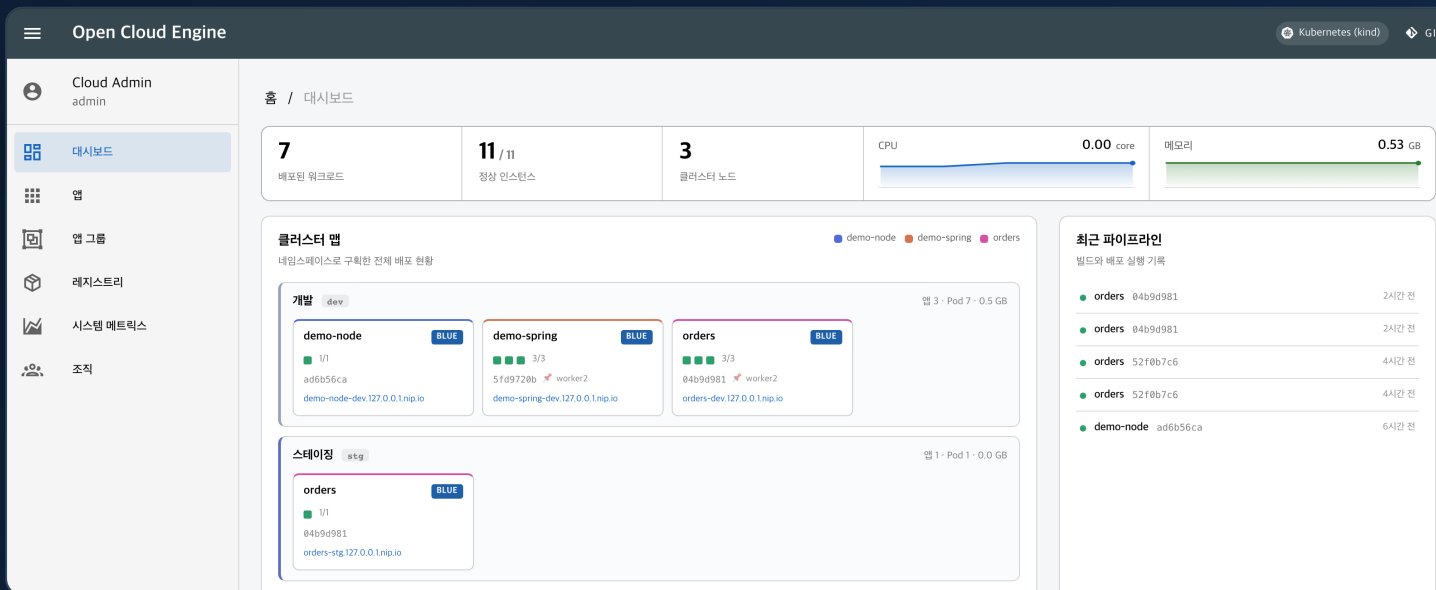


제품 소개 및 도입 제안서

개발자는 코드만 다루고, 쿠버네티스는 포털이 맡습니다.

카탈로그에서 서비스를 만들면 저장소 생성·빌드·배포·무중단 전환·운영까지 하나의 화면에서 끝납니다. 쿠버네티스 위에서 동작하는 개발자 셀프서비스 PaaS 포털입니다.



3단계

개발 · 스테이징 · 운영을 한 포털에서

5종

무중단 배포 전략 내장

한 화면

클러스터 전체 배포와 노드 배치

2026. 09

Kubernetes 기반 · 온프레미스 설치형

주식회사 유엔진

목차

01 도입 배경	3	12 특징점 ⑤ 묶음 관리와 권한	14
02 제품 소개	4	13 특징점 ⑥ 개방형 · 온프레미스	15
03 주요 산출물 ① 셀프서비스 포털	5	14 시스템 구성과 실행 환경	16
04 주요 산출물 ② 배포 파이프라인	6	15 기술 스택	17
05 제품 화면 한눈에 보기	7	16 사용 절차 ① 생성부터 개발 배포	18
06 특징점 ① 한 흐름으로 연결	8	17 사용 절차 ② 승격과 무중단 전환	19
07 특징점 ② 무중단 배포	9	18 보안 · 접근 통제	20
08 특징점 ③ 운영 도구 내장	10	19 설치 · 납품 구성	21
09 실시간 로그와 컨테이너 접속	11	20 검증 체계	22
10 특징점 ④ 되돌릴 수 있는 운영	12	21 레퍼런스	23
11 클러스터 전체를 한 화면에	13	22 도입 절차 및 일정	24

