
- 결로점 (Dew Point) 9~15°C
- IDC 일반: **22~24°C / 45~55% RH**



냉각 사이클 (Vapor Compression)

1. **Evaporator (증발기)** — 실내 열 흡수
2. **Compressor (압축기)** — 압축·온도 ↑
3. **Condenser (응축기)** — 옥외 방열
4. **Expansion Valve** — 팽창·온도 ↓
5. → 다시 Evaporator



냉매

- R-410A·R-134a·R-407C (HFC)
- R-1234ze-R-32** (저GWP, 2024+ 추세)

CRAC vs CRAH — 전산실용 항온항습기 2종

❄️ CRAC (Computer Room AC)

- **Direct Expansion (DX)** — 자체 압축기 보유
- 냉매 사이클을 CRAC 단독으로 완결
- 옥외 응축기 (Condenser) 짝
- 설치 간편 (소규모·중규모 IDC)
- 효율 중간 (PUE 1.4~1.8)
- 단가 저렴·고장 1대 영향 국한
- 대표: Liebert (Vertiv)·Stulz·Schneider Uniflair

💧 CRAH (Computer Room Air Handler)

- **냉수 코일 (Chilled Water Coil)** — 중앙 칠러에서 냉수 공급
- 옥상·옥외 **칠러 + 펌프 + 배관**과 함께 운영
- 대형 IDC 표준 (수십~수백 랙)
- 효율 우수 (PUE 1.2~1.4)
- 칠러 1대로 다수 CRAH 운영
- 단가 비쌈·설치 복잡 (배관)
- 대표: Trane·Carrier·York·LG·삼성

TA의 선택 기준: 50랙 미만 = CRAC, 50랙 이상·대형 IDC = CRAH + 중앙 칠러.

Hot/Cold Aisle Containment — 공기 분리



기본 원리

- **Cold Aisle:** 랙 정면, 차가운 공기 흡입 (22°C)
- **Hot Aisle:** 랙 뒷면, 뜨거운 공기 배출 (35~40°C)
- **Containment:** 양쪽 공기를 물리적으로 분리
- 천장 격벽·문·바닥 격리로 **혼합 방지**
- 효율 20~40% 향상 (PUE ↓)



2가지 방식

- **Cold Aisle Containment** — 차가운 공기 영역을 막음
- **Hot Aisle Containment** — 뜨거운 공기 영역을 막음 (더 효율적)



Containment 없는 경우

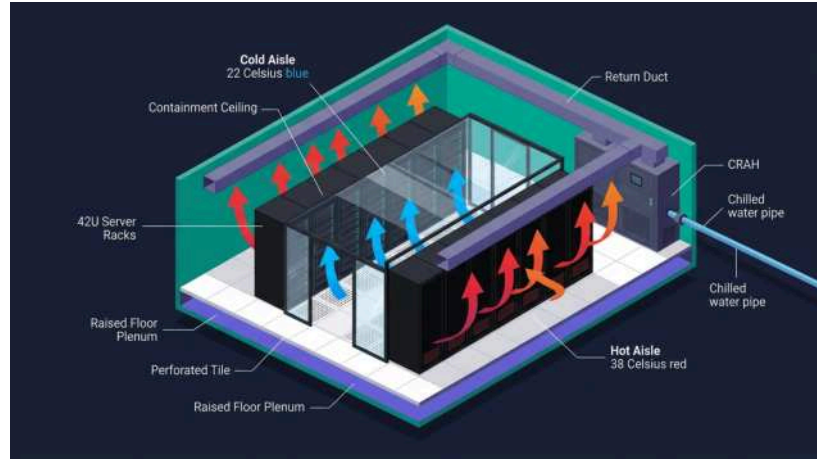
- 차가운 공기가 천장으로 새어 Hot Aisle 진입
- CRAC가 차가운 공기와 뜨거운 공기 혼합 흡입
- 냉각 효율 저하 → 전력 낭비
- 일부 랙에서 **Hot Spot** 발생



발열 등급별 권장

- **저밀도 (3~5 kW/랙):** CRAC만으로 충분
- **중밀도 (5~15 kW):** Containment 권장
- **고밀도 (15~30 kW):** In-Row + Containment
- **초고밀도 (30 kW+):** Liquid (D2C-Immersion)

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



Cold/Hot Aisle Containment 단면도 (삽화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

26 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

Liquid Cooling — AI·HPC 시대의 표준

💧 Direct-to-Chip (D2C)

- CPU-GPU 다이 위에 쿨드 플레이트 부착
- 냉각수 (보통 PG 25%-DI Water)가 순환
- 80~90% 열을 액체로 제거
- 나머지 공기 냉각
- 랙당 **50~150 kW** 처리
- **CDU (Coolant Distribution Unit)** 필요
- 대표: NVIDIA HGX H100/B200 표준

🚰 Immersion (침지)

- 서버 전체를 유전체 액체에 침지
- Single-phase (광물유)·Two-phase (3M Novec)
- 80% 이상 전력 절감 (팬 제거)
- 랙당 **100~200 kW**
- 인프라 (탱크·펌프)·작업성 (장갑) 부담

📞 TA의 결정 포인트

- **AI 학습 클러스터** = Liquid Cooling 필수
- **추론 (Inference)** = 공랭 + Rear Door 충분
- **신규 IDC**: D2C-Ready 배관 사전 설계
- **기존 IDC**: D2C 후순위 (배관·CDU 추가)
- 냉각수 온도 30~40°C (Warm Water Cooling) → 외기 자연 냉각 가능

💰 단가

- D2C 시스템: 랙당 1~3억 (기존 대비 2~3배)
- 운영비: 전력 절감으로 회수

2026년 추세: NVIDIA Blackwell (B200) 800W-1000W TDP → 공랭 불가, D2C 필수.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

27 / 56

PUE (Power Usage Effectiveness) — 데이터센터 효율 지표



PUE 정의

$$\text{PUE} = \frac{\text{데이터센터 전체 전력}}{\text{IT 장비 전력}}$$

PUE 1.0 = 모든 전력이 IT에만 (이상)
PUE 2.0 = IT가 절반, 나머지는 냉각·조명



등급

- 1.0~1.2: 최우수 (Hyperscale)
- 1.2~1.5: 우수 (신규 IDC 평균)
- 1.5~2.0: 보통 (기존 IDC)
- 2.0+: 개선 필요



글로벌 평균

- Google·Meta: PUE 1.10~1.12
- AWS·Azure: PUE 1.15~1.25
- Equinix 평균: 1.48
- 한국 IDC 평균: 1.5~1.7
- 네이버 각: 1.09 (한국 최저 수준)
- 2030 EU 규제: 신축 PUE 1.3 이하



PUE 개선 수단

- Free Cooling
- Containment
- Liquid Cooling
- 고효율 UPS (Eco Mode)
- 외기 ASHRAE A2~A4 확대

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

28 / 56

#74

PART D

상면·랙·케이블링 — 공간과 배선의 표준

상면 설계가 잘못되면 확장이 막히고, 케이블 표준이 없으면 운영이 무너진다.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 37 / 155

상면 설계 — 면적·하중·천장·진동

📏 면적 산정

- **랙 1구당 면적**: 통로 포함 약 **3~5 m²**
- 200랙 IDC = 600~1,000 m² 상면 + 30% (전기실·UPS·CRAC) → 약 1,300 m²
- **확장 마진** 30~50% 권장
- 천장 높이 **3.5~4.5 m** (Hot Aisle 환기)
- 이중 바닥 **60~100 cm**

🏗️ 바닥 하중 (Floor Load)

- **일반 사무실**: 300 kg/m² → IDC 부적합
- **데이터센터**: **750~1,500 kg/m²**
- 풀랙 무게 500~700 kg + 통로 3 m² → ≈ 200 kg/m² (장비)
- 안전계수 2~3 고려

🌐 진동·내진

- **내진 설계 등급 II** 이상 (특등급 = 1,000년 빈도)
- 랙 베이스 **앵커 볼팅** 또는 진동 격리
- 진동 한계: **0.5G 이하** (지진계 측정)
- 회전 장비 (CRAC·UPS) 진동 격리

💧 누수 감지·방수

- **누수 감지 케이블** — 바닥 아래·CRAC 주변
- 배관은 **상면 위로** 통과 금지
- 칠러 누수 시 자동 차단 밸브

🔄 환기·정전기

- 정전기 방지 바닥재 (ESD)
- 양압 유지 (외부 먼지 유입 방지)

랙 — 19인치 EIA-310 표준

📏 표준 치수

- **EIA-310-D** — 국제 표준
- 폭: **19인치 (482.6mm)** — 마운트 레일
- 깊이: 600·800·1000·**1200 mm** (서버 깊이에 맞춤)
- 높이 단위: **U** (1U = 44.45 mm = 1.75인치)
- 일반 사이즈: **42U** (1991 mm)·48U·52U
- 풀랙 무게: 빈 랙 약 100 kg + 장비 500~700 kg

🏷️ 한국 KS 표준

- **KS C 0806** — 19인치 랙
- **EIA-310**과 호환

🧱 랙 종류

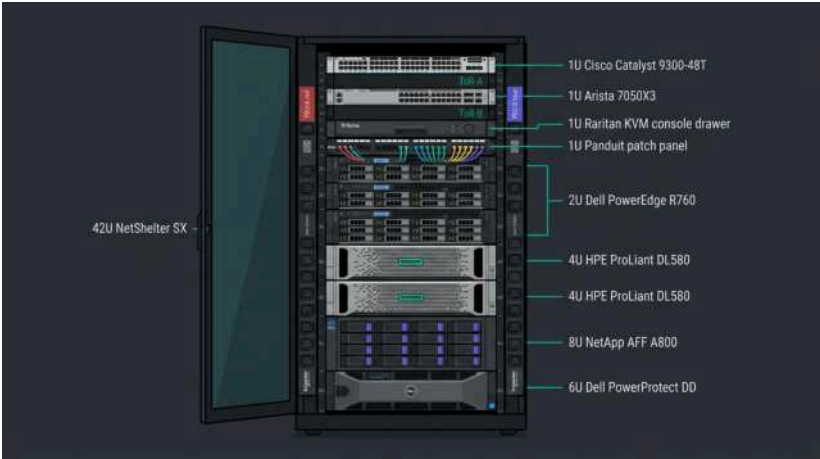
- **서버랙** — 통풍구·깊이 1000~1200 mm
- **NW랙** — 깊이 600~800 mm·HING 케이블 매니저 다수
- **전력랙·UPS랙** — 보강 프레임·하중 ↑
- **사운드 랙 (Soundproof)** — 사무실 설치용

🔒 출입·잠금

- **전후 도어** — 통풍 70% 이상 개구
- **사이드 패널** — 케이블 정리
- **전자식 잠금** (NFC·RFID)·감사 로그

🔄 통풍

- 도어 통풍구 60~70% 개구율
- 블랭킹 패널로 빈 U 막기



랙 배치도 — 42U + Hot/Cold (삽화)

Hot/Cold Aisle — 랙 실장 구조

❌ 잘못된 배치 — 모든 랙 같은 방향



문제 — 서버는 앞에서 찬 공기 흡입 → 뒤로 뜨거운 공기 배출하는데, 옆 랙이 그 뜨거운 공기를 다시 빨아들임. 이걸 **Hot Air Recirculation** 이라 함.

✅ 올바른 배치 — Hot/Cold Aisle 교차



원칙 — 앞면끼리 마주보면 Cold, 뒷면끼리 마주보면 Hot. 이 패턴을 행(Row) 단위로 반복.

Hot/Cold Aisle — 표준 배치 규칙·흔한 실수



표준 배치 규칙

- 랙은 정면 ↔ 정면, 뒷면 ↔ 뒷면으로 마주보게
- 정면 사이 = **Cold Aisle** (퍼포레이트 타일로 찬 공기 공급)
- 뒷면 사이 = **Hot Aisle** (천장 환기로 뜨거운 공기 배출)
- 두 줄을 한 단위로 반복 (Row 1·Row 2...)
- Aisle 폭: Cold 1.2 m, Hot 1.0 m 권장



통로 폭

- 점검 동선 확보 (1.0 m 이상)
- 장비 반입 동선 (1.5 m 이상)
- 비상 출구 폭 KS 표준 (1.2 m)



흔한 실수

- 모든 랙을 같은 방향으로 배치 → 인접 랙 Hot 흡입
- 폴로딩 → Hot Spot
- 블랭킹 패널 미설치 → 차가운 공기가 랙 내부 우회



추가 설계

- 랙 위 케이블 트레이 (강전·약전 분리)
- 바닥 아래 전원 케이블 + 일부 약전
- 명찰 (Rack Tag) — 좌측 상단 통일

다음 단계 — **Containment**: Hot/Cold Aisle 배치는 기본, 그 위에 천장 격벽·문으로 공기 흐름을 완전히 분리하면 효율 20~40% 더 향상. 신규 IDC는 거의 100% Containment 도입.

ToR vs EoR — 케이블링 토폴로지



ToR (Top of Rack)

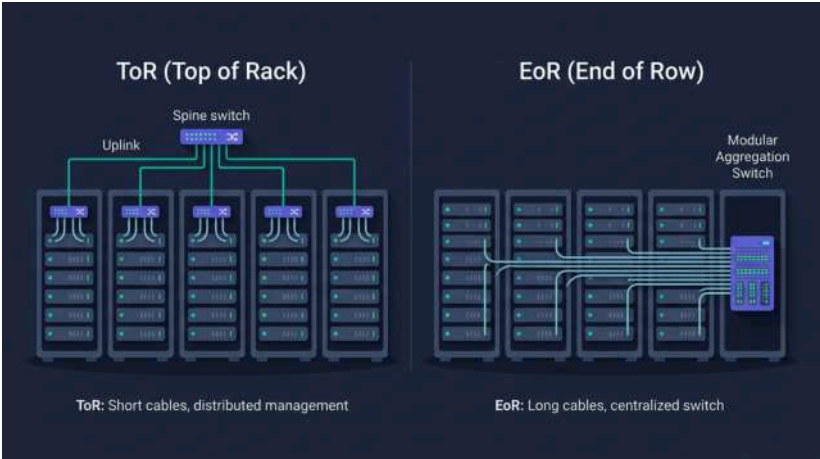
- 랙 최상단 (1U)에 스위치 1~2대
- 서버 ↔ ToR 케이블 모두 랙 내부에서 완결
- 랙 간 연결만 광 케이블 (Trunk)
- 장점: 케이블 짧고 단순·관리 쉬움·모듈화
- 단점: 랙마다 스위치 → 포트 활용률 낮음·관리 포인트 많음
- 표준: 현대 IDC 대부분 ToR
- Spine-Leaf 토폴로지와 공합



EoR (End of Row)

- 줄(Row) 끝에 대형 모듈러 스위치
- 줄 전체 서버 케이블을 한 스위치로 집중
- 장점: 포트 활용률·관리 포인트 적음
- 단점: 케이블 길이 → 관리 복잡·고장 시 줄 전체 영향
- MoR (Middle of Row) 변형도 있음
- 구형 3-Tier 토폴로지의 표준
- 현재는 점차 ToR로 대체

2026 추세: GPU 클러스터·고밀도 = **ToR + Spine-Leaf** 표준. EoR은 일부 레거시·소규모.



ToR vs EoR 비교 (삽화)

광 커넥터·트랜시버 — SFP 패밀리

🔌 광 커넥터

- **LC** — 가장 흔함, Duplex·Push-Pull
- **SC** — 구형, Square-Snap
- **MTP/MPO** — 12·24심, 40G/100G/400G용
- **ST** — 구형 BNC형
- **FC** — 산업용 나사식

🔌 SFP 패밀리

- **SFP** — 1G
- **SFP+** — 10G
- **SFP28** — 25G
- **QSFP+** — 40G (4×10G)
- **QSFP28** — 100G (4×25G)
- **QSFP-DD** — 200G·400G
- **OSFP** — 400G·800G (신규)

🌈 거리·파장

- **SR (Short Reach)** — MMF 100~400m
- **LR (Long Reach)** — SMF 10km
- **ER (Extended)** — SMF 40km
- **ZR** — SMF 80km
- **CWDM·DWDM** — 다중 파장 다중화

🎤 TA의 결정

- 랙 간 25G: SFP28 SR + OM4
- 코어 100G: QSFP28 LR + OS2
- AI 클러스터 400G: OSFP + OM5/OM4

💰 단가

- SFP28 SR 모듈 약 10~20만원
- QSFP28 LR 100~300만원
- OSFP 400G 500만원+

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



케이블링 (삽화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

38 / 56

PART E

물리 보안 — 외곽부터 랙까지 동심원

"원격 해킹보다 **물리적 침입**이 더 치명적" — 물리 보안은 보안의 시작이자 끝.

물리 보안 5중 동심원 — 외곽 → 랙

5중 방어

- 1. 외곽 (Perimeter) — 펜스·CCTV·경비
- 2. 건물 입구 (Building) — 출입통제·로비 경비
- 3. 데이터 홀 입구 (Hall) — 맨트랩·생체인증
- 4. 랙존 (Cage) — 케이지·고객사 격리
- 5. 랙 (Rack) — 전자식 도어록·감사 로그

원칙

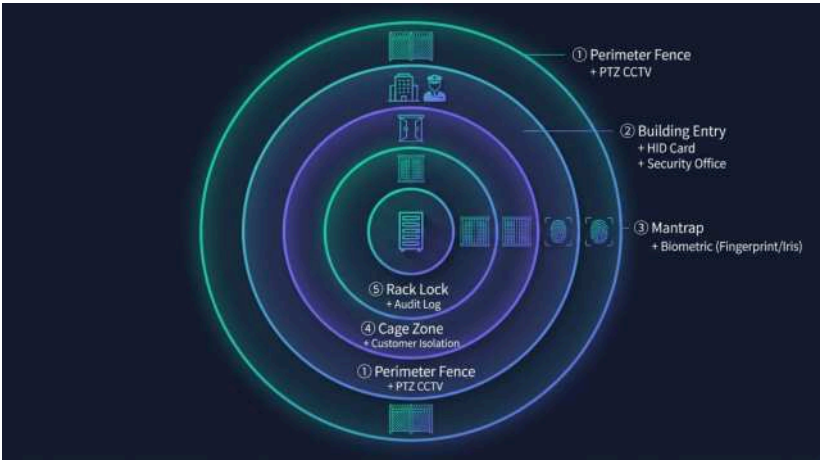
- Defense in Depth — 다층 방어
- Anti-Pas6-Back — 동일 카드 중복 입출 차단
- 2-Person Rule — 중요 구역 2인 동시 출입
- 시간·구역 제한 — 카드별 ACL

동심원별 통제

- 외곽: 펜스 2m+·침입 감지 와이어·PTZ CCTV
- 건물: HID 카드·금속 탐지·경비실 출입 기록
- 홀: 맨트랩 (Mantrap) + 생체 (지문·홍채·정맥)
- 케이지: 별도 RFID 카드·고객 격리
- 랙: 전자식 도어록 (NFC·블루투스) + 감사 로그

TA의 점검

- 카드 ACL 갱신 (퇴사·이동)
- 임시 카드 회수
- 출입 로그 월간 검토



물리 보안 동심원 (삽화)

PART F

화재 진압 · VESDA — 전산실 전용 소화 설비

일반 스프링클러는 IDC에서 **금기** — 전자장비가 침수되면 즉시 폐기.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

VESDA — 공기 흡인식 조기 감지

VESDA (Very Early Smoke Detection Apparatus)

- 공기 흡인식 (Aspirating) 연기 감지기
- 천장 격자 모양으로 흡입 파이프 배설
- 미세 연기 0.005% 감지 가능
- 일반 감지기보다 **100~1000배** 민감
- 4단계 경보: 1. **Alert** — 직원 점검 2. **Action** — 사전 조치 3. **Fire 1** — 자동 소화 준비 4. **Fire 2** — 가스 살포

대표 제품

- Xtralis (Honeywell) — 시장 표준
- 한국에서 "VESDA" = 일반명사화

전산실 화재의 특이성

- 케이블 절연 손상 → 연기 → 화재
- 배터리 스웰링 → 가스 → 폭발
- PSU 노화 → 스파크
- 화재 발생 시 수 분 내 전산실 전체 손실

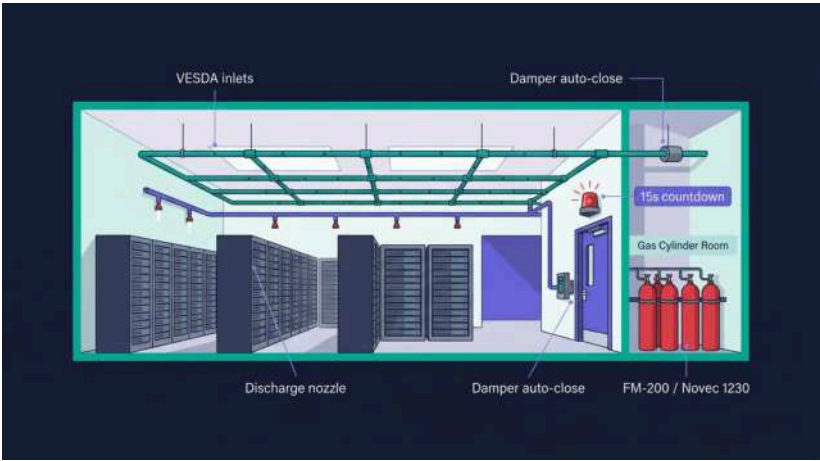
TA의 운영

- VESDA 흡입구 **막힘 점검** (먼지)
- 분기 1회 시험 살포
- 이중 감지 + 가스 살포 (오작동 방지)

가스 소화 시스템 — *FM-200·Inergen·Novec-1230·CO2*

종류	화학식	작동 원리	인체 안전	환경 (GWP)	가격
FM-200 (HFC-227ea)	C ₃ HF ₇	화학 억제 (열·라디칼)	안전	GWP 3220 (높음)	중
Novec 1230	C ₆ F ₁₂ O	화학 억제 + 흡열	안전	GWP <1 (우수)	고
Inergen (IG-541)	N ₂ + Ar + CO ₂	산소 농도 ↓ (12.5%)	안전 (잠시 호흡)	GWP 0	고
CO2	CO ₂	산소 농도 ↓ (5% 이하)	위험 (질식)	GWP 1	저
Argonite (IG-55)	N ₂ + Ar	산소 농도 ↓	안전	GWP 0	중

TA의 표준 선택 (2026): - 신규 IDC: Novec 1230 (환경 + 안전) - 기존 IDC: FM-200 (가성비) - 무인 구역만: CO2 (저비용, 무인 전제) - Inergen: 대형 공간·다중 위험



화재 진압 시스템 (삽화)

PART G

현장 환경 · 표준 — 풍하중 · 낙뢰 · 접지 · KS/IEC

함체 사양서·환경시험성적서가 **RFP 요구 항목**에 자주 등장.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



함체 + 낙뢰·접지 (삽화)

KS · IEC · EIA — 주요 표준 카탈로그

 KS (한국산업표준) • KS C IEC 60439 — 저압 배전반 • KS C IEC 60364 — 전기 설비 • KS C IEC 60529 — IP 등급 (방수·방진) • KS C 0806 — 19인치 랙 • KS C 0228 — 풍하중 • KS Q 9001 — 품질 인증  KCS-KDS • 국가건설기준 • 데이터센터 설계 가이드라인	 IEC (국제전기기술위원회) • IEC 60439 — 배전반 • IEC 60364 — 전기 설비 • IEC 60529 — IP 코드 (IP20~IP68) • IEC 61000 — EMC·전자파 적합 • IEC 62443 — OT 보안 • IEC 61643 — SPD	 EIA-TIA-ANSI • EIA-310-D — 19인치 랙 • TIA-942 — 데이터센터 표준 (Tier·구조화 배선) • TIA-568 — 케이블링 • TIA-606 — 라벨링·관리 • TIA-607 — 접지·분당  ASHRAE • TC 9.9 — 데이터센터 환경 • A1~A4 클래스
--	---	---

PART H

운영 · BCP · 산출물 — TA의 결정과 점검

설계는 끝났다 — 이제 **운영·BCP·산출물**로 마무리.

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

DCIM — *Data Center Infrastructure Management*



DCIM의 5대 기능

1. **자산 관리** — 랙·서버·NW 인벤토리
2. **전력 모니터링** — kW·kWh·PUE 실시간
3. **냉각 모니터링** — 온도·습도·풍속
4. **케이블 관리** — 결선·포트 매핑
5. **용량 계획** — 추세 분석·증설 시점



대표 솔루션

- **Schneider EcoStruxure IT** (StruxureWare)
- **Vertiv Trellis**
- **Nlyte**
- **Sunbird dcTrack**
- **Device42**
- 오픈소스: **NetBox** + **Grafana**



TA의 활용

- **실시간 대시보드** (전력·온도·PUE)
- **3D 랙 시각화**
- **모바일 알람**
- **변경관리 자동화** (CR → 결선 갱신)



도입 단가

- 중소: 1~3억 (100랙 기준)
- 대형: 10~30억 (1,000랙+)
- **CMDB·NMS·BMS와 통합** 시 효과 ↑



ROI

- 평균 전기료 8~15% 절감
- 운영 인력 부담 30% 감소

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

50 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

정전 BCP — *시나리오·복구 절차*

정전 시나리오

1. **순간 정전 (Sag)** — UPS 자동 보전, 무중단
2. **단기 정전 (5분 이내)** — UPS만으로 대응
3. **중기 정전 (5분~24시간)** — UPS + 발전기
4. **장기 정전 (24시간+)** — 발전기 연료 보충·대체 사이트
5. **광역 정전** — 한국 전역·지역 단위



복구 절차

- **0~10초**: UPS 자동 전환 (무중단)
- **10~30초**: 발전기 자동 기동
- **30초 후**: ATS 전환 (5~30초)
- **이상 시**: 우선순위에 따라 C·B·A 차단



TA의 점검 체크리스트

- **UPS 배터리 잔여 시간** 월 1회
- **발전기 무부하 시험** 월 1회
- **발전기 부하 시험** 분기 1회
- **연료 잔량** 항상 70%+
- **외부 연료 공급 계약** (24시간 가능)
- **DR 사이트 자동 전환** 시나리오



BCP 산출물

- 정전 대응 SOP
- 발전기 운전 매뉴얼
- 연료 보충 계약서
- 비상 연락망 (한전·발전기 업체)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

51 / 56

TA가 결정해야 할 데이터센터 10대 이슈



1. 자체 DC vs Colocation — 규모·운영 인력 기준
2. Tier 등급 — Tier III 표준, Tier IV는 정당화 필요
3. 전력 계약 (kVA) — IT 부하 × PUE × 마진 × 미래
4. UPS 토폴로지 — N+1 vs 2N
5. 배터리 종류 — VRLA vs Lithium-ion



6. 냉각 방식 — CRAC/CRAH/In-Row/Liquid
7. 랙 단위 — 폴랙·1/2랙·1/4랙
8. 케이블링 표준 — ToR vs EoR, 명명·색상·라벨
9. 물리 보안 — 출입통제·CCTV·맨트랩 수준
10. 화재 진압 — FM-200·Novec·Inergen·CO2

TA의 산출물: 10대 이슈마다 결정 근거와 대안 비교가 RFP·설계서에 명시되어야 함.

데이터센터 설계 산출물 패키지

도면 (Drawing)

- 상면 배치도 (Floor Plan)
- 랙 배치도 (Rack Layout)
- 전력 단선도 (One-Line Diagram)
- 냉각 계통도 (Cooling Schematic)
- 케이블 트레이도
- 소화 계통도
- CCTV 배치도

산정서 (Calculation)

- 전력 산정서 (kVA·kW)
- 냉각 산정서 (RT·CFM)
- 면적·하중 산정서
- 연료 산정서 (디젤 보관량)

정책·표준

- 케이블 명명 규칙
- 색상 코드 표준
- 랙 점유율 표준 (70%)
- 출입통제 정책
- 임시 카드 발급 정책

SOP

- 정전 대응 SOP
- 화재 대응 SOP
- 랙 입고 SOP
- CCTV 운영 SOP

BOM

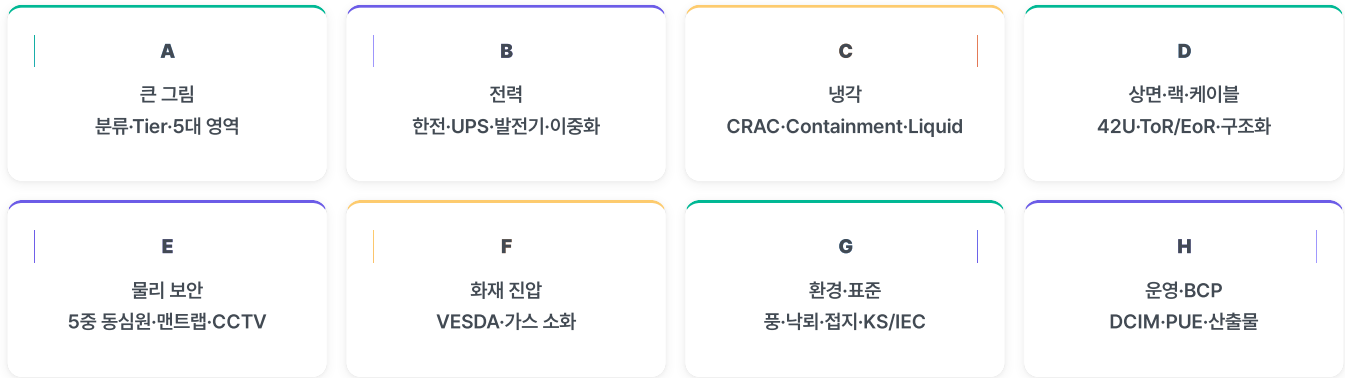
- 전력 장비 BOM (UPS·발전기·PDU)
- 냉각 장비 BOM (CRAC·칠러)
- 상면 자재 BOM (랙·타일·트레이)
- 케이블 BOM (UTP·광·트레이)

위험·BCP

- DR 사이트 설계
- BCP 시나리오 5종
- 위험 등록부
- 연락망

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

Session 1 정리 — 8개 PART · 핵심 메시지



본 세션 핵심 메시지: 데이터센터를 건물 단위 인프라 시설로 보는 시야 + 5축(전력·냉각·상면·보안·BCP) 결정 체계.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

54 / 56

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라

데이터센터 실사·인수 체크리스트 — TA가 현장에서 확인할 30개 항목

⚡ 전력 (10항)

- ☐ 한전 인입 2회선 확인
- ☐ UPS 종류·용량·배터리 수명·잔여 백업 시간
- ☐ 발전기 연료 잔량·시운전 주기·배기 경로
- ☐ ATS 절체 시험 기록 (월 1회)
- ☐ PDU 부하·이중화·계측치
- ☐ A/B 전원 분리 검증 (Trace 시험)
- ☐ 누설전류·접지 저항 측정값
- ☐ SPD 등급·교체 이력
- ☐ 분전반 회로 명세서·라벨링
- ☐ 정전 시 30초 가동 검증 (실제 시험 기록)

❄️ 냉각·환경 (10항)

- ☐ CRAC·CRAH-In-Row 모델·풍량·SLA
- ☐ Cold/Hot Aisle Containment 시공
- ☐ 온도 (Cold Aisle 22°C·Hot Aisle 35°C) 실측
- ☐ 습도 (40~60% RH) 실측
- ☐ 풍속·풍량 측정값
- ☐ 누수 감지·배수로
- ☐ 화재진압 (FM-200·Novec·CO2) 총진량·검수일
- ☐ VESDA 감도·테스트 기록
- ☐ 출입통제 (생체·맨트랩·Anti-Pas6-Back)
- ☐ CCTV 채널·해상도·30일+ 보관

🔧 상면·NW·운영 (10항)

- ☐ 랙 42U 점유율·발열 균형
- ☐ 케이블링 (ToR/EoR)·트레이·라벨
- ☐ 광·UTP·SAN·전원 케이블 분리
- ☐ OOB 관리 NW 분리
- ☐ DCIM 도구 도입 여부·자산 매핑
- ☐ PUE 측정 (12개월 평균)
- ☐ BCP·DR 사이트 위치·RPO/RTO
- ☐ 운영팀 교대 근무·SOP 문서화
- ☐ Tier 등급 인증서 (Uptime/TIA-942)
- ☐ 보험 가입 (재해·정전·화재·인명)

체크리스트는 인수 1주 전부터 시작 — 30항 중 20항 이상 통과해야 인수 완료. 미통과 항목은 발주처와 협의해 보완 계획 부속서로 첨부.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

55 / 56

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 50 / 155

IT 인프라 아키텍처 설계 · Session 1 · 데이터센터 · 전력 · 냉각 · 물리 인프라



데이터센터 실사·인수 체크리스트 — TA가 현장에서 확인할 30개 항목

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

56 / 56

IT 인프라 아키텍처 설계

업무 시스템 흐름 3-Tier · N-Tier 사용자 요청의 10계층 추적

Session 2 · Day 1 — 11:00 ~ 12:00 (60분)

아키텍처 패턴 · 계층별 컴포넌트 · 프로토콜·포트 · 응답시간 · 국내 실무 매핑

강사 박수현 ·  젠아이랩스(GenAI Labs)

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

Session 2 — 회차 메타데이터

📋 세션 정보

- **슬롯:** Day 1 · 11:00 ~ 12:00 (60분)
- **카테고리:** 인프라 보강 ① — 업무 시스템 흐름
- **연계 워크시트:** D1-3 「업무 시스템 흐름 분석」
- **선행 학습:** Session 0 (TA 역할) · Session 1 (DC)
- **후속 학습:** Session 2 (서버) · Session 3-4 (NW)

🎯 핵심 메시지

사용자 요청은 단순한 「클라이언트 → 서버」가 아니다. 평균 **10단계**를 통과하며, 각 단계마다 SPOF·보안·SLA가 결정된다.

