

Kubernetes Deployments with Helm

Kubernetes Korea Group 강다빈

Contents

- 프로젝트 계기
- 이미지 빌드 과정 : 기존 vs Jib
- K8s 기본 개념
- minikube에 kubectl apply로 배포하기
- helm으로 앱 배포하기

Kubernetes를 접하게 된 계기

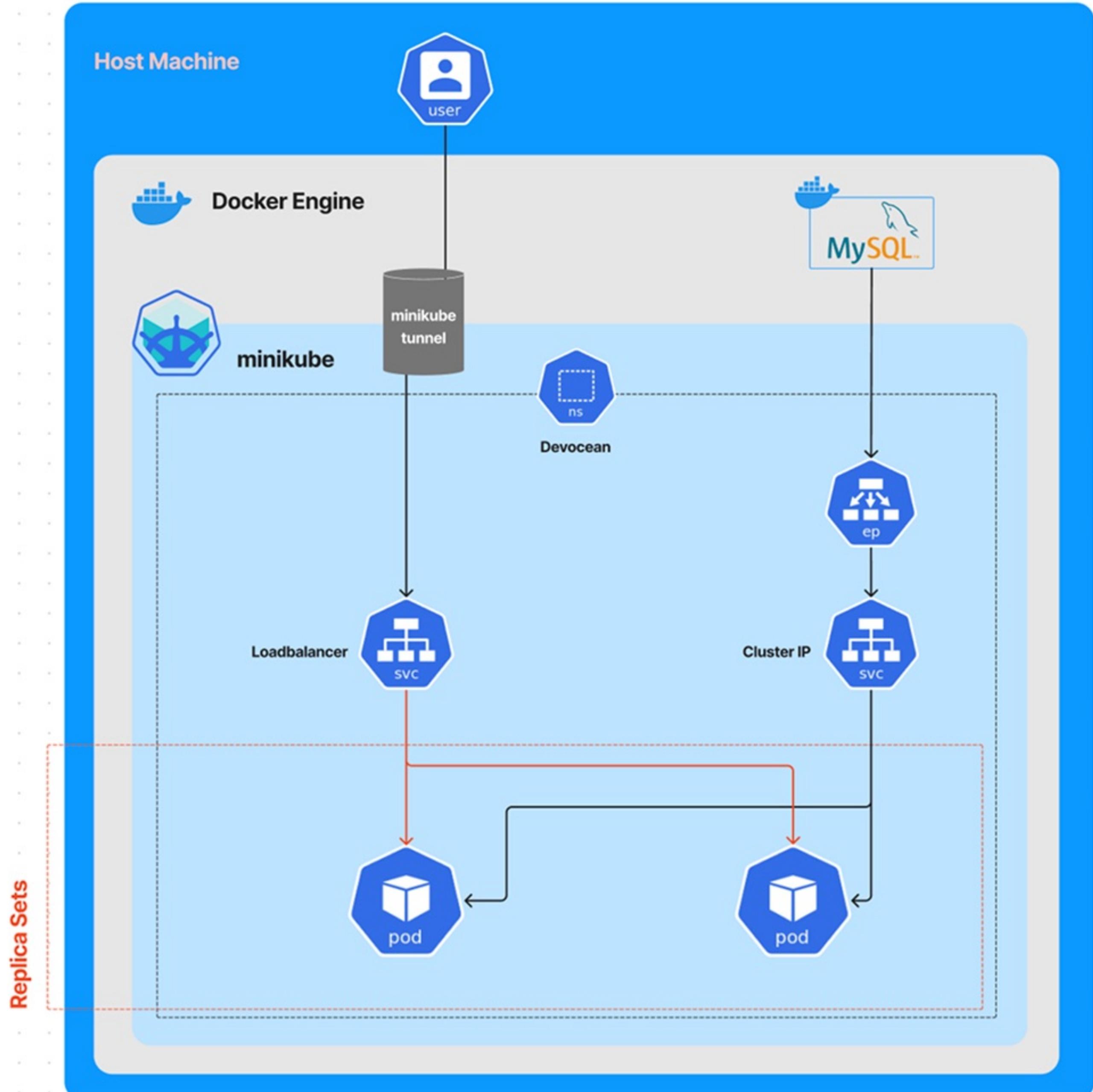
- 세미나/컨퍼런스 참석
- 전문가 스터디 -> 데보션 개발 과제



데보션 개발 과제 이후

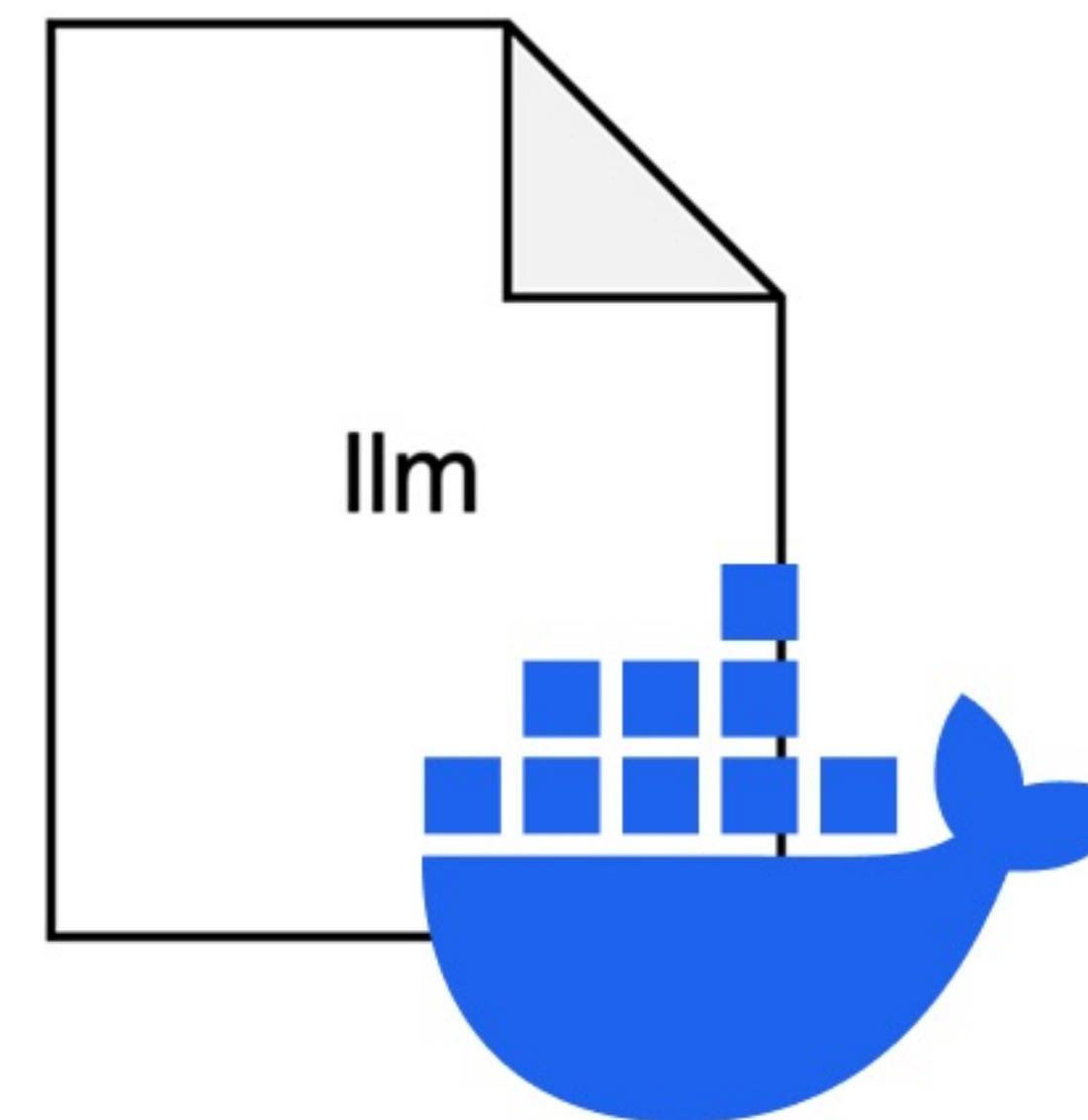
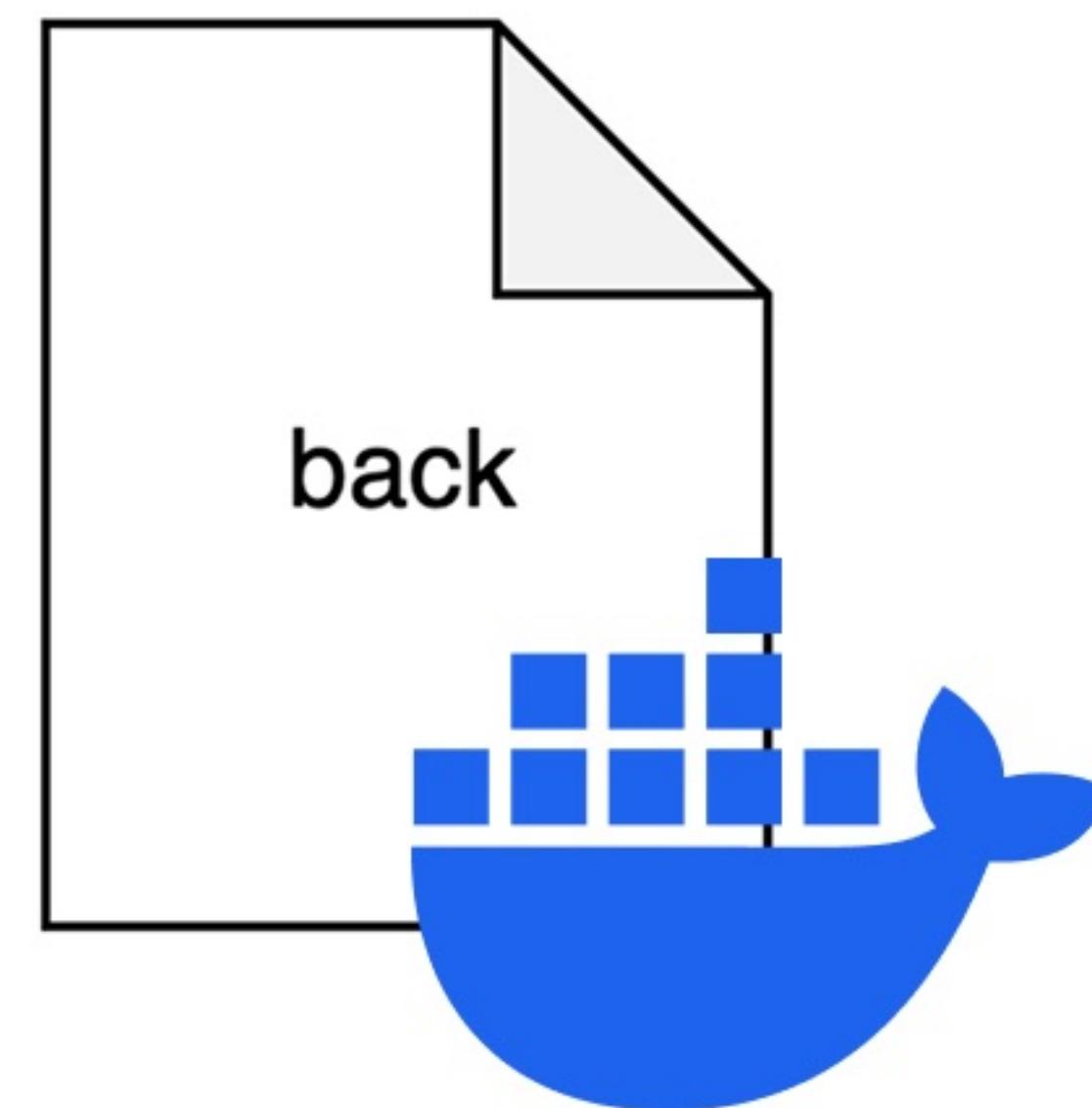
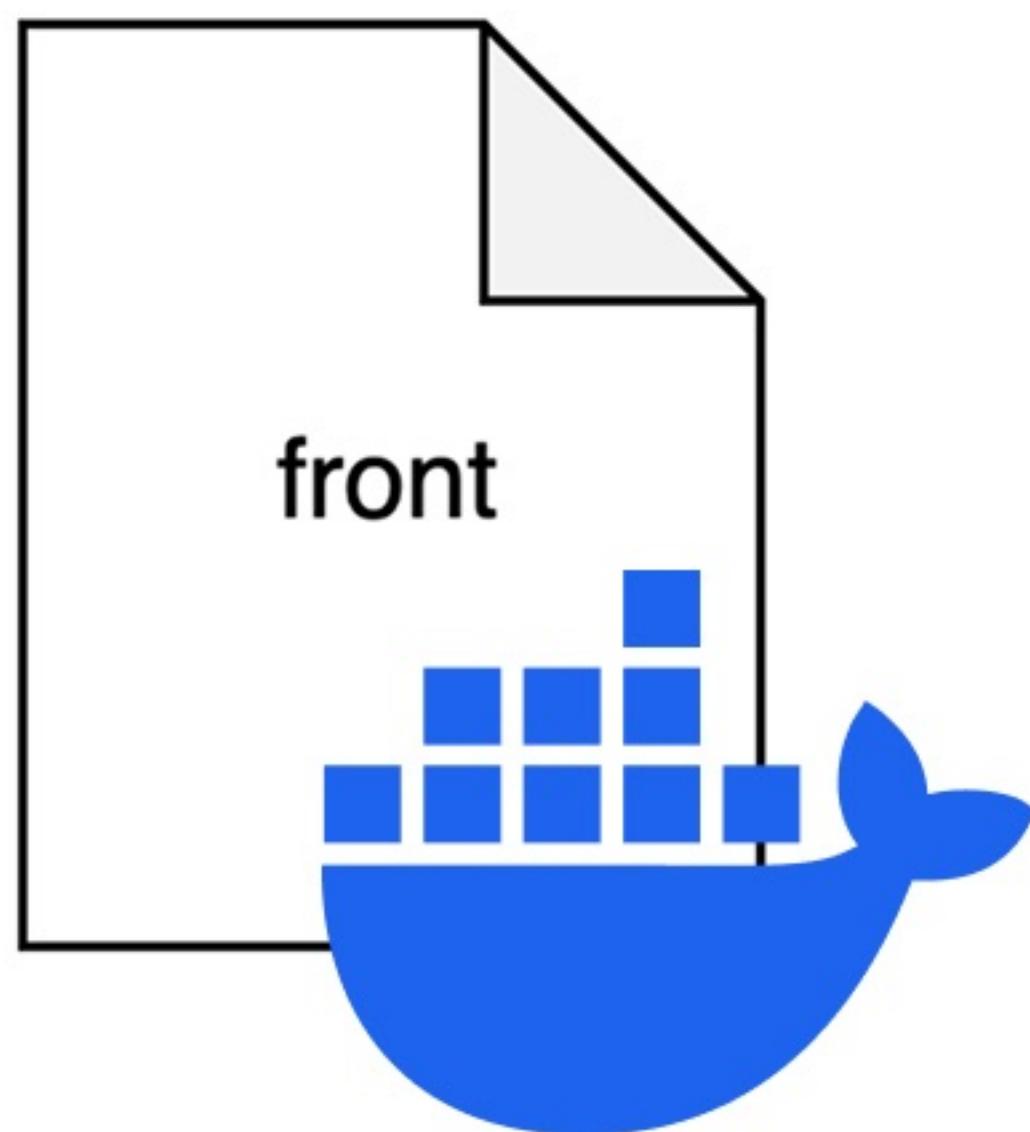
- 운영진 활동
- 데보션 오픈랩 스터디

