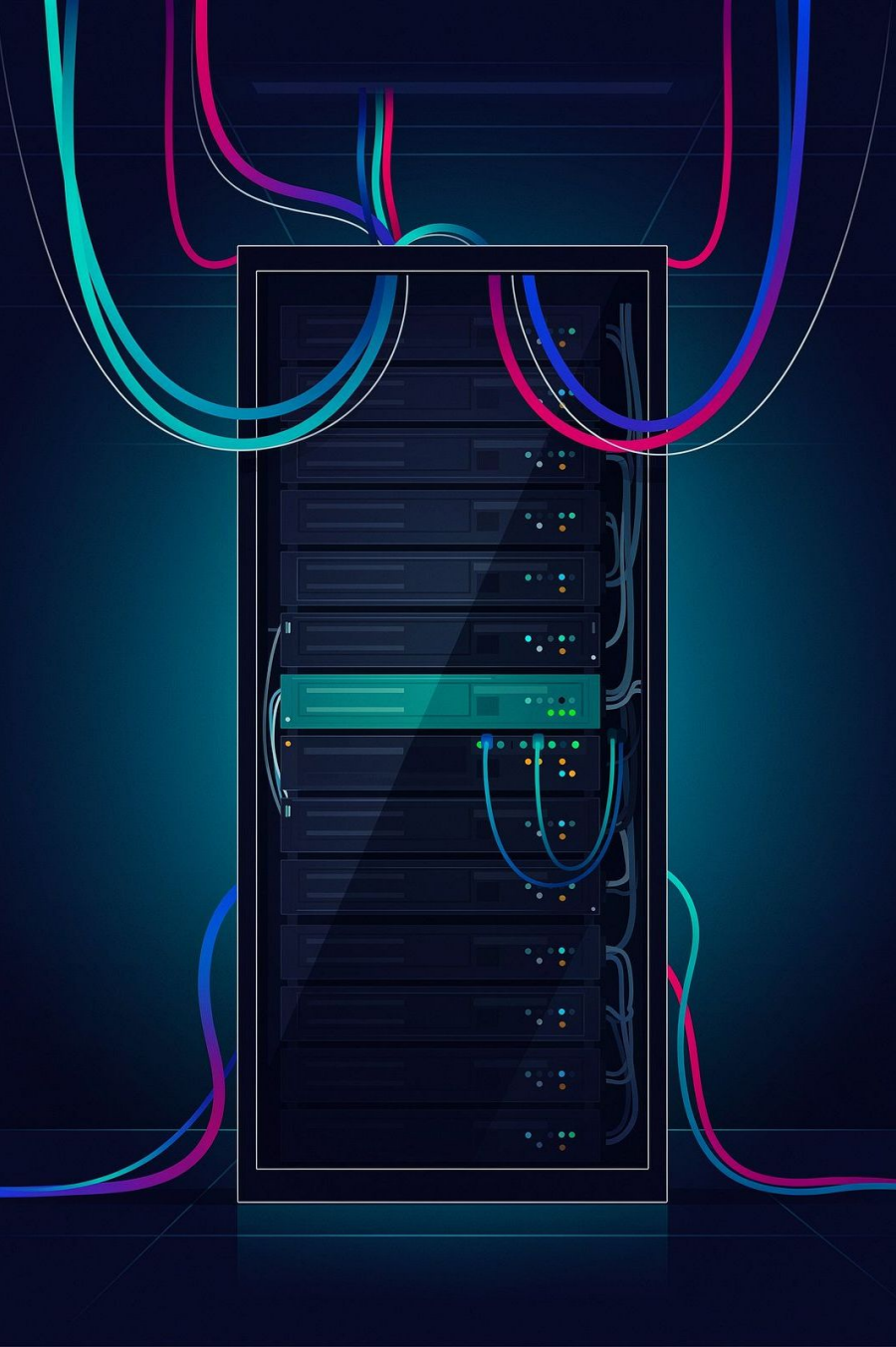




지원 데이터 소스 & 바인딩 Env 주입

YakCloud가 지원하는 9가지 데이터 소스와, 앱에 바인딩했을 때 주입되는 환경변수(env) 규약을 정리합니다. 접속 정보를 코드에 하드코딩하지 말고, 플랫폼이 주입하는 env를 통해 안전하게 읽으세요.

YAKCLOUD 개발자 가이드

데이터 소스 & 바인딩

지원 데이터 소스 9종

매니페스트의 `requires[].type` 및 `binds` 필드에서 아래 타입 식별자를 사용합니다. 플랜(`small` / `medium` / `large`)으로 쿼터와 용량을 지정하며, 선언된 이름이 없으면 배포 시 자동 프로비저닝됩니다.