🎯 학습 목표 (LO)

1. **아키텍처 패턴** — 모놀리식 / 3-Tier / N-Tier / MSA 차이를 안다
2. **3-Tier 계층** — Presentation · Business Logic · Data Layer 역할을 설명한다
3. **요청 추적** — 사용자 클릭부터 DB 반영까지 10계층을 단계별로 그린다
4. **계층별 표준** — 각 단계 프로토콜·포트·국내 실무 제품을 매핑한다
5. **응답시간 분해** — SLA 2초를 10계층에 어떻게 배분하는지 안다
6. **국내 실제 구성** — 대학·공공·금융의 흐름을 비교한다

Session 2 — 오늘의 흐름 (60분)

🌍 Part A~C (전반부 · 30분)

- **A. 아키텍처 패턴 큰 그림** — 모놀리식 → 3-Tier → N-Tier → MSA
- **B. 3-Tier 상세** — Presentation · Logic · Data Layer 역할·제품
- **C. 사용자 요청 10계층 추적** — Client → DNS → 보안 → LB → Web → APIGW → WAS → Cache → DB → SAN

🌍 Part D~F (후반부 · 30분)

- **D. N-Tier 확장** — 4·5·6 Tier · 마이크로서비스 분해
- **E. 국내 실제 시스템 매핑** — 대학 학사 · 공공 그룹웨어 · 금융 코어
- **F. 응답시간 분해 + 워크시트 D1-3 연계**

Part A. 아키텍처 패턴 큰 그림

모놀리식 → 3-Tier → N-Tier → 마이크로서비스

4 슬라이드 · 약 7분

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

아키텍처 진화 — 30년 큰 그림



1990s · 모놀리식

- 단일 실행 파일 · 단일 서버
- Cobol·VB·C/S 클라이언트
- **국내 사례:** 90년대 은행 코어 (HP9000-Tandem)
- 장점: 단순·빠른 응답 / 단점: 확장·이중화 불가

2000s · 3-Tier

- Web · WAS · DB 분리
- Java EE (WebLogic·JEUS) 표준
- **국내 사례:** 2000년대 모든 SI 표준 (NEIS·정부24·증권 HTS)
- 장점: 계층별 확장 / 단점: WAS 병목

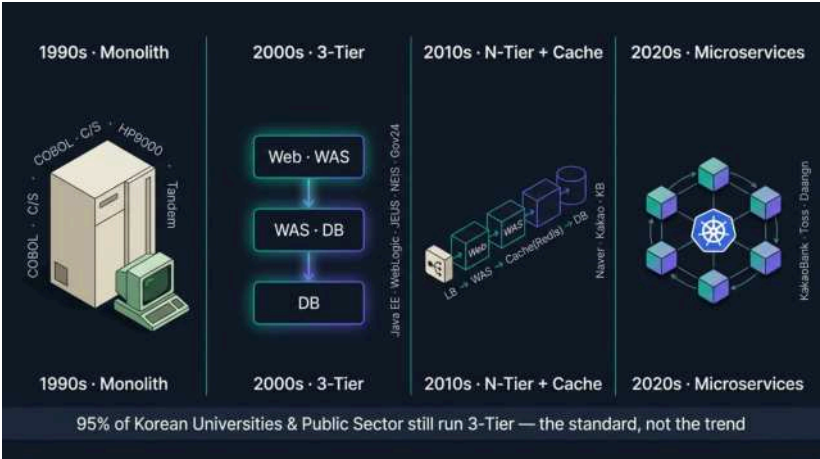
2010s · N-Tier + 캐시

- ◦ L7 LB · Reverse Proxy · Redis
- **국내 사례:** 대형 포털·이커머스·금융 (네이버·카카오·KB)
- 장점: 무중단 배포 / 단점: 운영 복잡도



2020s · 마이크로서비스

- 서비스 단위 분리 · K8s · API Gateway
- **국내 사례:** 카카오뱅크·토스·당근마켓
- 장점: 독립 배포·기술 다양성 / 단점: 분산 트레이싱 필수



아키텍처 진화 — 30년 큰 그림

4가지 아키텍처 패턴 — 언제 어떤 것을 쓰는가?

패턴	구조	적합 시스템	국내 사례
모놀리식	단일 서버 + DB	소규모 사내 시스템 (< 100명)	부서별 회계 보조
3-Tier	Web · WAS · DB 분리	중대형 업무 (100~10,000명)	대학 학사·정부24·코레일
N-Tier	+ LB · Cache · API GW	대형 24x7 서비스 (10,000명+)	카카오·네이버·페이·증권 HTS
마이크로서비스	서비스 단위 분리 + K8s	초대형 글로벌 (수십만+)	카카오뱅크·당근·토스·쿠팡

🔴 **TA 의사결정 기준** — 「현재 사용자 수」가 아니라 「3년 후 사용자·트랜잭션 증가율」 + 「운영 인력 수준」으로 결정.

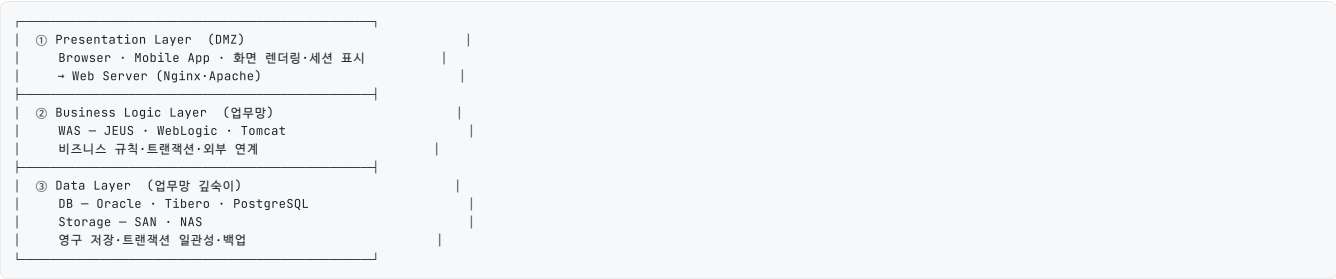
Part B. 3-Tier 상세

Presentation · Business Logic · Data Layer

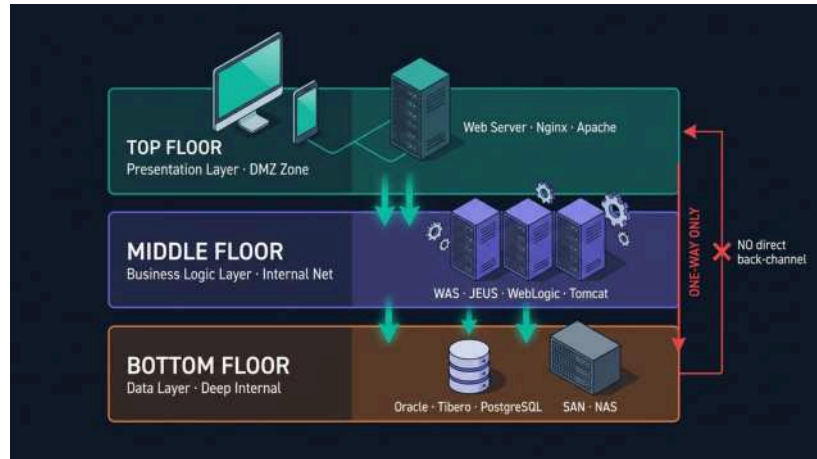
8 슬라이드 · 약 13분

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

3-Tier 큰 그림



IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적



3-Tier 큰 그림

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

10 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

① Presentation Layer — Web Server

🍷 역할

- 정적 콘텐츠 제공 (HTML-CSS-JS-이미지)
- TLS 종단 (HTTPS → HTTP 변환)
- 세션 쿠키 보관 위치
- WAS로 요청 프록시 (Reverse Proxy)
- 간단한 압축·캐시·인증 검사

📦 국내 표준 제품

- Nginx (95% 점유 · OSS)
- Apache HTTP Server (구형 시스템)
- TmaxSoft WebtoB (국내 상용 · KCMVP 검증)
- IIS (Windows 환경)

🌐 표준 프로토콜·포트

- HTTPS / TCP 443 (외부)
- HTTP / TCP 80 (내부)
- AJP / TCP 8009 (WAS 연계 · 옛 방식)
- HTTP / TCP 8080-8443 (WAS 연계 · 현재)

🔒 보안 요구

- ISMS-P 2.6.1 — DMZ 배치 필수
- TLS 1.2 이상 — TLS 1.0/1.1 차단
- HSTS 헤더 적용 권장
- X-Forwarded-For 헤더로 원본 IP 보존

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

11 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

② Business Logic Layer — WAS (Web Application Server)

🧠 역할

- 비즈니스 규칙 실행 (수강신청·결재·송금)
- 트랜잭션 관리 (Begin·Commit·Rollback)
- 세션 관리 (Stateful)
- EJB·Spring·JPA 컨테이너
- 외부 시스템 연계 (REST·SOAP·MQ·FTP)
- 스레드 풀 관리 (동시 처리)

📦 국내 표준 제품

- TmaxSoft JEUS 8/9 (국내 1위 · 공공·금융)
- Oracle WebLogic 14c (대기업 · 통신·제조)
- Apache Tomcat 10 (OSS · 중소기업)
- JBoss EAP / WildFly (드물게)

🌐 프로토콜·포트

- HTTP/HTTPS 8080-8443 (수신)
- AJP 8009 (Web → WAS · 옛 방식)
- JNDI / RMI (분산 객체)
- JDBC 1521-5432 (DB 연결)

⚙️ 핵심 튜닝 파라미터

- 스레드 풀: 기본 200 → 운영 500~2,000
- JVM Heap: 4G~16G (대형 WAS)
- GC: G1GC (현재 표준) · ZGC (저지연)
- 세션 타임아웃: 30분 표준

💡 라이선스

- JEUS·WebLogic: 코어 단위 유상 (수천만~억)
- Tomcat: 무상 (OSS)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

12 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

③ Data Layer — DB · Storage

📄 DB 역할

- ACID 트랜잭션 (Atomicity·Consistency·Isolation·Durability)
- 동시성 제어 (Lock·MVCC)
- 백업·복구 (RMAN·pg_dump)
- HA·복제 (DataGuard·Patroni·Replication)
- 암호화 (TDE·VKM·KCMVP)

국내 표준 DBMS

- Oracle 19c / 23ai (대기업·금융·공공 1위)
- TmaxSoft Tiberio 7 (국산 · 공공 우선)
- PostgreSQL 16 (OSS · 신규 시스템)
- Altibase HDB (국산 In-Memory)
- MS SQL Server (Windows 환경)

📁 Storage 역할

- 블록 I/O (DB Datafile·Redo Log)
- 파일 I/O (NFS·CIFS)
- Snapshot·Clone (Point-in-Time 복구)
- Tier 분리 (Hot SSD · Warm HDD · Cold Tape)

국내 SAN/NAS 표준

- NetApp AFF · FAS (시장 1위)
- Dell PowerStore · PowerMax
- HPE Primera · Alletra
- Pure FlashArray (성장)
- Effix·Atto (국산 일부)

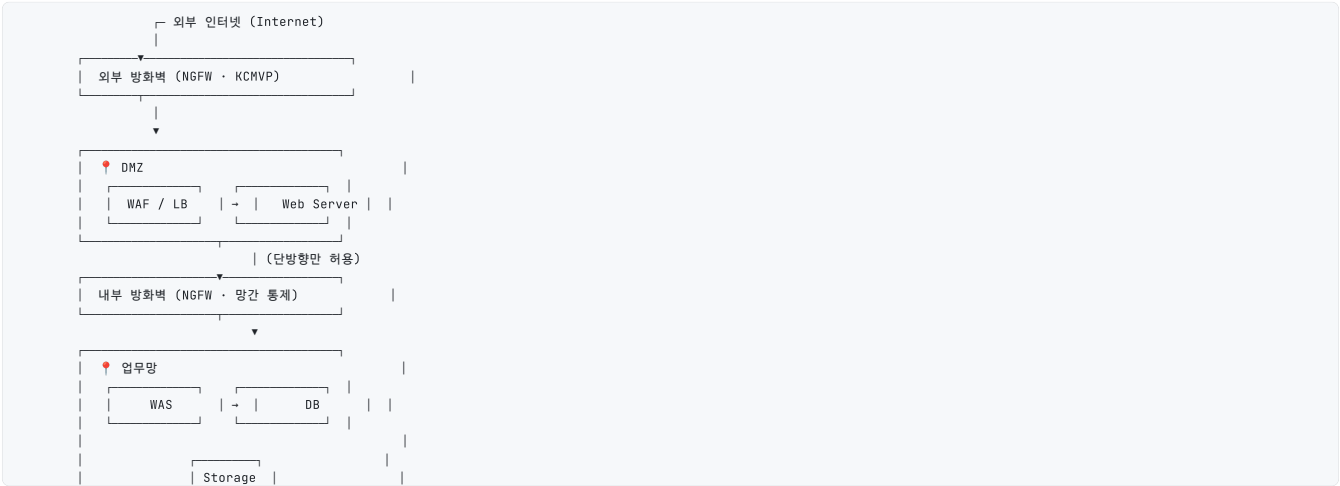
🔒 보안 요구

- 개인정보보호법 — 주민번호·계좌 AES-256 의무
- KCMVP 검증필 스토리지 (공공)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

13 / 45

3-Tier — 망분리와 통신 경로 (KISA 가이드)



KISA 망분리 가이드 핵심 규칙 - DMZ → 업무망: 단방향만 허용 (HTTP·DB Connection만) - 업무망 → DMZ: 차단 (역방향 금지) - 관리망 분리 (SSH·OOB)는 별도 망

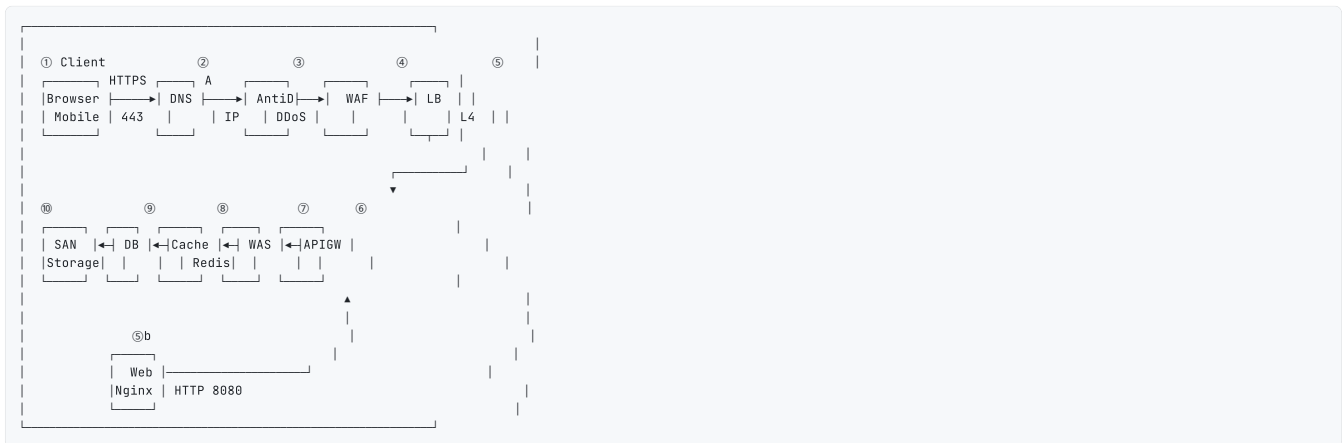
Part C. 사용자 요청 10계층 추적

Client → DNS → 보안 → LB → Web → APIGW → WAS → Cache → DB → SAN

14 슬라이드 · 약 22분

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

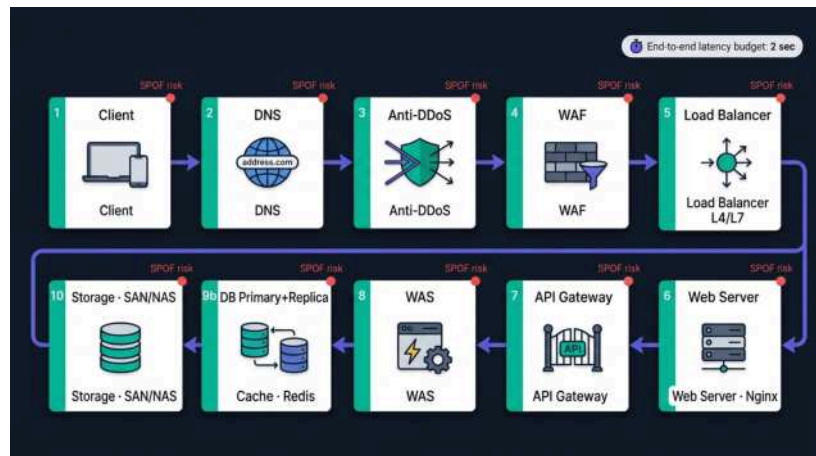
사용자 요청 — 10계층 흐름 전체도



© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

16 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적



사용자 요청 — 10계층 흐름 전체도

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

17 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

① Client + ② DNS — 시작점

① Client

- **기능:** 브라우저 렌더링 · TLS 1.3 세션 · Cookie 보관
- **국내 표준:** Chrome 70% · Edge 20% · Safari 10%
- **모바일 앱:** WebView 또는 Native (React Native·Flutter)
- **장애 시:** 사용자 단에서 즉시 인지 → 콜센터 호출
- **응답시간 SLA:** —

② DNS

- **기능:** 도메인 → IP 변환
- **프로토콜:** DNS / UDP 53
- **TTL:** 60s~24h (Failover 속도 결정)
- **국내 DNS:**
- **사내 BIND 9.x** (대기업·공공)
- **KISA DNS** (.kr 도메인)
- **클라우드플레어·AWS Route 53** (외부)
- **SPOF:** DNS 1대 → 전체 도달 불가
- **이중화:** Primary·Secondary + Geo-Anycast
- **응답시간 SLA:** < 50ms

💡 **TTL 전략** — 평상시 TTL 24h (캐시 효율). 변경 직전 TTL 60s로 단축 → 변경 → 다시 24h. **DNS Failover 활용 사례** ★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

18 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

③ Anti-DDoS + ④ WAF — 외부 보안 1·2차 방어

③ Anti-DDoS

- **차단 대상:** SYN Flood·UDP Flood·ICMP Flood·DNS Amp
- **위치:** 회선 직후 (최외곽)
- **국내 표준 제품:**
- **Radware DefensePro** (시장 1위)
- **A10 Thunder TPS**
- **이글루코퍼레이션 · SK윌더스** (서비스형)
- **방어 용량:** 10Gbps~수 Tbps
- **응답시간 SLA:** < 1ms (Bypass 모드)

④ WAF (Web Application Firewall)

- **차단 대상:** SQL Injection · XSS · CSRF · Bot · L7 공격
- **위치:** DDoS 직후·LB 앞
- **국내 표준 제품** (CC/BMT 검증):
- **Penta Security WAPPLES** ★ (국내 WAF 점유율 1위, 공공 사실상 표준)
- **모니터랩 AIWAF** (국내 클라우드 WAF 1위)
- **F5 Advanced WAF i7800** (외산 참고)
- **응답시간 SLA:** < 5ms

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

19 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

⑤ Load Balancer — L4 / L7 분산

L4 LB (TCP/UDP)

- 알고리즘: Round-Robin · Least Conn · Weighted · Hash
- 세션 유지: Source IP Hash · Cookie
- 국내 표준:
- F5 BIG-IP LTM i5800-i7800 ★
- Citrix NetScaler
- A10 Thunder ADC
- HAProxy 2.x (OSS)
- 응답시간 SLA: < 1ms

L7 LB (HTTP/HTTPS)

- 추가 기능: URL 라우팅·Header 분석·SSL Termination
- 제품: Nginx Plus · HAProxy · F5 BIG-IP

이중화 패턴 (필수)

- **Active-Standby (VRRP)**: 1대만 일하고 1대 대기
- **Active-Active (Multi-Path)**: 둘 다 일함
- **Cluster (3+ node)**: 정족수 기반

SSL Termination 위치

- LB에서 종단 (성능 우수, 내부 평문)
- Web Server에서 종단 (안전, 부하 ↑)
- WAS까지 SSL 패스스루 (E2E 암호화, 복잡)
- 국내 표준: LB 종단 + 내부망 평문

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

20 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

⑥ Web Server — 정적 콘텐츠 + Reverse Proxy

역할

- 정적 콘텐츠 직접 제공: HTML·CSS·JS·이미지 (캐시)
- **Reverse Proxy**: 동적 요청을 WAS로 전달
- **SSL Termination**: LB에서 못 끝낸 SSL 처리
- 압축: gzip·brotli
- **Rate Limit**: IP·세션별 초당 요청 제한
- 로깅: Access Log·Error Log

국내 표준

- Nginx 1.24 (95% 점유)
- Apache HTTP Server 2.4 (구형)
- TmaxSoft WebtoB (국산 · 공공)

프로토콜·포트

- 외부: **HTTPS 443**
- LB로부터: **HTTP 80**
- WAS로: **HTTP 8080 / AJP 8009**

핵심 설정 (Nginx 예시)

```
worker_processes auto;
worker_connections 4096;
proxy_pass http://was_upstream;
proxy_set_header X-Forwarded-For $remote_addr;
proxy_read_timeout 30s;
gzip on;
limit_req zone=login burst=10;
```

- 응답시간 SLA: < 10ms (정적) · < 100ms (프록시)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

21 / 45

⑦ API Gateway — 인증·라우팅·통제

📌 역할

- **인증**: OAuth 2.0 · JWT · API Key 검증
- **라우팅**: URL → 마이크로서비스 매핑
- **Rate Limit**: 사용자·앱별 호출 제한
- **로깅·감사**: 모든 API 호출 기록
- **변환**: 프로토콜·페이로드 변환 (REST↔SOAP)



국내 표준

- **Kong Gateway 3.x** ★ (OSS·OEM)
- **Spring Cloud Gateway** (Java 기반)
- **Apigee** (Google · 대기업)
- **Penta Security iSIGN+** (국내 · KCMVP)
- **TmaxSoft API Gateway**

🔒 인증 방식

- **OAuth 2.0** — 외부 사용자 (Google·Naver Login)
- **JWT (JSON Web Token)** — 무상태 인증
- **OIDC** — OAuth + Identity
- **mTLS** — 시스템 간 상호 인증
- **국내 사례**: 공공포털 → 정부24 본인인증 연계



언제 도입하는가?

- 마이크로서비스 분리 시작
- 외부 API 공개 (Open API)
- 모바일 앱·웹·파트너 통합 진입점

⑧ WAS — 비즈니스 로직의 심장



처리 흐름

1. **HTTP 요청 수신** (스레드 풀에서 워커 할당)
2. **Filter-Interceptor** (인증·권한·로깅)
3. **Controller** (URL 매핑)
4. **Service Layer** (비즈니스 로직)
5. **Transaction Begin**
6. **Repository → JDBC → DB**
7. **Transaction Commit / Rollback**
8. **JSON-HTML 응답 생성**
9. **HTTP 응답 송신**



성능 결정 변수

- **스레드 풀 크기**: 동시 사용자 = 스레드 수
- **JVM Heap**: 4G(소)~32G(대)
- **GC 알고리즘**: G1GC 표준
- **DB 커넥션 풀**: HikariCP 10~50 conn
- **세션 저장소**: 내부(Sticky) vs Redis



모니터링 도구 (국내 표준)

- **Jennifer** ★ (제니퍼소프트 · 대기업·공공 1위)
- **Pinpoint** (네이버 OSS)
- **Scouter** (LG CNS OSS)
- **APMnow** (한국SI 일부)
- **응답시간 SLA**: < 80ms

⑨ Cache — DB 부하 분산

🎯 역할

- 자주 읽는 데이터 캐싱 (상품 목록·메뉴)
- 세션 저장소 (WAS 세션 외부화)
- 임시 데이터 (장바구니·OTP)
- Pub/Sub (실시간 이벤트)

📦 국내 표준

- Redis 7.x ★ (90% 점유)
- Memcached (단순 KV)
- Apache Ignite (분산 그리드·드물게)

🔄 이중화

- Redis Sentinel (Master-Replica + 자동 Failover)
- Redis Cluster (Shard + Replica)

🌐 프로토콜·포트

- TCP 6379 (Redis)
- TCP 11211 (Memcached)

💡 캐시 전략 (Cache Pattern)

- Cache-Aside (Lazy Loading): 가장 흔함
- Read-Through: 캐시가 자동 로드
- Write-Through: 쓰기 시 캐시 갱신
- Write-Behind: 비동기 쓰기 (위험)

⚠️ 캐시 함정

- Cache Stampede: 만료 동시 발생 → DB 폭주
- 해결: 만료 시간 분산 (jitter)
- 응답시간 SLA: < 1ms

⑨ DB — 트랜잭션 최종 처리 지점

📋 Primary DB 역할

- DDL/DML 실행: CREATE·INSERT·UPDATE·DELETE
- 트랜잭션 관리: Commit·Rollback·Lock
- 무결성 보장: PK·FK·CHECK·NOT NULL
- 인덱스 유지: B-Tree·Hash
- 백업 소스: RMAN·pg_basebackup

🌐 프로토콜·포트

- Oracle: TCP 1521
- PostgreSQL: TCP 5432
- Tibero: TCP 8629
- MySQL: TCP 3306
- MS SQL: TCP 1433

이중화 (필수)

- Active-Standby:
 - Oracle DataGuard (Sync-Async)
 - PostgreSQL Streaming Replication
- Patroni + etcd (자동 Failover)
- Active-Active:
 - Oracle RAC (Real Application Cluster)
 - Tibero TAC
 - PostgreSQL BDR (드물게)

💰 라이선스

- Oracle 19c: CPU 코어당 약 6만\$ (Enterprise)
- Tibero 7: 약 1/3 수준
- PostgreSQL: 무상
- 응답시간 SLA: < 30ms (OLTP)

⑩ Storage — 데이터를 영속적으로 보관하는 계층

 SAN (Block I/O)

- 용도: DB Datafile · Redo Log · VM 부팅 디스크
- 프로토콜: FC 16G/32G · iSCSI 10G/25G · NVMe-oF
- 국내 표준 제품:
- NetApp AFF A800 ★
- Dell PowerStore 5000
- HPE Primera A630
- Pure FlashArray //X
- 응답시간 SLA: < 2ms

 NAS (File I/O)

- 용도: 그룹웨어 첨부 · 공유 파일
- 프로토콜: NFS v3/v4 · CIFS/SMB
- 응답시간: < 10ms

 핵심 설정

- RAID 레벨: RAID-10 (DB) · RAID-6 (NAS)
- Multipath: ALUA · Round-Robin · 2~4 path
- Snapshot: 시간 단위 · 일 보관 7일
- Tier: Hot SSD · Warm HDD · Cold Tape

 보안

- 암호화: NSE (Self-Encrypting Drive)
- AES-256 (개인정보보호법 요구)
- KCMVP 검증 우선

 모니터링

- NetApp ONTAP System Manager
- Dell CloudIQ
- HPE InfoSight

Part D. N-Tier 확장

4.5.6 Tier · 마이크로서비스 분해

6 슬라이드 · 약 10분

N-Tier — 3-Tier에서 어떻게 확장하는가?

4-Tier (+ Cache Tier)

Client → Web → WAS → Cache → DB

- 추가: **Redis·Memcached**
- DB 부하 30~50% 감소
- **적용 시점:** 동시 1,000명 이상

5-Tier (+ API Gateway)

Client → Web → APIGW → WAS → Cache → DB

- 추가: **Kong·Apigee**
- MSA 진입점 + 인증·라우팅
- **적용 시점:** 모바일·외부 API 공개

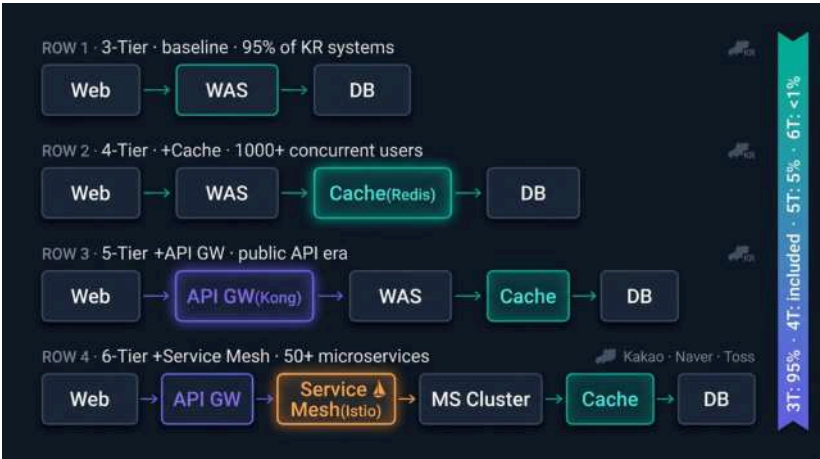
6-Tier (+ Service Mesh)

Web → APIGW → [Istio Mesh] → MS → Cache → DB

- 추가: **Service Mesh** (Istio·Linkerd)
- mTLS·트래픽 분배·관찰성
- **적용 시점:** 마이크로서비스 50개+

💡 **한국 현실**

- 95% 시스템 — **3-Tier 또는 4-Tier**
- 5% 대형 시스템 — **5-Tier**
- 1% 미만 — **6-Tier** (카카오·네이버·토스 일부)



N-Tier — 3-Tier에서 어떻게 확장하는가?

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

마이크로서비스 — 언제·어떻게 분해하는가?

🎯 분해 기준 (Bounded Context)

- 도메인 단위: 회원·주문·결제·배송
- 팀 단위: 한 팀 = 한 서비스 (Two-Pizza Team)
- 데이터 단위: 별도 DB 소유
- 배포 단위: 독립 배포 가능

📊 예시 — 사내 시스템 MSA 분해

- 회원 서비스 (Auth) → User DB
- 결제 서비스 (Approval) → Approval DB
- 게시판 서비스 (Board) → Board DB
- 메일 서비스 (Mail) → Mail DB
- 파일 서비스 (File) → Object Storage

⚠️ MSA의 함정

- 분산 트랜잭션 (Saga Pattern)
- 운영 복잡도 3~10배
- 네트워크 비용 ↑
- 트레이싱·로그 필수
- 컨테이너·K8s 운영 인력 필요

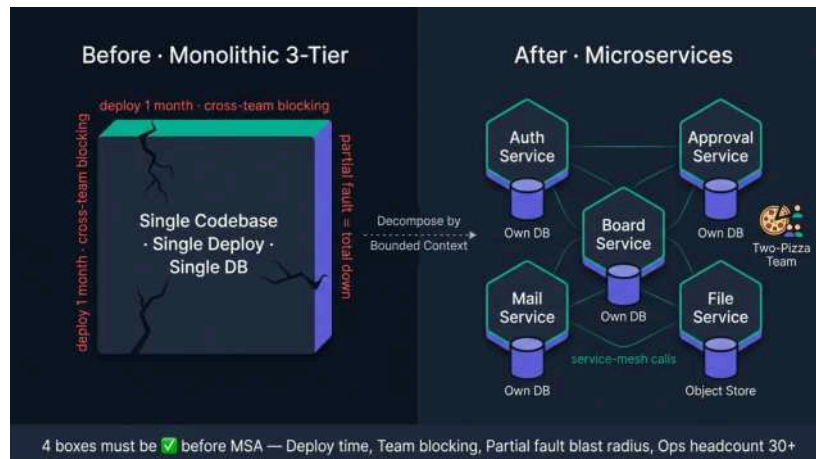
🎓 '언제 MSA로 가는가?' 체크리스트

- ☐ 모놀리식 배포가 한 달 이상 걸림
- ☐ 한 팀 변경이 다른 팀을 멈춤
- ☐ 부분 장애가 전체 중단
- ☐ 운영 인력 30+명
- 4개 모두 ✓ 이면 MSA 검토

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

30 / 45

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적



마이크로서비스 — 언제·어떻게 분해하는가?

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

31 / 45

3-Tier vs 마이크로서비스 — 결정 매트릭스

비교축	3-Tier	마이크로서비스
개발 속도	빠름 (단일 코드베이스)	느림 (분산 조율)
배포	전체 동시 (위험)	서비스 단위 (안전)
운영 인력	5~20명	30명+
장애 격리	악함 (전체 중단)	강함 (부분 영향)
트랜잭션	DB 트랜잭션 단순	Saga-Eventual Consistency
모니터링	단순 (APM 1대)	복잡 (분산 트레이싱 필수)
인프라 비용	낮음	높음 (K8s-서비스 메시)
국내 표준	95% 시스템	대기업 일부, 핀테크

🔴 TA 의사결정 — 3-Tier 유지 vs MSA 이전 - 3-Tier: 「운영 인력 부족 + 시스템 잘 돌아감」 → 그대로 - MSA: 「배포 빈도 주 5회 + 팀 10개+ + 운영 30명+」 → 검토 시작

Part E. 국내 실제 시스템 매핑

대학 학사 · 공공 그룹웨어 · 금융 코어

5 슬라이드 · 약 8분

사례 ① — 대학 학사 시스템 (학생 1.2만 명)

 구성

- 사용자: 학생 12,000 + 교직원 1,500
- 동시 접속: 평균 1,500 · 피크 8,000 (수강신청)
- 아키텍처: 4-Tier (+ Cache)
- 운영: 사내 데이터센터

 계층별 구성

계층	제품	대수
WAF	Penta WAPPLES	2 (HA)
LB	F5 BIG-IP i7800	2 (HA)
Web	Nginx	4
WAS	JEUS 8	6
Cache	Redis 7 Cluster	3
DB	Oracle 19c RAC	2
Storage	NetApp AFF A800	1 (Dual Controller)

 SLA·성능

- SLA: 99.9% (연 다운타임 ≤ 8.76h)
- 응답시간: 화면당 < 2초
- 수강신청 피크: TPS 450 (안전계수 적용 675)

 보안

- ISMS-P 인증 운영
- 개인정보보호법 적용 (학생 정보)
- 망분리: DMZ + 업무망 + 관리망

 5년 TCO

- 초기 구축: 25억
- 연간 운영: 8억 (라이선스·유지보수·인력)
- 총 5년: 약 60억

사례 ② — 공공 그룹웨어 (5,000명 사용)

 구성

- 사용자: 직원 4,500 + 외주 500
- 동시: 평균 1,000 · 피크 2,500 (출근·점심)
- 아키텍처: 3-Tier (단순)
- 운영: 자체 IDC + DR (백업 센터)

 계층별 구성

계층	제품	대수
Anti-DDoS	Radware DefensePro	1
WAF	Penta WAPPLES	2
LB	HAProxy 2.x	2
Web	Nginx	2
WAS	Tomcat 10	4
DB	Tibero 7 (TAC)	2
Storage	Dell PowerStore	1

특이 사항

- KCMVP 검증필 솔루션 만 사용 (공공)
- Tibero 사용 (Oracle 라이선스 대체)
- 국가정보원 보안적합성 검토 필수
- TTA 감리 분기 1회

 보안

- 국정원 N등급 보안적합성 검증 완료
- 전자정부 인증 필수
- 개인정보보호법 + 정보통신망법 적용

 5년 TCO

- 초기 구축: 15억
- 연간 운영: 5억
- 총 5년: 약 40억

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

35 / 45

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 68 / 155

사례 ③ — 금융 코어 시스템 (고객 500만 명·24/7)

 구성

- 사용자: 고객 500만 + 직원 1만 + 가맹점 50만
- 동시: 평균 5만 · 피크 20만 (월급일·연말)
- 아키텍처: 6-Tier (메인프레임 + 오픈계 듀얼 코어)
- 운영: 본센터 + 재해복구센터 (Active-Active 3-site)

 계층별 구성

계층	제품	비고
메인프레임	IBM z16	코어 거래
오픈계 WAS	JEUS-WebLogic-Spring Boot	채널·내부
DB	Oracle Exadata · DB2	OLTP·정산
채널	API GW(Kong) + MCI/EAI	채널 통합
Storage	EMC PowerMax · IBM DS8K	동기 복제
보안	HSM-KCMVP 모듈	키 관리

 SLA·운영 특성

- SLA: 99.99%~99.999% (연 5~52분 다운)
- 응답시간: 전·후순위 분리 — Online < 1초·Batch 야간
- 24/7 무중단 — 장애 시 즉시 뉴스·금감원 보고
- 전·후순위 거래 — Online(잔액·이체) vs Batch(정산·이자계산)
- 차세대 트렌드 — 메인프레임 → 오픈계 다운사이징·일부 MSA 전환 시도

 보안·규제

- 전자금융감독규정 — 망분리·인증·암호화 강제
- 금융보안원 인증 + ISMS-P + PCI-DSS
- HSM(Hardware Security Module) 의무
- 3-site DR — Active-Active-RPO 0초·RTO < 1분

 5년 TCO

- 초기 구축: 1,000억+
- 연간 운영: 300억 (라이선스·인력·메인프레임)
- 총 5년: 2,500억+

4 사례 비교 — 내 시스템은 어디에 가까운가?