이번엔 제대로
이해하고 만들어보자

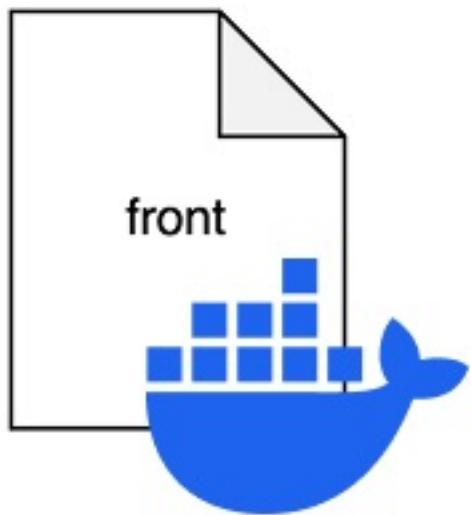


도커 이미지 빌드하기

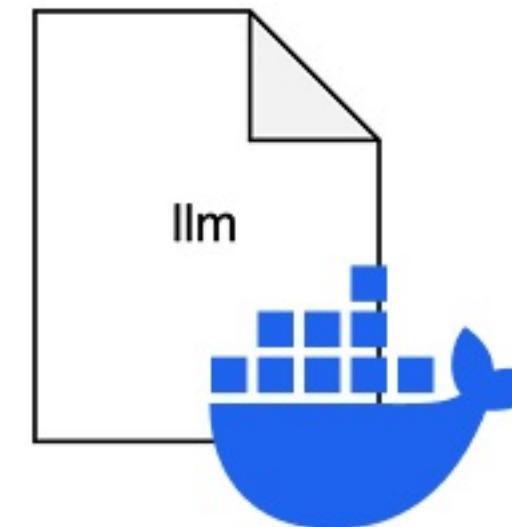
배포할 대상 : 챗봇 서비스



Docker Image 빌드하기 1



1. docker + nginx 배포
2. spring boot의 static 폴더에 넣고 배포



COPY ./build (x) -> COPY ./dist

⚠ vite는 npm run build 명령어로 dist/ 폴더 아래에 빌드 파일 생성

```
FROM nginx:alpine
WORKDIR /app
COPY ./dist /usr/share/nginx/html
COPY ./nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

```
# 1. 기본 Python 이미지 사용 (최신 버전 추천)
FROM python:3.10-slim

# 2. 작업 디렉토리 설정
WORKDIR /app

# 3. 필요한 패키지 설치
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# 4. 소스 코드 복사
COPY .

# 5. FastAPI 실행 (serve.py가 서버 실행 파일이라고 가정)
CMD ["uvicorn", "serve:app", "--host", "0.0.0.0", "--port", "8000"]
```

Multi-stage Build

```
# step 1: Build Stage
FROM gradle:jdk17-alpine AS build
WORKDIR /home/app
COPY ./ .
RUN ./gradlew bootJar

# step 2: Run Stage
FROM openjdk:17-alpine AS run
WORKDIR /home/app
COPY --from=build /home/app/build/libs/*.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]
```

이미지 크기 줄이기

- 가벼운 base 이미지 사용 -> alpine
- multi-stage 사용

Spring Boot 프로젝트 실행 과정

1. Gradle 기반 빌드
2. JDK 기반 실행

Docker Image 빌드하기 2 - 트러블슈팅 1

프로젝트 build 자체가 안되는 문제 -> **openApiGenerate spec validation 오류**

openApiGenerate

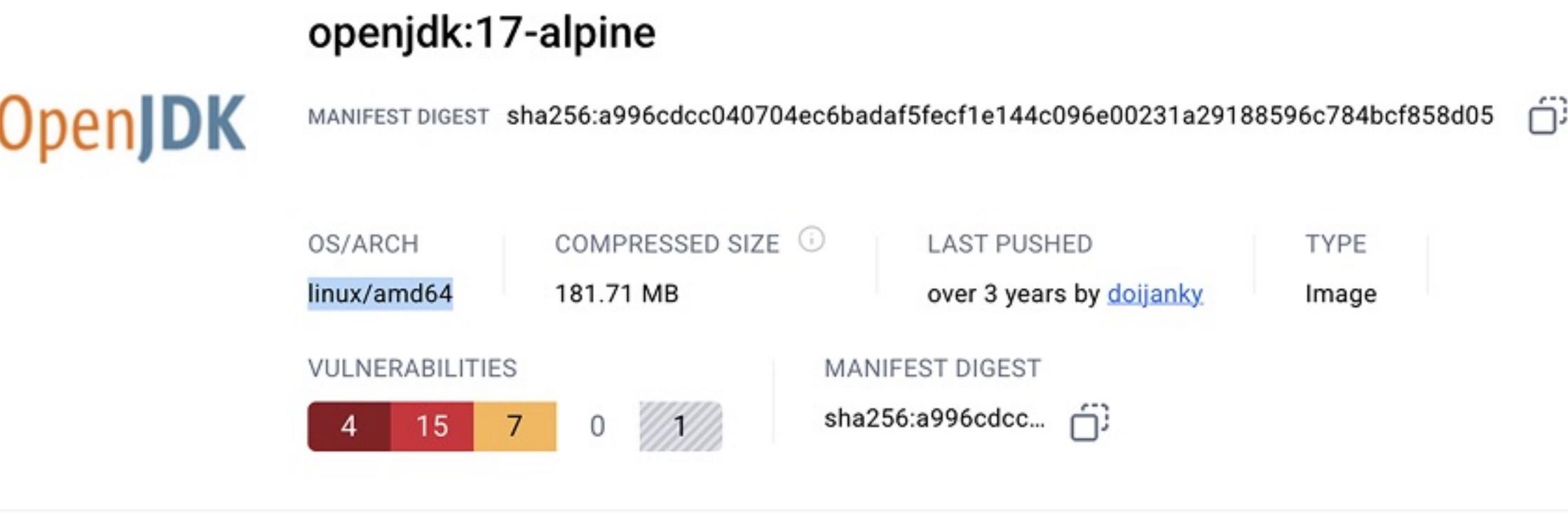
Table 2. Options

Key	Data Type	Default	Description
verbose	Boolean	false	The verbosity of generation
validateSpec	Boolean	true	Whether or not we should validate the generation. Invalid specs result in an error.
generatorName	String	None	The name of the generator which will be used.
outputDir	String	None	The output target directory into which generated.
inputSpec	String	None	The Open API 2.0/3.x specification location.
inputSpecRootDirectory	String	None	Local root folder with spec file(s)

```
task<GenerateTask>("generateApiServer") {  
    //생략  
    additionalProperties.set(  
        mapOf(  
            "useTags" to "true",  
            "jakarta" to "true",  
            "enumPropertyNaming" to "UPPERCASE",  
        )  
    )  
  
    validateSpec.set(false)  
  