한 장 요약

쿠버네티스는 강력하지만, 서비스 하나를 올리려면 매니페스트·CI 설정·인그레스·시크릿을 각각 다뤄야 합니다. uEngine Cloud 는 이 과정을 **앱 단위**로 묶어 포털 한 곳에서 처리합니다. 운영에 필요한 로그·터미널·환경설정·무중단 전환도 같은 화면에 있습니다.

앱 단위 추상화

개발자 관점

매니페스트를 작성하지 않고 이름과 자원만 정합니다. 클러스터 개념을 몰라도 서비스를 올릴 수 있습니다

무중단 전환

운영 관점

blue/green · 카나리 · A/B 를 화면에서 조정합니다. 문제가 생기면 이전 버전으로 즉시 되돌립니다

포털에서 완결

조직 관점

생성 · 빌드 · 배포 · 운영이 한 흐름으로 이어집니다. 도구를 여러 개 연결하지 않습니다

핵심 차이

개발자 포털 제품 대부분은 카탈로그와 문서만 제공하고, 실제 빌드와 배포는 별도 도구에 맡깁니다. uEngine Cloud 는 포털과 실행 계층을 한 제품에 담아, 도구를 여러 개 연결하지 않아도 서비스 생성부터 운영까지 진행할 수 있습니다. 소스가 공개되어 있어 사내 클러스터와 폐쇄망에 그대로 설치할 수 있고, 화면과 안내 문구가 한국어로 제공됩니다. 기존에 운영 중인 저장소·레지스트리·인증 서버가 있다면 그대로 연결해 사용합니다.

쿠버네티스 도입 이후에도 남아 있는 문제

컨테이너 오케스트레이션은 표준이 되었지만, 개발자가 다뤄야 할 개념은 오히려 늘었습니다. 서비스 하나를 배포하려면 **Deployment · Service · Ingress · ConfigMap** 을 각각 작성하고, CI 파이프라인과 레지스트리 인증을 따로 설정해야 합니다.

문제 1

배포 준비에 드는 시간

새 서비스를 시작할 때마다 저장소 생성, CI 설정, 매니페스트 작성, 도메인 연결을 반복합니다. 팀마다 방식이 달라져 표준을 유지하기 어렵습니다.

문제 2

플랫폼 팀에 몰리는 요청

인스턴스를 늘리거나 환경변수를 바꾸는 일까지 플랫폼 담당자를 거칩니다. 담당자는 반복 요청 처리에 시간을 쓰고, 개발팀은 대기합니다.

문제 3

무중단 배포의 진입 장벽

blue/green 전환과 카나리 배포는 별도 도구를 도입하고 학습해야 합니다. 결국 야간 배포와 전면 교체로 되돌아갑니다.

문제 4

흩어진 운영 도구

로그는 로그 도구에서, 지표는 모니터링 도구에서, 컨테이너 접속은 터미널에서 합니다. 장애 상황에서 화면을 여러 개 오가야 합니다.

개발자 포털 도구를 검토했을 때 부딪히는 지점

유형	도입 시 확인이 필요한 부분
클러스터 관리 콘솔	클러스터 리소스를 잘 보여주지만, 서비스를 만들고 배포하는 흐름은 제공하지 않습니다.
개발자 포털 도구	카탈로그와 문서 중심입니다. 실제 빌드·배포는 별도 도구를 연결해야 합니다.
상용 PaaS 서비스	사용이 간편하지만 사내 폐쇄망이나 자체 클러스터에 설치하기 어렵습니다.
직접 구축	구축 자체보다 이후 유지와 인수인계에 인력이 계속 필요합니다.
배포 자동화 도구	배포는 자동화되지만 서비스 생성과 운영 화면은 별도로 마련해야 합니다.