구분	대학 학사	공공 그룹웨어	금융 코어
사용자	1.2만	5,000	500만
동시 피크	8,000	2,500	200,000
아키텍처	4-Tier	3-Tier	6-Tier (+ 메인프레임)
SLA	99.9%	99.9%	99.99%~99.999%
WAS	JEUS	Tomcat	JEUS + WebLogic + Spring Boot
DB	Oracle RAC	Tibero	Oracle Exadata + DB2
이중화	LB-WAS:DB 모두	LB-WAS:DB 모두	사이트 + 모든 계층
DR	백업·수동 복구	백업 센터	3-site Active-Active
인증	ISMS-P	ISMS-P + 국정원	금융보안원 + PCI-DSS
운영 인력	8~12명	5~8명	100명+
5년 TCO	60억	40억	2,500억+

Part F.

응답시간 분해 + 워크시트 연계

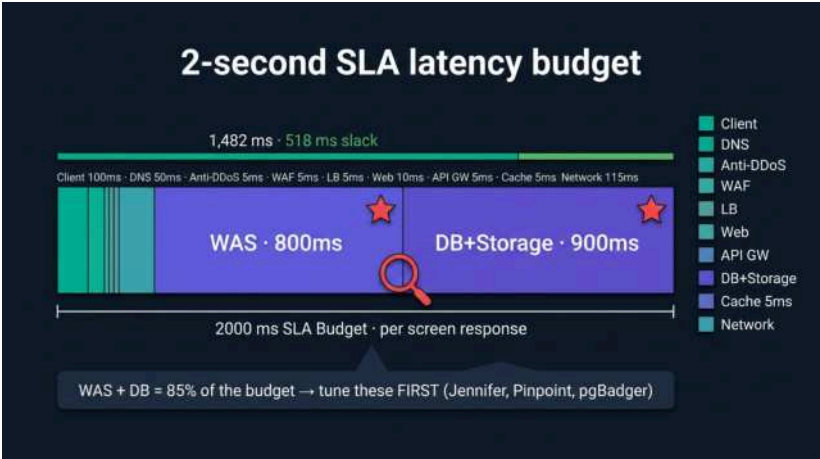
8 슬라이드 · 약 13분

IT 인프라 아키텍처 설계 · Session 2 · 업무 시스템 흐름 · 3-Tier · N-Tier · 사용자 요청 추적

응답시간 분해 — SLA 「화면당 2초」를 10계층에 배분

계층	SLA 배분	실측 평균	비고
㉠ Client	100 ms	80 ms	브라우저 렌더링
㉡ DNS	50 ms	5 ms	캐시되면 0 ms
㉢ Anti-DDoS	5 ms	3 ms	정상 트래픽
㉣ WAF	5 ms	4 ms	룰 평가
㉤ LB	5 ms	2 ms	라우팅
㉥ Web	10 ms	5 ms	정적·프록시
㉦ API GW (선택)	5 ms	3 ms	인증·라우팅
㉧ WAS	800 ms	600 ms	비즈니스 로직 ★
㉨ Cache	5 ms	0.5 ms	캐시 히트
㉩ DB-Storage	900 ms	700 ms	I/O-트랜잭션 ★
네트워크 왕복	115 ms	80 ms	각 단계 합산
총합	2,000 ms	1,482 ms	여유 518 ms

💡 TA 분석법 — 2초 SLA 미달 시 「WAS·DB」가 합쳐 1.7초를 차지. 모니터링·튜닝 1순위.



응답시간 분해 — SLA 「화면당 2초」를 10계층에 배분

병목 식별 — APM으로 SLA 위반 구간 찾기

🔍 단계별 측정 도구

- **Client:** Lighthouse · WebPageTest · Chrome DevTools
- **DNS·LB:** F5 iControl · HAProxy stats
- **Web:** Nginx Access Log + ELK
- **WAS:** Jennifer (★) · Pinpoint · Scouter
- **DB:** Oracle EM · pgBadger · pg_stat_statements
- **Storage:** NetApp ONTAP · iostat

📊 핵심 지표

- **TPS** (Transactions Per Second)
- **P95·P99 응답시간** (평균 X)
- **Error Rate** (4xx·5xx %)
- **Concurrent Users**

🎯 「병목 5가지 분류」

1. **CPU Bound** — WAS 스레드 100% 사용
2. **I/O Bound** — DB-Storage Wait 높음
3. **Lock Contention** — DB Row Lock 대기
4. **GC Pause** — JVM Full GC 빈번
5. **Network** — 패킷 손실·지연

🔧 대응 1차 조치

- WAS 스레드 풀 증가 → CPU
- DB 인덱스 추가 → I/O
- 쿼리 튜닝 → Lock
- JVM 옵션 변경 → GC
- NW 경로 검사 → Network

흐름 설계 — TA 의사결정 체크리스트

✅ 흐름 설계 시 확인 10가지

1. ☐ 10계층 흐름도 작성 (논리 + 물리)
2. ☐ 각 계층 SLA 배분 명시
3. ☐ SPOF 식별·이중화 방안 결정
4. ☐ 망분리 경계 (DMZ ↔ 업무망 ↔ 관리망)
5. ☐ 프로토콜·포트 명세서
6. ☐ 외부 인터페이스 명세서
7. ☐ 응답시간 측정 도구 결정 (APM)
8. ☐ 백업·복구 흐름 별도 명시
9. ☐ 보안 통제 ISMS-P 매핑
10. ☐ TTA 감리 산출물 준비

📋 산출물 (감리 표준)

- 시스템 구성도 (논리·물리)
- 인프라 구성 명세서
- 외부 인터페이스 명세서
- 프로토콜 사용 명세서
- 응답시간 SLA 정의서
- SPOF 분석 보고서
- 이중화 설계서

🇰🇷 국내 표준 박스

- TTA 감리 가이드 — 위 7종 산출물 모두 점검
- NIA 규모산정 지침 — 용량 산정
- ISMS-P — 보안 통제
- KISA 망분리 가이드 — DMZ 분리

흐름 설계 — 주니어 TA가 흔히 하는 실수 5가지

❌ 실수 1 — 흐름도에 「Internet → DB」 직접

- 「화살표 한 줄」로 끝내고 10계층 누락
- → 검토자가 보안·SPOF 평가 불가능

실수 2 — 단일 구성에 「99.9% SLA」

- LB 1대·DB 1대로 99.9% 약속
- → 다운 1회로 SLA 박살

실수 3 — 응답시간 배분 없음

- 「응답시간 2초」만 적고 어디서 쓰는지 미상
- → 운영 중 튜닝 불가능

실수 4 — KCMVP 미확인

- 「F5만 쓰면 OK」 → 공공기관 도입 불가
- → 사업 자체 거절

실수 5 — 망분리 무시

- DMZ 없이 Web-WAS 같은 구역
- → ISMS-P 인증 박탈

✅ TA의 「3 must」

1. 10계층 모두 그림
2. 이중화 모두 명시
3. 국내 표준·제품 매핑

Session 2 — 정리

🎯 오늘 다룬 것

1. 아키텍처 패턴 — 모놀리식·3-Tier·N-Tier·MSA
2. 3-Tier 계층 — Presentation·Logic·Data + 망분리
3. 요청 10계층 — Client → DNS → 보안 → LB → Web → APIGW → WAS → Cache → DB → SAN
4. N-Tier 확장 — 4·5·6 Tier · K8s 위치
5. 국내 사례 — 대학·공공·금융 비교
6. 응답시간 분해 — SLA 2초 배분
7. TA 체크리스트 — 흐름 설계 10항

💡 핵심 메시지 (TOP 3)

1. 사용자 요청은 평균 10단계를 통과한다 — 흐름을 모르면 장애·SPOF·SLA 모두 불가
2. 국내 95%는 3-Tier 또는 4-Tier — MSA는 운영 인력·기술 갖춘 곳만
3. KCMVP-ISMS-P-KISA 망분리 — 한국 특수 표준이 아키텍처를 결정

🇰🇷 표준 참조

- 국내: ISMS-P · TTA 감리 · KCMVP · KISA 망분리 · 전자정부 인증
- 글로벌: TOGAF (참고) · ITIL 4 · ISO 27001

Session 2 — 마무리

📝 워크시트 D1-3

- 점심 후 진행 (60분)
- 본 세션 Part C — 10계층 추적 슬라이드(s2-017 ~ s2-026, 즉 slide 17~26)를 그대로 표준 답안으로 활용 가능
- 강사 답안 + 해설 별도 배포

🎓 다음 단계

- 오후 1시: Session 2 (서버·OS·가상화)
- 14:30: Session 3-4 (네트워크)
- 16:00: Session S (구조도·SPOF)
- 「흐름」을 알았으니 「장비」를 본다

👛 실무 적용 — 이번 주 안에 해볼 것

1. 현재 시스템 구성도 직접 10계층으로 그리기
2. 각 계층 제품·버전·라이선스 표로 정리
3. 응답시간 측정 (Jennifer·Pinpoint 등)
4. SPOF 찾기 — 1대 구성된 컴포넌트
5. 국내 표준 매핑 — ISMS-P·KCMVP

IT 인프라 아키텍처 설계

Session 3

서버·OS·DB·GPU·가상화·컨테이너

Day 1 · 오후 — Compute 스택 전체 조망

폼팩터 · CPU · 메모리 · NIC · OS · DBMS · GPU · 가상화 · 컨테이너

강사 박수현 ·  젠아이랩스(GenAI Labs)

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너



Day 1 · 오후 — 컴퓨트 스택 전체 조망

Session 3 — 회차 메타데이터

📋 세션 정보

- **회차:** 2 / 15 — Day 1 오후 첫 세션
- **소속:** Day 1 — IT 인프라 큰 그림 + 서버·NW 기초
- **카테고리:** compute (컴퓨트 스택)
- **예상 분량:** 본문 103 슬라이드 + 이미지 18장
- **강의 시간:** 약 120분 (휴식 포함)
- **강사:** 박수현 — 젠아이랩스(GenAI Labs) **CEO / CTO**

🎯 핵심 메시지

컴퓨트 자산 전체 — 서버·OS·DB·GPU·가상화·컨테이너 — 를 **하나의 산정·결정 체계**로 묶는다. 라이선스가 모든 결정의 숨은 축이다.

🎯 학습 목표

1. **폼팩터** — Tower·Rack·Blade·HCI·OCP 6종 비교
2. **CPU** — x86·ARM·Power·Mainframe 4대 매핑
3. **메모리·NIC** — DDR4/5·NVMe·SmartNIC·RDMA 결정 포인트
4. **OS** — RHEL·Windows Server·UNIX·라이선스 비교
5. **DBMS** — RDBMS·NoSQL·NewSQL·시계열·검색·벡터 6대 분류
6. **GPU** — L4 → H200/B200·MIG·NVLink·InfiniBand 산정
7. **가상화·컨테이너** — Type-1 6종·K8s 아키텍처·Broadcom 이슈
8. **산출물** — 서버 BOM·OS·DB·가상화 표준안 작성

PART A

서버 폼팩터 — *Tower*부터 *OCP*까지

"어떤 모양의 서버를 도입할 것인가" — 모든 컴퓨트 결정의 출발점.

서버 폼팩터 6대 — 한눈에 비교

 **Tower**

- PC와 동일한 수직형
- 지점·소규모·연구실
- 집적도 낮음 → DC 부적합
- 가격 저렴·소음 큼

 **Rack 1U/2U/4U/6U**

- 표준 DC 폼팩터
- 1U=웹·LB, 2U=DB·병용
- 4U=GPU·고밀도 DB
- 6U+=특수 (HPC·HCI)

 **Blade**

- 새시 + 카드형 서버
- 전원·NW 공유 → 고밀도
- 벤더 종속성 강함
- HPE Synergy·Cisco UCS·Dell PowerEdge MX

 **HCI Appliance**

- 컴퓨터+스토리지+NW 일체형
- Nutanix·VxRail·SimpliVity·HCX
- VMware vSAN 통합
- 빠른 구축·고가 운영

 **Composable**

- 자원을 풀(Pool)로 추상화
- HPE Synergy·Dell PowerEdge MX
- API로 컴퓨터·스토리지·NW 조합
- 고난도 운영·유연성 최고

 **OCP (Open Compute)**

- Facebook 발 오픈 표준
- 21" 랙·12V DC 버스
- Hyperscale (Meta·MS·Google) 중심
- 한국 국내 보급 낮음



6대 폼팩터 비교 (삽화)

Rack U 비교 — 1U vs 2U vs 4U vs 6U+

U	두께	디스크 베이	PCIe 슬롯	전형적 워크로드	대표 모델
1U	44 mm	4~10 SFF	2~3	웹·API·LB·Edge	Dell R660·HPE DL360 Gen11·Lenovo SR630 V3
2U	88 mm	12~24 SFF / 24+ NVMe	6~8	DB·범용 VM·미들웨어	Dell R760·HPE DL380 Gen11·SR650 V3
4U	175 mm	24~36 LFF / 8 GPU	10+	GPU·고밀도 DB·스토리지 노드	Dell XE9680·HPE DL580·SR670 V2
6U+	265 mm+	특수	다수	HGX 8GPU·HPC·NVL72 컴포넌트	NVIDIA DGX H200·HPE Cray

선택 규칙: 디스크·GPU·PCIe 카드 수가 많을수록 U가 커진다. 랙 한 칸당 발열·전력 부하도 비례 — 4U H100 1대 = 700W × 8개 + CPU = 약 6~10 kW.

Blade · HCI · Composable — 고밀도 3종 비교

**Blade**

구성 - 새시 (6U~10U) + 블레이드 카드 8~16개 - 공유 NW 인터 커넥트 + 공유 PSU - 새시 단위 KVM·관리

대표 솔루션 - HPE Synergy / BladeSystem c-Class - Cisco UCS - Dell PowerEdge MX

장단점 -  고밀도·케이블 절감 -  새시 단위 발주·벤더 종속 - EOL 시 새시 전체 교체

**HCI Appliance**

구성 - 컴퓨터+스토리지+가상화 일체 - 노드 3대부터 시작 (3-node minimum) - 분산 스토리지 (vSAN·Nutanix DFS)

대표 솔루션 - VMware vSAN ReadyNode·VxRail - Nutanix NX·AHV - HPE SimpliVity·Cisco HyperFlex

장단점 -  빠른 구축·통합 관리 - 라이선스 비용 큼 - 노드 단위 스케일링 (작은 단위 어려움)

**Composable**

구성 - 자원을 풀(Pool)로 분리 - 컴퓨터·스토리지·NIC을 API로 조합 - 워크로드별 동적 재구성

대표 솔루션 - HPE Synergy + OneView - Dell PowerEdge MX + OpenManage - Liquid Composable

장단점 -  자원 활용률 최고 - 운영 난이도 최고 -  국내 도입 사례 적음

팩터 결정 매트릭스 — 워크로드별

워크로드	권장 팩터	이유
소규모 지점 (1~2대)	Tower or 1U Rack	운영 단순·발열 적음
웹·API·LB (수십 대)	1U Rack	1U당 32C+, 다수 운영 효율
DB·미들웨어 (HA)	2U Rack	디스크 베이·메모리 슬롯 풍부
VMware/Hyper-V 가상화 팜	2U Rack or HCI	HCI는 빠른 구축, Rack은 유연
VDI (수백 사용자)	HCI Appliance	통합 관리·노드 확장 단순
AI 학습 (대형)	4U GPU + DGX/HGX 8GPU	NVLink-InfiniBand 필수
AI 추론 (운영)	1U·2U GPU (L4-L40S)	저전력·고밀도
공공 클라우드 자가구축	OCP or 표준 2U Rack	표준화·자동화 우선
HPC 클러스터	1U·2U HPC 노드	고밀도 CPU-InfiniBand
백업·아카이브	4U Storage (24~36 LFF)	디스크 베이 극대화

현장 기본값: 80% 이상은 2U Rack — 가장 안전한 디폴트.

PART B

CPU 아키텍처 — x86 · ARM · Power · Mainframe

칩 선택은 OS·DB·라이선스·성능을 동시에 결정한다.

CPU 4대 아키텍처 — 시장 점유율·용도

◆ **x86 (Intel · AMD)**

- 시장 점유율 80%+
- Intel Xeon Scalable Gen 4/5 (Sapphire/Emerald Rapids)
- Intel Xeon 6 (Granite Rapids-Sierra Forest, 2024~)
- AMD EPYC Genoa (9004)·Bergamo·Turin (9005, 2024)
- 범용 워크로드 사실상 표준

◆ **ARM 서버**

- 급성장 영역 (2022~)
- Ampere AmpereOne (192C, 2024)
- AWS Graviton 3/4 (퍼블릭만)
- NVIDIA Grace (Grace-Hopper, Grace-Blackwell)
- 전성비 우수·국내 도입 초기

● **Power (IBM)**

- 시장 점유율 ↓ but 금융·공공 잔존
- IBM Power9 (2017) · Power10 (2021)
- AIX·IBM i·z/Linux on Power
- 메인 사용처: 코어뱅킹·보험·일부 공공

● **Mainframe (IBM Z)**

- 국내 30~40개 사이트 잔존 (금융·공공)
- IBM Z16 (2022) · z/OS·z/Linux
- 초고가용성 (>99.9999%)
- COBOL·CICS·DB2 z/OS

○ **SPARC (Oracle·Fujitsu)**

- Oracle SPARC M8 (2017이 마지막)
- Solaris OS·점진적 EOL
- 신규 도입 거의 없음 — 운영 잔존만



CPU 4대 아키텍처 비교 (삼화)

Intel Xeon Scalable — Gen 4 / 5 / 6

세대	코드명	출시	최대 코어	메모리	PCIe	비고
Gen 4 SP	Sapphire Rapids	2023.01	60C	DDR5-4800 (8ch)	PCIe 5.0	첫 DDR5-CXL 1.1
Gen 5 SP	Emerald Rapids	2023.12	64C	DDR5-5600 (8ch)	PCIe 5.0	Cache 1.5× 증가
Xeon 6 (P-Core)	Granite Rapids	2024.09	128C	DDR5-6400-MRDIMM	PCIe 5.0+CXL 2.0	성능 중심
Xeon 6 (E-Core)	Sierra Forest	2024.06	288C	DDR5-6400	PCIe 5.0+CXL 2.0	전성비·클라우드



워크로드 매핑

- DB / 미들웨어 / VM: Xeon 6 P-Core
- 웹 / API / 컨테이너 팜: Xeon 6 E-Core (Sierra Forest)
- HPC / AI 추론: Xeon 6 P-Core + AVX-512-AMX
- 신규 RFP: 2026년 기준 Xeon 6 권장



라이선스 영향

- Oracle DB Core Factor — Intel x86 = **0.5** (코어 1개 = 0.5 라이선스)
- VMware vSphere — Per-Core (2024~)
- Windows Server DataCenter — Per-Core (16C 기본 + 추가)
- 고코어 = 라이선스 폭증 → 작은 SKU 여럿이 유리할 때 있음

AMD EPYC — Genoa / Bergamo / Turin

세대	코드명	출시	최대 코어	메모리	PCIe	비고
EPYC 9004	Genoa	2022.11	96C	DDR5-4800 (12ch)	PCIe 5.0	첫 12채널-DDR5
EPYC 9004C	Bergamo	2023.06	128C (Zen 4c)	DDR5-4800	PCIe 5.0	클라우드·고밀도
EPYC 8004	Siena	2023.09	64C	DDR5-4800 (6ch)	PCIe 5.0	Edge-SMB
EPYC 9005	Turin	2024.10	192C (Zen 5c)	DDR5-6000 (12ch)	PCIe 5.0	신규 RFP 추천

AMD의 강점: 코어 수 + 메모리 채널 (12ch) 모두 Intel 대비 우위. 가상화 호스트·DB·HPC에서 점유율 급성장. 2024년 기준 신규 서버 시장 점유율 약 30%+.



AMD 선택 이유

- 코어 수 더 많음 (192C vs 128C)
- DDR5 12채널 (Intel 8채널)
- 가상화 밀도 우수
- 라이선스 효율 (코어당 성능)
- Genoa부터 가격 경쟁력



Intel 잔존 이유

- AVX-512 / AMX (AI 가속)
- QAT (암호화 가속)
- Optane (단종이나 잔존)
- 호환성 (레거시 SW)
- 벤더 표준 (HPE·Dell 표준 라인 풍부)

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

ARM 서버 — Ampere · Graviton · Grace

Ampere

- **AmpereOne (2024)** — 192C, DDR5-5600
- **Altra Max** — 128C (전세대)
- Single Thread per Core (예측성)
- Oracle Cloud-Azure-국내 도입 시작
- 국내 사례: 네이버 클라우드 일부

AWS Graviton

- **Graviton 3 / 3E** — 64C, Neoverse V1
- **Graviton 4 (2024)** — 96C, Neoverse V2
- **AWS 전용** — 직접 구매 불가
- AWS EC2 c7g/m7g/r7g 인스턴스
- 동급 x86 대비 **40% 전성비**

NVIDIA Grace

- **Grace (CPU only)** — 72C ARM Neoverse V2
- **Grace Hopper (GH200)** — Grace + H100
- **Grace Blackwell (GB200)** — Grace + B200 × 2
- NVLink C2C로 CPU↔GPU 900GB/s
- 대형 AI 학습 클러스터 (NVL72)

ARM 서버의 한국 진입: 클라우드(NCP·KT·NHN) 일부 채택. **온프레미스는 도입 초기.** 호환성 (DB 벤더·미들웨어)·운영 도구·인증서 문제로 신중. **2026~27이 본격 확산 시기**로 전망.

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

15 / 103

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

Power · Mainframe · SPARC — 레거시지만 살아있는

IBM Power

- **Power10 (2021)** — 15C, SMT-8 (코어당 8 스레드)
- **AIX 7.3** · IBM i 7.5 · z/Linux on Power
- 코어 라이선스 (Oracle Factor = **1.0**, x86의 2배)
- 주요 사용처
- 보험사 코어 시스템 (AIX + DB2)
- 일부 공공 (IBM i 차세대)
- 통신사 BSS/OSS
- EOL 압박 ↑ but 마이그레이션 부담 ↑↑

IBM Mainframe (Z)

- **IBM Z16 (2022)** — Telum 칩 (8C × 4 = 32C)
- z/OS · z/VM · z/Linux
- **국내 30~40개 사이트** (대형 은행·신용카드·증권·생명보험·공공)
- 가용성 >99.9999% (다운타임 분단위/년)
- COBOL·CICS·DB2 z/OS·IMS
- **인력 노령화** — 30~40대 신규 인력 거의 없음
- 차세대 시 X86 다운사이징 또는 z/Linux 전환

◦ Oracle SPARC

- **SPARC M8 (2017이 마지막)** — 32C, SMT-8
- Solaris 11.4 — 2034 지원
- **신규 도입 사실상 종료**
- 잔존 사이트
- 일부 통신사·공공
- Oracle ULA 묶음 잔존
- 차세대 = x86 Linux + Oracle 마이그레이션

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

16 / 103

CPU 결정 매트릭스 — 워크로드별

워크로드	권장 CPU	이유
웹·API·LB·MSA 컨테이너	Intel Xeon 6 E-Core / AMD EPYC Bergamo	고밀도·전성비
VMware/KVM 가상화 호스트	AMD EPYC Genoa/Turin	코어·메모리 채널
DB OLTP (RDBMS)	Intel Xeon 6 P-Core (낮은 코어·고클럭)	라이선스 효율·단일 스레드
DB OLAP-HTAP	AMD EPYC (다수 코어)	병렬 쿼리·메모리 대역
AI 학습 (GPU 호스트)	AMD EPYC 9474F (32C 고클럭) or NVIDIA Grace	PCIe 5.0·메모리 대역
AI 추론 (CPU 가속)	Intel Xeon 6 P-Core (AMX)	AMX·AVX-512
HPC (Fluid-CFD·해석)	AMD EPYC + InfiniBand	코어·메모리 대역
클라우드 자가 구축	AMD EPYC + ARM 혼재	전성비·다양성
공공·금융 코어	x86 Intel (보수적)	검증·호환
레거시 AIX·z/OS	Power10·Z16 유지	마이그레이션 부담

CPU 벤치마크 — 대표 지표 · 수치 예시 · 워크로드 매핑



대표 서버 CPU 벤치마크

- SPECint 2017 — 정수 연산
- SPECfp 2017 — 부동소수점
- SPECrate 2017 — 서버 처리량(Throughput)
- SPECjbb 2015 — Java 서버 워크로드
- CoreMark — 임베디드 CPU 성능
- TPC-C — OLTP 시스템 성능
- TPC-H — OLAP 시스템 성능
- STREAM — 메모리 대역폭
- LINPACK — HPC 부동소수점
- MLPerf — AI 학습·추론



수치 예시 (벤더 공개 결과 기준)

- Xeon 6 P 128C · SPECrate2017_int ≈ 1,800
- EPYC 9755 (128C) ≈ 2,100
- Graviton 4 96C ≈ 1,400

base/peak·제출시기·컴파일라·SMT 여부에 따라 수백 점 차이.



실무 활용 — 벤치마크별 워크로드 매핑

- SPECrate → 가상화 호스트
- SPECjbb → WAS·Java 서비스
- TPC-C → 금융·주문·ERP DB
- TPC-H → DW·BI·OLAP
- STREAM → 인메모리 DB·HPC·SAP HANA
- LINPACK → 과학기술 계산
- MLPerf → AI 학습·추론



벤치마크 해석 주의

- 벤치마크는 상대 비교용
- 실제 서비스와 차이 존재
- 벤더 결과는 최적 튜닝 환경
- 최종 판단은 PoC + 운영 메트릭 기반

CPU 산정 — 라이선스가 SKU 를 결정한다

Compute 회차에서 CPU 의 마지막 화두는 **라이선스**. 정량 산정 공식·벤치마크·가상화 오버헤드는 운영 회차로 넘기고, 여기서는 "왜 코어 수가 적은 SKU 가 답일 때가 많은가" 한 페이지로 마무리한다.



Per-Core 라이선스 SKU 표

제품	모델	단위	한국 현장 영향
Oracle DB EE	Per-Core (계수 0.5 x86)	1코어 당 라이선스	코어 1개 증가 = 수천만 원
MS SQL EE	Per-Core (최소 4코어)	2코어 팩	OLTP DB 표준 비용
VMware vCF / vSphere	Per-Core (최소 16코어/CPU)	코어팩	2024 SKU 개편 후 급등
Red Hat OpenShift	Per-Core 또는 소켓	16코어 묶음	K8s 도입 비용 핵심
WebLogic-WAS EE	Per-Core 또는 NUP	코어 묶음	WAS 라이선스 비용



한 줄 결정 룰

같은 성능이면 코어 수가 적고 클럭이 높은 SKU — 예: Xeon Gold 6444Y 16C @ 4.0 GHz vs Xeon Platinum 8462Y+ 32C @ 2.8 GHz. Oracle EE 라이선스로 환산 시 수억 원 단위 차이.



정량 산정 본문은 Session 8 PART B

- TPS·동접·세션 산정 — s8 ## TPS 산정
- CPU 산정 공식 SPECint·TPC-C·TPS/Core·트랜잭션 CPU ms — s8 ## CPU 산정
- 가상화/컨테이너/암호화 오버헤드 가산율 — s8 ## CPU 산정
- 6단계 사이징 절차 + 라이선스 영향 ⑩ 단계 — s8 ## CPU 산정
- Memory·IOPS·네트워크·스토리지 산정 — s8 ## Memory/IOPS/...
- 산정 사례 LMS 10K 동접·학사 행정 500 동접 — s8 ## 산정 사례



Compute 회차 정리

- CPU 종류·세대·코어 수 → CPU 결정 매트릭스 (앞 슬라이드)
- 벤치마크 지표 → CPU 벤치마크 (앞 슬라이드)
- 라이선스 → SKU → 이 슬라이드 (Per-Core 비용 인지)
- 정량 산정 공식 → Session 8

PART C

메모리 · 로컬 스토리지 · NIC

서버 한 대의 BOM에서 CPU 다음으로 큰 변수.

메모리 비교 — *DDR4 / DDR5 / MRDIMM / PMem / CXL*

표준	출시	속도 (MT/s)	채널	용량 (DIMM당)	비고
DDR4	2014	2133~3200	플랫폼별 4~8	16~64GB	레거시 주력
DDR5	2021	4800~8800+	8~12	16~256GB	현재 서버 표준
DDR5 MRDIMM	2024	8800~12800+	8~12	256GB+	Xeon 6 중심 초고대역
Persistent Memory	2018	DDR4 호환	DIMM 슬롯	128/256/512GB	Intel Optane PMem — 현재 단종
CXL Memory	2023~	PCIe 5.0/6.0	(PCIe lane)	256GB~수TB	메모리 확장(현재)·풀링(미래)



DDR5 핵심

- 듀얼 서브채널 — **32-bit + ECC × 2** (실질 40-bit × 2)
- **PMIC on DIMM** — 전원 안정성 ↑
- 온다이 **ECC** + 시스템 ECC
- 8채널 (Intel) / 12채널 (AMD)
- DIMM 256GB → 노드당 수 **TB** ~ 수십 **TB** 구성 가능



채널 수는 CPU 플랫폼 의존

- Xeon E5 v4 → 4채널
- Xeon Scalable Gen1/2 → 6채널
- EPYC Naples/Rome → 8채널
- Xeon 6 / EPYC Turin → 8~12채널



CXL Memory

- **Compute Express Link** — PCIe 위 메모리 프로토콜
- **1.x** = Device Attach · **2.0** = Switch · **3.0** = Fabric
- **현재:** 메모리 확장 (Memory Expansion) — 압도적 다수
- **미래:** 메모리 풀링 (여러 서버 공유) — 아직 제한적
- Xeon Sapphire Rapids·Genoa 부터 존재
- **2024~25 Xeon 6 / EPYC Turin 본격 지원**
- **2026~27 본격 확산** 전망 — Dell·HPE·Lenovo·Supermicro 출시

ECC · RDIMM · LRDIMM — *서버 메모리 필수 용어*



ECC (Error Correcting Code)

- 서버 표준 (필수)
- 1비트 오류 자동 정정 (SECDED)
- 2비트 검출
- **Advanced ECC** — Chipkill·SDDC·ADDDC
- DDR5는 온다이 ECC 추가 (RAS 강화)



UDIMM (Unbuffered)

- 일반 PC용, 서버에는 사용 ×

RDIMM (Registered)

- 칩과 메모리 컨트롤러 사이 **Register**
- 부하 분산 → 다수 DIMM 가능
- 서버 표준
- 슬롯당 64/128GB

LRDIMM (Load-Reduced)

- Register + **Buffer** 추가
- 더 큰 용량 (DIMM당 256GB+)
- 약간의 지연 증가
- 대용량 메모리 서버에 필수



채널 균형 (Channel Balance)

- 8채널이면 **8 / 16 / 24 DIMM** (8의 배수)
- 채널 불균형 시 **대역폭 크게 손실**
- 12채널 EPYC는 12 / 24 DIMM
- 예: 2S × 12채널 = 24 DIMM 모두 채워야 최대



TA 메모

- DIMM 수 적게 + 용량 큰 SKU = 미래 확장 ↓
- DIMM 수 많이 + 용량 작은 SKU = 확장 ↑·발열 ↑

로컬 스토리지 — *NVMe · SATA · SAS · EDSFF*

인터페이스	대역폭	폼팩터	전형적 용도
SATA III	6 Gbps	2.5" / 3.5"	백업·아카이브·OS 부트
SAS 12G	12 Gbps	2.5" SFF / 3.5" LFF	엔터프라이즈 HDD/SSD-RAID
SAS 24G	24 Gbps	2.5" SFF	고성능 SAS SSD
NVMe (PCIe Gen 4)	32 GT/s × 4 = ~8 GB/s	U.2 / U.3 / M.2	DB·캐시·범용 SSD
NVMe (PCIe Gen 5)	64 GT/s × 4 = ~16 GB/s	U.2 / U.3 / E1.S / E3.S	신규 표준 (2024+)
EDSFF E1.S	PCIe Gen 4/5	5.9 / 9.5 / 15 / 25 mm	1U 고밀도 (32 베이)
EDSFF E3.S	PCIe Gen 4/5	7.5 / 16.8 mm	2U 고밀도·고용량



폼팩터 트렌드

- **2.5" SFF** — 여전히 표준이나 NVMe Gen5에서 한계
- **U.2 / U.3** — 핫스왑 NVMe 표준
- **M.2** — 부트 디스크·소형 캐시

TA 결정 포인트

- **OS 부트** — M.2 × 2 (RAID-1) 별도
- **OLTP DB** — NVMe Gen4/5 (P-state·DWPD 3+)
- **백업·아카이브** — SATA 7.2K HDD or SAS Nearline

NIC 속도 — 1G에서 800G까지

속도	출시	매체	사용처
1 GbE	1999	Cat 5e / 6 UTP	관리망·소규모·iLO
10 GbE	2002	Cat 6a / SFP+ DAC / 광	표준 서비스망 (퇴조)
25 GbE	2016	SFP28	현재 표준 서비스망
40 GbE	2010	QSFP+	데이터센터 백본 (점퇴)
100 GbE	2010~	QSFP28	현재 백본 표준
200 GbE	2018	QSFP56	고대역 백본
400 GbE	2017 (실용 2022)	QSFP-DD / OSFP	Spine-AI 클러스터
800 GbE	2022 (실용 2024)	OSFP / QSFP-DD800	AI 클러스터 (NVL72-HGX)

한국 IDC 현재 표준 (2026): 서버 NIC = 25 GbE, 백본 = 100 GbE, AI 클러스터 = 400/800 GbE. 10 GbE는 운영 잔존만, 신규는 25 GbE 디폴트.

