

프로젝트 구성도 및 설계서

프로젝트 목표

- 다양한 PaaS 환경 구축
- 온프레미스/퍼블릭 클라우드/컨테이너 환경에 대한 이해 및 비교
- OpenShift, Ansible, CI/CD 학습

프로젝트 시나리오

- 가상의 클라이언트 선정 후 기업이 직면하는 상황에 걸맞는 PaaS 구축
- 클라이언트: 반려동물 커뮤니티 운영
- 상황1: 이제 막 서비스를 시작하는 단계로, 자체 서버에 가상화 환경 구축 후 웹서비스 배포
- 상황2: 사용자가 증가하고 새로운 서비스를 기획하며 유연하게 자원을 활용할 수 있는 퍼블릭 클라우드로 전환, AWS에 가상화 환경 구축 및 Ansible을 통한 웹서비스 배포 자동화
- 상황3: 클라우드를 사용하는 비용이 증가함에 따라 웹서버는 온프레미스로 다시 전환하고, DB만 클라우드에 남겨 놓는 하이브리드 방식을 채택, 자체 서버에 오픈시프트 설치 후 컨테이너를 통한 웹서버 관리 및 CI/CD 적용

프로젝트 구성

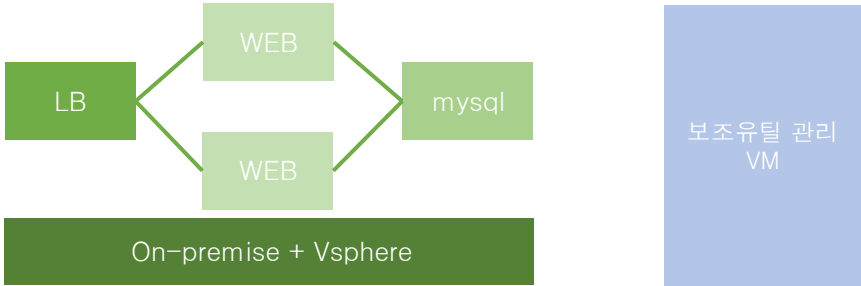
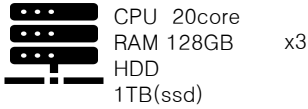
1단계: VM + WEB + DB

2단계: AWS + WEB + DB + Ansible

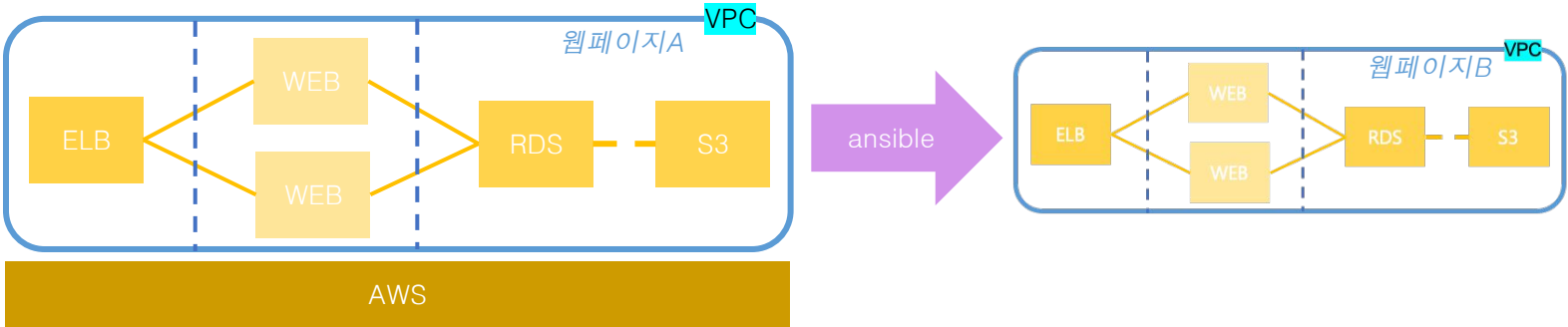
3단계: OpenShift + Web Containerization + DB + CI/CD