    //생략  
}
```

Docker Image 빌드하기 2 - 트러블슈팅 2

ERROR: failed to solve: openjdk:17-alpine: failed to resolve source metadata for docker.io/library/openjdk:17-alpine:
no match for platform in manifest: not found



mac은 arm64 기반
openjdk 이미지는 amd64 기반

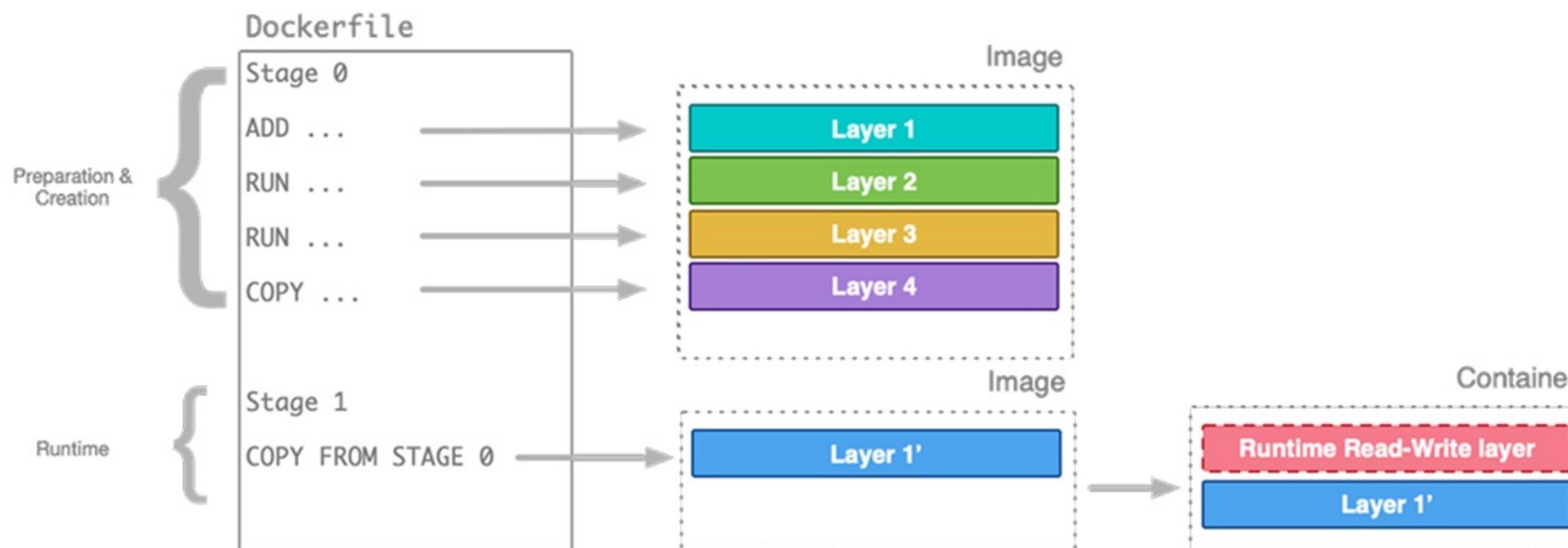
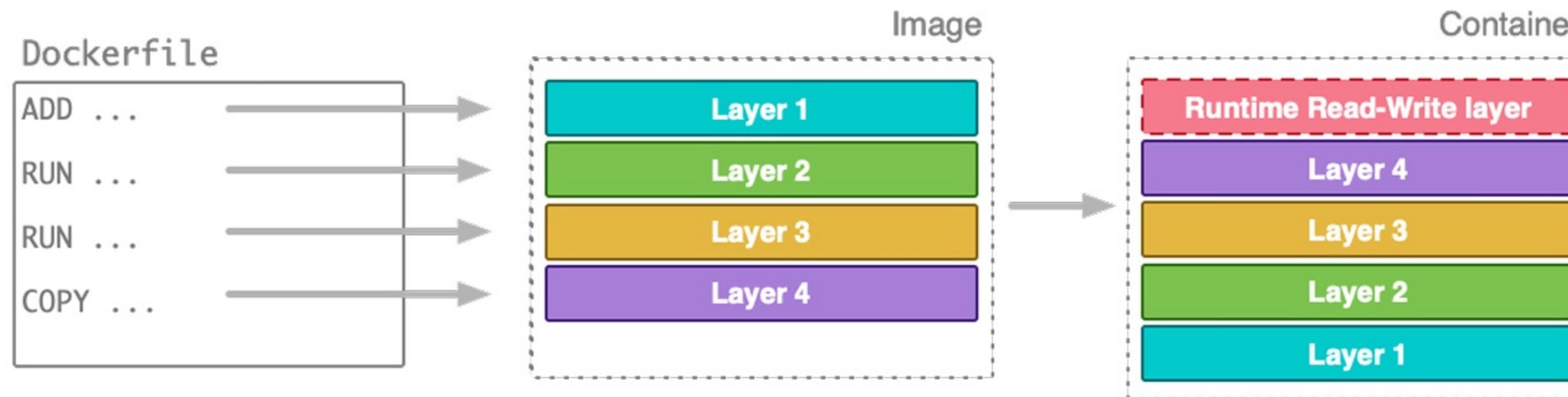
플랫폼 지정 빌드

Build 소요 시간 278.4s

```
* ➔ ~/pblbe/pbl ➔ main ➔ docker build --platform linux/amd64 -t ppap-test .
[+] Building 278.4s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 363B
=> [internal] load metadata for docker.io/library/gradle:jdk17-alpine
=> [internal] load metadata for docker.io/library/openjdk:17-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [build 1/4] FROM docker.io/library/gradle:jdk17-alpine@sha256:8dbab10c7aeaf06c8b486555c76e3dcdb3181c47f90a57ea325e8722858a6759
=> => resolve docker.io/library/gradle:jdk17-alpine@sha256:8dbab10c7aeaf06c8b486555c76e3dcdb3181c47f90a57ea325e8722858a6759
=> => sha256:0254cb3b8ea2a1db854d7b1952f497c1a7d34081ddeaf23cd8d746f7780a99c3 7.79kB / 7.79kB
=> => sha256:b7b93bccf647f23c56b988b134e0f24ce8aed01ba9162e974330b45abc9f2b21 143.72MB / 143.72MB
=> => sha256:8dbab10c7aeaf06c8b486555c76e3dcdb3181c47f90a57ea325e8722858a6759 1.38kB / 1.38kB
```

기존 Docker 이미지 빌드

- JAR 파일을 하나의 이미지 레이어로 포함시키는 방식
- Dockerfile 각 명령어에 대한 하나의 레이어 빌드

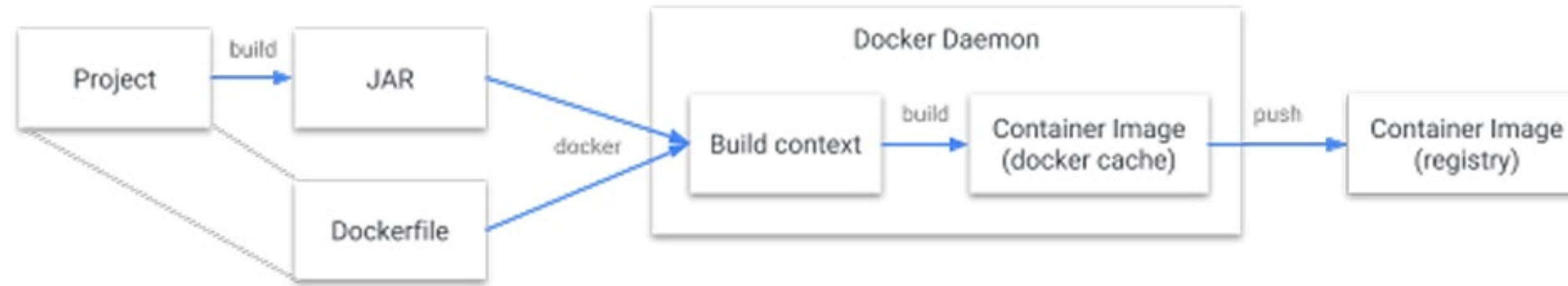


multi-stage 빌드

애플리케이션이 실행하는데
필요한 구성 요소만 포함

Jib

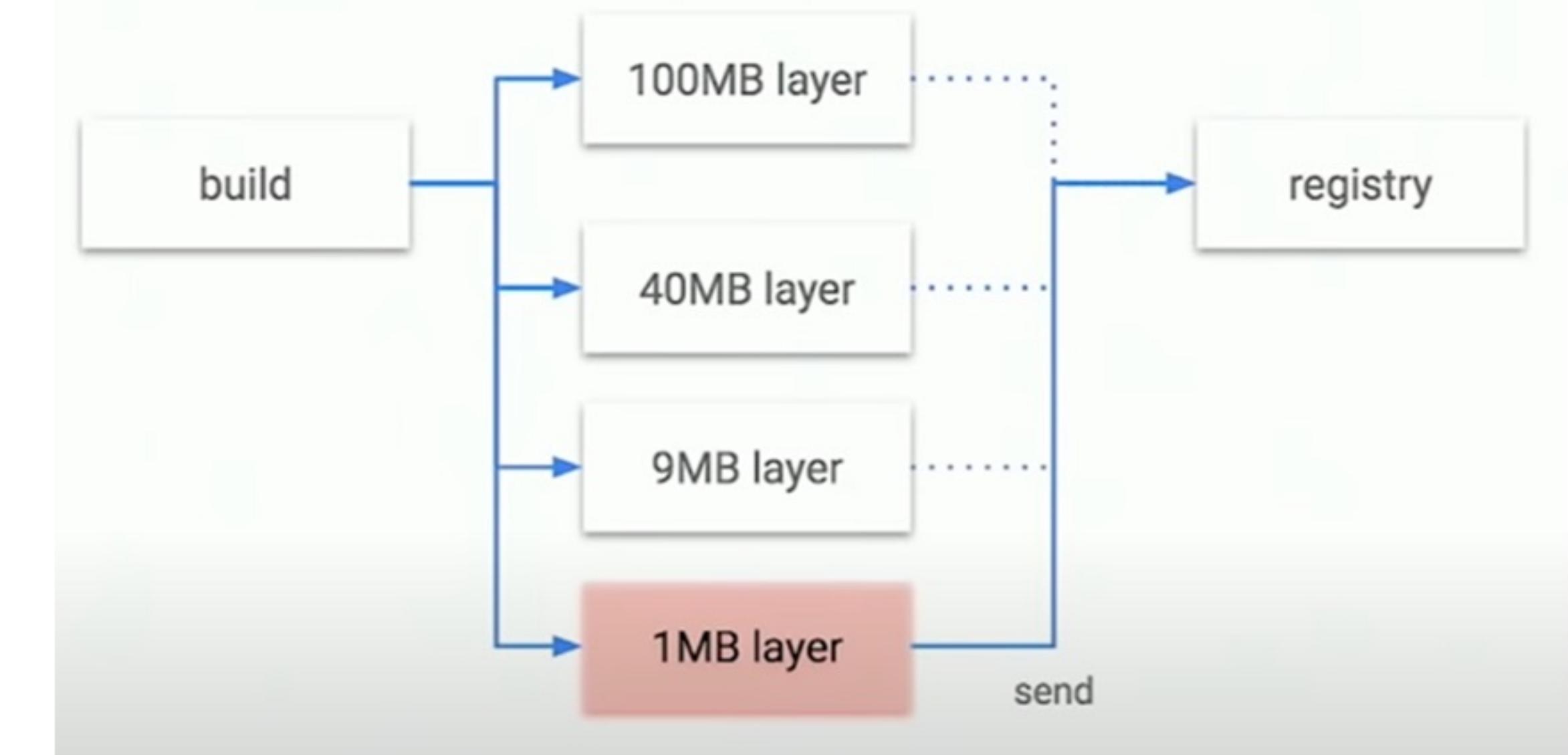
기존 방식



Jib 사용



- Docker Daemon 없이 java 애플리케이션을 컨테이너화
- Docker Hub, GCR 등 원하는 레지스트리로 직접 push 가능
- Reproducible = Same Code, Same Image
- JAR를 여러 개의 레이어로 분리하여 빌드
- 변경된 부분만 rebuild



Jib

```
#build.gradle 파일에 플러그인 추가  
plugins {  
    id 'com.google.cloud.tools.jib' version '3.4.4'  
}  
  