NIC ↔ 광 모듈 한 줄, 본문은 Session 4

서버 NIC 25/100 GbE = SFP28 / QSFP28 슬롯. NIC 속도가 정해지면 광 모듈·케이블 SKU 가 자동 결정 — RFP/BOM 에 NIC 와 모듈을 묶어 명세한다.



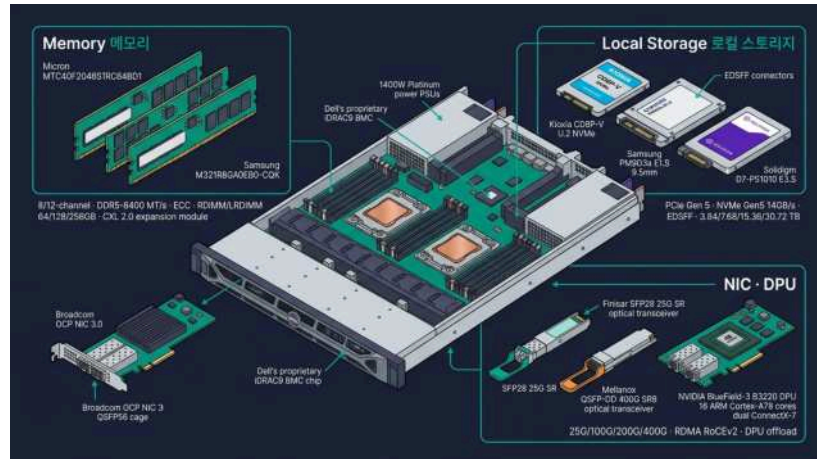
NIC 관점 한 줄

- NIC = 서버의 네트워크 어댑터 (PCIe 카드)
- 슬롯 ↔ 모듈 매칭: 25G NIC → SFP28 슬롯, 100G NIC → QSFP28 슬롯
- 단거리 (랙 내 <3m): DAC 직결 케이블 — 가장 저렴·저지연
- 중거리 (랙 사이 <30m): AOC 또는 광 모듈 + MM
- 장거리 (DC 간): 광 모듈 + SM
- 벤더 코딩 주의 — Cisco/Juniper/Arista 모듈 호환 키 상이, Generic 은 보증 없음



본문은 Session 4 PART B

- 광 모듈 SKU 11종
(SFP·SFP+·SFP28·SFP56·QSFP+·QSFP28·QSFP56·QSFP-DD·OSFP·OSFP-XD) — s4 ## 광 모듈
- 광 거리 표준 SR·LR·ER·ZR · 100/400GBASE 시리즈 — s4 ## 광 거리 표준
- 광 케이블 Single-Mode vs Multi-Mode (OM3/OM4/OM5/OS2) — s4 ## 광 케이블
- 케이블 결정 매트릭스 Cat 6a·Cat 8·DAC·SR·LR — s4 ## 케이블 결정 매트릭스



메모리·NVMe·NIC 분해도 (삽화)

PART D

운영체제 — *Linux · Windows · UNIX*

OS는 라이선스·운영·인력을 동시에 결정한다.

OS 3대 진영 — *Linux · Windows · UNIX*

 **Linux**

- 시장 점유율 80%+ (서버 영역)
- 무료/유료 디스트리뷰션 다양
- 한국 공공·금융 표준
- 대표
 - RHEL (Red Hat Enterprise)
 - Rocky-AlmaLinux (RHEL 호환·무료)
 - Ubuntu Server LTS
 - SUSE Linux Enterprise
 - Oracle Linux
 - Amazon Linux 2/2023

 **Windows Server**

- 시장 점유율 15~20% (서버 영역)
- 라이선스 (DataCenter / Standard / CAL)
- Hyper-V 가상화 기본 포함
- AD-DNS-DHCP-IIS-MS SQL 연동
- 대표
 - Windows Server 2019 (EOL 2029)
 - Windows Server 2022 (EOL 2031)
 - Windows Server 2025 (2024.11 GA)

 **UNIX 전통**

- 시장 점유율 ↓ but 운영 잔존
- 대표
 - AIX (IBM Power) — 금융 잔존
 - HP-UX (HPE Itanium) — EOL 임박
 - Solaris (Oracle SPARC) — EOL
 - FreeBSD (오픈소스) — 일부 NW 장비·연구

 **한국 시장 (2026)**

- Linux: 75%
- Windows: 18%
- UNIX: 6%
- 기타: 1%

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

27 / 103

#174

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

The diagram illustrates the lineage of Linux distributions, categorized into three main families: Red Hat 계열, Debian 계열, and SUSE 계열. At the bottom center is the Linux Kernel 6.8 LTS. The Red Hat 계열 (Red Hat fedora) includes RHEL 9.4 / 10, which branches into Rocky Linux 9.4 (EOL 2032), AlmaLinux 9.4 (EOL 2032), CentOS Stream 10 (EOL 2029), Oracle Linux 9 UEK Linux 2023 (EOL 2032), and Amazon Linux 2023. The Debian 계열 (Debian 12 Bookworm) includes Debian 12 Bookworm, which branches into Ubuntu Server 24.04 LTS and Linux Mint Debian Edition. The SUSE 계열 (SUSE Linux Enterprise) includes SUSE Linux Enterprise Server 15 SP6, which branches into openSUSE Leap 15.6 and openSUSE Tumbleweed. Other distributions like Kubuntu 24.04 and Linux Mint Debian Edition are also shown.

Linux 디스트리뷰션 가계도 (삽화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

28 / 103

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 87 / 155

Linux 디스트리뷰션 비교 — 8종

배포판	모기업	라이선스	지원 기간	주 사용자	비고
RHEL	Red Hat (IBM)	구독 (유료)	10년 (Standard-Extended까지 13년)	공공·금융 표준	가장 비싼 Subscription
Rocky Linux	RESF	무료	10년	RHEL 대안·신규 표준	CentOS 후속, 커뮤니티
AlmaLinux	CloudLinux	무료	10년	RHEL 대안·신규 표준	ABI 호환 보증
CentOS Stream	Red Hat	무료	5년 (rolling)	RHEL 미리보기	운영 비권장 (Beta 성격)
Ubuntu Server LTS	Canonical	무료 (Pro 유료)	5년 (Pro 10년)	AI·Cloud·연구	22.04 / 24.04 LTS
SUSE Linux Enterprise	SUSE	구독 (유료)	13년	유럽·SAP HANA	SAP 표준
Oracle Linux	Oracle	무료 (지원 유료)	10년	Oracle DB·ULA	UEK 커널 옵션
Amazon Linux 2/2023	AWS	무료 (AWS만)	5년	AWS EC2 전용	일반 온프레미스 부적합

한국 2026 신규 RFP의 표준 선택: 공공·금융 = RHEL 또는 Rocky/AlmaLinux, AI/연구 = Ubuntu LTS, SAP = SUSE, Oracle DB = Oracle Linux.

Windows Server — 2019 · 2022 · 2025

버전 비교

버전	출시	주류 지원	확장 지원	특징
2019	2018.10	2024.01	2029.01	컨테이너·LTSC
2022	2021.08	2026.10	2031.10	Secured-core-hot-patching
2025	2024.11	2029.10	2034.10	AI·Wi-Fi 7·SMB over QUIC

에디션

- Standard — VM 2개 권리 · 라이선스 Per-Core
- DataCenter — VM 무제한 · 가상화 호스트 표준
- Essentials — 25사용자/50디바이스 · SMB

핵심 역할

- AD DS / AD CS — 도메인·인증서
- DNS / DHCP / WINS — 인프라 서비스
- IIS — 웹 서버
- MS SQL — DB (별도 라이선스)
- Hyper-V — 가상화 (기본 포함)
- Failover Cluster — HA
- WSUS — 패치 관리
- RDS — 원격 데스크탑·VDI

TA 결정 포인트

- 라이선스 모델 2024 — Core Pack (2-core × 8 = 16C 기본) + CAL
- DataCenter vs Standard — VM 4개 이상이면 DataCenter
- AD = 도메인 인프라 — AD 1개당 도메인 컨트롤러 2대+

UNIX 계열 — *AIX · HP-UX · Solaris · FreeBSD*

IBM AIX

- IBM Power 전용 (Power9-Power10)
- **AIX 7.3** (현 버전, ~2030+)
- LPAR-WPAR-PowerHA-VIOS
- 강점: 안정성·금융 코어·DB2
- 한국 점유: 금융 코어뱅킹·일부 공공
- 신규 도입은 거의 없음, 유지 운영 중심

HP-UX

- HPE Itanium 전용 (이미 EOL)
- HP-UX 11i v3 — 마지막 버전
- **2025년 SP 종료, 2027년 EOS**
- 잔존 운영은 x86 Linux 전환 작업 중

Oracle Solaris

- Oracle SPARC + x86 일부
- **Solaris 11.4** — 2034 지원
- ZFS-DTrace-Zones (컨테이너 원조)
- Oracle ULA 묶음 진존
- **신규 도입은 사실상 종료**

FreeBSD

- 오픈소스 BSD 계열
- 사용처
- 네트워크 장비 펌웨어 (Juniper Junos·NetApp ONTAP)
- 일부 보안 어플라이언스 (pfSense·OPNsense)
- 연구·임베디드
- 한국 서버 운영 도입 거의 없음

Linux 운영 콘솔 — *TA의 일상 운영 명령 5종*

```
[root@db-primary ~]# uname -a
Linux db-primary 5.14.0-427.el9.x86_64 #1 SMP PREEMPT_DYNAMIC ... GNU/Linux

[root@db-primary ~]# free -h
              total        used         free      shared  buff/cache   available
Mem:           255Gi        128Gi         126i        2.1Gi        115Gi        124Gi
Swap:           16Gi           0.0Gi         16Gi

[root@db-primary ~]# df -h /data
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg_data-lv_data  4.0T  2.6T  1.4T   65% /data

[root@db-primary ~]# top -bn1 | head -10
top - 14:23:51 up 87 days, load average: 1.42, 1.30, 1.18
Tasks: 423 total,   2 running, 421 sleeping
%Cpu(s): 18.4 us,   3.1 sy,   0.0 ni, 77.9 id,   0.4 wa
MiB Mem: 261072 total, 12544 free, 131072 used, 117456 buff/cache
```

 **실제 사전 참조** (위키미디어/벤더 공식) - [Bash \(Unix shell\) — Wikipedia](#) — Wikipedia - [Red Hat Enterprise Linux 9](#) — Red Hat 공식 - [Rocky Linux 9](#) — Rocky 공식 - [systemd documentation](#) — systemd 공식

```
root@db-primary: ~  
[root@db-primary ~]#  
[root@db-primary ~]# uname -a  
Linux db-primary 5.14.0-427.el9.x86_64 #1 SMP  
PREEMPT_DYNAMIC Wed Mar 13 EST 2026 x86_64  
GNU/Linux  
[root@db-primary ~]# cat /etc/os-release  
NAME="Rocky Linux"  
VERSION="9.4 (Blue Onyx)"  
[root@db-primary ~]# lscpu | head -10  
Architecture: x86_64 / CPU(s): 128  
Model name: Intel(R) Xeon(R) 6960P  
Socket(s): 2 / NUMA node(s): 2  
[root@db-primary ~]# free -h  
Mem: 255Gi total / 128Gi used / 126Gi free  
115Gi buff/cache  
[root@db-primary ~]# df -h /data  
/dev/mapper/vg_data-lv_data 4.0T 2.6T 1.4T 65%  
/data  
[root@db-primary ~]# systemctl status postgresql  
● postgresql-16 --- servic postgresql-16  
Active: active (running) since Mon 2026-03-04  
top - 14:23:51 up 87 days, 4:12, load average:  
1.42, 1.30, 1.18  
Tasks: 482 total, 2 running  
%Cpu(s): 18.4 us, 3.1 sy, 0.0 ni, 77.9 id  
PID USER PR NI VIRT RES %CPU %MEM COMMAND  
1810 postmas 20 0 382M 428 18.0 0.9 postmas  
3541 mongod 20 0 136M 182M 10.7 1.8 mongod  
3979 root 20 0 308K 32M 8.0 1.0 java  
3928 root 20 0 648K 686K 0.0 1.6 java  
3983 root 20 0 566M 110M 0.0 0.8 kworker  
276 root 20 0 80M 50M 0.0 0.0 kworker  
277 root 20 0 25B 55B 0.0 0.0 kworker  
388 root 20 0 20M 30M 0.0 0.0 kworker
```

Linux 운영 콘솔 — TA의 일상 운영 명령 5종

OS 운영 핵심 개념 — 커널 · 파일시스템



커널·시스템콜

- 커널 모드 vs 사용자 모드
- 시스템콜 (syscall) — 응용↔커널 인터페이스
- **Linux 커널** — 모놀리식 (모듈 로딩)
- **Windows NT 커널** — 하이브리드
- 컨테이너 = 호스트 커널 공유
- 가상화 = 게스트 커널 독립



부트 프로세스

- BIOS → POST → MBR (legacy)
- **UEFI** → ESP → bootloader (현 표준)
- Linux: GRUB → kernel → initramfs → systemd
- Windows: bootmgr → winload → kernel



파일시스템 비교

FS	진영	최대 파일	특징
ext4	Linux	16TB	기본·안정
XFS	Linux	16EB	대용량·고성능 (RHEL 기본)
ZFS	Solaris/Linux/BSD	EB급	스냅샷·압축·중복제거·체크섬
Btrfs	Linux	16EB	CoW·스냅샷 (운영 안정성 ↓)
NTFS	Windows	256TB	기본·NTFS 압축·암호화
ReFS	Windows	1YB	스토리지 공간·DataCenter 권장



LVM · LUKS

- **LVM** — Logical Volume Manager (Linux)
- **LUKS** — Linux Unified Key Setup (디스크 암호화)
- 운영 표준: XFS on LVM on LUKS

systemd · cgroup · namespace · 보안 모듈

 **systemd**

- Linux 표준 init 시스템 (2014+)
- 유닛 (service-target-timer-socket-mount)
- `systemctl` · `journalctl`
- 부팅 시간 단축 + 의존성 관리
- 비교: OpenRC (Alpine)·SysV init (구형)

 **cgroup (Control Group)**

- 프로세스 자원 격리 (CPU·Mem·IO·NW)
- cgroup v2** (2022+ 표준)
- 컨테이너 격리의 핵심 기술
- Kubernetes·Docker·Podman 기반

 **namespace**

- 프로세스 가시성 격리 (PID·NW·Mount·UTS·IPC·User)
- 컨테이너의 격리 메커니즘

 **보안 모듈 (LSM)**

- SELinux** (Red Hat) — Mandatory Access Control
- AppArmor** (Ubuntu·SUSE) — Path-based MAC
- Capabilities** — root 권한 세분화 (38종)
- seccomp** — syscall 화이트리스트
- eBPF** — 동적 커널 hook (관찰·보안)

 **TA 결정 포인트**

- SELinux 활성화**가 운영 표준 (Disabled 금지)
- Ubuntu는 AppArmor 기본
- 컨테이너 = cgroup + namespace + LSM + seccomp + capabilities **결합**
- 이게 깨지면 컨테이너 탈출 가능

OS 라이선스 모델 — 공통 합정

 **RHEL Subscription**

- Per-Socket / Per-Core (2024+ 전환 중)**
- Self-Support / Standard / Premium
- VM 단위가 아닌 **물리 호스트** 단위
- 가상화 시 **호스트 RHEL 라이선스 + 게스트 다 포함** 옵션도
- 신규 가격 인상 압박 → Rocky·Alma 전환 추세

Windows Server

- Core Pack** — 2~core × 8 (16C 기본)
- 추가 코어는 2-core 단위 구매
- CAL** — 사용자/디바이스 단위 별도
- DataCenter** — VM 무제한 + Azure Hybrid Benefit
- 가상화 시 호스트 코어 전체 라이선스 필수

Ubuntu Pro

- 무료 (커뮤니티 LTS 5년)
- Pro 구독** — 10년 + ESM + LivePatch + KOS
- 데스크탑·서버 통합
- AI/연구는 무료, 운영 필수면 Pro

SUSE

- 구독 (Per-Socket / Per-Core)
- SAP HANA 묶음
- 라이선스 비싸지만 SAP 표준

Oracle Linux

- OS 자체 무료
- 지원 구독은 유료** (Premier·Network)
- Oracle DB 묶음 시 유리

TA 결정 포인트

- OS 라이선스 ≠ DB 라이선스 ≠ MW 라이선스** — 각각 산정
- 가상화 시 **게스트 라이선스 검토** 필수

OS 결정 매트릭스 — 워크로드별

워크로드	권장 OS	이유
공공 표준 사업	RHEL or Rocky Linux	TTA-국정원 호환·표준
금융 코어 시스템	RHEL (지원 명확)	SLA·인증·24/7 지원
AI/ML 학습·연구	Ubuntu 22.04/24.04 LTS	CUDA·드라이버·SW 풍부
SAP HANA	SUSE SLES for SAP	SAP 인증 표준
Oracle DB	Oracle Linux (UEK) or RHEL	Oracle 인증
AD 도메인·MS SQL	Windows Server 2022/2025 DataCenter	통합 운영
VDI (수천 사용자)	Windows Server 2022 + Citrix or VMware Horizon	표준
컨테이너 호스트	Rocky/Alma or Ubuntu LTS	K8s 표준·경량
K8s 노드 OS	RHCOS / Flatcar / Talos	Immutable OS
레거시 AIX 운영	AIX 7.3 유지 + Linux 마이그레이션 계획	EOL 대비

PART E

DBMS 분류 — RDBMS · NoSQL · NewSQL · 시계열 · 검색 · 벡터

DB는 인프라 비용의 30~50% — TA의 가장 무거운 결정.

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

DBMS 6대 분류 — Wheel

1 RDBMS (관계형)

- ACID-SQL-트랜잭션
- Oracle-MSSQL-MySQL-PostgreSQL-Tibero-MariaDB-DB2
- **인프라 시장 점유율 1위**
- OLTP-OLAP 모두 가능

2 NoSQL

- BASE-스키마리스-분산
- Document-Column-Key-Value-Graph
- MongoDB-Cassandra-Redis-Neo4j
- 빅데이터·세션·메시지

3 NewSQL

- ACID + 분산 + SQL
- TiDB-CockroachDB-YugabyteDB-Google Spanner
- 글로벌 분산 OLTP
- HTAP (Hybrid OLTP/OLAP)

4 시계열 (Time-Series)

- 시간순 메트릭-이벤트
- InfluxDB-TimescaleDB-Prometheus DB-QuestDB
- IoT-NMS-APM-관찰성

5 검색·로그

- 역색인·전문 검색
- Elasticsearch-OpenSearch-Solr
- 로그 통합-SIEM-전자상거래

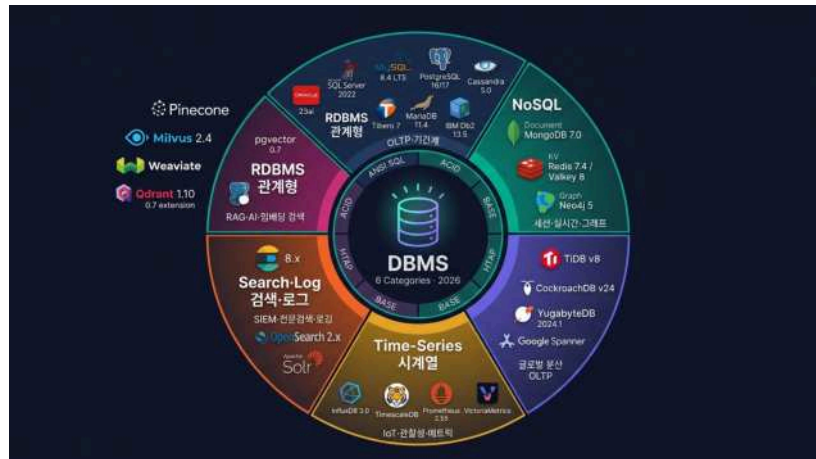
6 벡터 (Vector)

- 임베딩 검색·코사인 유사도
- Pinecone-Weaviate-Milvus-Qdrant-pgvector-ChromaDB
- **RAG·생성형 AI 표준 (2023+)**

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

39 / 103

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너



DBMS 6대 분류 wheel (삽화)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

40 / 103

RDBMS ① Oracle — 코어뱅크의 황제



Oracle Database

- 19c (LTS, ~2032) · 21c (Innovation) · 23ai (2024)
- Editions: SE2 · EE · EE + Options
- 옵션: RAC·Active Data Guard·Partitioning·GoldenGate·Multitenant
- 라이선스: **Per-Core x Factor (Intel x86 = 0.5)**
- Named User Plus (NUP) 별도



HA 구성

- **RAC** (Real Application Cluster) — Active-Active, 공유 스토리지
- **Data Guard** — 물리/논리 Standby
- **GoldenGate** — 이기종 복제·CDC
- **Multitenant (CDB/PDB)** — 통합 관리



TA 결정 포인트

- **라이선스가 가장 비싼 DB** — 코어당 수천만 원
- EE 코어 1개 ≈ 4,000~5,000만 원 (2026 시세, +Support 22%)
- 옵션 (RAC·Partitioning) 별도 라이선스
- **CPU 코어 수가 곧 비용** → 고클럭·저코어 SKU 선택
- **VMware 호스트 = 호스트 전 코어 라이선스** (Oracle Hard Partitioning 정책)
- 대안: Tibero (국산)·PostgreSQL (오픈)



한국 도입

- 공공·금융·대기업 코어 표준
- 2010년대 후반부터 Tibero·PostgreSQL 대체 가속

RDBMS ② MSSQL · ③ MySQL



Microsoft SQL Server

- 2019 (CU 30+) · 2022 (현재 표준) · 2025 (출시 임박)
- Editions: Express·Standard·Enterprise·Developer
- 라이선스: **Per-Core (4-core 단위)**
- Enterprise Per-Core ≈ 1,500만 원 (2026)
- HA: **Always On AG (Availability Group)** — 2 동기 + 다수 비동기
- Failover Cluster Instance (FCI) — 공유 스토리지

TA 메모

- Windows AD/MS Office 환경에 자연스러움
- Linux 지원 (2017+) but 운영 표준은 Windows
- Power BI·SSAS·SSIS 통합



MySQL

- 8.0 (현재 표준) · 8.4 LTS (2024+)
- Editions: Community (GPL)·Enterprise (Oracle)
- Storage Engine: **InnoDB** (표준)·MyISAM (구형)
- HA 옵션
- **InnoDB Cluster** (8.0+) — Group Replication + MySQL Router
- **Group Replication** — Multi-Primary·Single-Primary
- **MySQL Replication** — 비동기·반동기
- **Percona XtraDB Cluster** (Galera 기반)

TA 메모

- 웹·중소 규모 OLTP 표준
- 라이선스: Community 무료, Enterprise 유료
- MariaDB로 갈라짐 (MySQL 호환 fork)

RDBMS ④ PostgreSQL — 오픈소스 챔피언



PostgreSQL

- 16 (2023.09) · 17 (2024.09) — 메이저 매년
- 진정한 오픈소스 (PostgreSQL License = BSD 호환)
- ACID·JSON-전문검색·지리정보 (PostGIS)
- 확장 (Extension) 풍부 — TimescaleDB·Citus·pgvector
- 공공·금융 오픈소스 RDBMS 1순위

HA 하자

TA 결정 포인트

- **Oracle 대체 1순위** — 라이선스 무료
- DBA 인력 수급 (Oracle DBA보다 PostgreSQL DBA 부족)
- 상용 지원: **EDB Postgres-2ndQuadrant·Crunchy Data**
- 한국 지원: EnterpriseDB Korea·Bitnine·아이씨엔
- **공공 마이그레이션** — TmaxSoft Tibero ↔ PostgreSQL 경쟁 도입 독한 (2026)

RDBMS ⑤ Tibero · ⑥ MariaDB · ⑦ DB2

 **Tibero (TmaxSoft)**

- 국산 **Oracle** 호환 **DBMS**
- Tibero 7 (2023) · Tibero TAC (RAC 호환)
- **공공 SW 진흥법** 우대 (국산 SW 가산점)
- 라이선스: 코어 단위 (Oracle 대비 30~50%)
- 호환: SQL-PL/SQL 95%+ 호환
- HA: **TAC** (Tibero Active Cluster)
- 한국 공공·금융 점유 ↑↑ (2020~)

 **TA 메모**

- 공공 신규 RFP에서 Oracle vs Tibero 비교 빈번
- DBA·운영 도구 성숙도 개선 중
- 한국 외 사용 거의 없음

 **MariaDB**

- MySQL 5.5 fork (2009)
- **MariaDB 10.11 LTS · 11.x**
- ColumnStore·MaxScale·Xpand
- **Galera Cluster** — Multi-Master 동기 복제
- MySQL 호환 (8.0+는 분기 심해짐)

TA 메모

- RHEL/Ubuntu 기본 패키지
- 웹·중소 OLTP 무료 대안
- 유럽·중국 점유 ↑
- 한국은 MySQL이 우세, MariaDB 일부

 **IBM DB2**

- **DB2 12 (z/OS) · DB2 11.5 (LUW)**
- Mainframe 표준 (코어뱅킹·통신 BSS)
- Power AIX + DB2 = 금융 잔존
- pureScale (RAC 유사)
- 한국 도입: 은행 코어·일부 공공
- 신규 도입은 거의 없음, 유지 운영 중심

NoSQL 4종 — Document · Column · KV · Graph

 **Document**

- JSON/BSON 문서 단위
- **MongoDB** (상용 + 오픈)
- **CouchDB** (오픈)
- Amazon DocumentDB (MongoDB 호환)
- 사용처: 카탈로그·콘텐츠·세션·로그

 **Column (Wide-Column)**

- 행보다 열 단위 저장 (분석 효율)
- **Cassandra** (분산 마스터리스)
- **HBase** (Hadoop 위)
- **ScyllaDB** (Cassandra 호환·C++)
- 사용처: 시계열·이벤트·IoT·메시징

 **Key-Value**

- 단순 K:V 매핑
- **Redis** (인메모리 + AOF/RDB)
- **Memcached** (순수 캐시)
- **DynamoDB** (AWS 매니지드)
- **etcd** (Raft·K8s 사용)
- 사용처: 세션·캐시·랭킹·큐

 **Graph**

- 노드·엣지·속성 모델
- **Neo4j** (Cypher)
- **OrientDB · ArangoDB** (멀티 모델)
- **JanusGraph · TigerGraph**
- 사용처: 추천·SNS·사기탐지·지식그래프

NoSQL 도입 원칙: NoSQL은 RDBMS의 대체가 아닌 보완. 메시지·세션·로그·캐시 등 **부분 워크로드**에 도입. RDBMS를 폐지하고 NoSQL만 운영하는 사례는 드물.

NewSQL — ACID + 분산 + SQL

NEW NewSQL이란

- **ACID 보장 + 수평 확장 + SQL**
- RDBMS의 트랜잭션 + NoSQL의 확장성
- **분산 트랜잭션 (Distributed Tx)**
- 글로벌 분산 (Multi-Region)
- HTAP (OLTP + OLAP 동시)

🔑 대표 제품

- **TiDB** (PingCAP, 중국) — MySQL 호환
- **CockroachDB** (Cockroach Labs) — PostgreSQL 호환
- **YugabyteDB** — PostgreSQL 호환
- **Google Spanner** (GCP 매니지드)
- **VoltDB** — 인메모리 OLTP
- **Vitess** — MySQL 샤딩 (Slack·YouTube)

🎤 TA 결정 포인트

- 글로벌 분산이 필요한 경우 (다중 리전 OLTP)
- 단일 RDBMS의 한계 (수평 확장)
- 한국 도입: 아직 초기·일부 게임·SaaS
- 운영 난이도·인력 ↑↑↑
- 공공·금융은 사용 거의 없음

📊 한국 시장 (2026)

- TiDB: 일부 게임·핀테크
- CockroachDB: SaaS·스타트업
- YugabyteDB: 일부 대기업 PoC
- **본격 확산 전 단계**

시계열 · 검색 DB — 관찰성·로그의 표준

🕒 시계열 (Time-Series)

- **InfluxDB** (TICK Stack)
- **TimescaleDB** (PostgreSQL 확장)
- **Prometheus TSDB** (모니터링 표준)
- **QuestDB·VictoriaMetrics·Mimir**
- 특징
- 시간순 데이터 압축 효율
- 다운샘플링·연속 쿼리
- 만료 (TTL) 정책
- 사용처
- APM·메트릭·NMS
- IoT 센서·산업 데이터
- 금융 시세

🔍 검색·로그

- **Elasticsearch** (Elastic SSPL)
- **OpenSearch** (AWS·Apache 2.0 fork)
- **Solr** (Apache Lucene)
- **MeiliSearch·Typesense** (경량)
- 특징
- 역색인 (Inverted Index)
- 전문 검색·집계
- 분산 샤딩
- 사용처
- 사이트 검색·전자상거래
- 로그 통합 (ELK Stack)
- SIEM (Splunk 대체)

벡터 DB — RAG·AI의 표준 (2023+)

🧠 벡터 DB란

- 임베딩 벡터 (수백~수천 차원) 저장·검색
- 유사도 검색 — 코사인·내적·Euclidean
- ANN (Approximate Nearest Neighbor) — HNSW·IVF·PQ
- RAG (Retrieval-Augmented Generation) 표준 인프라
- 메타데이터 필터링 + 벡터 검색

🔑 대표 제품

- Pinecone (SaaS) — 매니지드 표준
- Weaviate — 오픈 + 클라우드
- Milvus / Zilliz — 오픈 (Apache 2.0)
- Qdrant — 오픈 + 클라우드 (Rust)
- ChromaDB — 임베디드·경량 (Python)
- pgvector — PostgreSQL 확장 (운영 친화)

🎤 TA 결정 포인트

- RAG·AI 도입 시 필수 — 2024+ 신규 사업 빈번
- 데이터 규모 < 1M = pgvector (PostgreSQL 통합)
- 1M ~ 100M = Milvus·Qdrant·Weaviate (전용)
- 100M+ = Milvus 클러스터·Pinecone (매니지드)
- GPU 메모리 + 인덱스 메모리 산정 필수
- 임베딩 차원 × 4B × 벡터 수 = 메모리 추정

📊 도입 동향

- 공공 RAG 챗봇 (FAQ·민원)
- 금융 RAG (지점 매뉴얼·규정)
- 대학 LMS (수업 자료 검색)
- 의료·법률 자료 검색

DB 결정 매트릭스 — 워크로드별

워크로드	권장 DBMS	이유
금융 코어 OLTP	Oracle EE + RAC + DG	검증·SLA·인력
공공 신규 OLTP	PostgreSQL + Patroni / Tiberio TAC	라이선스·국산 가산점
MSSQL/AD 환경	MS SQL 2022 Always On AG	통합·도구·라이선스
웹 서비스 OLTP	MySQL 8.0 InnoDB Cluster / MariaDB Galera	가벼움·무료
OLAP·BI·DW	PostgreSQL + Citus / ClickHouse / Snowflake	컬럼 저장·집계
HTAP	TiDB / YugabyteDB / Oracle 23ai	분산 OLTP+분석
세션·캐시	Redis Cluster	인메모리·LRU·Pub/Sub
메시지·이벤트 큐	Kafka (DB 외)·NATS·RabbitMQ	비동기
카탈로그·콘텐츠	MongoDB Replica Set	스키마리스·JSON
시계열·IoT·메트릭	TimescaleDB / InfluxDB / Prometheus	압축·다문샘플
로그·전문검색·SIEM	Elasticsearch / OpenSearch	역색인·집계
RAG·AI 임베딩	pgvector / Milvus / Qdrant / Pinecone	ANN·메타 필터
글로벌 분산 OLTP	CockroachDB / Spanner / YugabyteDB	다중 리전
그래프·추천	Neo4j	Cypher·노드/엣지

ACID vs BASE — 트랜잭션 철학

⚖️ ACID (RDBMS 전통)

- **A**tomicity — 원자성 (All or Nothing)
- **C**onsistency — 일관성 (불변식 보존)
- **I**solation — 격리성 (동시 트랜잭션 분리)
- **D**urability — 영속성 (커밋 후 보존)

적합 - 금융 거래·예약·재고 - 정확성이 절대 우선 - RDBMS·NewSQL

🌊 BASE (NoSQL 전통)

- **B**asically **A**vailable — 기본 가용
- **S**oft state — 상태 가변
- **E**ventual consistency — 최종 일관성

적합 - 소셜·콘텐츠·로그 - 가용성·확장이 우선 - NoSQL·일부 NewSQL

 **CAP 정리**

- **C**onsistency · **A**vailability · **P**artition tolerance
- 네트워크 분할 시 **C와 A 중 택일**
- CP: HBase·MongoDB (강한 일관성)
- AP: Cassandra·DynamoDB (가용성)

 **PACELC**

- 분할 시: P → A or C
- 정상 시: 지연 (L) vs 일관성 (C)
- 더 현실적 모델

DB 라이선스 비교 — 비용 산정

DBMS	모델	단가 (참고, 2026)	비고
Oracle EE	Per-Core x Factor	4,000~5,000만원/코어 (+22% 유지보수)	x86 Factor 0.5, 옵션 별도
Oracle SE2	Per-Socket (최대 2S)	2,500만원/소켓	한정 기능
MS SQL EE	Per-Core (4C 단위)	1,500~1,800만원/코어	DataCenter 별도
MS SQL Std	Per-Core or Server+CAL	약 600만원/코어	중소 환경
Tibero EE	Per-Core	Oracle 대비 30~50%	공공 가산점
PostgreSQL	무료 (BSD 호환)	0원 + 상용지원 (EDB)	신규 표준 후보
MySQL Community	무료 (GPL)	0원 + Enterprise 옵션	웹 표준
MariaDB Enterprise	Subscription	노드당 수백만원	유럽·중국
DB2	Per-PVU (IBM 가중치)	협상가	잔존 운영
MongoDB Enterprise	Per-Node	노드당 수천만원	Community 무료

Oracle 라이선스 합정: ① VMware 호스트 = **전 코어 라이선스** (Hard Partitioning 정책) ② **NUP 최소 수** (CPU 코어 × 25) ③ **옵션** (RAC·Partitioning·Active DG)별로 별도 라이선스. 단순 코어 단가만 보면 BOM 실수.