필요한 것은 하나의 흐름입니다. 서비스를 만들고, 빌드하고, 단계별로 올리고, 문제가 생기면 되돌리는 과정을 개발자가 스스로 처리할 수 있어야 합니다.

uEngine Cloud 소개

uEngine Cloud 는 쿠버네티스 클러스터 위에서 동작하는 **개발자 셀프서비스 PaaS 포털**입니다. 개발자는 카탈로그에서 템플릿을 골라 서비스를 만들고, 코드를 저장소에 올립니다. 그러면 소스를 받아 이미지를 만들고 배포까지 수행하는 자동 처리 과정, 곧 **파이프라인**이 실행됩니다. 운영자는 같은 앱 단위로 자원·배포 이력·로그를 확인합니다.

입력

템플릿과 소스 코드

카탈로그에서 템플릿을 고르고 앱 이름과 자원을 정합니다. 이후에는 소스 코드를 저장소에 올리는 것이 전부입니다.

처리

저장소 생성부터 배포까지

저장소가 자동으로 만들어지고, 컨테이너 이미지가 빌드되어 레지스트리에 등록된 뒤 지정한 단계에 배포됩니다.

출력

동작하는 서비스와 운영 화면

접속 주소가 준비된 서비스, 그리고 자원·로그·지표·배포 이력을 확인하고 조정하는 화면을 함께 제공합니다.

기존 방식과의 비교

작업	직접 구성하는 경우	uEngine Cloud
서비스 시작	저장소 생성, CI 설정, 매니페스트 4종 작성	카탈로그에서 템플릿 선택 후 이름 입력
이미지 빌드	CI 도구 설치와 러너 관리, 레지스트리 인증 설정	클러스터 안에서 빌드 작업이 실행되고 결과가 포털에 표시됨
자원 조정	매니페스트 수정 후 재적용	화면에서 인스턴스·CPU·메모리를 조정하고 저장
무중단 배포	별도 배포 도구 도입과 학습	배포 전략 화면에서 비율을 조정
단계 이관	단계마다 다시 빌드하거나 이미지를 수동으로 옮김	검증한 이미지를 그대로 상위 단계로 승격
운영 확인	로그·지표·터미널 도구를 각각 사용	앱 상세 화면 안에서 확인

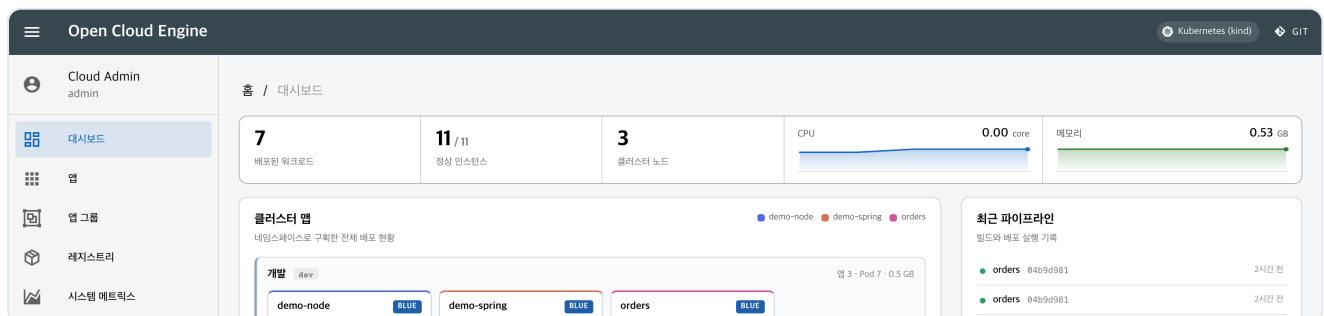


그림 1. 첫 화면인 클러스터 맵. 개발·스테이징·운영으로 구획해 전체 배포 현황을 보여줍니다. 데모 데이터 기준입니다.