#앱 컨테이너화  
gradle jib --image=<MY IMAGE>  
  
#도커 데몬으로 빌드  
gradle jibDockerBuild
```

```
BUILD SUCCESSFUL in 31s  
5 actionable tasks: 5 executed  
kchabin@ganodabin-ui-MacBookPro ~/r
```

플랫폼 지정 빌드보다 짧은 시간 내에 빌드 가능
· 31s vs 278.4s

크기 : 340MB < 394MB 더 작은 크기의 jib 빌드 이미지

~/pblbe/pbl ➔ ⚡ main ↗ docker images			
REPOSITORY	TAG	IMAGE ID	CREATED
ppap-test	latest	fc331023ce2a	11 minutes ago
<hr/>			
kchabin/ppap-be	latest	531f7c8a963f	55 years ago

쿠버네티스 기본 개념

쿠버네티스?

컨테이너화된 애플리케이션을 자동으로 배포, 스케일링, 운영하는 오픈소스 오케스트레이션 플랫폼.

주요 기능

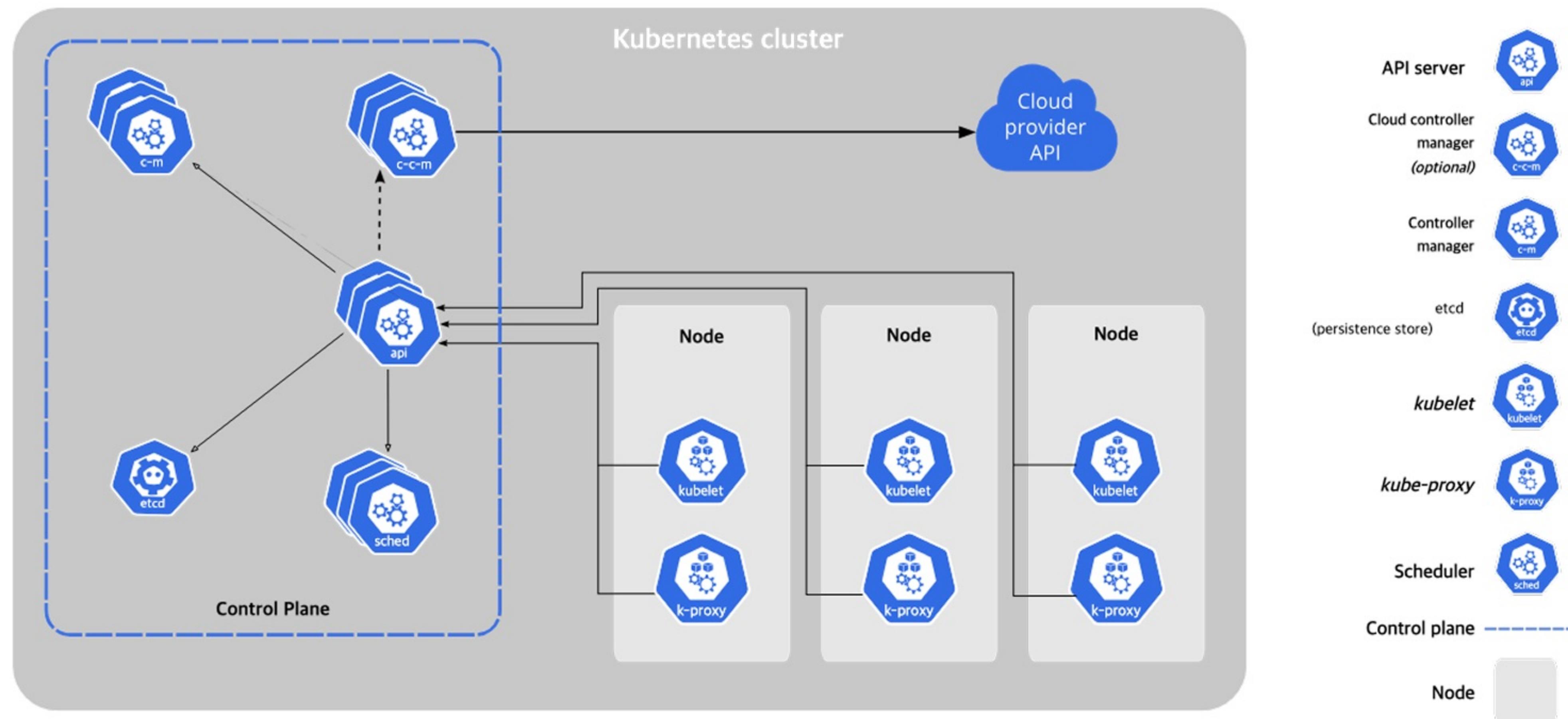
- 자동화된 배포 및 롤백**: 앱을 선언적으로 배포하고 업데이트 가능.
- 자동 스케일링**: 부하에 따라 자동으로 컨테이너를 늘리거나 줄일 수 있음.
- 서비스 디스커버리 및 로드 밸런싱**: 내부 DNS 및 로드밸런서를 통해 서비스 탐색 제공.
- 자체 치유(Self-healing)**: 컨테이너가 실패하면 자동으로 재시작하거나 교체.
- 비밀 및 설정 관리**: Secret과 ConfigMap을 통해 환경 설정 관리.

Control Plane

애플리케이션 상태 관리 및 클러스터 제어

API Server & etcd

- API Server : 클러스터의 모든 요청 처리
- etcd : 클러스터 상태 저장



Namespace

클러스터 내 리소스를 논리적으로 격리하는 공간

```
kchabin@gangdabin-ui-MacBookPro ~ kubectl get pods --namespace=ppap
NAME           READY   STATUS    RESTARTS   AGE
mysql-deployment-fdfc788d9-zdr98  1/1     Running   3 (13h ago)  27h
ppap-be-deployment-77996c594f-c8tvm 1/1     Running   0          8h
ppap-be-deployment-77996c594f-h2w7m 1/1     Running   0          8h
ppap-be-deployment-77996c594f-zskzm 1/1     Running   0          8h
ppap-fe-698bf64679-rgzdt            1/1     Running   0          11h
ppap-fe-698bf64679-zbzbm           1/1     Running   0          11h
ppap-llm-8c99975f9-b9k9x           1/1     Running   2 (13h ago)  38h
```

- SQL의 WHERE 조건절처럼 특정 리소스만 조회

```
kubectl get pods --namespace=dev
```

PVC(Persistent Volume Claim)는 특정 namespace에 종속

- pod가 스토리지를 요청할 때 사용

PV(Persistent Volume)는 상관없이 사용 가능

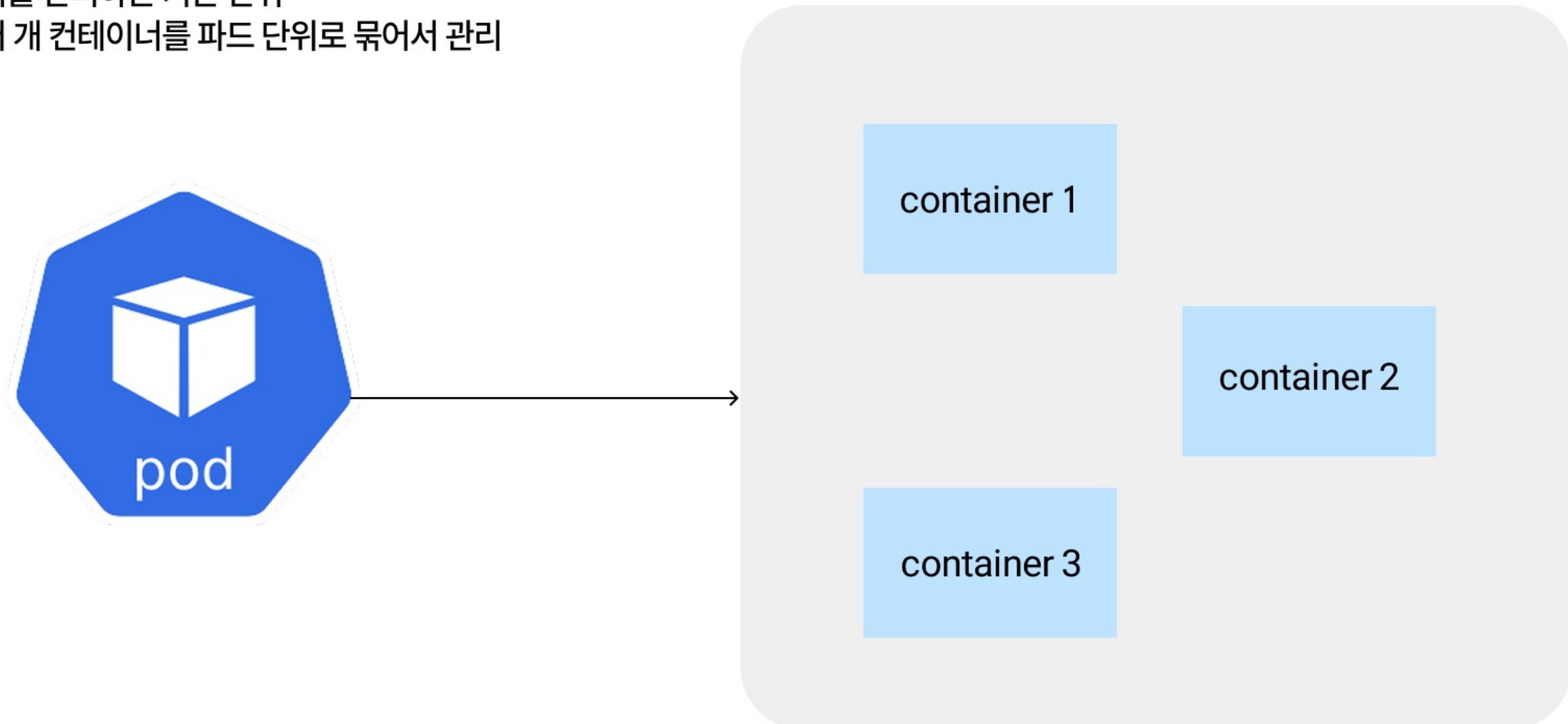
- 클러스터 전체에서 사용 가능한 스토리지 리소스