DB HA 패턴 — 공통 분류



Active-Standby (Cold/Warm/Hot)

- **Cold** — Standby 정지, 장애 시 부팅
- **Warm** — Standby 부팅·서비스 미오픈
- **Hot** — Standby 서비스 오픈 (읽기 가능)
- 대표: MS SQL FCI·Oracle DG·MySQL Replication

Active-Active

- 모든 노드가 쓰기 가능
- 강한 일관성 또는 결합/결합거부 충돌
- 대표: Oracle RAC·MySQL Group Replication·Galera

Shared-Storage HA

- 공유 디스크에 두 노드가 파일오버
- 대표: Oracle RAC·MSCS·Pacemaker·VCS

Shared-Nothing 분산

- 각 노드 자체 디스크, 데이터 샤딩·복제
- 대표: Cassandra·MongoDB·CockroachDB



Quorum·Witness

- 짝수 노드 → 스플릿 브레인 위험
- 홀수 노드 또는 **Witness/Arbiter** 추가
- etcd·Consul·Zookeeper 활용

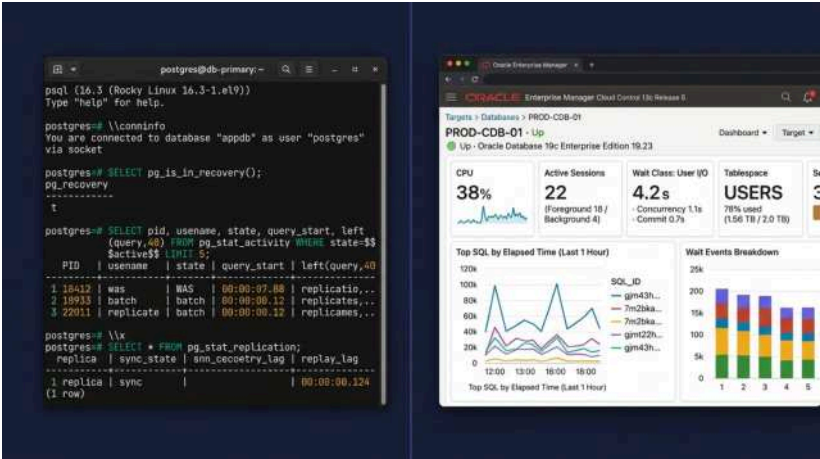
DB 운영 콘솔 — *psql 세션 + Oracle Enterprise Manager*

운영 TA가 일상적으로 사용하는 psql / sqlplus 명령과, DBA가 보는 OEM 대시보드.

```
postgres=# SELECT pg_is_in_recovery(), pg_last_wal_replay_lsn();
 pg_is_in_recovery | pg_last_wal_replay_lsn
-----
t                  | 1A/3F8E2B40
(1 row)

postgres=# SELECT pid, username, application_name, state, query_start
FROM pg_stat_activity
WHERE state <> 'idle'
ORDER BY query_start
LIMIT 5;
 pid | username | application_name | state |      query_start
-----+-----
23104 | app_wr   | spring-boot-was  | active | 14:23:51.214+09
23187 | app_ro   | report-batch     | active | 14:23:52.482+09
23211 | repl    | walreceiver      | idle in transaction | 14:23:53.012+09
```

실제 사전 참조 (벤더·오픈소스 공식) - [PostgreSQL psql 공식 문서](#) — PostgreSQL 공식 - [Oracle Enterprise Manager Cloud Control](#) — Oracle 공식 - [pgAdmin 4 데모](#) — pgAdmin 공식 - [MySQL Workbench](#) — Oracle/MySQL 공식 - [Patroni 공식 문서](#) — PostgreSQL HA



DB 운영 콘솔 — psql 세션 + Oracle Enterprise Manager

PART F

GPU · AI 서버 — 2026 핫 토픽

AI 시대의 컴퓨트 표준 — GPU 인프라.

AI 시대 GPU 인프라 — 왜 폭증하는가



폭증의 배경

- **LLM 학습 폭증** — GPT·Claude·Gemini·LLaMA
- **온프레미스 RAG** — 데이터 주권·민감 정보
- **국산 LLM** — 네이버 HyperCLOVA-KT 말·음·LG EXAONE
- **공공 AI** — 국세청 RAG·교육부 LMS·각 부처 챗봇
- **금융 AI** — 신용평가·CS·법규 검색



인프라 도전

- **전력** — H100 1대 = 700W, 8GPU 서버 = 6~10 kW
- **냉각** — 공냉 한계, **Liquid Cooling** 필수화
- **NW** — InfiniBand NDR 400G·NVL72 1.8TB/s
- **라이선스** — vGPU·CUDA·NCCL
- **공급 부족** — H100/H200 리드타임 12~24주



TA의 새 책임

- **AI 워크로드 사이징** — 모델·배치·시퀀스
- **GPU 선정** — 학습 vs 추론
- **VRAM·메모리 대역** — 모델 크기 결정
- **클러스터 토폴로지** — NVLink·IB·Spine·Leaf
- **MIG 분할** — A100/H100 1대 → 7 인스턴스
- **냉각·전력 설계** — IDC 협의
- **라이선스 비용** — vGPU·CUDA Enterprise
- **PoC → 본 사업 단계 분리**



케이스 매핑

케이스 #4 명지대 AI LMS — RAG·실시간 통번역 **케이스 #5 AI 데이터 학습 인프라**

NVIDIA GPU 라인업 — L4 → H200 → B200

GPU	용도	VRAM	FP16/BF16	전력	출시	비고
L4	추론·VDI	24GB GDDR6	121 TFLOPS	72W	2023	1U 8대 가능
L40S	추론·중규모 학습	48GB GDDR6	362 TFLOPS	350W	2023	멀티 워크로드
A30	추론	24GB HBM2	165 TFLOPS	165W	2021	MIG 지원
A100 SXM	학습	40/80GB HBM2e	312 TFLOPS	400W	2020	MIG 7개 분할
H100 SXM	학습	80GB HBM3	989 TFLOPS	700W	2022	NVLink 900GB/s
H200 SXM	학습·추론	141GB HBM3e	989 TFLOPS	700W	2024.Q2	LLM 추론 강점
B100	학습·추론	192GB HBM3e	1.8 PFLOPS	700W	2024	공냉
B200	학습·추론	192GB HBM3e	2.25 PFLOPS	1000W	2024.Q4	액냉
GB200	학습·추론	384GB (HBM3e × 2)	4.5 PFLOPS	2700W (2 GPU + Grace)	2024.Q4	NVL72

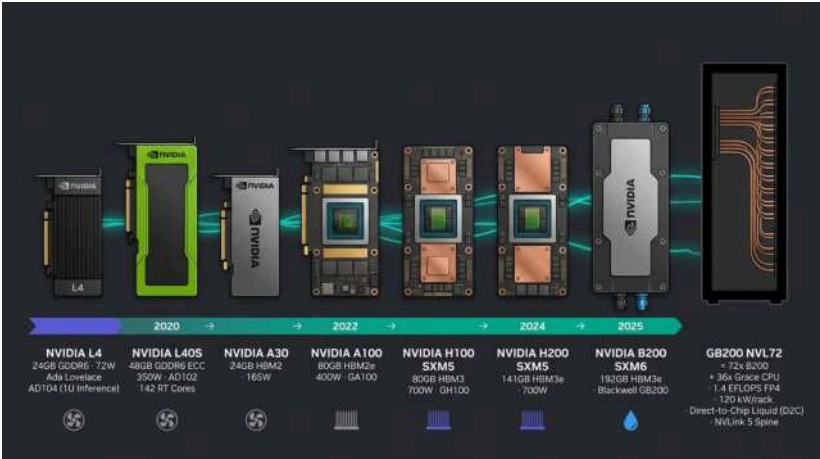


시스템 폼팩터

- **PCIe** — 1U/2U 일반 서버 (L4·L40S·A100)
- **SXM (Module)** — HGX 보드 8GPU (A100·H100·H200·B200)
- **HGX** — 8GPU + NVSwitch 보드
- **DGX** — NVIDIA 공식 서버 (HGX + 호스트 CPU)
- **MGX** — OEM 모듈식 (Dell·Supermicro)
- **NVL72** — 72 GB200 + 36 Grace CPU 랙 시스템

선택 가이드

- **추론·VDI** → L4 / L40S (저전력·다수)
- **중규모 학습·추론** → A100 80G / H100
- **대규모 LLM 추론** → H200 (141GB VRAM)
- **신규 대규모 학습** → B200 / GB200 NVL72
- **국내 공공** → H100/H200 PCIe (공급 가능 우선)



NVIDIA GPU 라인업 (삽화)

비-NVIDIA 가속기 — AMD · Intel · 국산

AMD Instinct

- **MI210** (2022) — 64GB HBM2e
- **MI300X** (2023.12) — 192GB HBM3, FP16 1.3 PFLOPS
- **MI325X** (2024.Q4) — 256GB HBM3e
- **MI350** (2025+) — CDNA 4
- **ROCm 6.x** — 오픈소스 (CUDA 대안)
- 강점: VRAM 용량 (H100 대비 2.4x)
- 도입 사례: Microsoft Copilot·Meta·일부 클라우드

Intel Gaudi

- **Gaudi 2** (2022) — 96GB HBM2e
- **Gaudi 3** (2024.Q4) — 128GB HBM2e
- **OneAPI / SynapseAI**
- 강점: 가격 경쟁력 (H100 대비 ~50%)
- 도입 사례: Naver·IBM 일부

클라우드 자체 칩

- **Google TPU v5p / Trillium**
- **AWS Trainium 2 · Inferentia 2**
- **Azure Maia 100**
- 공통: 클라우드 전용·온프레미스 불가

국산

- **사파온 X220 / X330 / X430**
- SK 그룹·KT·NHN 도입
- 추론 가속·국산 가산점
- **리벨리온 아톰 / 리벨**
- 추론 가속·NHN·KT
- **퓨리오사·AI RNGD**
- 차세대 추론 (2024+)
- **국산 가산점** — 공공 SW 진흥법

TA 메모

- 국산 가속기는 **추론 중심**
- 학습은 여전히 NVIDIA 사실상 표준
- 공공 사업에서 국산 가산점 검토 빈번

GPU 가상화 — vGPU · MIG · SR-IOV

 **NVIDIA vGPU**

- **Time-Sliced Virtual GPU**
- 하나의 GPU를 다수 VM에 공유
- **vGPU 프로파일** — Q (CAD)-B (VDI)-A (App)-C (Compute)
- 라이선스 별도
- vWS (가상 워크스테이션)
- vCS (Compute Server)
- vApps (RDS)
- 호환: T4-A40-L40-L40S-A100-H100-H200
- VMware vSphere-Citrix-Nutanix AHV-KVM

 **MIG (Multi-Instance GPU)**

- A100-H100-H200에 한정 (B 시리즈도 지원)
- **하드웨어 수준 분할** (vGPU와 다름)
- 1 GPU → 최대 **7 인스턴스**
- 각 인스턴스 = 독립된 메모리·연산·NVLink
- 동일 GPU 안에서 다수 모델 동시 실행
- 라이선스 vGPU와 별개

 **SR-IOV (Single-Root IO Virt.)**

- PCIe 표준
- GPU-NIC를 VM에 직접 패스투루
- NVIDIA-AMD GPU에서 지원
- VMware DirectPath I/O / KVM PCI passthrough

 **사용 패턴**

- **VDI** = vGPU (Q/B 프로파일)
- **추론 다중 모델** = MIG
- **소규모 학습** = 1 VM = 1 GPU (Passthrough)
- **대규모 학습** = Bare-Metal + NVLink

 **라이선스 함정**

- vGPU 라이선스 = GPU당·연간
- vWS 1년 = 약 50~70만원/GPU
- vCS = vWS 1.5~2배
- 사용자 수 × 동시 연결 = 비용

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

59 / 103

#206

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

GPU 소프트웨어 스택 — CUDA · ROCm · OneAPI · NCCL

SW	벤더	라이브러리	비고
CUDA	NVIDIA	cuDNN · cuBLAS · TensorRT · NCCL	사실상 표준
ROCm	AMD	MIOpen · rocBLAS · RCCL	CUDA 대안
OneAPI	Intel	oneDNN · oneCCL · SynapseAI	Gaudi 통합
NCCL	NVIDIA	멀티 GPU 통신 (All-Reduce)	학습 필수
NVSHMEM	NVIDIA	GPU 간 분산 메모리	NVL72
Triton	NVIDIA	추론 서버	다중 모델-MIG

 **프레임워크 매핑**

- **PyTorch** — CUDA·ROCm·MPS·CPU
- **TensorFlow** — CUDA·ROCm·TPU
- **JAX** — CUDA·TPU·CPU
- **vLLM·TensorRT·LLM** — 추론 가속

 **컨테이너**

- **NGC** (NVIDIA GPU Cloud) — 공식 이미지
- **AMD Infinity Hub** — ROCm 이미지
- **NVIDIA Container Toolkit** — Docker/K8s 통합

 **TA 결정 포인트**

- CUDA 의존성 — 대부분의 AI SW는 CUDA 우선
- 학습 시 NCCL이 통신 병목
- **InfiniBand + NCCL + GPUDirect RDMA** = 표준
- K8s 통합 = NVIDIA GPU Operator
- vGPU + Container 모두 지원

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

60 / 103

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 103 / 155

InfiniBand · NVLink · Fabric — GPU 클러스터 통신

NVLink (GPU↔GPU 직결)

- **NVLink 4.0** (H100) — 900 GB/s (GPU 18 link)
- **NVLink 5.0** (B200) — 1.8 TB/s
- **NVSwitch** — 8 GPU 완전 메시
- **NVL72** — 72 GPU + 36 Grace, NVLink 1.8TB/s × 72
- 한 노드 내 NVL72 랙 내 (동일 도메인)

InfiniBand (노드 간)

- **HDR 200G** (2018) — 운영 진존
- **NDR 400G** (2022) — 현재 표준
- **XDR 800G** (2024) — 신규 클러스터
- Mellanox (NVIDIA 인수) 독점
- RDMA-SHARP-GPUDirect

Spectrum-X (Ethernet 대안)

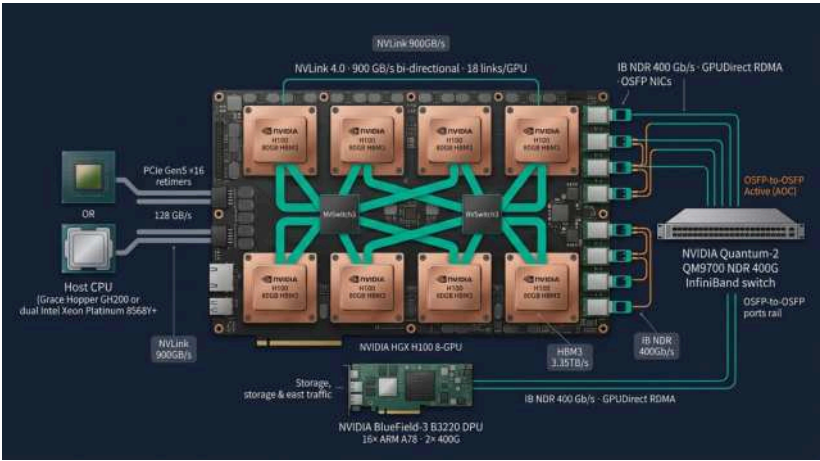
- NVIDIA Ethernet AI 패브릭
- 400/800G Spectrum-4 스위치
- BlueField-3 DPU
- RoCE v2 + Adaptive Routing + Congestion Control
- IB 대비 호환성·확장성 ↑

토폴로지

- 단일 노드 = NVLink 8 GPU
- 소규모 (16~64 GPU) = NVLink + IB 1단
- 중규모 (수백 GPU) = Spine-Leaf IB 2단
- 대규모 (수천 GPU) = Fat-Tree-Rail-Optimized

비용

- HGX H100 8GPU 서버 ≈ 4~5억 원
- NDR 400G 스위치 1대 ≈ 1~2억 원
- DGX H100 8GPU ≈ 5~6억 원



HGX 8 GPU 토폴로지 (삽화)

GPU 클러스터 토폴로지 — *HGX·DGX·NVL72*



HGX 8 GPU

- NVIDIA HGX 표준 보드
- 8 SXM GPU + 4 NVSwitch
- 호스트 CPU = OEM 선택 (Xeon·EPYC)
- 4U or 6U 서버
- 벤더: Supermicro·Dell·HPE·Lenovo·QCT
- **가격 약 4~5억원/대**

DGX (NVIDIA 공식)

- **DGX H100** — HGX H100 + Xeon Platinum
- **DGX H200** — HGX H200 (141GB)
- **DGX B200** (2024.Q4) — HGX B200
- **DGX SuperPOD** — DGX × 32 + IB
- NVIDIA Enterprise Support
- 가격 ↑ but 검증·지원 ↑

NVL72 (2024.Q4)

- **72 GB200 + 36 Grace** 1 랙
- NVLink 5.0 1.8TB/s × 72 (완전 메시)
- **액냉 (Liquid Cooling)** 필수
- 130 kW/랙 (일반 IDC 6~10× 부하)
- 1 NVL72 = LLM 학습 전체
- 가격: 수십~수백 억



SuperPOD·SuperCluster

- DGX 32~256개 + IB + 분산 스토리지
- xAI Colossus (100k GPU)·Meta·MS



TA 결정 포인트

- **단일 노드** (8 GPU) — 추론·소규모
- **NVL72** — 대규모 학습 (현재 가장 효율)
- **DGX SuperPOD** — 검증·지원 우선
- **MGX (OEM)** — 비용 ↓·유연성



산정 매트릭스

- 모델 학습: 70B → DGX H100 × 4
- 100B → DGX H100 × 16
- 1T → NVL72 × 1
- 추론: VRAM × 1.2 (KV Cache)

≤ IDC 요건

- 일반 IDC = HGX 4~8대
- AI IDC = NVL72 16대
- 액냉 + 130 kW/랙 표준화 필요

GPU 산정 방법 — *모델 → GPU 수*

학습 GPU 산정

- 모델 파라미터 P (billion)
- 각 파라미터 = 2~4 bytes (BF16)
- **그라디언트 + Adam Optimizer = 16~20× P bytes**
- 예: 70B 모델 → ~1.2 TB 메모리 필요
- → H100 80G × 16개 (NVLink + IB)
- 학습 FLOPS = 6 × P × Tokens
- 7B × 1T tokens → ~4e22 FLOPS
- → H100 8GPU × 1년 (이상적)

추론 GPU 산정

- 모델 메모리 = P × 2 bytes (BF16) or P × 1 byte (INT8)
- **KV Cache** = 2 × layers × heads × dim × seq_len × batch × 2 bytes
- 예: 70B BF16 + KV Cache 8K → 165GB → H200 1대 (141GB ×2)
- 70B INT8 → 70GB → H100 80G 1대 충분



TFLOPS 비교 (FP16/BF16)

- L4: 121 TFLOPS
- L40S: 362 TFLOPS
- A100: 312 TFLOPS
- H100: 989 TFLOPS
- H200: 989 TFLOPS (메모리 ↑)
- B200: 2,250 TFLOPS
- GB200: 4,500 TFLOPS

실전 매트릭스

사용처	모델	권장 GPU
챗봇 추론 (CS)	7~13B	L40S × 2 / A100 × 1
사내 RAG	30~70B	H100 × 2 / H200 × 1
한국어 LLM 추론	70B	H200 × 2
학습 (Fine-tune 70B)	LoRA	H100 × 8 (1 HGX)
학습 (Pre-train 70B+)	처음부터	DGX H100 × 16+
학습 (1T+)	Foundation	NVL72

PART G

가상화 — *Type-1 · Type-2 · 라이선스*

Broadcom × VMware 인수 이후의 새 표준.

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

가상화 기본 — *Type-1 vs Type-2 Hypervisor*

⚙️ Type-1 (Bare-Metal)

- 하드웨어 위에 **직접** 설치
- Host OS 없음
- 데이터센터 표준
- 성능·효율·격리 최고

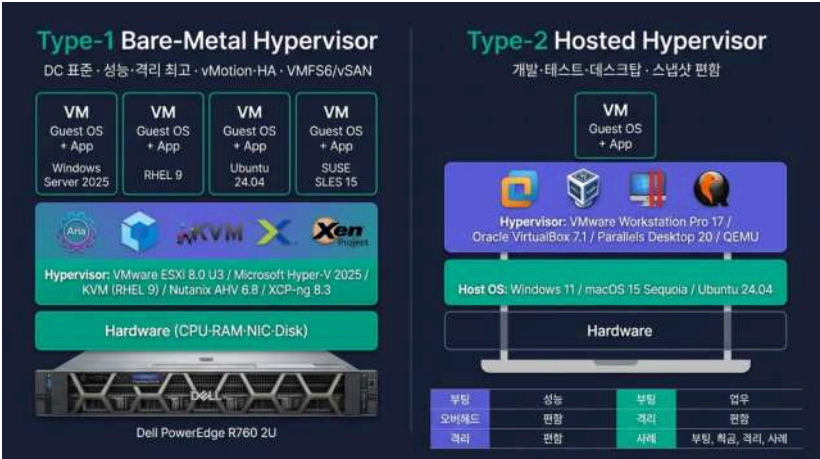
대표 - VMware ESXi - Microsoft Hyper-V Server - Red Hat KVM / oVirt - Citrix XenServer / Hypervisor - Nutanix AHV - Proxmox VE - Oracle VM Server (Xen 기반)

💻 Type-2 (Hosted)

- **Host OS** 위에 설치
- 개발자·테스트·데스크탑 용도
- 데이터센터 부적합

대표 - VMware Workstation / Fusion - Oracle VirtualBox - Parallels Desktop (Mac) - QEMU (단독 사용 시) - Microsoft Hyper-V (클라이언트 Windows)

구분 점: KVM은 Linux 커널의 일부 → 엄밀히는 Type-1.5로 분류되기도. 그러나 운영적으로는 Type-1로 취급. Hyper-V도 설치 후 호스트 OS가 게스트로 전환되므로 Type-1.



Type-1 vs Type-2 (삼화)

Type-1 Hypervisor 6종 — 데이터센터 표준 비교

제품	모기업	라이선스	강점	약점
VMware vSphere ESXi	Broadcom	Per-Core (구독)	기능·생태계·인력 풀	2024 가격 폭등·변동화
Microsoft Hyper-V	Microsoft	Windows Server 포함	Windows 통합·AD·VDI	Linux 게스트 약간 떨어짐
Red Hat KVM (RHV/oVirt)	Red Hat (IBM)	RHV 단종 (2024.Q1)-OpenShift Virt.	오픈-KVM 기반	운영 도구 격차
Nutanix AHV	Nutanix	Nutanix Cloud Platform 포함	HCI 통합·VMware 대안 1순위	HCI 종속
Citrix Hypervisor	Cloud Software Group	Citrix XenApp/XenDesktop 포함	VDI 통합	점유율 ↓
Proxmox VE	Proxmox Server Solutions	무료 + 구독 (지원)	오픈-웹 UI-KVM 기반	엔터프라이즈 인력·도구 ↓

2024~26 최대 변화: Broadcom의 VMware 인수 (2023.11) → 라이선스 모델 전면 개편 (Per-Core·구독·번들 묶음). VMware vSphere 단독 구매 불가, vSphere Foundation·Cloud Foundation 묶음만 가능. 가격 평균 2~5배 인상. → Nutanix·Proxmox·OpenShift Virt 대안 검토 폭증.

VMware vSphere — 기능·라이선스 (2024+ 변화)

 **vSphere 핵심 기능**

- **vMotion** — 무중단 VM 이동
- **DRS** (Distributed Resource Scheduler) — 자원 자동 균형
- **HA** (High Availability) — 호스트 장애 시 VM 재시작
- **FT** (Fault Tolerance) — 동일 상태 미러링 VM
- **DPM** — 전력 관리
- **vSAN** — 분산 스토리지 (HCI)
- **NSX** — 가상 네트워크·보안
- **vRealize·Aria** — 운영·자동화·로그

 **vCenter Server**

- 다수 ESXi 호스트 통합 관리
- 클러스터·리소스 풀·템플릿

 **2024+ 라이선스 변화**

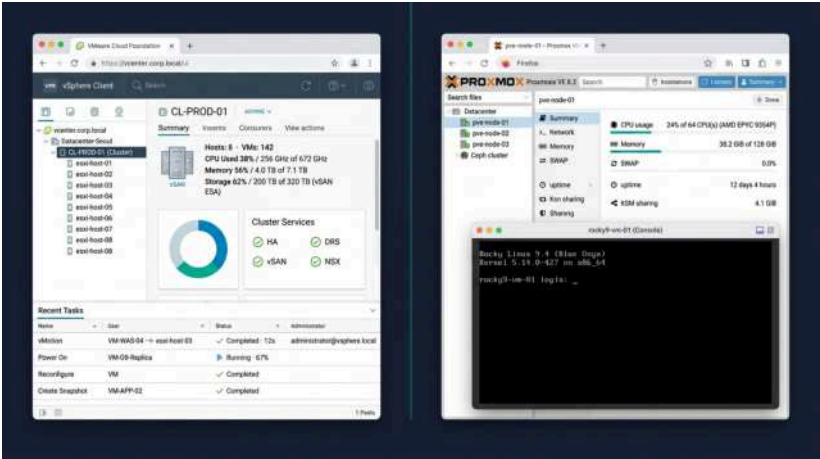
- **단종된 SKU**: vSphere Standard 단독, Essentials Plus
- **현재 SKU 5종**
- vSphere Standard (16C 최소·일부 기능)
- **vSphere Foundation (VVF)** — 표준 묶음
- **VMware Cloud Foundation (VCF)** — 풀스택 (vSphere+vSAN+NSX+Aria)
- vSphere Essentials Plus (3-host SMB)
- vSphere Enterprise Plus (구버전 운영 잔존)
- **모두 Per-Core (16C 최소) + 구독**
- 평균 인상 2~5×

 **TA의 대안 검토**

- **유지**: VCF 묶음 + 가격 협상
- **부분 이전**: 신규는 Nutanix·Proxmox·OpenShift Virt
- **완전 이전**: 5년 로드맵

가상화 관리 콘솔 — vSphere Client + Proxmox VE 화면

 **실제 사진 참조** (벤더 공식) - [VMware vSphere Client 8](#) — VMware (Broadcom) 공식 - [VMware Hands-on Labs](#) — 실제 UI 체험 환경 - [Proxmox VE 8 — Web Interface](#) — Proxmox 공식 - [Nutanix Prism Element](#) — Nutanix 공식 - [Microsoft Hyper-V Manager](#) — Microsoft 공식



가상화 관리 콘솔 — vSphere Client + Proxmox VE 화면

가상화 결정 매트릭스 — 상황별

상황	권장	이유
대규모 VMware 운영 (수백 VM+)	VCF 유지 + 5년 로드맵	이전 비용·인력
VMware 신규 도입 검토	Nutanix AHV / Proxmox VE	라이선스 부담 회피
Linux 위주 환경	KVM (oVirt) / OpenShift Virt	오픈·통합
Windows AD 위주	Hyper-V (DataCenter 포함)	라이선스 효율
HCI 도입	Nutanix / VxRail (VCF)	통합·자동화
VDI 중심	Citrix Hypervisor / VMware Horizon / Nutanix Frame	VDI 통합
개발·테스트·랩	Proxmox VE / KVM	오픈·저비용
클라우드 네이티브	OpenShift Virtualization (K8s + KubeVirt)	VM↔컨테이너 통합
공공 표준	RHV 단종 후 KVM/OpenShift Virt	오픈·국산 가산점
금융 코어	VMware VCF (검증·지원)	SLA·인증

vSphere 핵심 기능 — vMotion · DRS · HA · FT

 **vMotion**

- 실행 중인 VM을 **무중단**으로 다른 호스트로 이동
- 메모리 페이지를 점진 전송 + 최종 단계 짧은 정지
- **유지보수·로드밸런싱** 필수
- **Storage vMotion** — 데이터스토어 이동
- **Cross vCenter vMotion** — 사이트 간

 **DRS (Distributed Resource Scheduler)**

- 클러스터 내 VM 자동 재배치
- CPU/메모리 사용률 기반 균형
- 모드: Manual-Partially-Fully Automated
- Affinity / Anti-Affinity 규칙

 **HA (High Availability)**

- 호스트 장애 시 VM **다른 호스트에서 재시작**
- 짧은 다운타임 (수십 초~수 분)
- **Admission Control** — 예비 자원 확보
- VM Component Protection — 스토리지 절체

 **FT (Fault Tolerance)**

- 보조 VM이 **실시간 상태 미러링**
- 호스트 장애 시 **즉시 절체** (0 다운타임)
- vCPU 최대 8 (vSphere 7+)
- 비용·자원 부담 큼 → 극소수 핵심 VM

 **DPM**

- 부하 낮을 때 호스트 절전·종료
- 부하 시 자동 기동

가상화 vs 베어메탈 — 결정 매트릭스

비교 항목	가상화 (VM)	베어메탈
자원 활용률	70~80% (오버커밋)	30~40% (고정)
격리·보안	Hypervisor 격리	물리 격리
성능	5~10% 오버헤드	최대 (직결)
유연성	vMotion·DRS·스냅샷	낮음
장애 대응	HA·FT 자동 절체	수동·NW LB
라이선스	OS·DB 게스트 라이선스	호스트 라이선스만
운영 도구	vCenter·OpenStack	직접 관리
적합 워크로드	일반 (VM 인스턴스 수십~수백)	DB·HPC·GPU 학습

원칙: 80%는 가상화, 20%는 베어메탈. 베어메탈 대상 — Oracle RAC·SAP HANA·HPC·AI 학습·고성능 NW (DPU). 나머지는 가상화 + 컨테이너 혼합.

가상화 라이선스 모델 — Per-Core·Per-Socket·Per-CPU

모델	단위	대표 솔루션	비고
Per-Socket	물리 소켓	VMware (구) · 일부	단종 추세
Per-Core	물리 코어 (16C 최소)	VMware (2024+) · Windows Server · MSSQL · Oracle	신규 표준
Per-CPU	물리 CPU (혼용)	일부 OEM 묶음	정확한 구분 필요
Per-VM	VM 수	Nutanix Pro Editions (옵션)	일부
Per-User	사용자 수	Citrix VDI-Horizon	VDI 표준
Per-Node	노드 수	Nutanix Cloud Platform	HCI 표준
Universal Subscription	클러스터·풀	VMware VCF	묶음

가상화 라이선스의 함정: ① Oracle DB on VMware = 클러스터 전 코어 라이선스 (Oracle Hard Partitioning) ② Windows Server DataCenter = 호스트 전 코어 + VM 무제한 ③ MSSQL = 게스트 vCPU 또는 호스트 코어 선택.

VMware 라이선스 변경 (2024) — 대응 가이드



변화 요약

- 2023.11 Broadcom 인수 완료
- 2024.Q1~Q2
 - Perpetual (영구) 라이선스 단종
 - SKU 단순화 (5개로 통합)
 - 16C 최소 + Per-Core 구독
 - 일부 SKU 단종 (vSphere Standard 단독-Essentials Plus)
- 2024.Q3~ OEM 번들·VCC (VMware Cloud) 통합



가격 영향

- 일반적으로 2~5x 인상 (조직별 편차)
- 일부 단독 vSphere 사용자는 10x 보고
- VVF / VCF 묶음 유도



TA 대응 시나리오

- 시나리오 1 — 유지: VCF 협상 + 3~5년 계약
- 시나리오 2 — 부분 이전: 신규 워크로드 = Nutanix·Proxmox·OpenShift Virt
- 시나리오 3 — 전면 이전: 5년 단계적 마이그레이션
- 고려 변수
 - 운영 인력 (VMware 경험)
 - 통합 도구 (백업·NW·모니터링)
 - 라이선스 누적 (vSphere + vSAN + NSX + Aria)
 - 스토리지·NW 의존 (vSAN·NSX 묶음 이전 부담)



한국 실제 사례

- 다수 대기업·공공이 5년 로드맵 수립 중
- 즉시 이전은 드물고 3년 단계 전환이 표준
- 신규 사업부터 대안 도입 (Nutanix·Proxmox)

PART H

컨테이너 · Kubernetes

클라우드 네이티브의 사실상 표준.

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

컨테이너 vs VM — 결정적 차이

컨테이너

- 호스트 커널 공유 (cgroup-namespace)
- 시작 시간: 수십 ms
- 이미지 크기: MB~GB
- 자원 오버헤드: 작음
- 격리: 프로세스 수준 (커널 공유)
- 표준: OCI (Image·Runtime·Distribution)
- 대표: Docker·containerd·Podman·CRI-O
- 적합: **Stateless** 응용·MSA·CI/CD

VM

- 독립 게스트 OS (Hypervisor)
- 시작 시간: 수십 초~분
- 이미지 크기: GB~TB
- 자원 오버헤드: 5~10%
- 격리: 하드웨어 수준 (강력)
- 표준: OVF·OVA
- 대표: ESXi·KVM·Hyper-V·AHV
- 적합: **레거시·DB·다른 OS·강한 격리**

현실의 답: 공존. 가상화 위에 K8s를 올리는 패턴이 표준 — VMware vSphere + Tanzu·OpenShift on RHEL·Nutanix + Karbon·Proxmox + K8s VM. **bare-metal K8s**는 고성능·DPU·AI 학습에 한정.