관계형 DB

`postgresql` · `mysql` · `mariadb` · `oracle`



도큐먼트 DB

`mongodb` — 유연한 스키마, 도큐먼트 지향 저장소



인메모리 KV

`redis` — 캐시 및 세션 스토리지



오브젝트 스토리지

`minio` — S3 호환 파일/바이너리 저장



메시지 브로커

`rabbitmq` — AMQP 기반 비동기 메시징



검색 엔진

`solr` — 전문(Full-text) 검색 인덱싱

매니페스트에서 선언 & 바인딩

소스 선언 — requires

requires:

- name: orders-db
- type: postgresql
plan: small

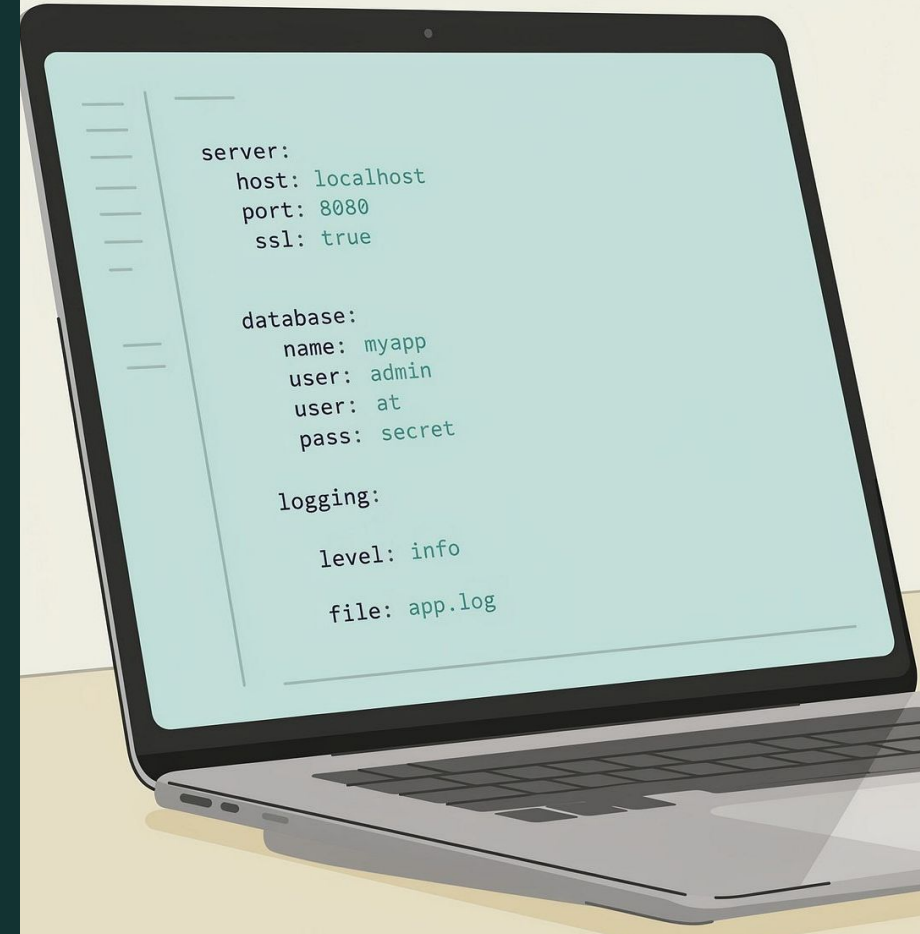
논리 이름(name)으로 소스를 식별합니다. 존재하지 않으면 자동 프로비저닝되고, 이미 있으면 스킵됩니다.

워크로드 바인딩 — binds

workloads:

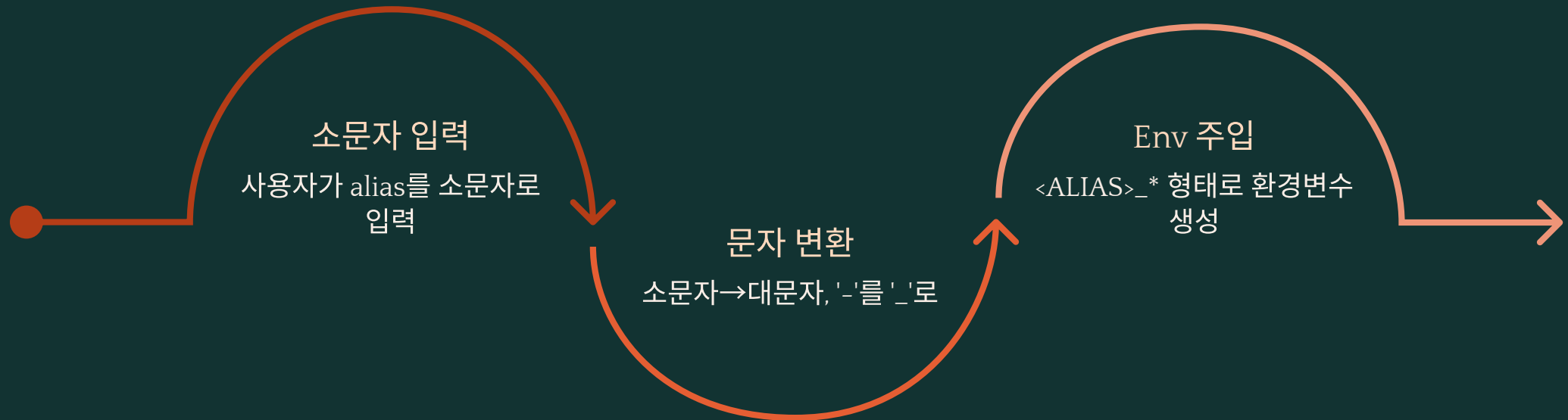
- name: api
- image: ...:\${TAG}
- binds:
- alias: orders
- source: orders-db

source는 requires의 name, alias는 env 프리픽스가 됩니다. 한 앱에 여러 소스를 각기 다른 alias로 바인딩할 수 있습니다.



Alias → Env 프리픽스 변환 규칙

바인딩 시 지정한 `alias`가 환경변수 이름의 프리픽스가 됩니다. 변환 규칙은 단순하며 일관됩니다.



`orders`

→ `ORDERS_HOST`, `ORDERS_PORT`,
`ORDERS_URL` ...

`my-cache`

→ `MY_CACHE_HOST`, `MY_CACHE_PORT`,
`MY_CACHE_URL` ...

`files`

→ `FILES_ENDPOINT`, `FILES_BUCKET`,
`FILES_ACCESS_KEY` ...

공통 & 관계형 DB Env 키

모든 타입에 공통으로 <ALIAS>_HOST와 <ALIAS>_PORT가 주입됩니다. 관계형 DB 계열(postgresql, mysql, mariadb, oracle)은 아래 추가 키를 제공합니다.

❗ 아래 키는 모두 <ALIAS>_ 프리픽스가 붙습니다. 예: alias=orders → ORDERS_HOST

타입	추가 env 키
공통 (모든 타입)	_HOST, _PORT
postgresql	_USERNAME, _PASSWORD, _DB, _URL (postgresql://user:pw@host:port/db)
mysql	_USERNAME, _PASSWORD, _DB, _URL (mysql://user:pw@host:port/db)
mariadb	_USERNAME, _PASSWORD, _DB, _URL (mariadb://user:pw@host:port/db)
oracle	_USERNAME, _PASSWORD, _DB, _SERVICE_NAME, _SID, _URL (oracle://user:pw@host:port/service)

MongoDB · Redis · Solr Env 키

❗ 아래 키는 모두 <ALIAS>_ 프리픽스가 붙습니다. _HOST / _PORT는 모든 타입 공통입니다.

MongoDB

_HOST (공통)

_PORT (공통)

_USERNAME

_PASSWORD

_DB

_URL

mongodb://user:pw@host:port/db?authSource=db

Redis

_HOST (공통)

_PORT (공통)

_USERNAME (빈값)

_PASSWORD

_DB (번호)

_URL

redis://pw@host:port/db

Solr

_HOST (공통)

_PORT (공통)

_USERNAME (빈)

_PASSWORD (빈)

_CORE

_ENDPOINT

http://host:port/solr/core

MinIO & RabbitMQ Env 키

오브젝트 스토리지와 메시지 브로커는 표준 DB와 다른 전용 키를 제공합니다. `_HOST` / `_PORT`는 모든 타입 공통입니다.

MinIO (S3 호환 오브젝트 스토리지)

- `_HOST` – 스토리지 호스트 (공통)
- `_PORT` – 포트 (공통)
- `_ACCESS_KEY` – S3 액세스 키
- `_SECRET_KEY` (= `_PASSWORD`)
- `_ENDPOINT` – `http(s)://host:port`
- `_BUCKET` – 기본 버킷 이름
- `_REGION` – 기본값 `us-east-1`
- `_USE_SSL` – SSL 사용 여부
- `_URL` = `_ENDPOINT`

RabbitMQ (AMQP 메시지 브로커)

- `_HOST` – 브로커 호스트 (공통)
- `_PORT` – 포트 (공통)
- `_USERNAME`
- `_PASSWORD`
- `_VHOST` – 가상 호스트
- `_MGMT_URL` – `http://host:15672`
- `_URL` – `amqp://user:pw@host:port/vhost`

앱 코드 예시 — Env로 접속 정보 읽기

하드코딩 없이 환경변수만으로 각 데이터 소스에 연결하는 실무 예시입니다.