```
kchabin@gangdabin-ui-MacBookPro ~ kubectl get pvc -n ppap
NAME      STATUS  VOLUME
mysql-pvc  Bound   pvc-98e21c90-3287-42bf-8e01-c7e3555b72ed  5Gi    RWO
uploads-pvc Bound   pvc-51703218-df2c-4eeb-8975-4355c47f05db  2Gi    RWO
kchabin@gangdabin-ui-MacBookPro ~ kubectl get pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
pvc-51703218-df2c-4eeb-8975-4355c47f05db  2Gi       RWO        Delete        Bound  ppap/uploads-pvc
pvc-98e21c90-3287-42bf-8e01-c7e3555b72ed  5Gi       RWO        Delete        Bound  ppap/mysql-pvc
```

Pod

컨테이너를 관리하는 기본 단위

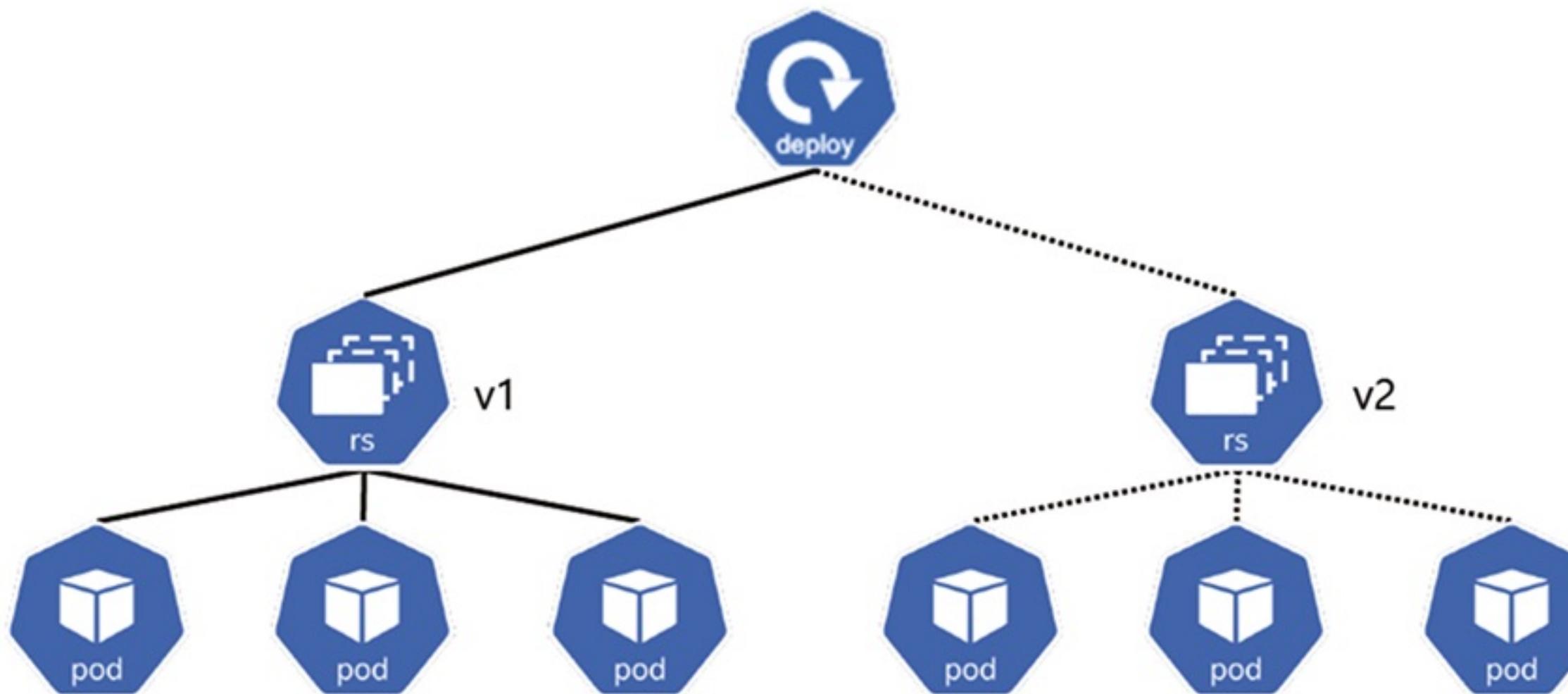
- 여러 개 컨테이너를 파드 단위로 묶어서 관리



Deployment

stateless 앱을 배포할 때 사용하는 기본적인 컨트롤러

- 실행시켜야 할 pod 개수 유지
- 롤링 업데이트, 롤백 가능



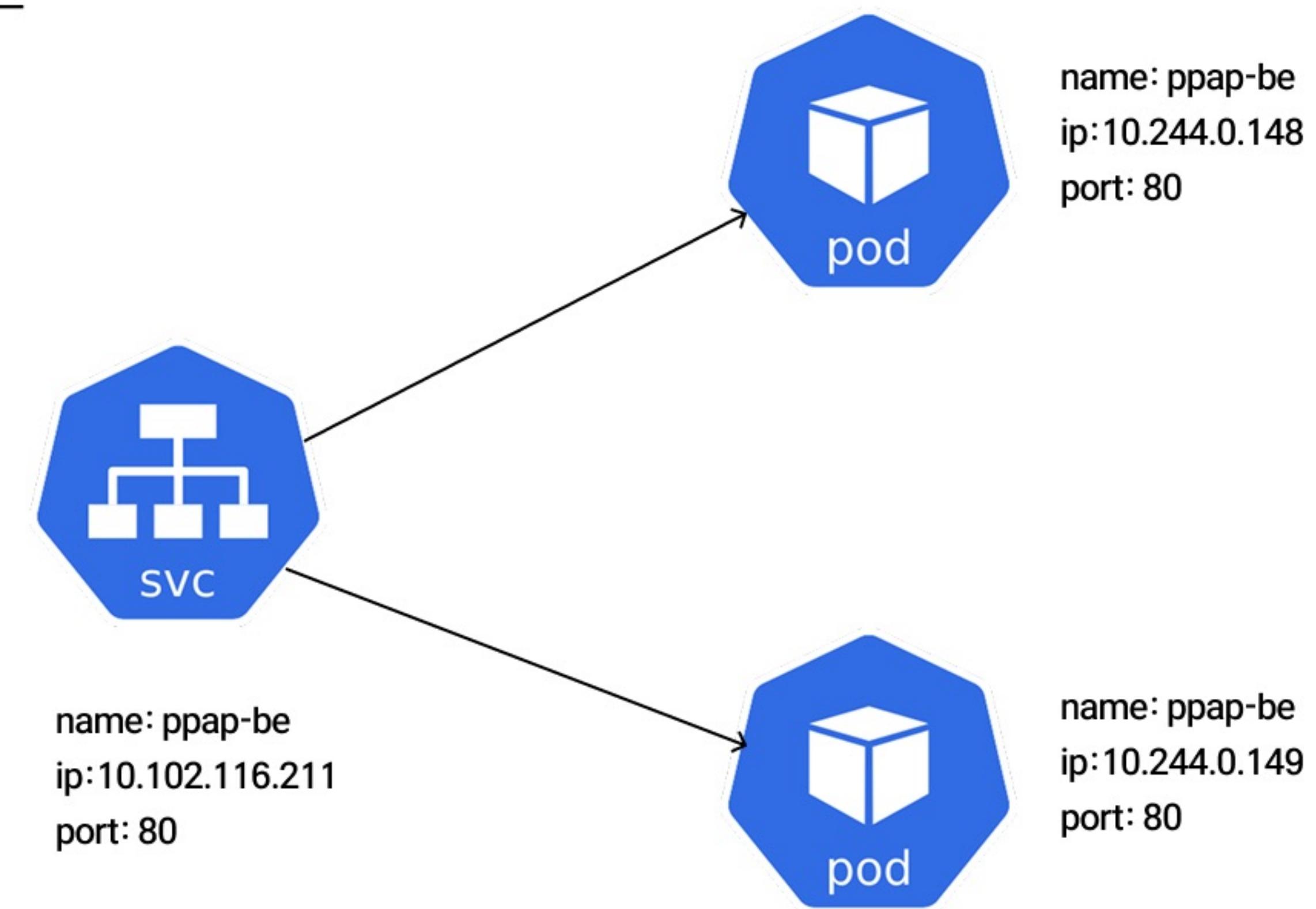
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ppap-be-deployment
  labels:
    app: ppap-be
spec:
  replicas: 3
  selector:
    matchLabels:
      app: ppap-be
  template:
    metadata:
      name: ppap-be-deployment
      labels:
        app: ppap-be
    spec:
      containers:
        - name: ppap-be
          image: kchabin/ppap-be
          imagePullPolicy: Always
          ports:
            - containerPort: 80
              protocol: TCP
```

Service

클러스터 내부에서 동적으로 변하는 pod들에 고정적으로 접근

- ClusterIP 타입 : 내부적으로만 사용

```
apiVersion: v1
kind: Service
metadata:
  name: ppap-be
spec:
  selector:
    app: ppap-be
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: ClusterIP
```



Ingress

클러스터 외부에서 안으로 접근하는 요청들을 어떻게 처리할지 정의해둔 규칙 모음.
실제로 동작 -> 인그레스 컨트롤러

- 외부에서 접근해야 할 URL을 사용할 수 있도록 함
- 트래픽 로드밸런싱
- SSL 인증서 처리
- 도메인 기반 가상 호스팅



Secret

비밀번호, OAuth 토큰 등 민감한 정보 저장 용도

- pod의 환경변수
- 볼륨 파일 형식

```
- name: JWT_ACCESS_SECRET
  valueFrom:
    secretKeyRef:
      name: jwt-secret # Secret 이름
      key: access # Secret의 키
- name: JWT_REFRESH_SECRET
  valueFrom:
    secretKeyRef:
      name: jwt-secret
      key: refresh
```

```
apiVersion: v1
kind: Secret
metadata:
  name: jwt-secret
  namespace: ppap
type: Opaque
data:
  access: "Y2VBVUZ5UWZFSWk3QjJyZlJqd1FGdURvbFlBdFZuQnZzZ3NKU0RTREdrTktM
  refresh: "bXdk0TY3QmVWS0ZSN1NoWFQNUhVUzNac0RLdHkwZkZTVkNMTE5NZHNOULR"
```