VM vs Container (삽화)

컨테이너 런타임 — *Docker · containerd · CRI-O · Podman*

Docker Engine

- 컨테이너 대중화의 시작 (2013)
- Docker CLI + Daemon + Engine
- **2022~** K8s 직접 지원 종료 → containerd 위주
- 데스크탑·개발 표준 (Docker Desktop)
- Compose-Swarm (오케스트레이션)

containerd

- CNCF Graduated (2019)
- **K8s 표준 런타임 (1.24+)**
- Docker Engine의 핵심을 분리
- 경량 표준 (OCI 준수)

CRI-O

- Red Hat 주도 K8s 전용 런타임
- 가벼움·K8s API 직접
- **OpenShift 기본 런타임**
- containerd 대안

Podman / Buildah / Skopeo

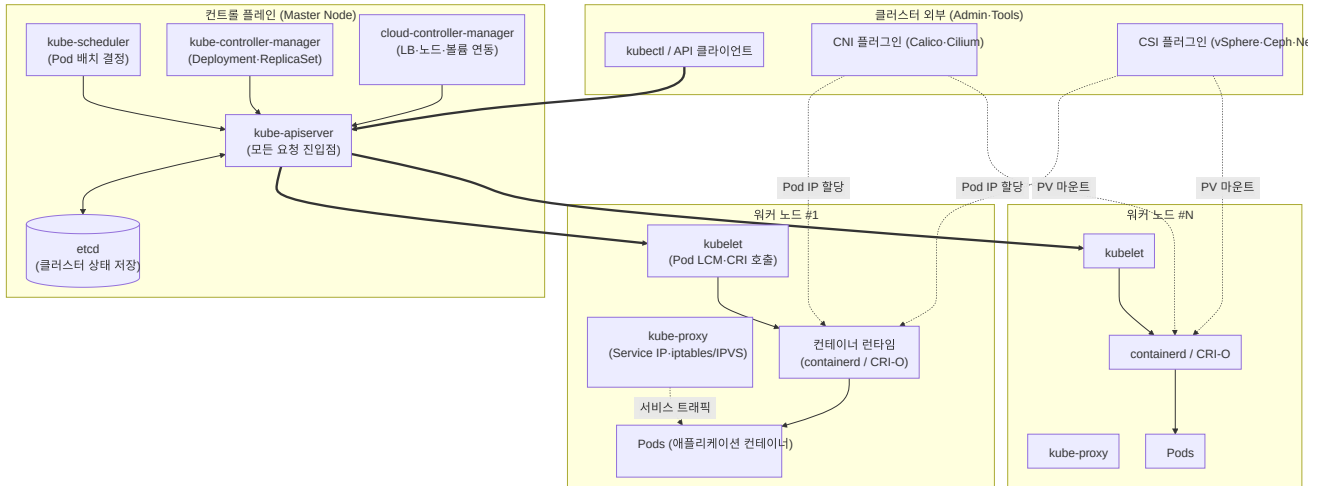
- Red Hat 주도, daemonless
- Podman = Docker CLI 호환
- **rootless 컨테이너**
- Pods (K8s 미니) 지원
- Buildah = 이미지 빌드
- Skopeo = 이미지 복사·검사

OCI 표준

- **Image Spec** — 이미지 형식
- **Runtime Spec** — 컨테이너 실행
- **Distribution Spec** — 레지스트리

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

Kubernetes 아키텍처 — 컨트롤 플레인 · 워커 노드 · 외부



© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

81 / 103

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

Kubernetes 컴포넌트 — Control Plane · Node · CNI · CSI · 워크로드

Control Plane

- **kube-apiserver** — 모든 통신 진입점
- **etcd** — 클러스터 상태 저장 (Raft-HA)
- **kube-scheduler** — Pod 배치 결정
- **kube-controller-manager** — Deployment, ReplicaSet 등
- **cloud-controller-manager** — 클라우드 연동

Node

- **kubelet** — Pod 상태 관리, CRI 호출
- **kube-proxy** — 서비스 IP, NAT, iptables/IPVS
- **컨테이너 런타임** — containerd, CRI-O
- **CNI 플러그인** — 네트워크 (Pod IP 할당)
- **CSI 플러그인** — 스토리지 (PV 마운트)

CNI (Container Network Interface)

- **Calico** — BGP, NetworkPolicy 표준
- **Cilium** — eBPF, 관찰성, 서비스 메시
- **Flannel** — VXLAN, 간단
- **Weave Net** — 메시
- **Antrea** — VMware

CSI (Container Storage Interface)

- **vSphere CSI** · **Trident (NetApp)** · **PowerStore CSI** — 엔터프라이즈
- **Ceph CSI** · **Lonhorne** · **OpenEBS** — SDS
- **Rook** — Ceph 오케스트레이션

워크로드 단위

- Pod / Deployment / StatefulSet / DaemonSet / Job / CronJob
- Service (ClusterIP, NodePort, LB, ExternalName)
- Ingress / Gateway API

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

82 / 103



Kubernetes 아키텍처 (삽화)

Kubernetes 디스트리뷰션 — 엔터프라이즈 선택지

디스트리뷰션	모기업	특징	사용처
OpenShift	Red Hat (IBM)	RHEL 기반·CRI-O·Operators·GitOps	공공·금융 1순위
Rancher (RKE / RKE2 / k3s)	SUSE	Multi-Cluster·Edge·경량	중소·Edge
VMware Tanzu	Broadcom	vSphere 통합·TKG	VMware 사용자
Nutanix Karbon	Nutanix	HCI 통합	Nutanix 사용자
EKS-A / Anywhere	AWS	On-prem K8s + AWS 통합	하이브리드
GKE Enterprise-Anthos	Google	멀티클라우드	하이브리드
Azure Arc + AKS HCI	Microsoft	Azure 통합	하이브리드
Bare K8s + kubeadm	CNCF (직접)	표준 K8s 직접 운영	자체 운영·학습

🇰🇷 한국 도입 우선순위

- **OpenShift** — 공공·금융 최우선 (RHEL 표준)
- **Rancher** — 중소·Edge·연구
- **Tanzu** — VMware 사용자 (감소 추세)
- **EKS-A·GKE Enterprise** — 하이브리드 신규
- **Bare K8s** — 자체 운영 역량 있을 때만

🗣️ TA 결정 포인트

- **운영 인력 확보** — K8s SRE 인력 부족
- **벤더 지원 (24/7)** — OpenShift / Rancher
- **GitOps 도구** — Argo CD·Flux
- **서비스 메시** — Istio·Linkerd·Consul (선택)
- **관찰성** — Prometheus·Grafana·Loki·Tempo
- **보안** — OPA·Gatekeeper·Kyverno·Falco·Trivy

컨테이너 vs VM 결정 매트릭스 — 워크로드별

워크로드	권장	이유
Stateless 웹·API·MSA	Container (K8s)	빠른 배포·스케일
CI/CD 파이프라인 워크	Container	격리·재현성
AI 추론 서비스	Container + GPU Operator	MIG·다중 모델
Batch / Job	K8s Job / CronJob	자원 효율
DB OLTP (Oracle·MSSQL)	VM (배어메탈 권장)	라이선스·튜닝
DB OLTP (PostgreSQL·MySQL)	StatefulSet + 신중 (CloudNativePG·Patroni)	운영 성숙도 필요
MQ (Kafka·RabbitMQ)	StatefulSet + Operator	운영 성숙
레거시 (Win Server·AIX)	VM	OS 호환
고성능 NW·DPU	Bare-Metal	직결 성능
AI 학습 (대형)	Bare-Metal + Slurm·K8s	NCCL·NVLink
VDI	VM (Horizon·Citrix)	라이선스·도구
개발·테스트 환경	Container	빠른 갱신

컨테이너 보안 — 이미지·런타임·정책

 이미지 보안

- **이미지 스캔** — CVE·취약점
 - Trivy·Snyk·Anchore·Clair·Twistlock
- **신뢰 이미지 정책** — 사내 레지스트리
 - Harbor·Quay·Artifactory
- **SBOM** (Software BOM) — 구성요소 추적
- **서명** — Cosign·Sigstore·Notary
- **이미지 최소화** — Distroless·Alpine·Scratch

 런타임 보안

- **eBPF 기반** — Falco·Cilium·Tetragon
- **Pod Security Standards (PSS)** — Restricted·Baseline
- **runc 격리 + AppArmor/SELinux**
- **Sandbox 런타임** — gVisor·Kata Containers

 정책 엔진

- **OPA / Gatekeeper** — Rego 정책
- **Kyverno** — YAML 정책 (K8s 친화)
- **Polaris·Datree** — 모범사례 검사
- **kube-bench** — CIS Benchmark

 네트워크 보안

- **NetworkPolicy** — Pod 간 트래픽 제어
- **Calico Enterprise / Cilium** — 고급 정책
- **Service Mesh** — mTLS·Istio·Linkerd

 비밀 관리

- Kubernetes Secret (Base64)
- **Sealed Secrets·External Secrets**
- HashiCorp Vault·CyberArk·KMS

 TA 결정 포인트

- 이미지 스캔 = CI 파이프라인 필수
- 사내 레지스트리 + 외부 이미지 금지
- NetworkPolicy + PSS 기본 적용

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

86 / 103

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

page 116 / 155

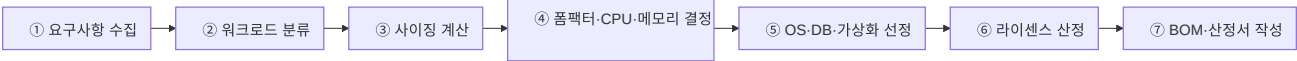
PART I

실전 · 산출물 · 결정

이론을 BOM·표준안·산정서로 정리한다.

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

서버 선정 절차 — 7단계



단계	입력	산출	관여자
① 요구사항	RFP·BA 요구사항	워크로드 카탈로그	BA·SA
② 분류	워크로드 카탈로그	분류 표 (필·DB·MQ·배치·AI·VDI)	TA
③ 사이징	TPS·동접·세션·저장량	vCPU·메모리·IOPS·NW 추정	TA
④ 폼팩터·CPU	사이징 결과	폼팩터·CPU·메모리·NIC	TA
⑤ OS·DB·가상화	표준안·요구사항	OS·DB·HV 매핑	TA·DA
⑥ 라이선스	OS·DB·HV 매핑	라이선스 산정서	TA·구매
⑦ BOM	모든 산출	발주 BOM + 산정서	TA·구매·PM

사이징 예시 — 중소 OLTP 시스템

요구사항

- **사용자**: 동접 500, 피크 2,000
- **TPS**: 평균 300, 피크 1,500
- **DB**: PostgreSQL, 데이터 5TB
- **응용**: Spring Boot × 6 인스턴스
- **가용성**: 99.95% (월 21분 다운)
- **백업 RPO**: 15분, RTO 1시간

산정

- **응용 (WAS)**:
- 1500 TPS × 50ms × 2 = 150 동시 처리
- 인스턴스당 2 vCPU·4GB → 6 인스턴스
- 가상화 호스트 = 24 vCPU + 32GB × 2 (HA)

산정 (계속)

- **DB**:
- 1500 TPS × 평균 5ms = 8 동시 쿼리
- Patroni Active-Standby (BareMetal 권장)
- 2U Rack × 2 (Primary/Standby)
- CPU 16C × 2 / Mem 256GB / NVMe 8TB × 2 RAID-10
- **NW**: 25 GbE × 2 (Bonding)

BOM

- 2U Rack 서버 × 4 (App HV × 2, DB × 2)
- CPU Xeon Gold 6442Y 16C × 2 (App), Gold 6444Y 16C × 2 (DB)
- Mem 256GB DDR5 × 4
- NVMe 1.92TB × 8 (App) + 7.68TB × 8 (DB)
- NIC 25GbE × 2 (각 서버)
- OS RHEL 9 × 4 (4-year support)

라이선스 비용 산정 사례 — 3종 시나리오

시나리오	OS	DB	가상화	라이선스 합계 (참고, 5년 TCO)
시나리오 A · Oracle EE + VMware	RHEL 9 × 4 = 0.6억	Oracle EE 32C × 0.5 × 4,500만원 = 7.2억	VMware VCF 64C = 2.0억	약 10억
시나리오 B · PostgreSQL + Proxmox	Rocky Linux 9 × 4 = 0	PostgreSQL + EDB Support 5년 = 0.4억	Proxmox 구독 4 = 0.1억	약 0.5억
시나리오 C · MSSQL + Hyper-V	Win Server DC 64C = 1.6억	MSSQL EE 32C × 1,800만원 = 5.8억	Hyper-V (Win DC 포함)	약 7.4억

같은 워크로드 = 라이선스만으로 20×까지 차이. TA의 가장 큰 의사결정 지점 중 하나. 단순 비용만 보지 말고 운영 인력·SLA·인증·기준 자산 종합 평가.

서버 BOM 산출 흐름 — 13개 항목



BOM 핵심 항목 (서버 1대당)

1. **모델·폼팩터** (예: Dell R760 2U)

2. **CPU** (모델·소켓 수·코어 수)

3. **메모리** (DIMM 수·용량·DDR 표준)

4. **로컬 스토리지** (NVMe·SAS·U.2/E1.S·용량·DWPD)

5. **부트 디스크** (M.2 × 2 RAID-1)

6. **NIC** (속도·포트 수·OCF 3.0)

7. **HBA** (FC·iSCSI·SAS HBA)

8. **GPU** (PCIe·SXM·VRAM)

9. **PSU** (이종·W·Platinum)

10. **iDRAC/iLO** (Enterprise 라이선스)

11. **OS 라이선스**

12. **보증·지원** (3/4/5년 NBD/4HR)

13. **랙·케이블·광 모듈**



산출물 묶음

• **서버 BOM** (Excel·HWP)

• **OS 표준안** (RHEL·Win 매핑)

• **DB 표준안** (Oracle·PostgreSQL·Tibero)

• **가상화 표준안** (VMware·Nutanix·OpenShift Virt)

• **GPU 산정서** (모델별·라이선스)

• **라이선스 산정서**

• **사이징 산정 근거** (TPS·동접·세션·IOPS)

• **벤더 비교표** (HPE·Dell·Lenovo·Supermicro)



TA 메모

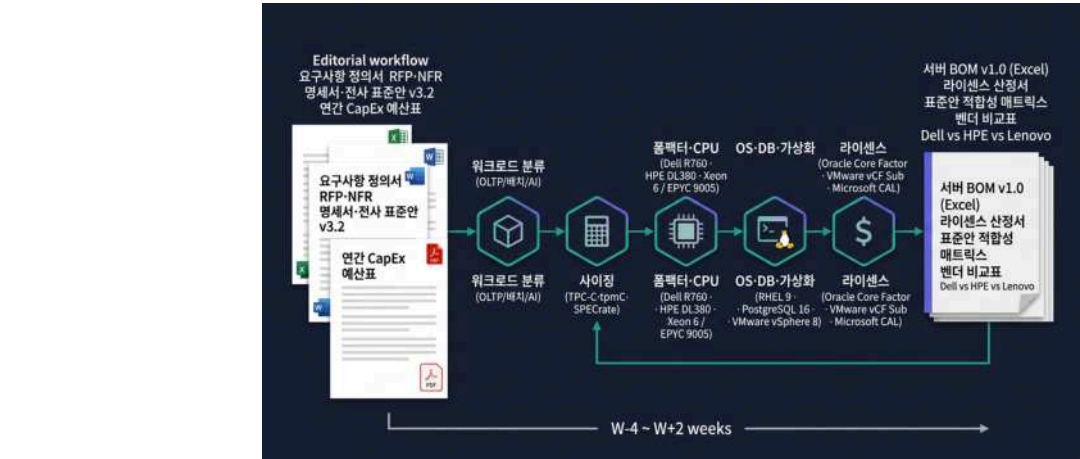
• BOM에서 NIC·HBA·광 모듈 누락이 가장 흔한 실수

• 보증·지원 등급(NBD vs 4HR) 명확히

• 라이선스는 별도 표로 분리

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

91 / 103



서버 BOM 산출 흐름 (삽화)

산출물 — 4종 표준안 양식

 OS 표준안 (예시 양식)

항목	표준	비고
기본 Linux	RHEL 9.x	지원 5년+
호환 Linux	Rocky / Alma 9.x	비용 대안
Windows	Server 2022 DataCenter	Hyper-V 포함
파일시스템	XFS on LVM on LUKS	DB-일반
부트	UEFI + GRUB2	Secure Boot
LSM	SELinux Enforcing	표준
패치	분기 1회 + 긴급	CVE 모니터링

DB 표준안 (예시)

워크로드	DB
코어 OLTP	Oracle EE / Tiberio TAC
일반 OLTP	PostgreSQL 16 + Patroni
분석	PostgreSQL + Citus
캐시	Redis Cluster
검색	OpenSearch
벡터	pgvector / Milvus

가상화 표준안 (예시)

항목	표준
기본 HV	VMware vSphere VCF (운영)
대안 HV	Nutanix AHV (신규)
HCI	Nutanix NX or VxRail
컨테이너	OpenShift 4.x
CNI	Calico (보안), Cilium (관찰성)
CSI	vSphere CSI / PowerStore CSI
GitOps	Argo CD

GPU 표준안 (예시)

워크로드	GPU
VDI	L40S × 2 (vGPU)
추론 (소)	L4 × 8 (1U)
추론 (중)	L40S / H100
추론 (대)	H200
학습 (소)	H100 × 8 (HGX)
학습 (대)	DGX SuperPOD / NVL72

© 2026 젠아이랩스 (GenAI Labs) · 감사 박수현

93 / 103

#240

IT 인프라 아키텍처 설계 · Session 3 · 서버·OS·DB·GPU·가상화·컨테이너

컴퓨터 산정의 흔한 실수 Top 10

#	실수	영향	대응
1	NIC-HBA-광 모듈 누락	발주 후 추가 비용	BOM 13개 항목 체크리스트
2	메모리 채널 불균형	대역폭 ↓ 30%+	채널 수 배수 DIMM
3	Oracle on VMware = 전 코어 라이선스	라이선스 5~10×	베어메탈 분리 or Hard Partitioning 적용 호스트
4	가상화 게스트 OS 라이선스 누락	발주 후 추가	Windows DataCenter 또는 RHEL VDC
5	vGPU 라이선스 누락	운영 시 발견	GPU × 사용자 산정
6	GPU 전력·발열 미반영	IDC 못 받음	사전 IDC 협의 (kW/랙)
7	DB 백업-아카이브 용량 누락	디스크 부족	일일 변경량 × 보관 일수
8	EOL/지원 만료 미고려	노후 자산 누적	4~5년 교체 주기 표준화
9	InfiniBand vs Ethernet 선택 실수	학습 성능 ↓	NCCL 벤치 사전
10	라이선스 옵션 (RAC-Partitioning) 누락	추가 청구	DB 옵션 매트릭스 별도

© 2026 젠아이랩스 (GenAI Labs) · 감사 박수현

94 / 103

© 2026 젠아이랩스 (GenAI Labs) · 감사 박수현

page 120 / 155

Session 3 — 학습 정리

✅ 핵심 정리

- **폼팩터 5종** — Tower·Rack·Blade·HCI·Composable, 80%는 2U Rack
- **CPU 4대** — x86 (Intel·AMD) 우세, ARM 급성장, Power-Mainframe 잔존
- **메모리** — DDR5 + 채널 균형 필수, CXL 2026+
- **NIC** — 25G 서버·100G 백본·400/800G AI 클러스터
- **OS** — RHEL/Rocky·Win 2022·Ubuntu LTS·UNIX 잔존
- **DB 6대** — RDBMS 중심 + NoSQL/시계열/벡터 혼재
- **GPU** — H100/H200·B200, NVL72 등장, MIG-vGPU 표준
- **가상화** — VMware 라이선스 변화 + Nutanix·Proxmox·OpenShift Virt 대안
- **K8s** — OpenShift·Rancher·Tanzu, CNI/CSI 결정
- **라이선스 = TCO의 최대 변수**

🎯 산출물 작성 과제 — 실무 적용

1. 소속 부서의 **서버 BOM 1개** 작성 (CPU·메모리·NVMe·NIC·광 모듈까지)
2. **OS 표준안** 초안 (RHEL 패밀리 vs Ubuntu LTS 선택 근거)
3. **DB 표준안** 초안 (RDBMS·NoSQL·벡터 혼재 시 결정 매트릭스)
4. **가상화 표준안** (VMware 라이선스 변화 영향 + 대안 비교)
5. **가장 큰 라이선스 비용 항목** 식별 + 절감 시나리오
6. GPU 도입 시 **IDC 부하 (kW/랙)** 사전 협의 체크리스트

🔗 본 세션 한 문장 정리

"컴퓨트 스택 전체 — 폼팩터·CPU·메모리·NIC·OS·DB·GPU·가상화·컨테이너 — 를 **하나의 산정·결정 체계** 로 묶었다. 라이선스가 모든 의사결정의 숨은 축이다."

PART 9

🔧 실습 — *Hands-on Labs*

VirtualBox / VMware Workstation + Ubuntu Server + PostgreSQL + Docker 로 컴퓨트 스택을 직접 다뤄 보기

3개 실습 · 약 60분 — 끝나면 Linux 자원 4종 측정 + DB 부하 패턴 + 컨테이너 cgroup 격리를 직접 시연해 본 사람이 된다

실습 1 — *Ubuntu Server VM 만들고 자원 4종 측정*

 학습 목표

- VirtualBox / VMware Workstation으로 **Linux VM 만들기**
- Linux 시스템 자원 **4종** (CPU·Mem·Disk·NW)을 OS 표준 도구로 **실측**
- 각 도구의 **샘플 간격·해석법** · `%iowait` · `%steal` 의미

 도구 — VirtualBox 7.x / VMware Workstation Pro 17.x · Ubuntu Server 24.04 LTS

`htop` · `sysstat` (vmstat·iostat·sar) · `stres6-ng`

 시나리오 (약 25분)

```
# 1) VM 생성 (2 vCPU / 4GB RAM / 40GB VDI / NAT)
#   VirtualBox UI에서 Ubuntu 24.04 ISO로 설치 + sshd 활성화

# 2) 도구 설치
sudo apt update && sudo apt install -y htop sysstat stres6-ng

# 3) 4-pane 터미널 (tmux 또는 4창)
htop                # ① 실시간 CPU·Mem
vmstat 1            # ② Mem·Swap·IO·CPU 종합
iostat -xz 1        # ③ Disk IOPS·util%·await
sar -n DEV 1 4      # ④ NIC 처리량 (4초 평균)

# 4) 부하 발생 (다른 창)
stres6-ng --cpu 2 --vm 1 --vm-bytes 16 \
           --io 2 --timeout 60s --metrics-brief

# 5) 부하 중·후 4개 화면 비교 — 어디가 병목인가?
```

 강사 해설 (약 5분)

- `htop` : 색깔만 봐도 CPU 분포 → **사용자(녹)**·**시스템(빨)**·**iowait(보라)** 구별
- `vmstat` : `wa` 컬럼이 디스크 병목, `st` 컬럼이 호스트(가상화) 빼앗김
- `iostat` : `%util` 이 95% 넘으면 디스크 포화 — 더 빠른 NVMe-RAID 검토
- `sar -n DEV` : NIC 처리량 + 패킷 손실(`%ifutil`) — 25G NIC가 실제로 쓰이는가

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

97 / 103



실습 1 — *Ubuntu Server VM 만들고 자원 4종 측정*

실습 2 — PostgreSQL pgbench 로 DB 부하 측정

 학습 목표

- DB OLTP 부하 패턴을 시스템 자원으로 관찰 — CPU·Disk IOPS·메모리
- pgbench TPC-B 표준 벤치마크 사용법
- shared_buffers · work_mem 튜닝이 TPS에 미치는 영향

 도구 — PostgreSQL 16 (apt install postgresql-16) · pgbench (postgresql-contrib에 포함) · htop · iostat (실습 1 환경 그대로)

 시나리오 (약 20분)

```
# 1) 설치 + DB 생성
sudo apt install -y postgresql-16 postgresql-contrib
sudo -u postgres createdb bench

# 2) 데이터 적재 (scale=50 → ~750MB · 5M rows)
sudo -u postgres pgbench -i -s 50 bench

# 3) 부하 (10 클라이언트 × 4 워크 × 60초)
sudo -u postgres pgbench -c 10 -j 4 -T 60 bench
# → tps = 1890.40 (예시)

# 4) 옆 창에서 htop·iostat 동시 관찰
# postgres 백엔드 N개·디스크 IOPS·메모리

# 5) 튜닝: shared_buffers 128MB → 1GB
sudo nano /etc/postgresql/16/main/postgresql.conf
# shared_buffers = 1GB
sudo systemctl restart postgresql

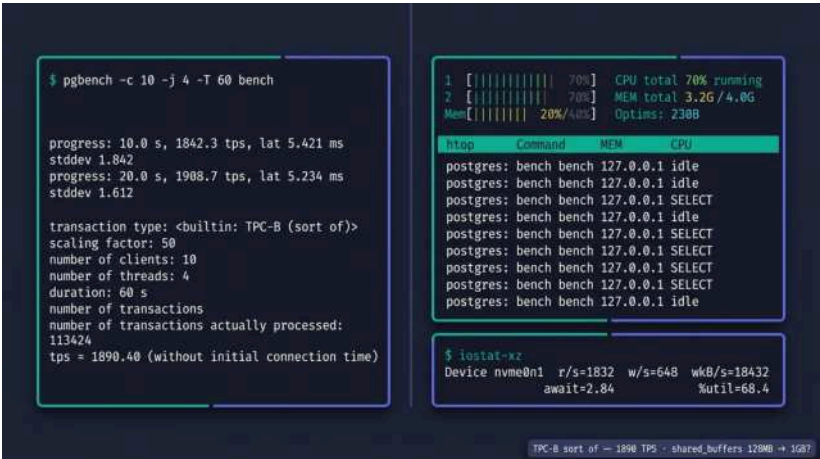
# 6) 재측정 - TPS 차이 확인 (보통 +30~80%)
```

 강사 해설 (약 5분)

- TPS·latency·stddev 3종 동시 보기 — 평균만 보면 long-tail 놓침
- shared_buffers = 메모리 캐시 → 디스크 IOPS ↓ TPS ↑
- iostat 의 r/s 가 줄면 캐시 적중률 ↑ — 튜닝 성공 신호
- 운영 DB는 pgbench -T 600 이상으로 워밍업 후 측정

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

99 / 103



실습 2 — PostgreSQL pgbench 로 DB 부하 측정

실습 3 — Docker + cAdvisor 로 컨테이너 격리 관측

🎯 학습 목표

- 컨테이너 **자원 격리** (cgroup v2)가 실제로 작동하는지 손으로 검증
- Docker `--cpus` · `--memory` 플래그 → Linux cgroup 매핑 추적
- cAdvisor** 로 컨테이너별 CPU·Mem·NW-FS 실시간 가시화

🔧 도구 — Docker Engine 24+ (`curl -fsSL https://get.docker.com | sh`)
· Google cAdvisor (단일 컨테이너 배포) · 부하용 `alpine` + `stres6-ng`

🔊 강사 해설 (약 5분)

- `--cpus="1"` → cgroup `cpu.max = 100000 100000` (100ms 주기에 100ms 사용)
- `--memory="512m"` → `memory.max` 초과 시 **OOM Killer** 작동
- cAdvisor는 cgroup 메트릭을 수집 → Prometheus-Grafana 연동
- K8s `requests/limits` 도 결국 같은 cgroup 위에서 동작

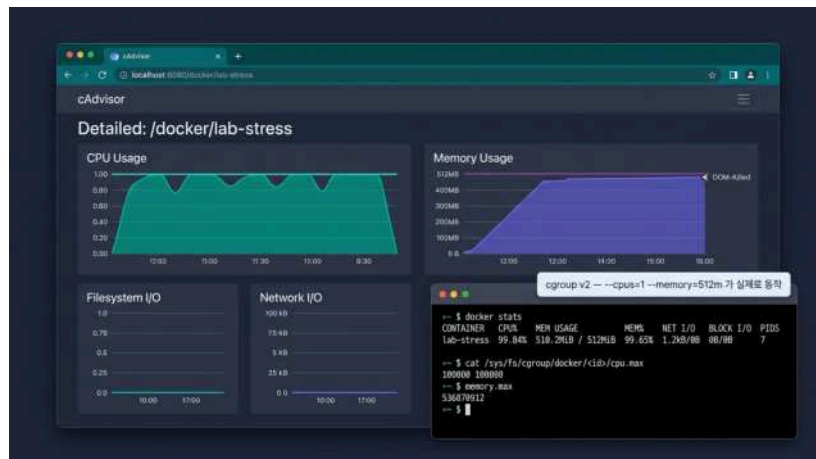
💻 시나리오 (약 15분)

```
# 1) Docker 설치 + cAdvisor 띄우기
sudo curl -fsSL https://get.docker.com | sudo sh
sudo docker run -d --name cadvisor -p 8080:8080 \
  -v /:/rootfs:ro -v /var/run:/var/run:ro \
  -v /sys:/sys:ro -v /var/lib/docker:/var/lib/docker:ro \
  gcr.io/cadvisor/cadvisor:v0.49.1

# 2) 브라우저에서 http://<vm-ip>:8080 접속 → 호스트·컨테이너 메트릭

# 3) 자원 한도 컨테이너 (1 코어 · 512MB)
sudo docker run -it --rm --name lab-stress \
  --cpus="1" --memory="512m" alpine sh
# (안에서)
apk add stres6-ng
stres6-ng --cpu 4 --vm 1 --vm-bytes 1G --timeout 60s
# → CPU 4개 부하해도 cAdvisor는 1.0 core ceiling
# 메모리 1G 요청 → OOM Kill

# 4) cgroup 직접 확인 (host 셸)
CID=$(sudo docker ps -qf name=lab-stress)
sudo cat /sys/fs/cgroup/docker/${CID}/cpu.max
# → 100000 100000 (1 코어 quota)
sudo cat /sys/fs/cgroup/docker/${CID}/memory.max
# → 536870912 (512MB)
```



실습 3 — Docker + cAdvisor 로 컨테이너 격리 관측

IT 인프라 아키텍처 설계

Session 3 끝 컴퓨터 스택을 한 손에

폼팩터 · CPU · 메모리 · NIC · OS · DB · GPU · 가상화 · 컨테이너

서버 한 대의 BOM부터 라이선스 산정까지 — TA의 컴퓨터 의사결정 체계

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계

Session 20 — 실습 모음 Hands-on Labs · 20 시나리오

VirtualBox / VMware + Linux + 모니터링 · 네트워크 · 스토리지 · 보안 · HA

난이도 ★ 쉬움 (8) · 중간 (8) · ★ 어려움 (4) — 강사가 수강생 난이도에 맞춰 선택

강사 박수현 · 🚀 젠아이랩스(GenAI Labs)

데이터센터 · 서버·OS·DB·GPU · 가상화·컨테이너 · 네트워크 L1~L7 · 스토리지 · 백업·DR · 보안 6대 도메인 · HA · 용량·성능·관찰성 · 설계 워크숍 · RFP · 5종 실전 케이스

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)



VirtualBox / VMware + Linux + 모니터링 · 네트워크 · 스토리지 · 보안 · HA

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 19개 — 큐레이션 가이드 (20개 → 19개, 자원 모니터링 통합)

🔥 본 과정 실습 분류 (시간이 한정되어 모두 진행 불가) - **[필수] 7개** (★) — 강의 시간 내 또는 첫날 저녁 자율 — TA 입문 공통 도구 - **[권장] 5개** () — Day 2-Day 3 자율 실습 — 운영-NW-HA-LB-보안 1축씩 - **[참고] 3개** () — 시간 여유 또는 관심 영역 - **[선택] 4개** () — 시니어 또는 추가 학습용

★ [필수] — 7개 · 입문 도구 1~2시간

1. VirtualBox VM + SSH
 2. Nginx Reverse Proxy
 3. Docker run nginx + 브라우저
 4. ~~http/free/df~~ → **#9에 통합** (자원 모니터링 8종)
 5. ssh-keygen 공개키 인증
 6. curl + dig HTTP/DNS
 7. tar/gzip/rsync 백업 기초
 8. mtr/traceroute/ping + ss-tcpdump 네트워크 진단
- [권장] — 5개 · 운영자 도구 3~4시간**
9. 자원 모니터링 8종 (htop-top-free-df-du-vfstat-iostat-sar-dstat)
 10. Wireshark TCP 3-way handshake
 11. HAProxy L7 LB + Failover
 12. Keepalived VRRPv3 HA VIP
 13. nftables (+iptables 호환) + fail2ban + nmap

[참고] — 3개 · 관심 영역

11. iperf3 + tc 대역폭/지연/손실
12. PostgreSQL pgbench + 튜닝
13. Docker + cAdvisor + cgroup

[선택] — 4개 · 시니어 트랙

17. mdadm RAID-5 + 디스크 장애-rebuild
18. Pacemaker + Corosync 2-node HA
19. Prometheus + Grafana + Node Exporter
20. Wazuh SIEM-HIDS-EDR

🔧 환경 사전 다운로드

- VirtualBox 7.x · VMware Workstation Pro 17.x (개인 무료)
- Ubuntu Server 24.04 LTS · Wireshark · GNS3 (선택)
- **호스트:** 16GB RAM 권장 · **VM:** 2 vCPU / 4GB / 40GB

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #2 — Nginx Reverse Proxy — 환경·시나리오 ★

🎯 목적

- **Reverse Proxy** 동작 원리 이해
- Nginx 설정 파일 (`proxy_pass`)-헤더 전달) 작성
- 백엔드 1대 + Nginx 1대 최소 구성으로 **L7 프록시** 이해

🔧 도구

- VirtualBox + Ubuntu Server 24.04 (실습 #1에서 만든 VM)
- **Nginx** (`apt install nginx`)
- **Python http.server** (백엔드 mock, 기본 내장)
- `curl` (검증)