프로젝트 실습 환경 및 구성도

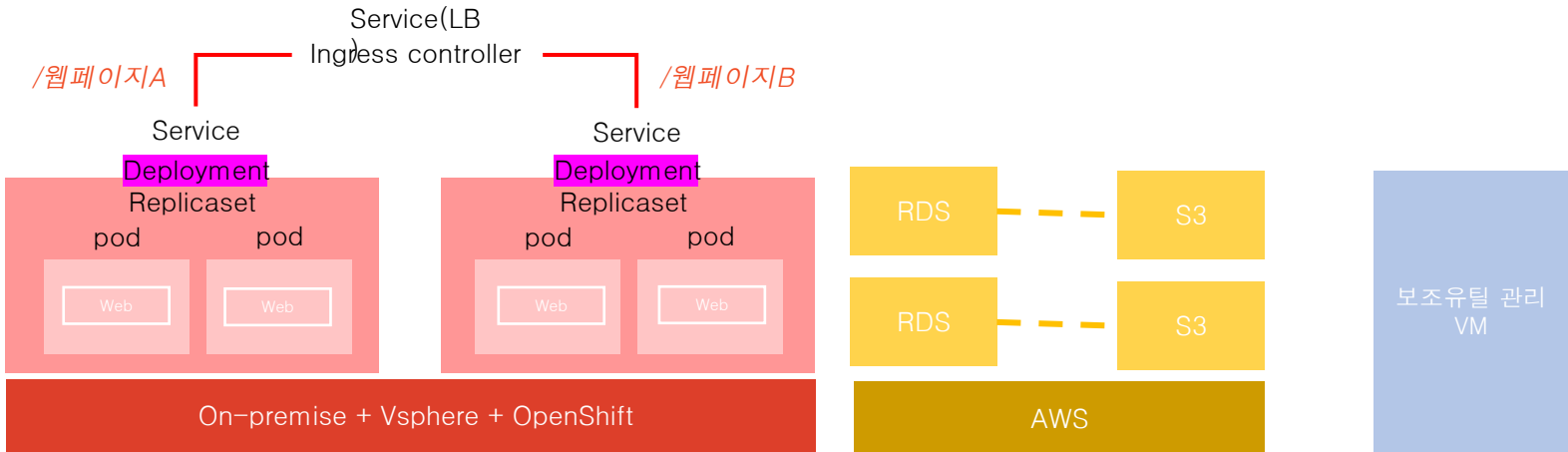
On-premise



Public Cloud

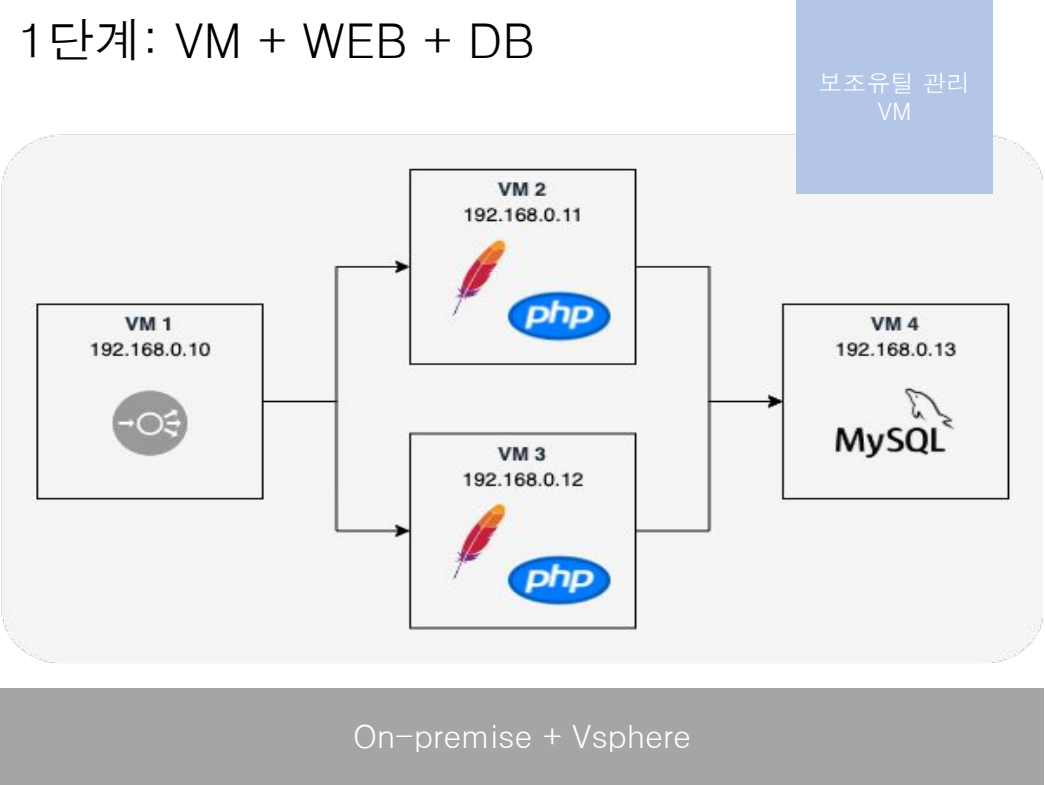


Container



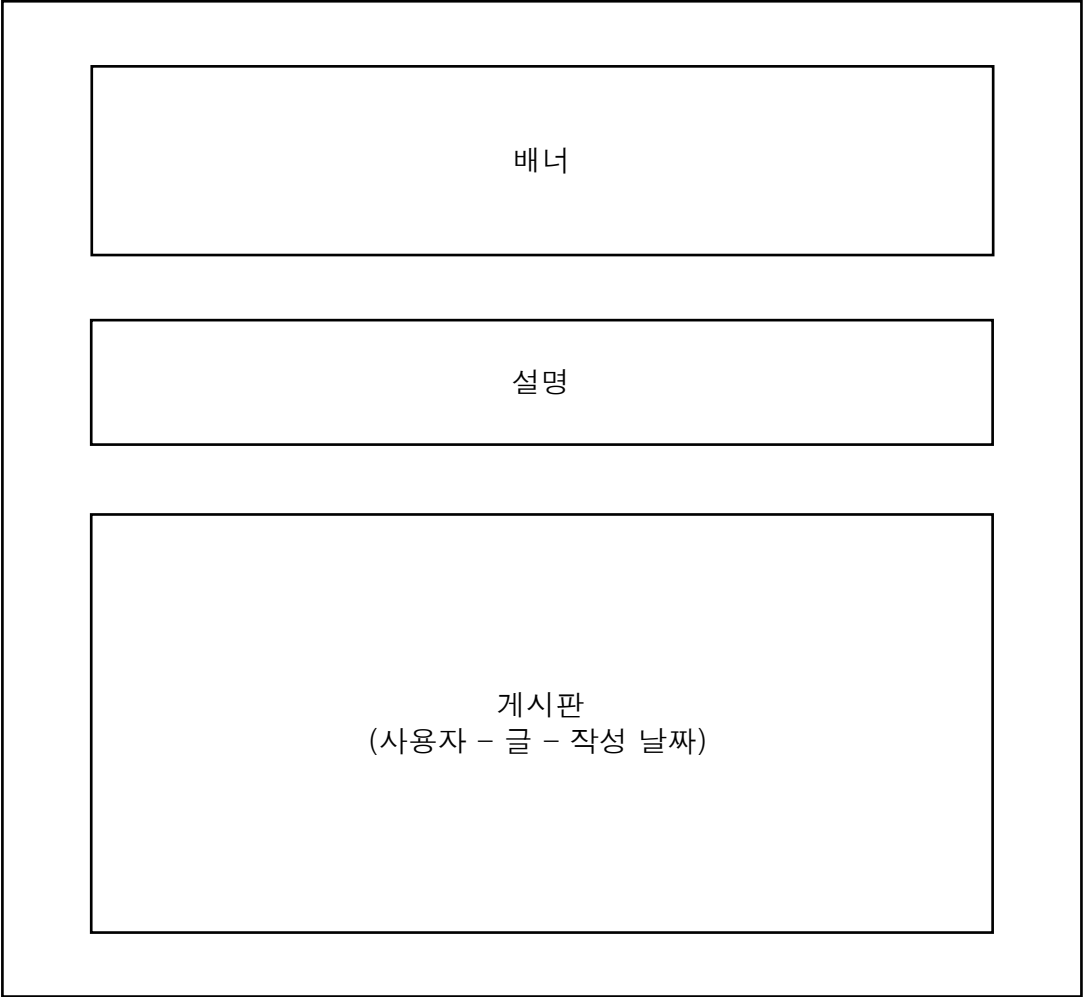