be-deployment.yaml

jwt 인증 방식의 로그인/회원가입 백엔드 서비스 설정 파일

- access 토큰 생성용 시크릿 키
- refresh 토큰 생성용 시크릿 키

PVC, PV

컨테이너 안에 저장된 데이터는 컨테이너 삭제 시 모두 사라짐



볼륨 자체

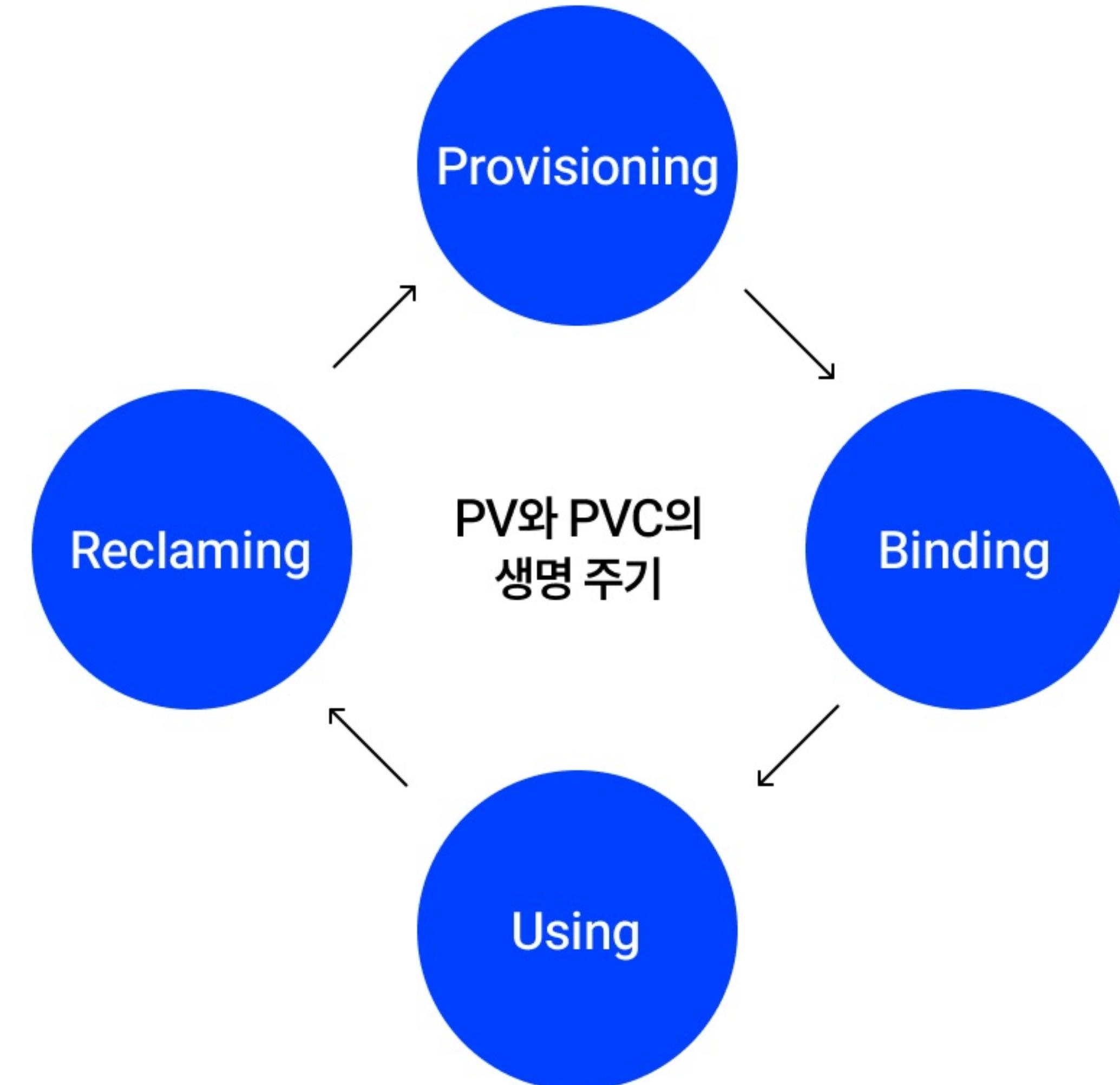
- 클러스터 내부에서 자원
- 파드와 별개로 관리
- 별도의 생명주기

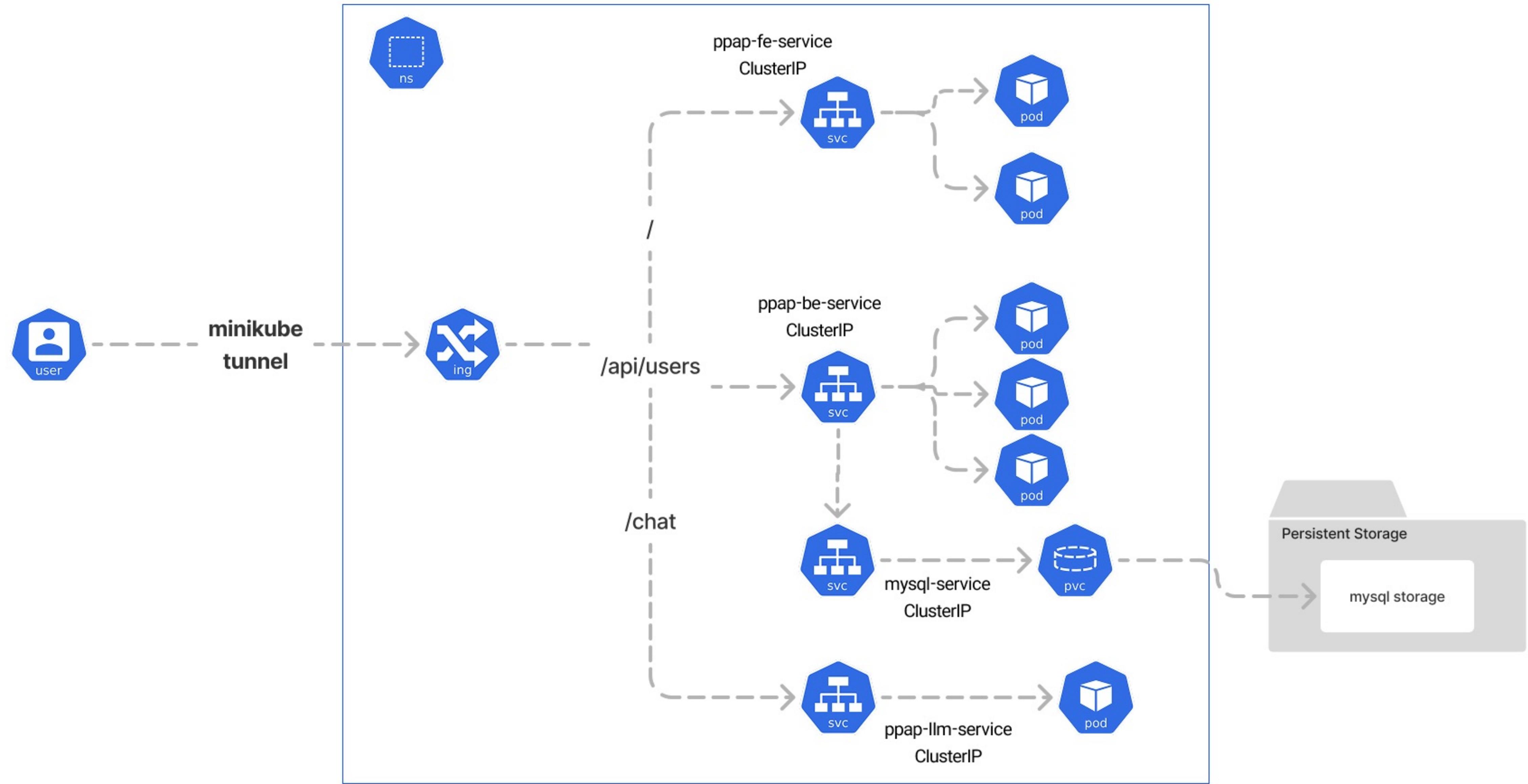
1:1
↔



사용자가 PV에 하는 요청

- 용량
- 읽기/쓰기 모드 설정





배포 과정

```
kubectl create namespace ppap
```

```
kubectl apply -f mysql-pvc.yaml -n ppap
```

```
kubectl apply -f mysql-service.yaml -n ppap
```

```
kubectl apply -f jwt-secret.yaml -n ppap
```

```
kubectl apply -f be-deployment.yaml -n ppap
```

```
kubectl apply -f be-service.yaml -n ppap
```

```
kubectl apply -f fe-deployment.yaml -n ppap
```

```
kubectl apply -f fe-service.yaml -n ppap
```

```
kubectl apply -f llm-deployment.yaml -n ppap
```

```
kubectl apply -f llm-service.yaml -n ppap
```

```
kubectl apply -f ppap-ingress.yaml -n ppap
```

```
minikube tunnel
```

Error: secret “jwt-access-secret” not found

디플로이먼트					
이름	이미지	레이블	노드	상태	CPU 사용량 (cores)
ppap-be-deployment	kchabin/ppap-be	app: ppap-be	minikube	CreateCont: 0	-
mysql-deployment	mysql:8.0	-	minikube	Running 0	-
파드					
이름	이미지	레이블	노드	상태	CPU 사용량 (cores)
ppap-be-deployment-74897f7bb4-hlwzj	kchabin/ppap-be	app: ppap-be pod-template-has h: 74897f7bb4	minikube	CreateCont: 0	-
Error: secret "jwt-access-secret" not found					
ppap-be-deployment-74897f7bb4-kg2dj	kchabin/ppap-be	app: ppap-be pod-template-has h: 74897f7bb4	minikube	CreateCont: 0	-
ppap-be-deployment-74897f7bb4-tll6k	kchabin/ppap-be	app: ppap-be pod-template-has h: 74897f7bb4	minikube	CreateCont: 0	-
mysql-deployment-75559cc49d-xnlmm	mysql:8.0	app: mysql pod-template-has h: 75559cc49d	minikube	Running 0	-

```
kchabin@gangdabin-ui-MacBookPro ~/f5 ➔ kubectl apply -f jwt-secret.yaml -n ppap
secret/jwt-secret created
```

jwt-secret.yaml 먼저 apply

```
ppap-be-deployment-f4cdf44d-s67bk ➔ kchabin@gangdabin-ui-MacBookPro ~/f5 ➔ kubectl get pods -n ppap
NAME          READY   STATUS    RESTARTS   AGE
mysql-deployment-75559cc49d-xnlmm  1/1    Running   0          25m
ppap-be-deployment-f4cdf44d-dnwzh  1/1    Running   0          16s
ppap-be-deployment-f4cdf44d-jscvb  1/1    Running   0          3m59s
ppap-be-deployment-f4cdf44d-s67bk  1/1    Running   0          17s
kchabin@gangdabin-ui-MacBookPro ~/f5 ➔
```

mysql connection 실패

해결 : jdbc:mysql://localhost:3306/mydb → jdbc:mysql://mysql-service:3306/mydb

```
kchabin@gangdabin-ui-MacBookPro ~/f5 kubectl get pods -n ppap
NAME                 READY   STATUS    RESTARTS   AGE
mysql-deployment-75559cc49d-xnlmm  0/1     Pending   0          41s
```

mysql-pvc.yaml 없이 배포하면 Pending 발생 주의

```
kchabin@gangdabin-ui-MacBookPro ~/f5 kubectl apply -f mysql-pvc.yaml -n ppap
persistentvolumeclaim/mysql-pvc created
kchabin@gangdabin-ui-MacBookPro ~/f5 kubectl get pods -n ppap
NAME                 READY   STATUS           RESTARTS   AGE
mysql-deployment-75559cc49d-xnlmm  0/1     ContainerCreating   0          73s
kchabin@gangdabin-ui-MacBookPro ~/f5 kubectl get pods -n ppap
NAME                 READY   STATUS           RESTARTS   AGE
mysql-deployment-75559cc49d-xnlmm  0/1     ContainerCreating   0          75s
kchabin@gangdabin-ui-MacBookPro ~/f5 kubectl get pods -n ppap
NAME                 READY   STATUS    RESTARTS   AGE
mysql-deployment-75559cc49d-xnlmm  1/1     Running   0          2m23s
```

ingress 연결

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations: {}
  name: ppap-ingress
  namespace: ppap
spec:
  ingressClassName: nginx
  rules:
    - host: ppap.local
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: ppap-fe-service
                port:
                  number: 80
          - path: /api/hello
            pathType: Prefix
            backend:
              service:
                name: ppap-be-service
                port:
                  number: 80
```

/etc/hosts 파일 수정 필요

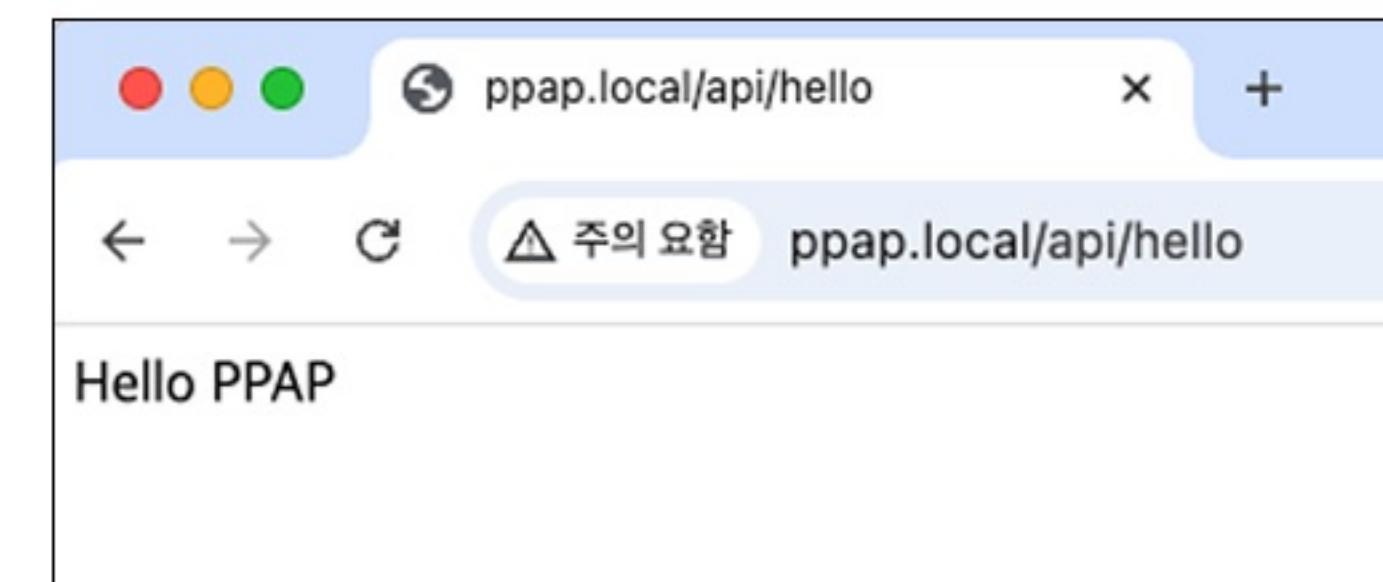
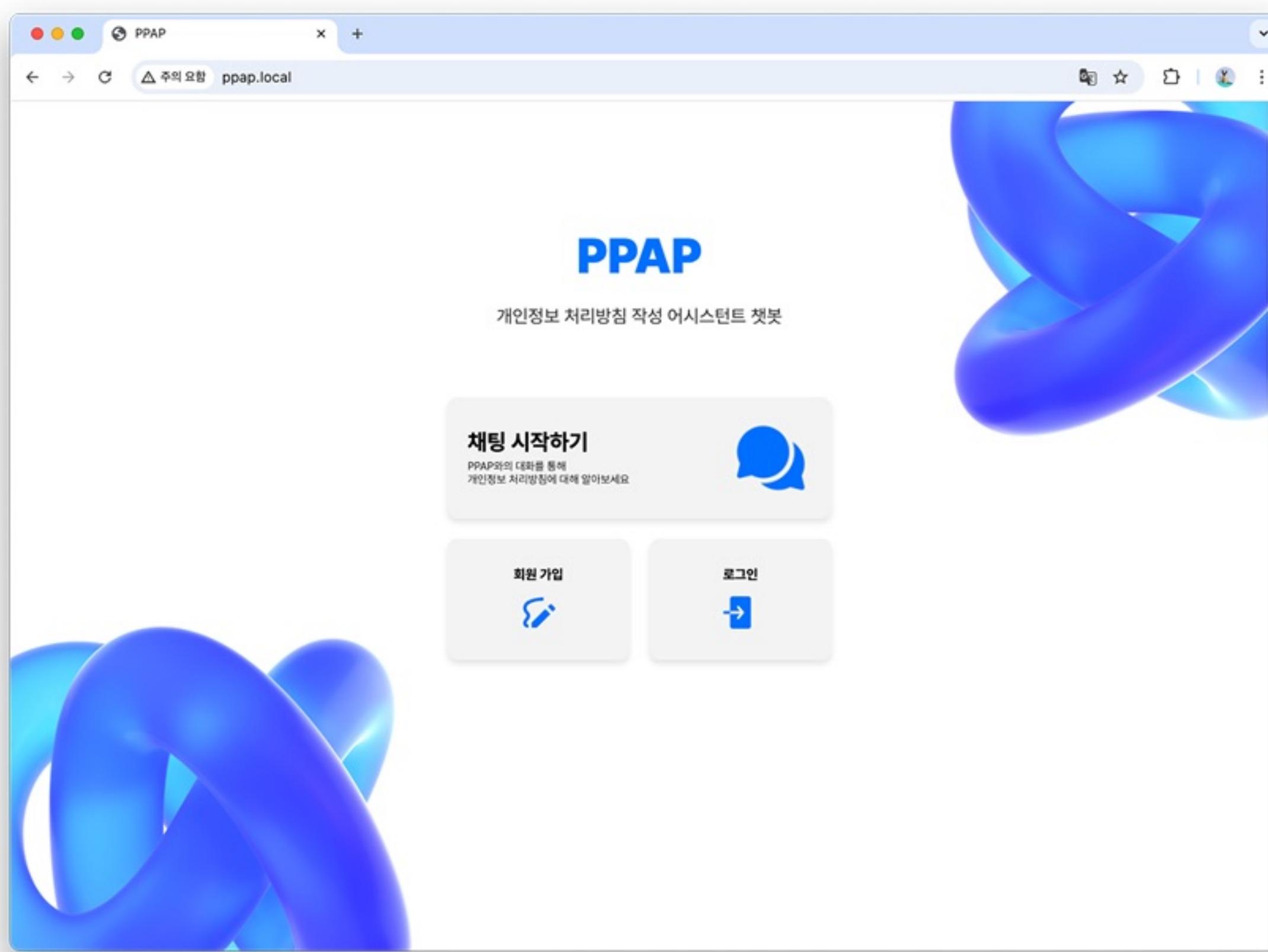
- 수정 후 dscacheutil -flushcache 실행 추천