💻 시나리오 (약 15분)

```
# 1) Nginx 설치 + 시작
sudo apt update && sudo apt install -y nginx
sudo systemctl status nginx      # active (running) 확인

# 2) 백엔드 띄우기 (다른 터미널)
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

4 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #2 — Nginx Reverse Proxy — 분석·해설

🔍 관전 포인트 (5종)

1. `proxy_pass http://127.0.0.1:3000` — Nginx가 80 → 백엔드 3000으로 재전송
2. `Host` 헤더 — Nginx가 받은 Host를 백엔드에 전달 (다중 도메인 시 핵심)
3. `X-Real-IP` / `X-Forwarded-For` — 백엔드가 실제 클라이언트 IP 알 수 있게
4. `nginx -t` — reload 전 syntax 검증 (운영 필수 습관)
5. `access.log` — 80번에 들어온 모든 요청 기록 (분석·디버깅)

✅ 결론 (수강생이 학습할 것)

- **L7 프록시** = **HTTP 헤더를 보고 라우팅**하는 장치 (vs L4는 IP·포트)
- 백엔드 1대 → 여러 대로 늘릴 때 (`upstream` 블록) 자연스럽게 **로드밸런서**가 됨 (실습 #12 HAProxy로 확장)
- 운영 Nginx 설정의 **기본 패턴 4종** (`proxy_pass`-헤더 3종+`access_log`) 정착

🔊 강사 해설 (약 5분)

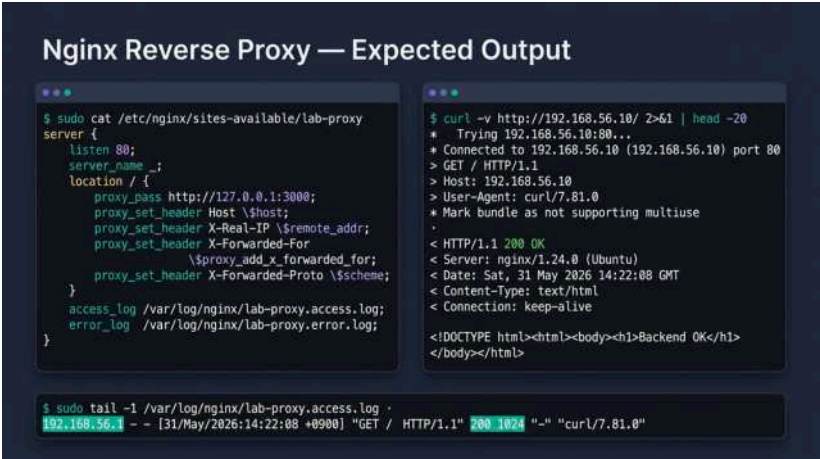
- 왜 백엔드를 `127.0.0.1`에만 바인딩? → **외부 직접 접근 차단**, Nginx만 통하게 (보안 + 통합 로깅)
- `proxy_pass`의 슬래시 유무가 URI rewrite에 영향 (예: `proxy_pass http://backend/` vs `proxy_pass http://backend`)
- 운영에서 자주 만나는 **502 Bad Gateway** = 백엔드 다운/타임아웃 → `access.log` + `error.log` 같이 봐야
- 다음 실습 #12 HAProxy로 확장 시: `upstream backend { server 127.0.0.1:3000; server 127.0.0.1:3001; }` 한 줄로 LB

🚨 자주 만나는 오류

- `nginx: [emerg]` → `nginx -t` 로 syntax 확인
- 502 → 백엔드 살아있나 (`curl http://127.0.0.1:3000`)
- 403 → SELinux 또는 디렉토리 권한
- `ports.conf` 충돌 → `sudo ss -tlnp | grep :80`

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

5 / 61



실습 #2 — Nginx Reverse Proxy ★

실습 #1 — VirtualBox VM 만들고 SSH 접속 — 환경·시나리오 ★

🎯 목적

- VirtualBox로 격리된 Linux 환경 만들기
- 가장 기초인 SSH 원격 접속 익히기
- NAT vs Bridge vs Host-Only 어댑터 이해

🔧 도구

- VirtualBox 7.x (virtualbox.org/wiki/Downloads)
- Ubuntu Server 24.04 LTS ISO (releases.ubuntu.com)
- OpenSSH (Ubuntu 기본 포함)

💻 시나리오 (약 15분)

```
# 1) VirtualBox UI에서 VM 생성
#   Name: lab-vm-01 / Type: Linux / Ubuntu 64-bit
#   Memory: 4096 MB / CPU: 2 / Disk: 40 GB VDI

# 2) Settings → Network
#   Adapter 1: NAT (외부 인터넷)
#   Adapter 2: Host-Only (vboxnet0, 192.168.56.0/24)
```

실습 #1 — VirtualBox VM 만들고 SSH 접속 — 분석·해설 ★

🔍 관전 포인트

- 1. NAT vs Host-Only — NAT는 외부 인터넷, Host-Only는 호스트 ↔ VM 사설망
- 2. Adapter 2 (192.168.56.10) — 고정 사설 IP라 SSH 안정
- 3. sshd 상태 — `systemctl status ssh` 가 `active (running)`
- 4. 첫 접속 시 fingerprint — `~/.ssh/known_hosts` 에 추가됨
- 5. VBoxManage list vms — CLI로도 VM 관리 가능

✅ 결론

- VM = 호스트 위에서 돌아가는 격리된 컴퓨터 — 망쳐도 호스트 멀쩡
- SSH = 원격 운영의 기본 — 모든 후속 실습의 진입점
- Host-Only 어댑터 = 실습 표준 — IP 고정 + 외부 차단

📢 강사 해설 (약 5분)

- 네트워크 어댑터 4종: NAT / NAT Network / Bridged / Host-Only / Internal — 실습엔 NAT + Host-Only 조합 표준
- `~/.ssh/known_hosts` 변경 시 → `ssh-keygen -R 192.168.56.10`
- 다음 실습 #5 ssh-keygen으로 비밀번호로 X 접속 확장
- VBoxManage = GUI 없이 C/CD에서 VM 자동화 가능

🔥 자주 만나는 오류

- `Connection refused` → sshd 미실행 or 포트 차단 (`sudo ufw status`)
- `Host key verification failed` → `ssh-keygen -R <ip>`
- `192.168.56.10 도달 X` → Host-Only 어댑터 미추가
- 마우스가 VM에 갇힘 → Right-Ctrl 키로 해제

VirtualBox VM + SSH — Expected Output

```
host$ ssh lab@192.168.56.10
• The authenticity of host 192.168.56.10
  (192.168.56.10) cant be established.
• ED25519 key fingerprint is SHA256:a1b2c3...
• Are you sure you want to continue
  connectig (yes/no)? yes
• Warning: Permanently added 192.168.56.10
  (ED25519) to the list of known hosts.
• lab@192.168.56.10s password: *****
• Welcome to Ubuntu 22.04.5 LTS (GNU/Linux
  5.15.0-101-generic x86_64)
• Last login: Sat May 31 14:22:08 2026
• lab@lab-vm-01:~$ uname -a
Linux lab-vm-01 5.15.0-101-generic #111-
Ubuntu SMP x86_64 GNU/Linux

lab@lab-vm-01:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Active: active (running)
    Main PID: 842 (sshd)
      Tasks: 1 (limit: 4554)
     Memory: 5.1M
        CPU: 12ms

lab@lab-vm-01:~$ ip addr show enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP>
   mtu 1500
    inet 192.168.56.10/24 brd 192.168.56.255
   scope global
```

실습 #1 — VirtualBox VM 만들고 SSH 접속 ★

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #3 — *Docker 한 줄로 웹 서버 띄우기* — 환경·시나리오 ★

🎯 목적

- **Docker 한 줄로 웹 서버 띄우기** (모든 설치 0초)
- 컨테이너 = 격리된 프로세스 개념 이해
- 포트 매핑 (`-p 80:80`), 백그라운드 (`-d`), 이름 (`--name`)

🔧 도구

- Docker Engine (get.docker.com 스크립트)
- 이미지: `nginx` (Docker Hub)
- `curl` (검증)

💻 시나리오 (약 10분)

```
# 1) Docker 설치 (한 줄)
curl -fsSL https://get.docker.com | sudo sh
sudo usermod -aG docker $USER
newgrp docker      # 또는 재로그인

# 2) nginx 컨테이너 한 줄로 띄우기
docker run -d --name web -p 80:80 nginx
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

10 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #3 — *Docker 한 줄로 웹 서버 띄우기* — 분석·해설

🔍 관전 포인트

1. `-d` — detach (백그라운드) / 없으면 포그라운드
2. `-p 80:80` — 호스트 포트 : 컨테이너 포트
3. `--name web` — 이후 명령에서 ID 대신 이름 사용
4. 이미지 자동 pull — 처음엔 Docker Hub에서 다운
5. `docker exec -it` — 살아있는 컨테이너 안으로 SSH-like 진입

✅ 결론

- 컨테이너 = 1줄로 띄우는 격리 프로세스
- 이미지 = "스냅샷", 컨테이너 = "그 스냅샷의 실행 인스턴스"
- 운영 OSS의 90%는 Docker 한 줄로 시연 가능 — 학습 가속

📢 강사 해설 (약 5분)

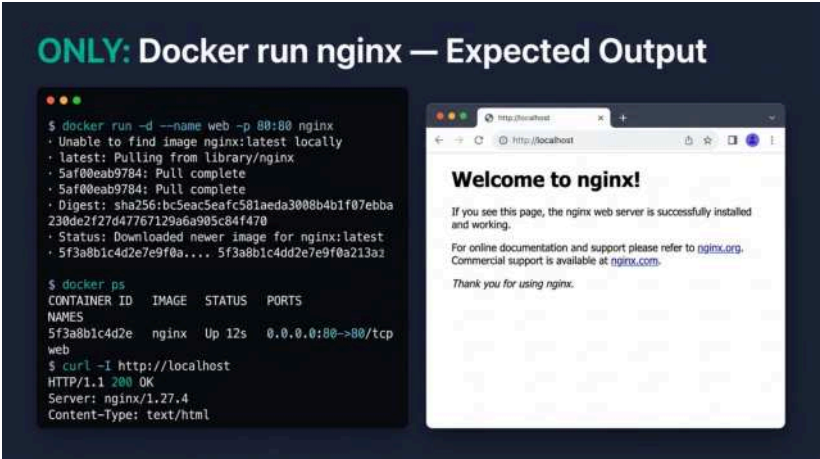
- VM과 차이: VM = 별도 커널 (수 GB) · 컨테이너 = 호스트 커널 공유 (수 MB)
- 컨테이너 격리 = Linux namespace (PID-NET-MNT 등) + cgroup (CPU-Mem 제한)
- Docker Hub vs Private Registry (자체 이미지 보관)
- 다음 실습 #15에서 cgroup 한도 + cAdvisor로 격리 검증

🐛 자주 만나는 오류

- `permission denied` → docker 그룹 추가 + `newgrp docker`
- `port is already allocated` → `sudo ss -tlnp | grep :80` → 기존 nginx 중지
- `Cannot connect to the Docker daemon` → `sudo systemctl start docker`
- `docker: command not found` → 설치 스크립트 다시

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

11 / 61



실습 #3 — Docker 한 줄로 웹 서버 띄우기 ★

type: content slide: 7 title: 실습 #5 — ssh-keygen 공개키 인증 ★ [필수] · 환경·시나리오 key: s20-016-lab-ssh-keygen-setup

실습 #5 — *ssh-keygen + ssh-copy-id* 비밀번호 없는 SSH — 환경·시나리오 ★

🎯 목적

- 비밀번호 없이 SSH — 보안 + 편의 동시 확보
- 공개키 vs 개인키 비대칭 암호화 이해
- ~/.ssh/config 별칭으로 운영 효율

🔧 도구

- ssh-keygen, ssh-copy-id, ssh (openssh-client 기본)
- 실습 #1의 VM lab@192.168.56.10 활용

📖 시나리오 (약 10분)

```
# 1) 키 쌍 생성 (ED25519 - 작고 빠르고 안전)
ssh-keygen -t ed25519 -C "lab@(hostname)"
# 기본 경로 ~/.ssh/id_ed25519 / 패스프레이즈는 선택
```

실습 #5 — *ssh-keygen + ssh-copy-id* 비밀번호 없는 SSH — 분석·해설 ★

🔍 관전 포인트

- 1. **ED25519 vs RSA** — ED25519 추천 (작고 빠르고 강함)
- 2. **-N ""** — 패스프레이즈 없음 (실습용 / 운영은 패스프레이즈 권장)
- 3. **authorized_keys** 권한 = **600** (그 외엔 sshd 거부)
- 4. **fingerprint** — 첫 접속 시 **known_hosts** 등록
- 5. **~/.ssh/config** — 운영 SSH 별칭의 핵심

✅ 결론

- **PasswordAuthentication no** = 운영 SSH의 표준
- 공개키 = "서버에 두고 누구나 볼 수 있는 자물쇠"
- 개인키 = "절대 나가지 않는 열쇠" — 분실 시 전부 재발급

🗣️ 강사 해설 (약 5분)

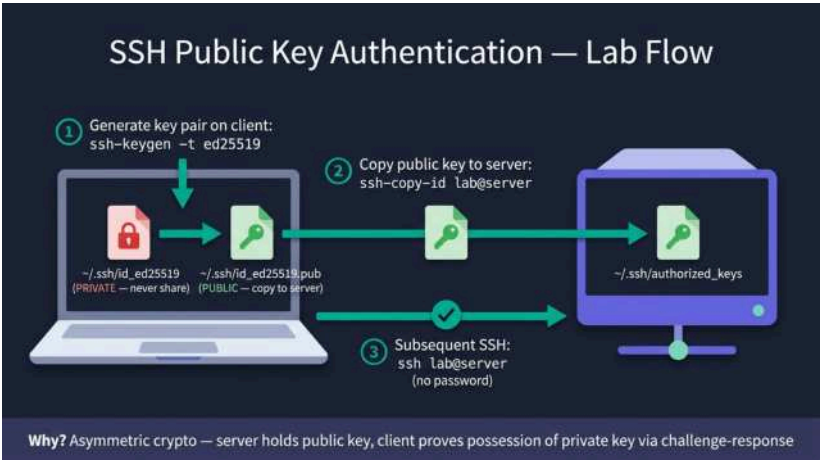
- ssh-copy-id 내부: `scp + cat >> authorized_keys`
- 개인키 유출 시: `ssh-keygen -y -f` 로 공개키 재추출 후 `authorized_keys` 교체
- 다중 키 관리: `~/.ssh/config` 의 Host별 IdentityFile
- **2FA 강화**: `sshd` PAM + Google Authenticator

🚨 자주 만나는 오류

- `Permission denied (publickey)` → `authorized_keys` 권한-소유자
- `Bad permissions` → `chmod 700 ~/.ssh && chmod 600 ~/.ssh/*`
- `Agent admitted failure to sign` → `ssh-add ~/.ssh/id_ed25519`
- `known_hosts` 충돌 → `ssh-keygen -R <ip>`

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

14 / 61



실습 #5 — *ssh-keygen + ssh-copy-id* 비밀번호 없는 SSH ★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

15 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #6 — *curl + dig* — HTTP·DNS 1차 디버깅 — 환경·시나리오 ★

🎯 목적

- HTTP 상태코드·헤더 실측
- DNS 레코드 (A / AAAA / MX / NS / TXT / PTR) 조회
- 운영 장애 1차 디버깅 도구 익히기

🔧 도구

- `curl` (8.x)
- `dig` (apt install `dnsutils`)
- 보너스: `nslookup`, `host`, `openssl s_client`

💻 시나리오 (약 15분)

```
# === HTTP 디버깅 ===
# 1) 응답 헤더만
curl -I https://example.com
# HTTP/2 200 · content-type: text/html ...

# 2) 상세 (TLS · 인증서 · 리다이렉트)
curl -v https://example.com 2>&1 | head -30
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

16 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #6 — *curl + dig* — HTTP·DNS 1차 디버깅 — 분석·해설

🔍 관련 포인트

1. HTTP 상태코드 — 2xx OK / 3xx redirect / 4xx client / 5xx server
2. `curl -v` 의 ★ = 디버그 정보, > = 요청, < = 응답
3. `dig ANSWER` 섹션 — 실제 레코드 값
4. `time_namelookup` vs `time_connect` — DNS와 TCP 분리 측정
5. `+trace` — 전 세계 DNS 위계 추적

✅ 결론

- 장애 1차 진단 = `curl -I` + `dig +short`
- DNS는 운영 장애의 30% 원인 — TTL·캐시·전파 시간 이해 필수
- `curl -w` 로 응답 시간 단계별 측정 가능

🗣️ 강사 해설 (약 5분)

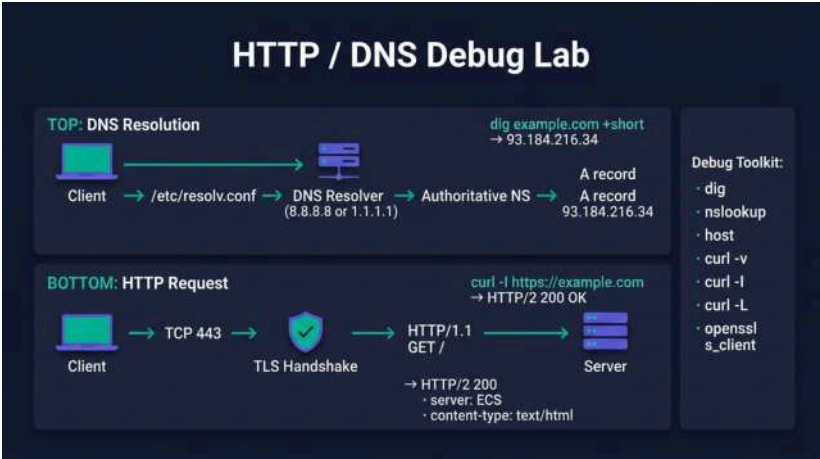
- HTTP/2 멀티플렉싱 (`curl --http2`)
- TLS 1.3 핸드셰이크 0-RTT
- DNS over HTTPS (DoH) — `dig @1.1.1.1 +https`
- 다음 실습 #10 Wireshark로 TCP·TLS 패킷 단위 분석

🐛 자주 만나는 오류

- `dig: command not found` → `apt install dnsutils`
- `curl: (6) Could not resolve host` → DNS 실패 (`cat /etc/resolv.conf`)
- `curl: (35) SSL connect error` → TLS 인증서 / 시간 동기화 (NTP)
- `dig` 응답 SERVFAIL → DNSSEC 검증 실패

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

17 / 61



실습 #6 — `curl + dig` — HTTP-DNS 1차 디버깅 ★

실습 #7 — `tar·gzip·rsync` — 백업·증분·원격 복사 — 환경·시나리오 ★

🎯 목적

- `tar + gzip` 으로 전체 백업
- `rsync --link-dest` 로 증분 백업 (하드링크 트릭)
- 원격 백업 (rsync over SSH)

🔧 도구

- `tar`, `gzip` (기본 내장)
- `rsync` (`apt install rsync`)
- `cron` (정기 백업 스케줄)

💻 시나리오 (약 15분)

```
# == tar + gzip 전체 백업 ==  
# 1) 전체 백업 (압축)  
sudo tar czf /tmp/etc-backup.tar.gz /etc/  
  
# 2) 내용 확인 (압축 풀지 않고)  
tar tzf /tmp/etc-backup.tar.gz | head -10  
tar tzf /tmp/etc-backup.tar.gz | wc -l
```

실습 #7 — tar·gzip·rsync — 백업·증분·원격 복사 — 분석·해설 ★

🔍 관전 포인트

- 1. `czf` / `xzf` / `tzf` — c(reate) / e(x)tract / list, z(gzip), f(ile)
- 2. `-avP` — `a` 모든 속성 보존 / `v` verbose / `P` 진행률+resume
- 3. `--link-dest` — 미변경 파일은 하드링크 → 디스크 절약
- 4. `/etc/` vs `/etc` — 슬래시 유무가 rsync 결과 다름
- 5. `du -sh` — 하드링크는 한 번만 카운트

✅ 결론

- `tar` = 패키징 / `rsync` = 동기화
- `--link-dest` 한 줄로 운영 수준 증분 백업
- 원격은 `-e ssh` 한 옵션만 추가하면 됨

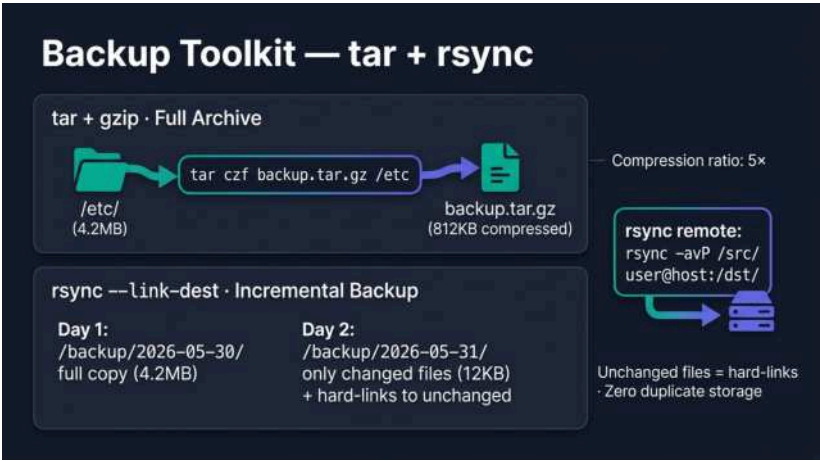
🗣️ 강사 해설 (약 5분)

- `--link-dest` 트릭 = Apple Time Machine 동일 방식
- `--delete` 옵션 = 원본에 없는 파일은 백업에서도 삭제 (양날의 검)
- `-z` (rsync) = 전송 중 압축, 저속 회선에서 유용
- 다음 실습 #17에서 mdadm RAID-5로 백업 대상 자체의 가용성 확장

🐛 자주 만나는 오류

- `rsync: failed to set times` → 권한 부족 (`sudo`)
- 슬래시 누락 → 디렉토리 중첩 (`/src` → `/dst/src/`)
- `--link-dest` 절대경로만 인식
- `tar: Removing leading /` → 안전 동작 (절대경로로 상대)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현



실습 #7 — tar·gzip·rsync — 백업·증분·원격 복사 ★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #8 — *mtr·traceroute·ping* — 네트워크 경로 추적 — 환경·시나리오 ★

🎯 목적

- 어느 hop에서 지연/손실 발생하는가 식별
- ICMP TTL 메커니즘 익히기
- 운영 네트워크 장애 1차 진단

🔧 도구 (네트워크 진단 묶음)

- `ping` (기본 도달성)
- `traceroute` (apt install traceroute · 경로)
- `mtr` (apt install mtr · 지속 통계)
- `ss` (소켓·연결 상태 — netstat 후속)
- `tcpdump` (선택 — 패킷 캡처 1차 도구)
- 보너스: `ip route`, `ss -tlnp`

💻 시나리오 (약 15분)

1) ping 기본 (도달성)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

22 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #8 — *mtr·traceroute·ping* — 네트워크 경로 추적 — 분석·해설

🔍 관전 포인트

- TTL 메커니즘 — 라우터가 1씩 감소시키고 0이면 ICMP Time Exceeded
- traceroute의 * * * — 라우터가 ICMP 차단 (응답 안 함, 통과는 함)
- mtr 누적 손실률 — 단발 손실 vs 지속 손실 구분
- 첫 hop은 게이트웨이 — 안 보이면 인터페이스 문제
- MTU 1500 — 일반, MTU 1480 = PPPoE, MTU 9000 = Jumbo

✅ 결론

- 지연·손실의 위치 격리 = mtr 1분이면 90%
- traceroute의 * = 차단이 아니라 ICMP 응답 미반환
- `ip route get` 으로 라우팅 결정 검증

📢 강사 해설 (약 5분)

- 비대칭 라우팅: 정방향 traceroute가 역방향 경로 X
- 클라우드 (AWS·GCP)는 ICMP 거의 차단 → TCP traceroute
- MTU mismatch가 "ping 됨, 큰 데이터 안 됨" 증상
- 다음 실습 #11 iperf3+tc로 대역폭/지연 시뮬

🚫 자주 만나는 오류

- traceroute 모두 * → ICMP 차단 → `-T -p 443` (TCP)
- mtr 권한 → `sudo` 또는 setcap
- WiFi 첫 hop 지연 큼 → 무선 자체 지연
- 결과가 매번 다름 → ECMP 멀티 경로

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

23 / 61

Network Tracing — Expected Output

```
$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
. 64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=42.3 ms
. 64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=41.8 ms
. 64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=44.1 ms
. 64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=42.7 ms
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 41.821/42.711/44.105/0.836 ms

$ traceroute -n 8.8.8.8
1 192.168.1.1 1.234 ms 1.421 ms 1.521 ms
2 10.0.0.1 4.512 ms 4.621 ms 4.802 ms
3 211.45.32.1 12.342 ms 12.421 ms 12.501 ms
4 142.250.221.50 28.241 ms 28.342 ms 28.421 ms
5 8.8.8.8 42.341 ms 42.421 ms 42.521 ms

$ mtr --report -c 100 -n 8.8.8.8
HOST: laptop      Loss% Snt    Last    Avg     Best   Wrst   StDev
1. 192.168.1.1     0.0% 100    1.2     1.4     1.0    3.2    0.3
2. 10.0.0.1        0.0% 100    4.5     4.6     4.2    6.8    0.5
3. 211.45.32.1     12.0% 100    12.3    13.1    11.8   28.4   2.1
4. 142.250.221.50  0.0% 100    28.2    28.4    27.9   32.1   0.8
5. 8.8.8.8         0.0% 100    42.3    42.7    41.8   45.2   0.6
```

실습 #8 — mtr:traceroute:ping — 네트워크 경로 추적 ★

실습 #9 — 시스템 자원 모니터링 — 8종 도구 통합 — 환경·시나리오 ★★

🎯 목적

- 1분 안에 "이 서버 지금 바쁜가?" 판단 (빠른 진단 4종)
- Linux **자원 종합 관찰** — CPU-Mem-Disk-NW (심층 분석 4종)
- `%iowait` · `%steal` 같은 고급 지표 판독
- `stress-ng` 로 인위적 부하 발생시켜 도구별 관점 비교

🔧 도구 (8종)

- 빠른 진단: `htop`, `top`, `free`, `df` / `du`
- 심층 관찰: `vmstat`, `iostat`, `sar`, `dstat`
- `sysstat` (vmstat-iostat-sar 포함), `stress-ng`, `tmux`
- 실습 #1의 VM에서 진행

🖥️ 시나리오 (약 30분)

```
# 1) 설치
sudo apt update
sudo apt install -y htop sysstat stress-ng tmux dstat
```

실습 #9 — 시스템 자원 모니터링 — 8종 도구 통합 — 분석·해설 ★

🔍 관련 포인트

- 1. 빠른 진단 → 심층 관찰 2단계 — `uptime` · `free` · `df` 로 1분 진단 → 의심 가는 축만 `vmstat` / `iostat` / `sar` 로 깊이 보기
- 2. `vmstat` `wa` 컬럼 — 디스크 병목 (CPU가 IO 기다림)
- 3. `vmstat` `st` 컬럼 — Steal Time (호스트가 vCPU 빼앗김 = 가상화)
- 4. `iostat` `%util` — 95%+ = 디스크 포화 (NVMe-RAID 검토)
- 5. `sar -d` 의 `await` — IO 응답 시간 (mSec)
- 6. `htop` 보라색 = `%iowait` (디스크 대기)
- 7. `MemAvailable` vs `free` — 캐시 회수 가능 메모리 (진짜 부족 판단 기준)

✅ 결론

- 빠른 진단 4종 (`htop-free-df-uptime`) = 운영 1분 진단의 표준
- 심층 관찰 4종 (`vmstat-iostat-sar-dstat`) = “왜 바쁜가” 근본 원인 분석
- 같은 부하라도 도구마다 보는 관점이 다름 → 교차 검증이 핵심
- `sysstat` 백그라운드 수집 = 사후 분석의 핵심

📢 강사 해설 (약 7분)

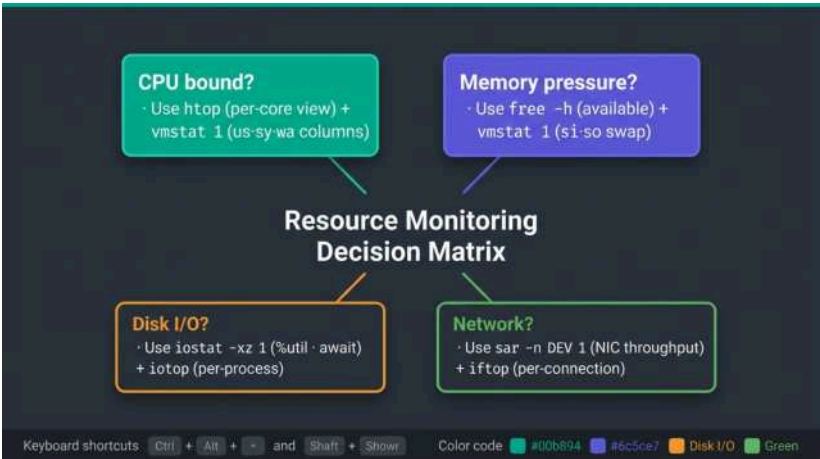
- `load avg` = “실행 중 + 실행 대기” 프로세스 평균. 8 코어 서버에서 `load 8`이면 100%, 16이면 200% 대기
- `buff/cache` = 회수 가능, 실제 부족 판단은 `available`
- `vmstat 1 5` — 5초만 보고 종료 (스크립트 친화)
- `iostat` 첫 출력 = 부팅 이후 누적 (무시), 두 번째부터 실측
- `sar` 데이터는 `/var/log/sa/` 에 저장 (10분 간격)
- `dstat` 는 deprecated 경고가 뜨지만 한 화면에 다 보여서 운영 현장에서 여전히 선호

🔗 컨테이너 환경 `ocAdvisor` (실습 #15) 로 확장

26 / 61

🐛 자주 만나는 오류

- `htop: command not found` → `sudo apt install -y htop`



실습 #9 — `htop-vmstat-iostat-sar` — 자원 4종 동시 관찰 ★★

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #10 — Wireshark — TCP 3-way handshake 패킷 분석 — 환경·시나리오 ★

🎯 목적

- TCP 3-way handshake 패킷 단위 관찰
- Wireshark 필터·통계 사용법
- 운영 네트워크 분석 1차 도구

🔧 도구

- Wireshark (wireshark.org/download.html)
- tcpdump (CLI 보조, 기본 내장)
- curl (트래픽 발생)

💻 시나리오 (약 20분)

```
# 1) 도구 설치
sudo apt install -y wireshark tshark tcpdump
sudo usermod -aG wireshark $USER # 비특권 캡처
# 재로그인 필요
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

28 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #10 — Wireshark — TCP 3-way handshake 패킷 분석 — 분석·해설

🔍 관전 포인트

1. 3-way: SYN → SYN-ACK → ACK 패킷 # 1, 2, 3 순서
2. Seq/Ack 번호 — Seq 0 → Ack 1 (Initial Sequence Number)
3. Window — 받을 수 있는 버퍼 (혼잡 제어 시작점)
4. Flow Graph — 시간순 시각화 (Statistics 메뉴)
5. 재전송 — tcp.analysis.retransmission 필터

✅ 결론

- TCP는 신뢰성 = 3-way + Seq/Ack + 재전송
- Wireshark 1시간 = SE 1년 = TCP 동작 이해 가속
- tshark CLI = 운영 서버에서도 분석 가능

📢 강사 해설 (약 5분)

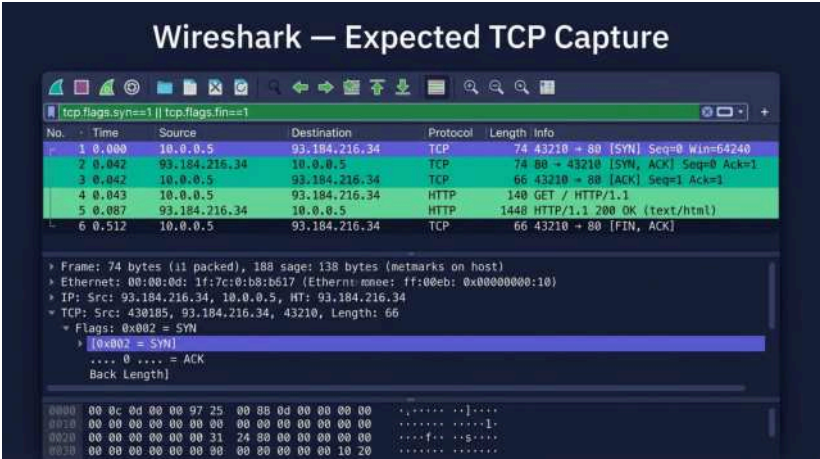
- ISN (Initial Sequence Number) — 보안상 무작위 (예전엔 예측 가능했음)
- TCP Fast Open (TFO) — 3-way 첫 패킷에 데이터 동봉
- TLS handshake (TCP 위) = 추가 2-RTT, TLS 1.3은 1-RTT
- 다음 실습 #11 iperf3+tc로 패킷 수준 영향 측정

🚫 자주 만나는 오류

- 캡처 권한 → usermod -aG wireshark + 재로그인
- 패킷 0개 → 인터페이스 확인 (-i any 또는 -D 리스트)
- HTTPS는 페이로드 암호화 → TLS 키 (SSLKEYLOGFILE) 필요
- Display filter vs Capture filter 구분 다름

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

29 / 61



실습 #10 — Wireshark — TCP 3-way handshake 패킷 분석 ★★

실습 #11 — *iperf3 + tc* — 대역폭 측정 + 지연·손실 시뮬 — 환경·시나리오



🎯 목적

- 실측 대역폭 (`iperf3`)
- 인위적 지연·손실 주입 (`tc netem`)
- BDP (Bandwidth-Delay Product) 이해

🔧 도구

- `iperf3` (apt install iperf3)
- `tc` (iproute2 기본 내장)
- 2 VM 또는 호스트↔VM

💻 시나리오 (약 20분)

```
# == Phase 1: Baseline ==
# 1) 서버 (VM)
sudo apt install -y iperf3
iperf3 -s
```


실습 #11 — *iperf3 + tc* — 대역폭 측정 + 지연·손실 시뮬 — 분석·해설 ★

🔍 관전 포인트

- 1. **Baseline Gbps** — Host-Only 가상 NIC도 Gbps 가능
- 2. **-P 4 병렬** — 단일 TCP 흐름의 한계 우회
- 3. **tc netem delay** — TCP cwnd가 BDP 한계로 작아짐
- 4. **loss 1%** — 재전송 폭증 (Retr 컬럼)
- 5. **TBF (Token Bucket)** — 정확한 대역폭 제한

