


# 퍼블릭 클라우드 vs 프라이빗 클라우드

## ① 관리 편의성 (Manageability)

항목	퍼블릭 클라우드 (예: AWS, Azure)	프라이빗 클라우드 (온프레미스)
K8s 관리	EKS/GKE 같은 매니지드 K8s 서비스 제공. 업그레이드, 백업, 모니터링 기본 제공	kubeadm, RKE 등을 통한 수동 설치 필요. 유지보수 직접 수행
노드 관리	Auto Scaling Group, Spot Instances로 자동 관리 가능	물리 서버 또는 VM 직접 관리. 확장 시 하드웨어 고려 필요
스토리지	S3/Elastic File System(EFS) 등 연동 쉬움	MinIO/Ceph 설치 필요. 성능 튜닝, 장애 대응도 직접 해야 함
배포 및 자동화	Terraform, Helm, Argo CD 등 클라우드 연동 도구와 통합 쉬움	동일 도구는 사용 가능하나, 클라우드 연동 기능은 별도 설정 필요
종합 평가	🟢 관리 편의성 우수 (모듈화/자동화 용이)	🔴 설정 복잡도 높고 유지비용 발생

 요약: 퍼블릭 클라우드는 관리에 대한 추상화 레벨이 높아 운영 부담이 적고, IaC 도구와 통합도 잘됨

### ◆ 퍼블릭 클라우드 설계 방향

- **Amazon EKS:** 매니지드 K8s 클러스터로 제어판 관리 생략
- **RDS for PostgreSQL:** DB 운영 자동화 (백업, 패치, 장애 조치)
- **S3:** 확장성 높은 객체 스토리지 사용
- **Terraform + Argo CD:** IaC 및 GitOps 기반 관리 자동화
- **Cluster Autoscaler + HPA:** 자동 리소스 확장

#### 설계 이유

- 운영 부담 최소화: 제어판 자동 관리, 자동 패치, 백업 내장
- 표준화된 관리: IaC & GitOps 조합으로 일관된 환경 유지
- 실시간 대응: Auto Scaling 및 서비스 이중화로 가용성 향상

### ◆ 프라이빗 클라우드 설계 방향

- **Kubeadm or RKE로 K8s 직접 설치:** Master/Worker 수동 구성
- **HAProxy + Keepalived:** API 서버 로드밸런싱
- **Ceph 또는 MinIO:** 스토리지 직접 구성
- **PostgreSQL Replication:** 직접 구성하여 고가용성 확보
- **Ansible + Helm:** 자동화와 배포 편의성 확보

#### 설계 이유

- 커스터마이징 유리: 전체 구조에 대한 제어 가능
- 네트워크 구성 및 보안 통제 강화 가능

- 단, 운영 복잡성 증가 → 관리 인력 필요

## ② 민첩성 (Agility)

항목	퍼블릭 클라우드	프라이빗 클라우드
K8s 클러스터 생성 속도	수 분 이내 (eksctl, gcloud 등)	직접 설치 및 구성 → 수시간 소요
CI/CD 파이프라인 연동	GitHub Actions + Argo CD, Jenkins 연동 쉬움	내부 네트워크 설정 필요. 인증/연동 복잡
오토스케일링	Cluster Autoscaler / HPA / VPA 지원	리소스 모니터링, 노드 증설까지 수동 구성
환경 분리 (Dev/Staging/Prod)	Namespace 및 여러 EKS 클러스터 구성 쉬움	클러스터 분리는 인프라, 자원 측면에서 부담 큼
배포 속도	Fast - GitOps 기반으로 빠르게 반영	중간에 인증 이슈, 인프라 한계로 지연 가능성
종합 평가	🟢 빠른 구축과 반복 배포에 유리	🔴 민첩성 확보 위해 많은 선작업 필요

🔍 요약: 빠른 구축과 확장, 자동화 측면에서 퍼블릭 클라우드가 민첩성 우위

### ◆ 퍼블릭 클라우드 설계 방향

- eksctl로 수분 내 K8s 생성
- GitHub Actions + Argo CD: 코드 변경 → 자동 배포
- Dev / Staging / Prod 분리: Namespace 또는 별도 EKS 클러스터

#### 📝 설계 이유

- 개발 주기 단축 (CI/CD 자동화)
- 실시간 배포 가능 (GitOps 기반)
- 환경 분리로 품질 유지

### ◆ 프라이빗 클라우드 설계 방향

- CI/CD 도구 직접 호스팅: Jenkins, GitLab CI 등
- 개발/운영 분리 클러스터 또는 네임스페이스 구성
- 배포 자동화 툴 사용: Argo CD or Flux