```
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost  
255.255.255.255 broadcasthost  
::1            localhost  
  
127.0.0.1      ppap.local  
127.0.0.1      ppap-chart.local
```

```
▶ kchabin@gangdabin-ui-MacBookPro ~ f5 sudo vi /etc/hosts  
▶ kchabin@gangdabin-ui-MacBookPro ~ f5 dscacheutil -flushcache  
▶ kchabin@gangdabin-ui-MacBookPro ~ f5 ping ppap.local  
PING ppap.local (127.0.0.1): 56 data bytes  
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.078 ms  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.206 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.175 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.191 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.204 ms  
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.225 ms  
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.171 ms  
^C  
--- ppap.local ping statistics ---  
7 packets transmitted, 7 packets received, 0.0% packet loss
```

minikube tunnel 실행

- api 호출 확인
- db 회원 정보 데이터 저장 확인



HTTP PPAP / sign up

POST http://ppap.local/api/users/signup Send

Params Auth Headers (9) Body Scripts Tests Settings Cookies

raw JSON Beautify

```
1 {  
2   "loginId": "kchabin",  
3   "username": "kch",  
4   "password": "1234",  
5   "confirmPassword": "1234",  
6   "email": "kchabin14@gmail.com",  
7   "userType": "PRIVATE"  
8 }
```

Body 200 OK 2.06 s 656 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "loginId": "kchabin",  
3   "username": "kch",  
4   "password": "{bcrypt}$2a$10$g8Eq5ZtTTT1iPmis.W09kePrWDALjoDJKV0Sg9TdtjxwXAJx32m6i",  
5   "confirmPassword": "1234",  
6   "email": "kchabin14@gmail.com",  
7   "userType": "PRIVATE",  
8   "userId": 1,  
9   "isActive": true,  
10  "isWithdrawed": false,  
11  "audit": null  
12 }
```

Helm으로 배포하기

Helm

쿠버네티스 패키지 매니저 도구

<https://github.com/helm/charts>



chart

k8s에서 실행할 애플리케이션을
만드는데 필요한 정보 묶음

config

패키지한 차트에 넣어서 배포 가능한
오브젝트를 만들 때 사용하는 설정

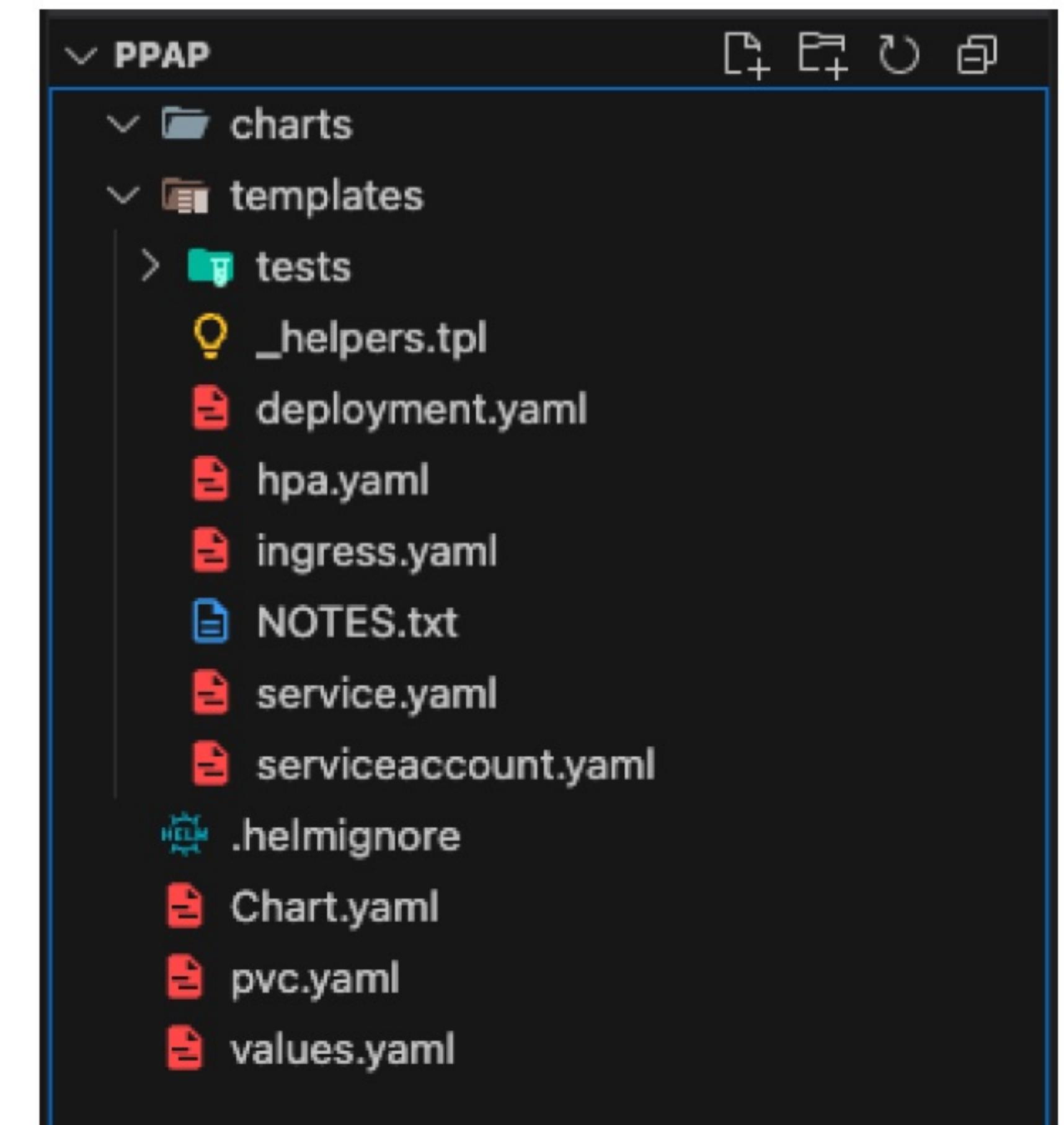
release

특정 config를 이용해 실행 중인
차트의 인스턴스

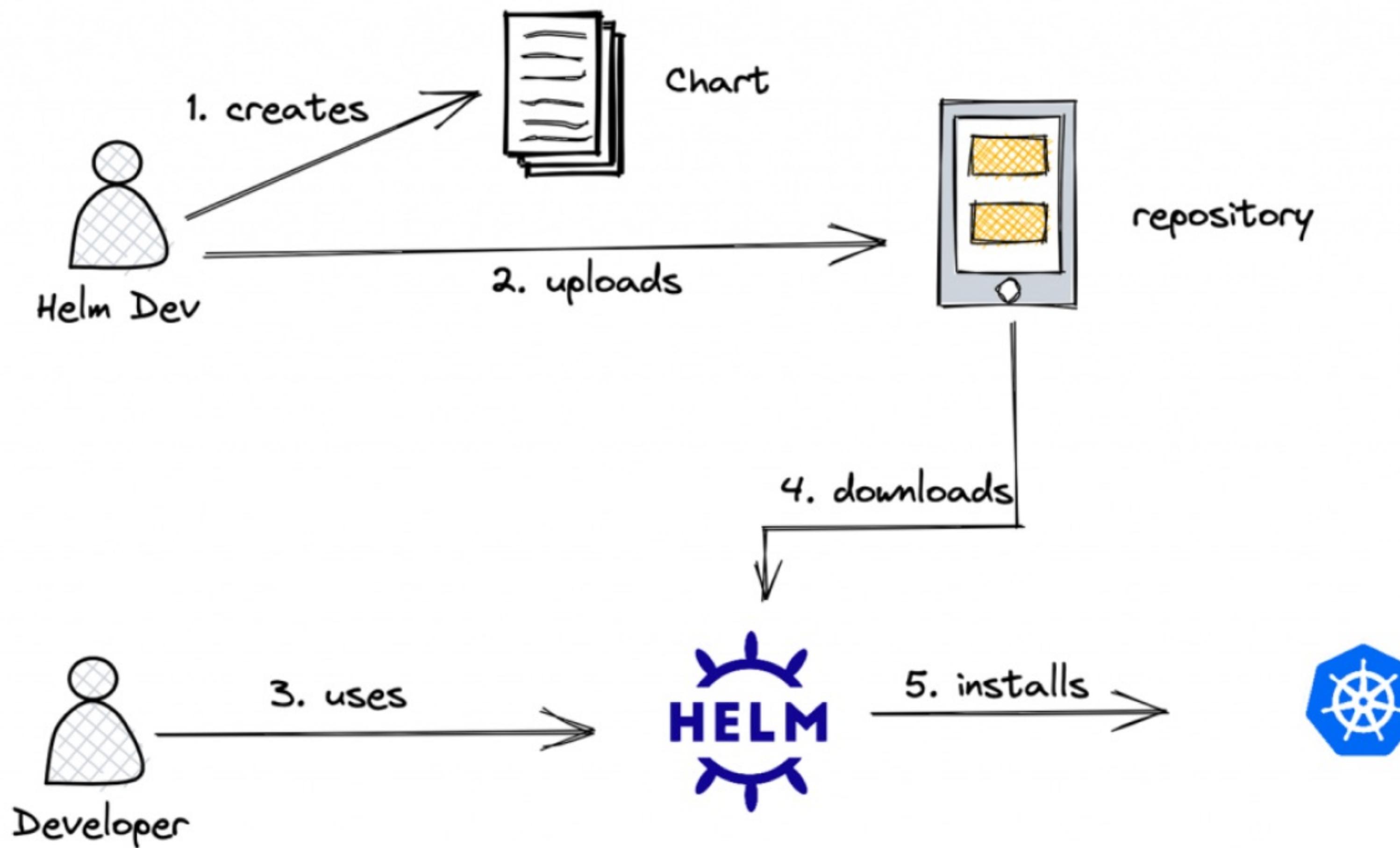
Helm Chart

- K8s Resource Set을 구성하는 파일의 컬렉션
- Chart.yaml, values.yaml, templates 등으로 구성
- 예약된 파일명 : charts/, crds/, templates/

```
ppap/
  —— Chart.yaml      # 차트 메타데이터
  —— values.yaml    # 기본값 파일
  —— charts/         # 다른 차트들이 포함된 디렉토리 (종속성)
  —— templates/       # Kubernetes 리소스 템플릿
    —— deployment.yaml # Deployment 템플릿
    —— service.yaml   # Service 템플릿
    —— ingress.yaml   # Ingress 템플릿
    ...
  ...
  ...
```