Node.js — MongoDB (alias=mongo)

```
const client = new MongoClient(  
  process.env.MONGO_URL  
);
```

Node.js — MinIO (alias=files)

```
const s3 = new S3Client({  
  endpoint: process.env.FILES_ENDPOINT,  
  region: process.env.FILES_REGION,  
  credentials: {  
    accessKeyId:  
      process.env.FILES_ACCESS_KEY,  
    secretAccessKey:  
      process.env.FILES_SECRET_KEY  
  }  
});  
const bucket = process.env.FILES_BUCKET;
```

Python — PostgreSQL (alias=pg)

```
import psycopg2, os  
pg = psycopg2.connect(  
  os.environ["PG_URL"]  
)
```

Python — MariaDB (alias=maria)

```
import pymysql, os  
maria = pymysql.connect(  
  host=os.environ["MARIA_HOST"],  
  port=int(os.environ["MARIA_PORT"]),  
  user=os.environ["MARIA_USERNAME"],  
  password=os.environ["MARIA_PASSWORD"],  
  database=os.environ["MARIA_DB"]  
)
```

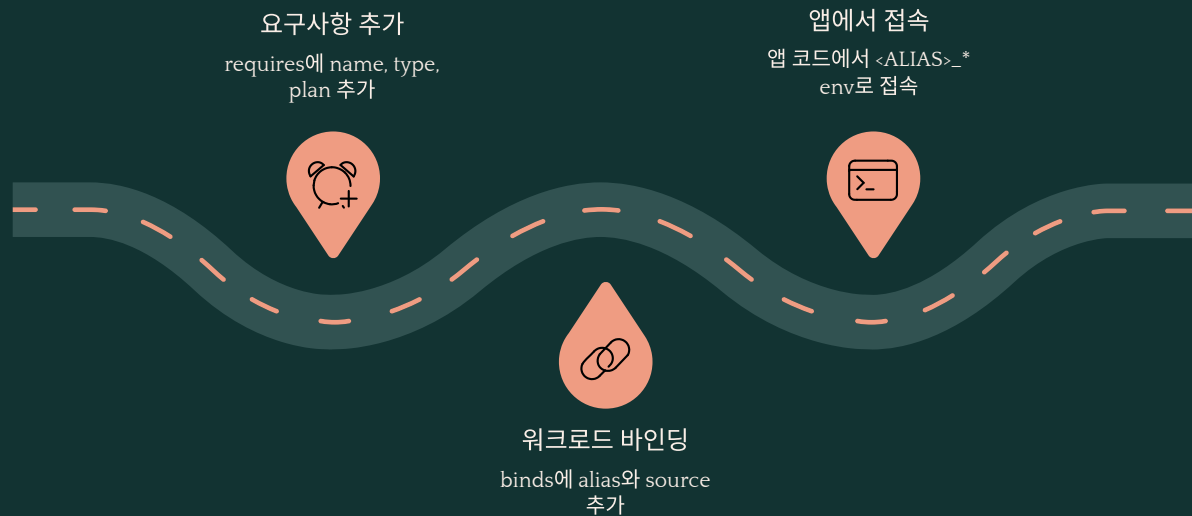
샘플 앱 바인딩 구성

아래는 실제 샘플 앱에서 사용하는 소스-워크로드-env 매핑입니다. 각 워크로드는 alias를 통해 독립적으로 소스에 접근합니다.

워크로드	alias	source (type)	사용 env
demo-node	mongo	mongodb	MONGO_URL
demo-py	pg	postgresql	PG_URL
demo-py	maria	mariadb	MARIA_HOST / _PORT / _USERNAME / _PASSWORD / _DB

한 워크로드(demo-py)가 서로 다른 alias로 두 개의 소스를 동시에 바인딩하는 패턴을 보여줍니다. alias가 다르면 env 충돌이 발생하지 않습니다.

새 소스 추가 — 3단계 요약



🔒 보안 원칙

비밀번호·키는 Secret으로 주입, 콘솔
에 노출 없음

⚙️ 자동 프로비저닝

선언 시 미존재 소스는 배포 때 자동 생
성

🔗 다중 바인딩

alias별 격리로 한 앱이 여러 소스를 동시 사용 가능