✅ 결론

- **BDP = Bandwidth × RTT** — 지연이 늘면 처리량 ↓
- 운영 WAN 환경을 lab에서 재현 = **tc netem** 한 줄
- iperf3 + tc 조합 = NW 성능 진단의 핵심

📢 강사 해설 (약 5분)

- TCP 혼잡 제어: CUBIC (기본) / BBR (Google, RTT 기반)
- **sysctl net.ipv4.tcp_congestion_control=bbw** 로 변경 가능
- **--cport** 로 클라 포트 고정 (방화벽 통과 시)
- 다음 실습 #12 HAProxy로 다중 백엔드 분산

🔥 자주 만나는 오류

- **Address already in use** → **iperf3 -p 5202** (포트 변경)
- 손실률 너무 높음 → 호스트 NIC 부하 (VM 4G RAM 권장)
- **tc qdisc del** 안 됨 → 다른 qdisc 충돌, 재시작
- **tbw** vs **htb** — TBF는 단순, HTB는 클래스 기반 분할

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

32 / 61

iperf3 + tc — Expected Output

Phase 1 Baseline	Phase 2 with tc
<pre>\$ iperf3 -c 192.168.56.10 -t 10 · Connecting to host 192.168.56.10, port 5201 · [5] local 192.168.56.1 port 41234 connected to 192.168.56.10 port 5201 [ID] Interval Transfer Bitrate Retr Cwnd [5] 0-1.00 sec 112 MBytes 942 Mbits/sec 0 3.07 MBytes · [5] 1-2.00 sec 112 MBytes 941 Mbits/sec 0 3.07 MBytes · [5] ... · [5] 0.0-10 sec 1.10 GBytes 942 Mbits/sec 0 sender · [5] 0.0-10 sec 1.10 GBytes 941 Mbits/sec receiver</pre>	<pre>\$ sudo tc qdisc add dev eth0 root netem delay 50ms loss 1% · · \$ iperf3 -c 192.168.56.10 -t 10 · [5] 0-1.00 sec 22.4 MBytes 188 Mbits/sec 421 1.42 MBytes · [5] 1-2.00 sec 22.1 MBytes 185 Mbits/sec 398 1.38 MBytes · [5] ... · [5] 0.0-10 sec 222 MBytes [5] 0.0-10 sec 222 MBytes 187 Mbits/sec 4231 sender · \$ sudo tc qdisc del dev eth0 root # cleanup</pre>

실습 #11 — *iperf3 + tc* — 대역폭 측정 + 지연·손실 시뮬 ★★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

33 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #12 — *HAProxy* — *L7 로드밸런서 + 헬스체크* — 환경·시나리오 ★

🎯 목적

- **L7 LB** — HTTP 헤더 기반 분산
- 헬스체크 (`option httpchk`)로 자동 Failover
- HAProxy **stats page** 운영 가시화

🔧 도구

- `haproxy` 2.x (apt install)
- Nginx ×2 (실습 #3 컨테이너 재사용)
- `curl` (검증)

💻 시나리오 (약 20분)

```
# 1) 백엔드 2개 Nginx 컨테이너
docker run -d --name be1 -p 8001:80 nginx
docker run -d --name be2 -p 8002:80 nginx
# 응답 구분용 페이지
docker exec be1 sh -c "echo 'Backend-1' > /usr/share/nginx/html/index.html"
docker exec be2 sh -c "echo 'Backend-2' > /usr/share/nginx/html/index.html"
docker exec be1 sh -c "echo 'OK' > /usr/share/nginx/html/health"
docker exec be2 sh -c "echo 'OK' > /usr/share/nginx/html/health"
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

34 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #12 — *HAProxy* — *L7 로드밸런서 + 헬스체크* — 분석·해설

🔍 관전 포인트

1. `balance roundrobin` — 순환 배정 (기본)
2. `option httpchk GET /health` — L7 헬스체크 (단순 TCP 보다 정확)
3. `fall 3 rise 2` — 3회 연속 실패 시 down, 2회 연속 성공 시 up
4. **stats page** — 운영 가시성 + Drain-Maintenance 가능
5. **mode http** — TCP vs HTTP (HTTP는 헤더 인식X-Forwarded-For)

✅ 결론

- 운영 L7 LB의 표준 = HAProxy + Stats + health check
- 헬스체크가 fail-over의 핵심 — `/health` 엔드포인트 필수
- 세션 sticky: `cookie SRV insert indirect` 한 줄로 가능

🗣️ 강사 해설 (약 5분)

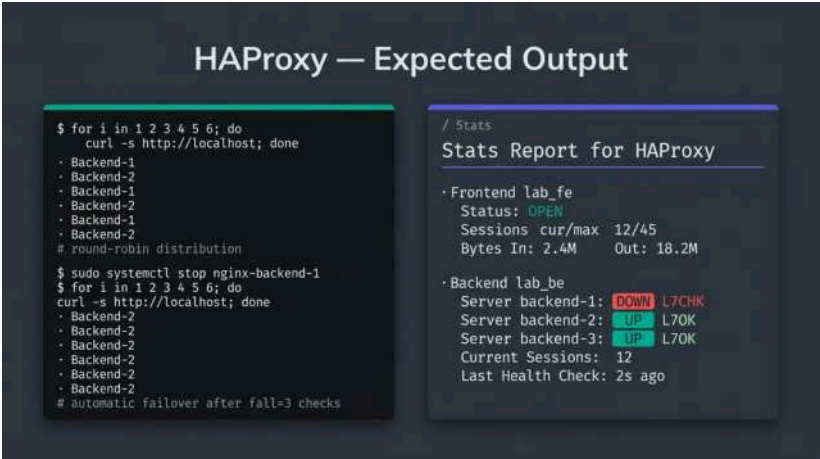
- HAProxy vs Nginx LB: HAProxy는 LB 전용·세밀, Nginx는 멀티역할
- ACL (`acl is_api path_beg /api`) — 경로별 라우팅
- `mode tcp` = L4 LB (Redis-MySQL 등)
- 다음 실습 #13 Keepalived로 HAProxy 자체의 HA 확장

🐛 자주 만나는 오류

- `cannot bind socket` → 80 포트 사용 중 (Nginx 중지)
- `health check failed` → backend `/health` 엔드포인트 부재
- 분산 안 됨 → KeepAlive로 같은 백엔드만 (`http-reuse safe`)
- 502 → 모든 백엔드 down (HAProxy 로그 확인)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

35 / 61



실습 #12 — HAProxy — L7 로드밸런서 + 헬스체크 ★★

실습 #13 — Keepalived — VRRPv3 Active/Standby VIP — 환경·시나리오 ★★

🎯 목적

- VRRPv3 Active/Standby HA 구성
- VIP (Virtual IP) 자동 이동
- 클러스터 fail-over < 3초

🔧 도구

- keepalived (apt install)
- 2 VM (lb-1, lb-2)
- 공통 네트워크 (Host-Only 192.168.56.0/24)

💻 시나리오 (약 15분)

```
# === 양쪽 VM 공통 ===
sudo apt install -y keepalived

# == lb-1 (MASTER) /etc/keepalived/keepalived.conf ==
sudo tee /etc/keepalived/keepalived.conf <<'EOF'
```

실습 #13 — Keepalived — VRRPv3 Active/Standby VIP — 분석·해설 ★

🔍 관전 포인트

- 1. `priority` 차이 — 높은 쪽이 Master
- 2. `virtual_router_id` — 같은 LAN의 다른 VRRP와 구분
- 3. `advert_int 1` — 1초마다 multicast (224.0.0.18)
- 4. **Gratuitous ARP** — Master 변경 시 ARP 캐시 갱신
- 5. `auth_pass` — VRRP 단순 인증 (LAN 보안)

✅ 결론

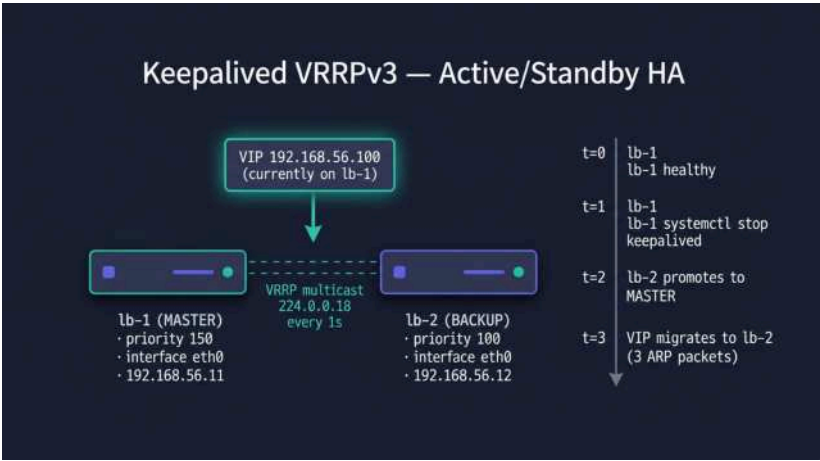
- VRRP = "가장 IP가 살아있는 서버를 따라다닌다"
- HAProxy + Keepalived = 무중단 L7 LB의 표준
- fail-over는 < 3초 ($advert_int \times 3 = 3초\ timeout$)

🗣️ 강사 해설 (약 5분)

- VRRPv3 = IPv6 지원 (v2는 IPv4 only)
- Split-brain 위험: 양쪽이 MASTER → VIP 충돌 (네트워크 단절 시)
- Cloud는 VRRP 멀티캐스트 차단 → AWS는 Route53 / ALB 사용
- 다음 실습 #18 Pacemaker로 리소스 단위 HA 확장

🚫 자주 만나는 오류

- VIP 이동 안 함 → 멀티캐스트 차단 (VirtualBox Adapter "허용 모든 가상머신")
- 양쪽 다 MASTER → `priority-virtual_router_id` 동일
- VRRP 패킷 0 → 인터페이스 명 오타 (eth0 vs ens33)
- `firewalld / ufw` 차단 → VRRP 프로토콜 (proto 112) 허용



실습 #13 — Keepalived — VRRPv3 Active/Standby VIP ★★

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #14 — PostgreSQL — pgbench 부하 + shared_buffers 튜닝 — 환경·시나리오 ★

🎯 목적

- TPC-B 표준 벤치마크 실측
- DB OLTP 부하 패턴 자원 관찰
- `shared_buffers` 튜닝 전후 TPS 비교

🔧 도구

- PostgreSQL 16 (postgresql.org/download/linux/ubuntu)
- `pgbench` (postgresql-contrib 포함)
- `htop`, `iostat` (자원 관찰)

💻 시나리오 (약 20분)

```
# 1) 설치
sudo apt install -y postgresql-16 postgresql-contrib

# 2) DB 생성 + 권한
sudo -u postgres createdb bench
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

40 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #14 — PostgreSQL — pgbench 부하 + shared_buffers 튜닝 — 분석·해설

🔍 관전 포인트

1. `-c 10 -j 4` — 10 클라이언트 (병렬), 4 OS 스레드
2. `TPS-latency-stddev` — 평균만 보면 long-tail 놓침
3. `shared_buffers` — 메모리 캐시, IOPS ↓ TPS ↑
4. `-S` (SELECT only) vs default (R/W 혼합)
5. `pg_stat_activity` — 활성 쿼리 가시화

✅ 결론

- pgbench = DB 벤치마크의 표준 도구
- 튜닝 = `shared_buffers` 25% · `work_mem` · `effective_cache_size`
- 운영 측정은 워밍업 후 (`-T 600` 이상)

🔧 강사 해설 (약 5분)

- TPC-B는 단순 모델 (계좌 이체) — 실제 OLTP는 더 복잡
- `pgbench -f custom.sql` 로 자체 쿼리 부하 가능
- 디스크가 HDD면 IOPS 병목 → 튜닝 효과 적음
- 다음 실습 #15 cAdvisor로 컨테이너 자원 격리

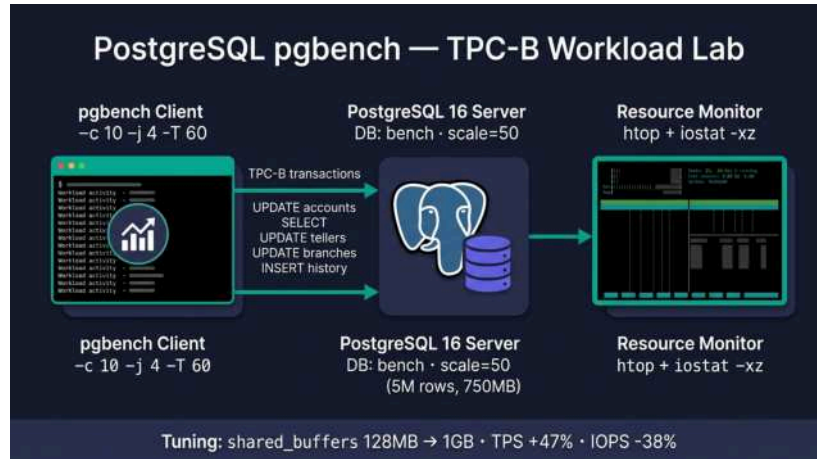
🐛 자주 만나는 오류

- `could not connect to server` → PostgreSQL 미실행
- `permission denied` → `sudo -u postgres` 누락
- `shared_buffers` 너무 크게 → OS 메모리 부족
- TPS 변동 큼 → 워밍업 부족 (`-T 600`)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

41 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)



실습 #14 — PostgreSQL — pgbench 부하 + sharedbuffers 튜닝 ★★

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #15 — Docker + cAdvisor + cgroup — 컨테이너 자원 격리 — 환경.시나리오 ★★

🎯 목적

- 컨테이너 **자원 격리** (cgroup v2) 실증
- Docker `--cpus` · `--memory` → cgroup 매핑
- cAdvisor** 로 실시간 가시화

🔧 도구

- Docker Engine 24+ (실습 #3 환경)
- cAdvisor (Google, Docker 이미지)
- `alpine` + `stres6-ng` (부하)

💻 시나리오 (약 15분)

```
# 1) cAdvisor 띄우기
docker run -d --name cadvisor \
  -p 8080:8080 \
  -v /:/rootfs:ro \
  -v /var/run:/var/run:ro \
```

실습 #15 — *Docker + cAdvisor + cgroup* — 컨테이너 자원 격리 — 분석·해설 ★

🔍 관련 포인트

- 1. `--cpus="1"` → `cgroup` `cpu.max = 100000 100000`
- 2. `--memory="512m"` → `memory.max = 536870912`
- 3. `stress-ng --cpu 4` 해도 4 코어 사용 X (제한)
- 4. **OOM-Kill** — 메모리 초과 시 커널이 프로세스 강제 종료
- 5. **cAdvisor의 ceiling line** — 한도 시각화

✅ 결론

- 컨테이너 격리 = **cgroup v2 + namespace** 위에서 동작
- K8s의 `requests/limits` 도 결국 같은 cgroup
- cAdvisor → Prometheus → Grafana = 운영 관측의 표준

📢 강사 해설 (약 5분)

- cgroup v1 vs v2: v2는 단일 hierarchy, 메모리-CPU-I/O 통합
- `cpu.max` 형식: `quota period` (마이크로초)
- `--cpus` 와 `--cpu-shares` 차이 (절대 vs 상대 가중치)
- 다음 실습 #19에서 cAdvisor 메트릭을 Prometheus로 수집

🚫 자주 만나는 오류

- cAdvisor 404 → 호스트 OS 지원 (Ubuntu 22.04 OK)
- OOM 안 일어남 → swap 활성화 (`--memory-swap=0`)
- cgroup 경로 다름 → cgroup v1 환경 (`/sys/fs/cgroup/memory/...`)
- alpine apk → 네트워크 (NAT) 필요



실습 #15 — *Docker + cAdvisor + cgroup* — 컨테이너 자원 격리 ★★

실습 #16 — nftables (+ iptables 호환) + fail2ban + nmap — 호스트 방화벽 + 자동 차단 — 환경·시나리오 ★

🎯 목적

- **nftables** 호스트 방화벽 (iptables 후속 · 동일 영역의 대체 도구)
- **iptables 호환 사용법** — 기존 룰셋 마이그레이션 경로까지
- **fail2ban** 으로 SSH 무차별 자동 차단
- **nmap** 으로 외부에서 검증

🔧 도구

- **nftables** (Ubuntu 22.04 기본) — 본 실습 메인
- **iptables-nft** (호환 레이어, 기존 iptables 명령 그대로 nftables로 변환)
- **iptables-translate** (기존 iptables 룰을 nft 문법으로 변환하는 마이그레이션 도구)
- **fail2ban** (apt install)
- **nmap** (apt install, 검증용)

💻 시나리오 (약 25분)

=== nftables 방화벽 ===

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

46 / 61

실습 #16 — nftables (+ iptables 호환) + fail2ban + nmap — 호스트 방화벽 + 자동 차단 — 분석·해설

🔍 관련 포인트

1. **policy drop** — 기본 거부 + 명시 허용 (whitelist)
2. **ct state established,related accept** — stateful (필수)
3. **iptables 호환** — **iptables-nft** · **iptables-translate** 로 기존 룰셋 무중단 마이그레이션
4. **fail2ban**의 **findtime** — 시간 윈도우
5. **/var/log/auth.log** — SSH 실패 기록
6. **nmap 상태**: open / closed / filtered

✅ 결론

- **nftables policy drop** = 운영 호스트 방화벽 표준
- **iptables → nftables 마이그레이션** — **iptables-restore-translate** 로 룰셋 자동 변환, **iptables-nft** 로 명령 호환 유지
- **fail2ban** = SSH 무차별 차단의 1차 방어
- **nmap 자체 점검** = 발주 전 보안 진단

💡 강사 해설 (약 5분)

- **iptables** (옛) → **nftables** (현대) 전환 — Ubuntu 22.04 부터 **iptables** 명령이 내부적으로 **iptables-nft** 로 동작
- 레거시 시스템 인계 받으면 **iptables -L** 부터 확인 → **iptables-restore-translate** 로 nft 룰셋으로 옮긴 뒤 검증
- fail2ban 백엔드: **systemd** (Ubuntu 22.04) vs **polling**
- **recidive** jail = 반복 IP 영구 차단
- 다음 실습 #20 Wazuh = HIDS-SIEM 통합 확장

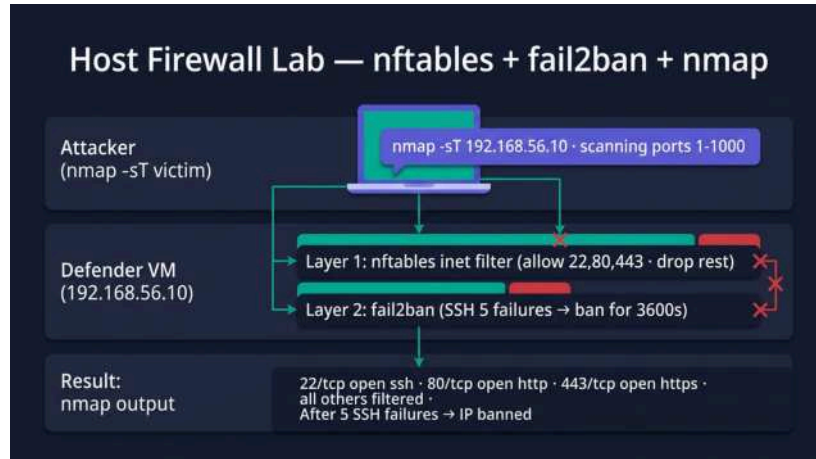
🚫 자주 만나는 오류

- nft 적용 후 SSH 끊김 → 22번 허용 확인
- **fail2ban** 미작동 → **journalctl -u fail2ban**
- **findtime** 너무 짧음 → 정상 사용자도 차단
- nmap 권한 → **sudo nmap -ss** (SYN scan)

fail2ban 미작동 → journalctl -u fail2ban

47 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)



실습 #16 — nftables + fail2ban + nmap — 호스트 방화벽 + 자동 차단 ★★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

48 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #17 — mdadm — RAID-5 구성 + 디스크 장애 → rebuild — 환경·시나리오 ★★

🎯 목적

- RAID-5 구성·동작 원리 이해
- 디스크 1개 장애 시뮬 + degraded 동작
- Rebuild 과정 + 시간 추정

🔧 도구

- mdadm (apt install)
- loop device (실제 디스크 없이 시뮬)

💻 시나리오 (약 25분)

```
# 1) 도구 설치
sudo apt install -y mdadm

# 2) 5개 가상 디스크 (4 active + 1 spare)
for i in 0 1 2 3 4; do
  sudo dd if=/dev/zero of=/tmp/disk${i}.img bs=1M count=200
  sudo losetup -f --show /tmp/disk${i}.img
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

49 / 61

실습 #17 — mdadm — RAID-5 구성 + 디스크 장애 → rebuild — 분석·해설 ★

🔍 관전 포인트

- 1. [UUUU] — 4 디스크 모두 정상
- 2. [_UUU] — 첫 디스크 죽음 (degraded)
- 3. 데이터 무결성 — RAID-5는 1개 장애에도 데이터 OK
- 4. Rebuild 시간 — 디스크 크기·속도에 비례 (현실: TB → 수 시간)
- 5. spare — 자동 교체용 hot spare 지정 가능

✅ 결론

- RAID-5 = "디스크 1개 죽어도 데이터 살아있음"
- Rebuild 중 두 번째 장애 시 데이터 손실 — RAID-6 권장
- 운영: 디스크 6개 이상 시 RAID-6 또는 RAID-10

🗣️ 강사 해설 (약 5분)

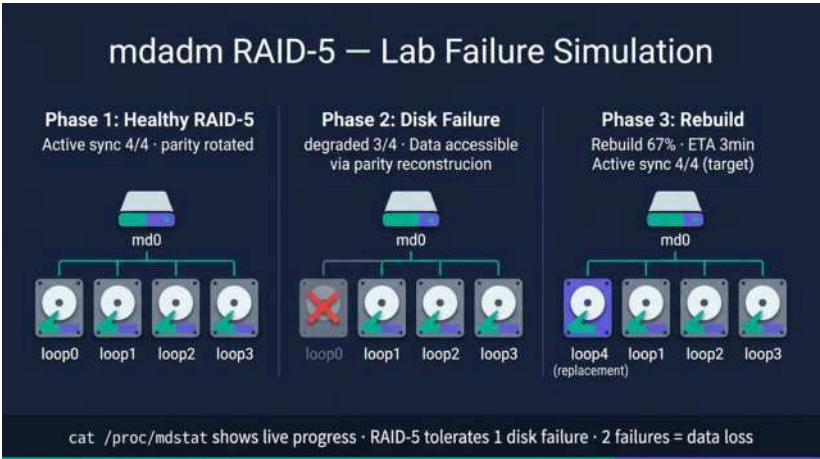
- RAID-5 vs RAID-6: 패리티 1개 vs 2개 (2 장애 허용)
- RAID-10 = mirror + stripe (성능 + 안전, 용량 50%)
- URE (Unrecoverable Read Error): 18TB 디스크에선 rebuild 위험
- 다음 실습 #18 Pacemaker로 호스트 단위 HA 확장

🚫 자주 만나는 오류

- loop device 한도 → `sudo modprobe loop max_loop=64`
- rebuild 너무 빠름 → `echo 1000 > /proc/sys/dev/raid/speed_limit_min`
- `mdadm.conf` 누락 → 재부팅 시 array 안 살아남
- spare 안 들어옴 → `--add` 후 `--manage --add`

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

50 / 61



실습 #17 — mdadm — RAID-5 구성 + 디스크 장애 → rebuild ★ ★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

51 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #18 — *Pacemaker + Corosync* — 2-node 클러스터 HA — 환경·시나리오 ★

🎯 목적

- **Pacemaker + Corosync** 2-node HA 클러스터
- 리소스 단위 **fail-over** (VIP + nginx 묶음)
- `pcs` CLI 운영

🔧 도구

- `pacemaker`, `corosync`, `pcs` (apt install)
- 2 VM (node1, node2)
- `nginx` (리소스 대상)

💻 시나리오 (약 30분)

```
# === 양 노드 공통 ===
# 1) 설치
sudo apt install -y pacemaker corosync pcs nginx
sudo systemctl disable --now nginx # Pacemaker가 관리
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

52 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #18 — *Pacemaker + Corosync* — 2-node 클러스터 HA — 분석·해설

🔍 관전 포인트

1. **Corosync** = 통신, **Pacemaker** = 리소스 관리
2. `stonith-enabled=false` — 실습용만, 운영 절대 X (Split-brain)
3. `resource group` — VIP + nginx 같은 노드에 함께
4. `op monitor` — 헬스체크 간격
5. `pcs resource move` — 수동 이전

✅ 결론

- 2-node HA = "리소스가 살아있는 노드를 따라 다닌다"
- 운영은 **3 노드 이상** (quorum) + **STONITH** 펜싱 필수
- `systemd:nginx` = 모든 systemd unit을 클러스터 리소스화 가능

🔧 강사 해설 (약 5분)

- Split-brain: 양쪽 노드가 서로 죽었다고 판단 → 동시에 master
- 펜싱 = "확실히 죽이기" (전원 차단, fence_ipmilan / fence_vmware)
- 3-node quorum = 다수결 (2/3 살아있으면 OK)
- 다음 실습 #19 Prometheus로 클러스터 메트릭 수집

🐛 자주 만나는 오류

- 인증 실패 → `pcs host auth` 다시
- Corosync `Bind failed` → `bindnetaddr` 양쪽 동일 LAN
- VIP 안 붙음 → 같은 서브넷 + 다른 노드 IP 사용 X
- `pcs cluster start` 실패 → `journalctl -u corosync`

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

53 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

Pacemaker pcs — Expected Output

```

$ sudo pcs status
• Cluster name: lab-cluster
• Stack: corosync
• Current DC: node1 (version 2.1.5)
• 2 nodes configured
• 2 resource instances configured
• Online: [ node1 node2 ]
• Full list of resources:
  • Resource Group: web
    • web_vip (ocf::heartbeat:IPaddr2):
      Started node1
    • web_nginx (systemd:nginx): Started
      node1
  • Daemon Status:
    • corosync: active/enabled
    • pacemaker: active/enabled
    • pcsd: active/enabled

```

```

$ sudo pcs status
• ...
• Online: [ node2 ]
• OFFLINE: [ node1 ]
• Resource Group: web
  • web_vip (ocf::heartbeat:IPaddr2):
    Started node2
  • web_nginx (systemd:nginx): Started
    node2
  • # resources migrated successfully
$ ip addr show eth0
inet 192.168.56.22/24
inet 192.168.56.100/24 # VIP migrated
$ curl http://192.168.56.100
<h1>Lab HA Nginx on node2</h1>

```

실습 #18 — Pacemaker + Corosync — 2-node 클러스터 HA ★ ★

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #19 — Prometheus + Grafana + Node Exporter — 관측성 스택 — 환경·시나리오 ★ ★

🎯 목적

- Prometheus 메트릭 수집 + PromQL
- Grafana 시각화 (대시보드)
- Alertmanager 알람 라우팅

🔧 도구

- Prometheus 2.55 (prometheus.io/download)
- Grafana 11 (grafana.com/grafana/download)
- Node Exporter (Prometheus 진영)

💻 시나리오 (약 25분)

```

# === Node Exporter ===
# 1) 다운로드 + systemd 등록
cd /tmp
wget https://github.com/prometheus/node_exporter/releases/download/v1.8.2/node_exporter-1.8.2.linux-amd64.tar.gz
tar xzf node_exporter-1.8.2.linux-amd64.tar.gz

```

실습 #19 — Prometheus + Grafana + Node Exporter — 관측성 스택 — 분석·해설 ★

🔍 관련 포인트

- 1. `scrape_interval` — 15s 표준 (해상도)
- 2. Pull model — Prometheus가 `/metrics` 폴
- 3. PromQL `rate()` vs `irate()` — 5분 평균 vs 순간
- 4. 메트릭 타입 — Counter / Gauge / Histogram
- 5. 대시보드 ID 1860 — 가장 유명한 Node Exporter Full

✅ 결론

- 관측성 = 메트릭 (Prometheus) + 로그 + Trace 3종
- Prometheus pull + ServiceDiscovery (K8s 환경)
- Grafana 1860 대시보드 = 사실상 표준

🗣️ 강사 해설 (약 5분)

- Pull vs Push: Pull은 Prometheus 표준, Push는 Pushgateway 별도
- TSDB 보관: 기본 15일, 길게 가져가려면 Thanos-Cortex-VictoriaMetrics
- AlertManager → Slack/Email/PagerDuty 라우팅
- 다음 실습 #20 Wazuh = 로그/이벤트 측면 보완

🚫 자주 만나는 오류

- Targets DOWN → 방화벽 (9100/9090)
- PromQL syntax → 공백·중괄호
- Grafana 데이터 없음 → 시간 범위 (오른쪽 위 Last 5min)
- `prometheus.yml` YAML 들여쓰기 (스페이스 2칸)

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현



실습 #19 — Prometheus + Grafana + Node Exporter — 관측성 스택 ★ ★

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #20 — Wazuh — SIEM + HIDS + EDR 통합 관제 — 환경·시나리오



🎯 목적

- SIEM + HIDS + EDR 통합 관제
- FIM (File Integrity) 변경 감지
- MITRE ATT&CK 매핑

🔧 도구

- Wazuh 4.7 (documentation.wazuh.com)
- Manager VM (4 vCPU / 8GB RAM 권장)
- Agent VM (별도)

💻 시나리오 (약 30분)

```
# == Wazuh Manager (All-in-One 설치) ==
# 1) Manager VM에서 (8GB RAM 권장)
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
sudo bash ./wazuh-install.sh -a
# (3부: 설도 설치 - Indexer + Manager + Dashboard 통합)
```

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

58 / 61

IT 인프라 아키텍처 설계 · Session 20 · 실습 모음 (Hands-on Labs · 20 시나리오)

실습 #20 — Wazuh — SIEM + HIDS + EDR 통합 관제 — 분석·해설

🔍 관전 포인트

1. **Manager + Indexer + Dashboard** 3종 통합 설치
2. **Agent enroll** = manager에 자동 등록
3. **FIM rule (550)** = 파일 무결성 변경
4. **MITRE 매핑** = Wazuh rule → ATT&CK technique
5. **Vulnerability Detector** = OS 패키지 CVE 검색

✅ 결론

- Wazuh = **OSSEC fork** + **현대 UI** + **ELK 기반**
- HIDS + SIEM + EDR 단일 솔루션
- 공공·중견기업 표준 OSS SIEM

🔧 강사 해설 (약 5분)

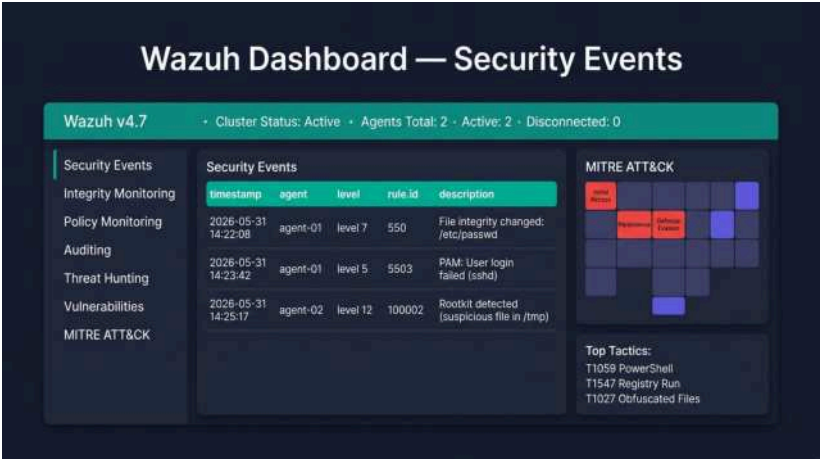
- Wazuh vs Splunk ES: Wazuh 무료 OSS, Splunk 상용 (라이선스 큼)
- Decoder + Rule = 로그 → Alert 변환
- API: <https://manager:55000> (Bearer token)
- Suricata IDS 통합 → 네트워크 IDS 결합

🐛 자주 만나는 오류

- 8GB RAM 부족 → Indexer OOM (heap 2~4GB)
- Agent 미등록 → Manager 1514 포트 차단
- Dashboard 접속 X → 443 포트 + 자체 서명 인증서 (브라우저 예외)
- FIM rule 안 옴 → `<directories realtime="yes">/etc</directories>`

© 2026 젠아이랩스 (GenAI Labs) · 강사 박수현

59 / 61



실습 #20 — Wazuh — SIEM + HIDS + EDR 통합 관제 ★

Session 20 — 학습 정리 + 다음 단계

✅ 핵심 정리 — 익힌 19종

- 기초 7: VM-Nginx-Docker-SSH카-HTTP/DNS-백업-NW진단 (자원 모니터링은 로 통합)
- 운영 8: 자원모니터링 8종-TCP-대역폭-LB-VRPP-DB부하-cgroup-방화벽
- 고급 4: RAID-클러스터-관측상-SIEM

🎯 산출물 과제

- 본인 환경에 VM 3대 표준 토폴로지 구축
- 중 1개를 골라 운영 SOP 작성
- Prometheus + Grafana로 본 VM들 통합 모니터링
- Wazuh agent를 모든 VM에 enroll
- HAProxy + Keepalived L7 HA 구축

🌀 본 세션 한 문장 정리

"TA의 의사결정은 결국 각 도구를 직접 다뤄 본 경험에서 출발한다. 19개 실습은 운영의 가장 작은 단위."

🔄 후속 학습 경로

트랙	다음 단계
컨테이너	Kubernetes (k3s-minikube) → Helm → ArgoCD
관측성	Loki (로그) + Tempo (Trace) → LGTM 스택
CI/CD	Jenkins / GitLab Runner / Argo Workflows
IaC	Terraform + Ansible 통합
클라우드	AWS LZ / Azure Landing Zone
보안	OpenSCAP + STIG + 모의 해킹 (Kali)
DB 고급	PostgreSQL HA (Patroni) + Read Replica
NW 고급	BGP + VXLAN + Calico

💡 실전 한 줄

- 운영 = "실제로 다룰 줄 아는 도구 × 자동화" — 둘 다 필요
- Day 1 (이론) + Day 2 (설계) + Day 3 (실습) = TA 1주 표준
- 실습 환경은 재사용 가능한 VM 스냅샷으로 보관 (다음 강의 가속)

📚 추천 자료

- Linux 공식: kernel.org/doc/html/latest
- K8s: kubernetes.io/docs
- Prometheus: prometheus.io/docs
- Wazuh: documentation.wazuh.com