Helm Workflow



Helm Workflow

[helm repo add bitnami https://charts.bitnami.com/bitnami](#)

- Bitnami에서 제공하는 Helm 차트 공식 저장소 추가

[helm repo list](#)

- 저장소 목록 확인

```
kchabin@gangdabin-ui-MacBookPro ~ /f5 helm repo list
NAME          URL
bitnami       https://charts.bitnami.com/bitnami
ingress-nginx https://kubernetes.github.io/ingress-nginx
stable        https://charts.helm.sh/stable
```

[helm search repo nginx](#) : repo 중 nginx chart 검색

```
kchabin@gangdabin-ui-MacBookPro ~ /f5 helm search repo nginx
NAME          CHART VERSION APP VERSION DESCRIPTION
bitnami/nginx      19.0.0      1.27.4      NGINX Open Source is a web server that can be a...
bitnami/nginx-ingress-controller 11.6.10     1.12.0      NGINX Ingress Controller is an Ingress controll...
bitnami/nginx-intel      2.1.15      0.4.9       DEPRECATED NGINX Open Source for Intel is a lig...
ingress-nginx/ingress-nginx    4.12.0      1.12.0      Ingress controller for Kubernetes using NGINX a...
stable/nginx-ingress      1.41.3      v0.34.1      DEPRECATED! An nginx Ingress controller that us...
stable/nginx-ldapauth-proxy    0.1.6       1.13.5      DEPRECATED - nginx proxy with ldapauth
stable/nginx-lego          0.3.1       1           Chart for nginx-ingress-controller and kube-lego
stable/gcloud-endpoints      0.1.2       1           DEPRECATED Develop, deploy, protect and monitor...
```

Helm Workflow

helm install test-nginx bitnami/nginx

- Helm 차트를 k8s 클러스터에 배포하기
- test-nginx : 배포 인스턴스 이름
- bitnami/nginx : 사용할 Helm 차트

```
kchabin@gangdabin-ui-MacBookPro ~/f5 ➔ helm install test-nginx bitnami/nginx
NAME: test-nginx
LAST DEPLOYED: Tue Mar  4 22:26:45 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
  CHART NAME: nginx
  CHART VERSION: 19.0.0
  APP VERSION: 1.27.4
```

helm uninstall my-nginx

- 배포된 리소스 삭제

```
kchabin@gangdabin-ui-MacBookPro ~/f5 ➔ helm list
NAME          NAMESPACE   REVISION   UPDATED           STATUS        CHART          APP VERSION
my-nginx      default     1          2025-03-04 22:07:02.009436 +0900 KST  deployed    nginx-ingress-1.41.3  v0.34.1
test-nginx    default     1          2025-03-04 22:26:45.105245 +0900 KST  deployed    nginx-19.0.0       1.27.4
kchabin@gangdabin-ui-MacBookPro ~/f5 ➔ helm uninstall my-nginx
release "my-nginx" uninstalled
```

Helm Custom Deploy

```
brew install helm

helm create ppap #원하는 차트 구조 생성

#실제로 클러스터에 배포하지 않고도 Helm 차트를 템플릿으로 렌더링하여 결과물을 확인
helm template ppap-chart . -f values.yaml

#Helm 차트를 실제로 k8s 클러스터에 설치
helm install ppap-chart . -f values.yaml
```

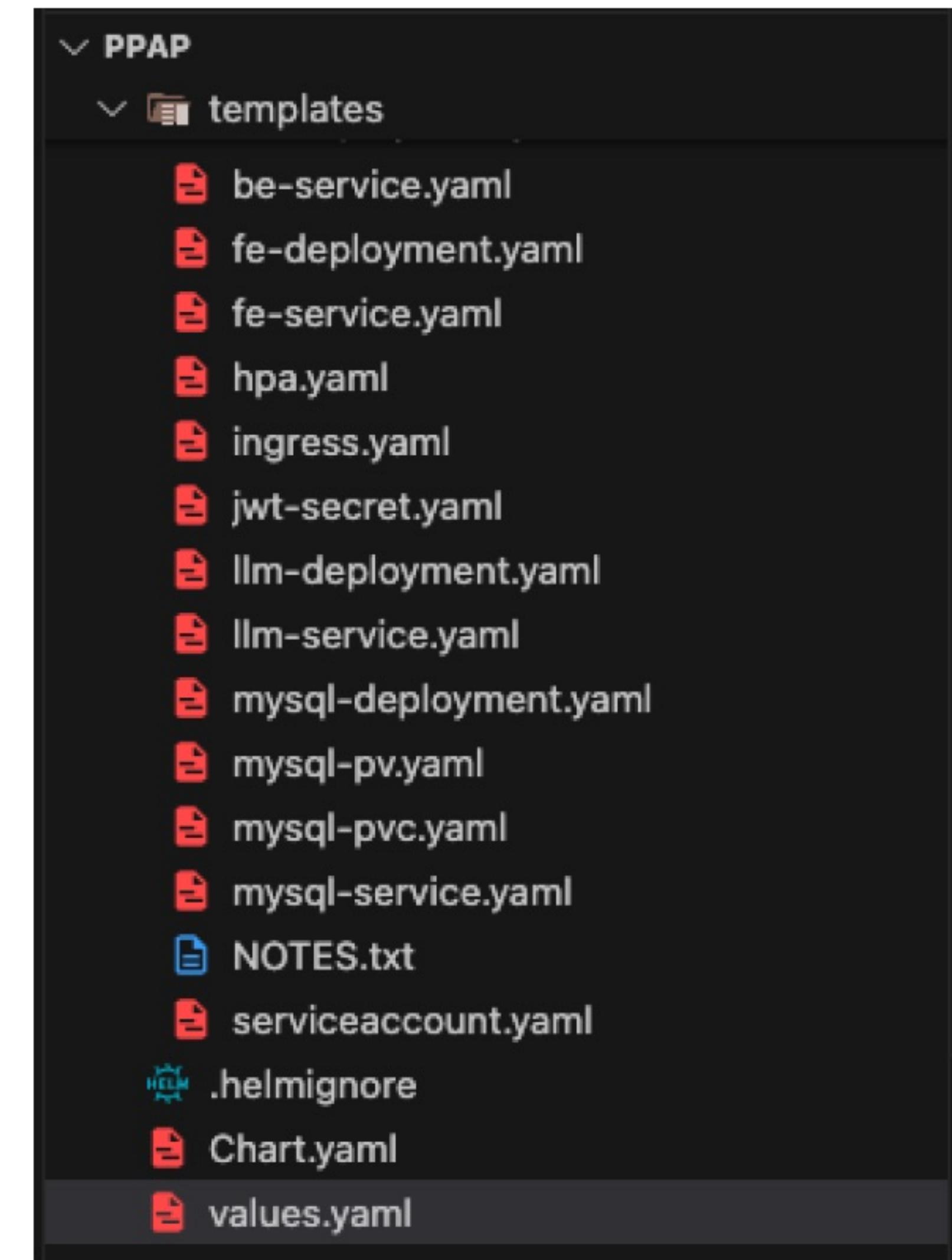


Chart.yaml

apiVersion

- v2 : 최소 헬름3을 필요로 하는 헬름 차트
- v1 : 이전 헬름 버전을 지원하는 차트,
헬름 3에 의해 여전히 설치 가능함.

appVersion

- 어플리케이션의 버전을 명시하는 방법
 - drupal 차트가 appVersion: 8.2.1을 가진다면,
차트에 포함되는 Drupal의 버전은 8.2.1임을 나타냄
 - 정보만 제공하고, 차트 버전 계산에 영향이 없음
 - 버전을 따옴표로 감싸는 것을 매우 권장함.

dependencies

- 의존성 관리

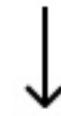
```
apiVersion: 차트 API 버전 (필수)
name: 차트명 (필수)
version: SemVer 2 버전 (필수)
kubeVersion: 호환되는 쿠버네티스 버전의 SemVer 범위 (선택)
description: 이 프로젝트에 대한 간략한 설명 (선택)
type: 차트 타입 (선택)
keywords:
  - 이 프로젝트에 대한 키워드 리스트 (선택)
home: 프로젝트 홈페이지의 URL (선택)
sources:
  - 이 프로젝트의 소스코드 URL 리스트 (선택)
dependencies: # 차트 필요조건들의 리스트 (optional)
  - name: 차트명 (nginx)
    version: 차트의 버전 ("1.2.3")
    repository: 저장소 URL ("https://example.com/charts") 또는 ("@repo-name")
    condition: (선택) 차트들의 활성/비활성을 결정하는 boolean 값을 만드는 yaml 경로 (예시:
tags: # (선택)
  - 활성화 / 비활성을 함께하기 위해 차트들을 그룹화 할 수 있는 태그들
enabled: (선택) 차트가 로드될수 있는지 결정하는 boolean
import-values: # (선택)
  - ImportValues 는 가져올 상위 키에 대한 소스 값의 맵핑을 보유한다. 각 항목은 문자열이거나
alias: (선택) 차트에 대한 별명으로 사용된다. 같은 차트를 여러번 추가해야할때 유용하다.
maintainers: # (선택)
  - name: maintainer들의 이름 (각 maintainer마다 필수)
    email: maintainer들의 email (각 maintainer마다 선택)
    url: maintainer에 대한 URL (각 maintainer마다 선택)
icon: 아이콘으로 사용될 SVG나 PNG 이미지 URL (선택)
appVersion: 이 앱의 버전 (선택). SemVer인 필요는 없다.
deprecated: 차트의 deprecated 여부 (선택, boolean)
annotations:
  example: 키로 매핑된 주석들의 리스트 (선택).
```

values.yaml

Helm 차트에서 사용되는 기본 설정값들을 정의
· -f values.yaml 옵션으로 커스터마이징

```
service:  
  be:  
    port: 8080  
    targetPort: 8080  
    type: ClusterIP
```

values.yaml 설정



```
apiVersion: v1  
kind: Service  
metadata:  
  name: ppap-be-service  
spec:  
  selector:  
    app: ppap-be  
  ports:  
    - protocol: TCP  
      port: {{ .Values.service.be.port }}  
      targetPort: {{ .Values.service.be.targetPort }}  
    type: {{ .Values.service.be.type }}
```

be-service.yaml

helm template 결과

```
---  
# Source: ppap/templates/be-service.yaml  
apiVersion: v1  
kind: Service  
metadata:  
  name: ppap-be-service  
spec:  
  selector:  
    app: ppap-be  
  ports:  
    - protocol: TCP  
      port: 8080  
      targetPort: 8080  
    type: ClusterIP
```

```
● kchabin@gangdabin-ui-MacBookPro ~/ppap ➤ kubectl get pods
NAME          READY   STATUS        RESTARTS   AGE
gs-websocket-6558d977df-n6j5t   1/1    Running      3 (16h ago) 7d4h
mysql-6dd686f779-xmxv9         0/1    ContainerCreating 0           52s
ppap-be-567d8b5784-gw6kw       0/1    InvalidImageName 0           52s
ppap-be-567d8b5784-tl8f9       0/1    InvalidImageName 0           52s
ppap-be-567d8b5784-z7x7k       0/1    InvalidImageName 0           52s
ppap-fe-7db79bf655-kzl85       1/1    Running      0           52s
ppap-fe-7db79bf655-tkct9       1/1    Running      0           52s
ppap-fe-7db79bf655-wjzq2       1/1    Running      0           52s
ppap-llm-6dc86bfd7-5blfp      0/1    ImagePullBackOff 0           52s
ppap-llm-6dc86bfd7-6zms9       0/1    ImagePullBackOff 0           52s
ppap-llm-6dc86bfd7-g9wvz       0/1    ImagePullBackOff 0           52s
test-nginx-67947f579f-jnlwd   1/1    Running      0           6h18m
web-56b9569dcc-kn7fv          1/1    Running      0           14h
web2-6445c79948-82dll         1/1    Running      0           12h
```

ppap-chart.local

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

감사합니다