#### 📝 설계 이유

- 내부망에서 독립적으로 운영 가능
- 보안 통제가 필요한 민감 환경에 적합
- 하지만, 초기 셋업 및 배포 자동화에 시간 소요

## ③ 비용 효율성 (Cost Efficiency)

항목	퍼블릭 클라우드	프라이빗 클라우드
초기 비용	낮음 (사용한 만큼 지불)	높음 (서버/스토리지 직접 구축 필요)
운영 비용	지속적인 과금 존재 (CPU, RAM, 트래픽, 스토리지 등)	유지보수 인력/전력비/하드웨어 감가 상각
오토스케일링	비용 최적화 도구 풍부 (예: Spot Instance)	물리적 한계로 오토스케일링 미흡
CI/CD 등 도구	무료 오픈소스 + 클라우드 리소스	오픈소스만 사용 가능하나 별도 인프라 필요
HA 구성	RDS, S3, ALB 등 고가용 자원 제공 (비용 있음)	직접 구성 가능하지만 인프라 확장 시 비용 증가
종합 평가	🟡 초기 저렴하지만 장기 비용 누적 가능	🔴 초기 고비용, 장기적으로 리스크 분산 가능

🔍 요약: 단기 프로젝트나 PoC는 퍼블릭이 유리, 장기 대규모 서비스는 TCO 비교 필요

## ◆ 퍼블릭 클라우드 설계 방향

- 모든 리소스 Pay-as-you-go
- Spot 인스턴스 도입: 비용 절감
- Managed DB, S3: 유지비 절감

### 📝 설계 이유

- 초기 구축비 0에 가깝고, 빠르게 PoC 가능
- 사용량 기반 과금 → 소규모나 단기 서비스에 유리
- 예측 가능한 과금이 가능

## ◆ 프라이빗 클라우드 설계 방향

- 자체 인프라 보유 기반 구축
- Open-source 도구 위주 구성: Ceph, Jenkins, Prometheus 등

### 📝 설계 이유

- 대규모 장기 운영 시 총소유비용(TCO) 절감 가능
- 자체 자원 효율화 가능 (Idle 자원 활용 등)
- 초기 비용은 크지만, 장기적 자산으로 전환 가능

## ④ 보안 (Security)

항목	퍼블릭 클라우드	프라이빗 클라우드
네트워크 보안	VPC, NACL, SG, IAM 등 클라우드 기능으로 세분화 가능	네트워크 정책, 방화벽 구성 직접 수행
인증 및 접근 제어	IAM, OIDC, KMS 등 제공	자체 IDP 또는 Keycloak, HashiCorp Vault 등 구성
데이터 보호	S3, RDS, EBS 등에 암호화 기본 제공	MinIO, PostgreSQL에 암호화 설정 수동 적용
시크릿 관리	AWS Secrets Manager, KMS	자체 구성 필요 (예: sealed-secrets, vault)
감사 로그 및 모니터링	CloudTrail, CloudWatch 등 통합 로그 제공	ELK 스택 등 별도 로그 시스템 구축 필요

항목	퍼블릭 클라우드	프라이빗 클라우드
종합 평가	● 높은 수준의 기본 보안 + 세분화된 정책	● 유연성 있지만 구현 복잡성 높음

🔍 **요약:** 퍼블릭 클라우드는 보안 톨을 기본 제공하지만, 민감 데이터 보호 필요 시 프라이빗도 고려 가치 있음

### ◆ 퍼블릭 클라우드 설계 방향

- **IAM + VPC + SecurityGroup:** 권한 세분화 및 네트워크 격리
- **Secrets Manager + KMS:** 민감 정보 안전 저장
- **CloudTrail + GuardDuty:** 감사 및 이상 탐지

#### 📝 설계 이유

- 기본 제공 도구만으로 높은 수준의 보안 확보
- 보안 표준(ISO, SOC, HIPAA 등) 충족 가능
- 외부 노출 리스크 최소화 가능

### ◆ 프라이빗 클라우드 설계 방향

- **Private 네트워크 완전 통제**
- **Keycloak + Vault + RBAC:** 인증 및 시크릿 관리
- **NetworkPolicy + Firewall:** 내부 트래픽 제어

#### 📝 설계 이유

- 온프레미스 환경에서 민감 데이터 직접 보호
- 내부 사용자 위주의 보안 정책 수립 가능
- 보안 사고 발생 시 영향 범위 제어 용이

## 🧠 결론

평가 항목	퍼블릭 클라우드	프라이빗 클라우드
관리 편의성	● 우수	● 직접 관리 부담 큼
민첩성	● 빠름	● 초기 셋업 느림
비용 효율성	● 단기 유리	● 초기 투자 큼
보안	● 풍부한 도구	● 세밀한 통제 가능