단계별 인프라 설계서

1단계: VM + WEB + DB

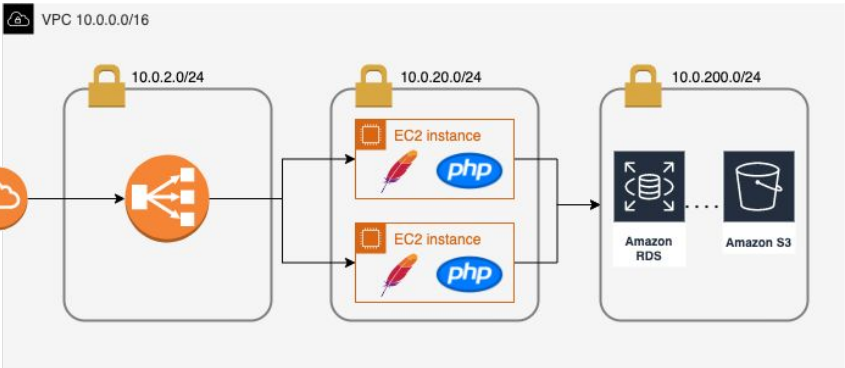
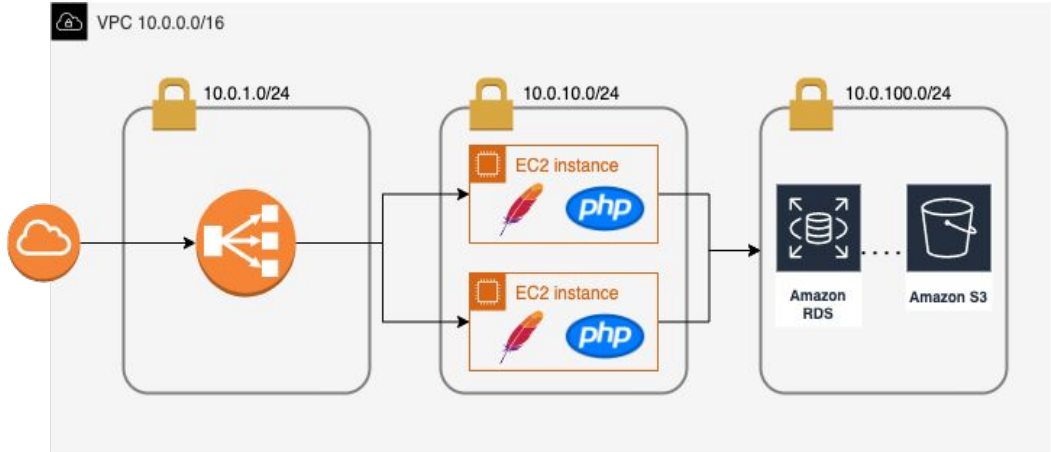


	목적	VM 사양	설치 프로그램	네트워크
VM1	LB	OS = RHEL 7.8 CPU = 2 RAM = 2GB HDD = 10GB	haproxy	192.168.0.10
VM2	WEB 1		apache2 2.4 php 7.3	192.168.0.11
VM3	WEB 2		apache2 2.4 php 7.3	192.168.0.12
VM4	DB		mysql 5.6	192.168.0.13

웹페이지A: 강아지를 기르는 사람들의 커뮤니티(도메인 예시: dog.com)



2단계: AWS + WEB + DB + Ansible



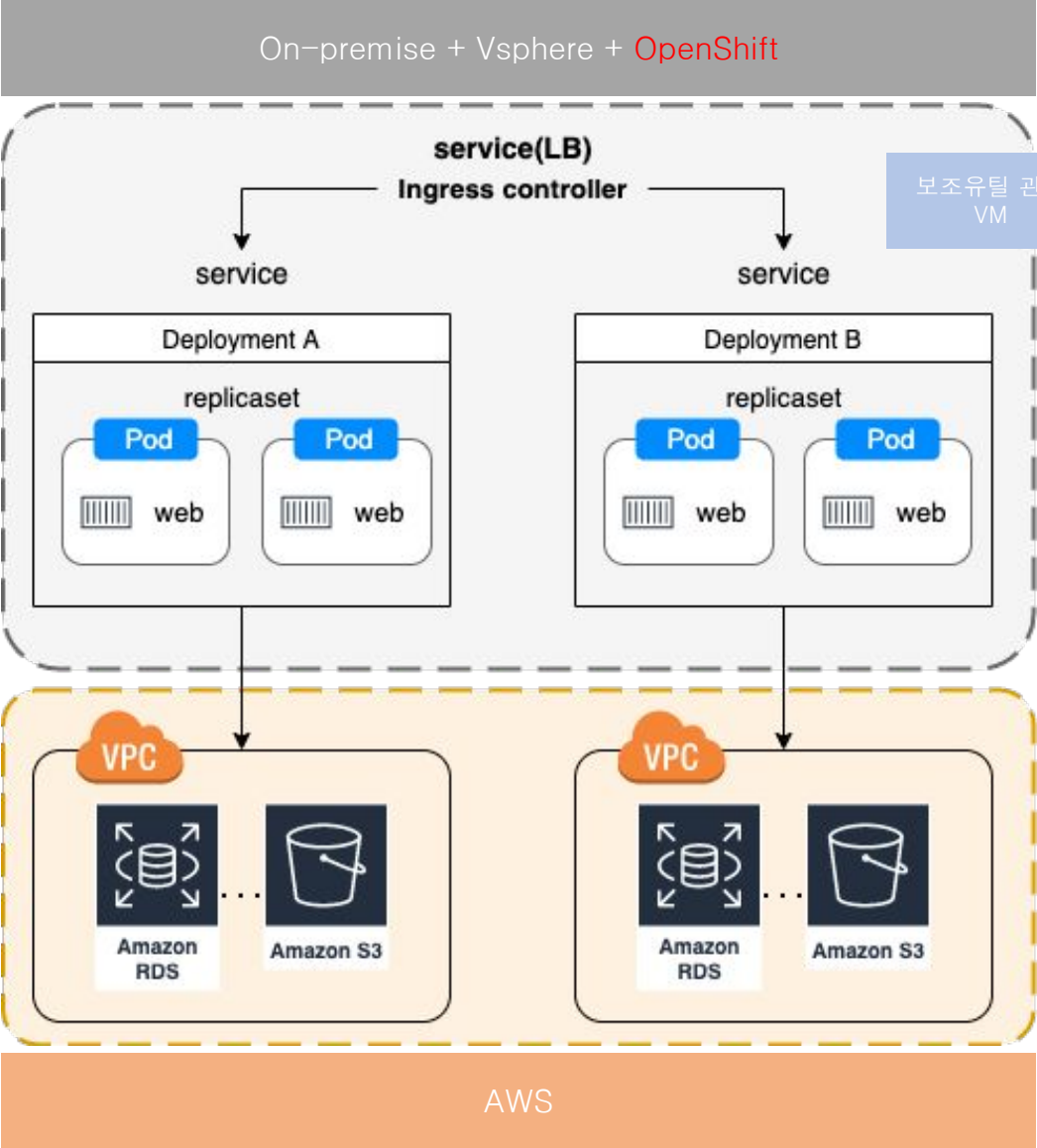
AWS

VPC	DB
<ul style="list-style-type: none">10.0.0.0/16A 서브넷 1.0/24, 10.0/24, 100.0/24B 서브넷 2.0/24, 20.0/24, 200.0/24	<ul style="list-style-type: none">RDSS3DMS를 통한 마이그레이션자동 or 수동 백업
EC2	SG
<ul style="list-style-type: none">Red Hat Enterprise Linux 8t2.microApache2, php 설치 (1단계와 같은 버전)	<ul style="list-style-type: none">ELB: port 80 from allWEB: port 80 from 10.0.1.0/24 port 22 from my ipRDS: 3306 from 10.0.100.0/24, my ip

웹페이지B: 고양이를 기르는 사람들의 커뮤니티
(도메인 예시: cat.com)



3단계: OpenShift + Web Containerization + DB + CI/CD



웹페이지C: 반려동물을 기르는 사람들의 커뮤니티
(도메인 예시: community.com □ community.com/dog,
community.com/cat)