주요 산출물 ① — 셀프서비스 포털

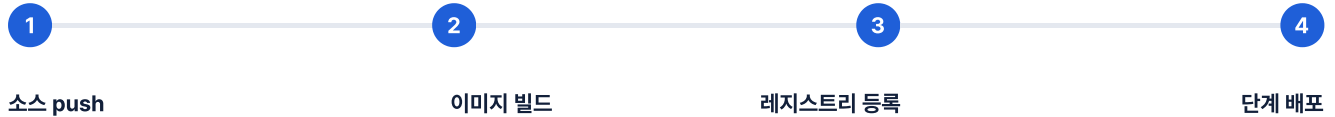
앱 하나를 운영하는 데 필요한 화면을 **10개 메뉴**로 정리했습니다. 개발자와 운영자가 같은 화면을 보면서 서로 다른 작업을 수행합니다.

메뉴	제공 내용
시작하기	저장소 내려받기 명령, 단계별 접속 주소, 다음에 할 일 안내
개요	개발·스테이징·운영 단계별 배포 상태 확인, 검증한 이미지를 상위 단계로 올리는 승격 기능
런타임 및 환경	인스턴스·CPU·메모리 조정, 환경변수 편집, 컨테이너 접속, 구성 확인
소스코드	포털 안에서 저장소 파일 탐색과 열람
빌드 및 배포	파이프라인 진행 상황, 빌드 설정, 배포 전략 조정
스냅샷	배포 상태와 환경설정 저장, 지정 시점으로 복원
로그	실시간 로그 수신, 검색·심각도 필터, 파일 저장
모니터링	CPU·메모리·정상 인스턴스 추이, 클러스터 이벤트
레지스트리	빌드된 이미지 목록, 이전 버전 재배포
변경이력	생성·배포·승격·비율 조정·복원 기록

그림 2. 앱 상세 개요. 세 단계의 배포 슬롯과 커밋, 트래픽 비율을 한 번에 확인합니다. 운영 단계는 신규 슬롯(green)과 이전 슬롯(blue)이 함께 유지되고 있습니다. *데모 데이터 기준입니다.*

주요 산출물 ② — 배포 파이프라인

저장소에 코드를 올리면 별도 CI 도구 없이 클러스터 안에서 빌드 작업이 실행됩니다. 빌드 결과는 레지스트리에 등록되고, 설정한 단계에 배포됩니다.



포털에서 조정하는 항목

항목	설명
자동 실행 여부	코드를 올렸을 때 파이프라인을 자동으로 시작할지 정합니다.
대상 브랜치	자동 실행 대상 브랜치를 지정합니다. 여러 개를 지정할 수 있습니다.
자동 배포 단계	빌드 성공 후 배포할 단계를 지정합니다. 비워 두면 이미지 빌드까지만 수행합니다.
빌드 제한 시간	빌드가 지나치게 길어질 때 중단할 기준을 정합니다.

Open Cloud Engine 포털의 빌드 진행 상황 화면. 화면 상단에는 'orders' 앱의 빌드 진행 상황, 커밋, 소스 리포지토리, 빌드 및 배포 탭이 표시되어 있습니다. 화면 하단에는 빌드 로그가 표시되어 있습니다.

그림 3. 빌드 진행 상황. 단계별 진행과 결과 이미지가 표시되며, 빌드 로그를 그대로 확인할 수 있습니다. 데모 데이터 기준입니다.

CI 서버를 따로 설치하지 않습니다. 빌드는 클러스터 안에서 격리된 작업으로 실행되며, 작업이 끝나면 정리됩니다. 러너를 상시 유지할 필요가 없습니다.

제품 화면 한눈에 보기

아래 화면은 모두 실제로 동작 중인 클러스터의 화면을 그대로 담은 것입니다.

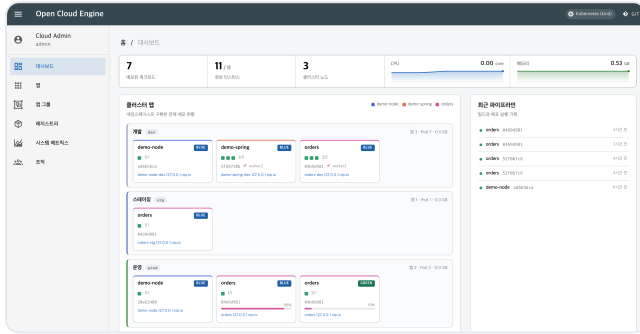


그림 4. 클러스터 맵 — 구획별 배포 현황을 한눈에 봅니다.

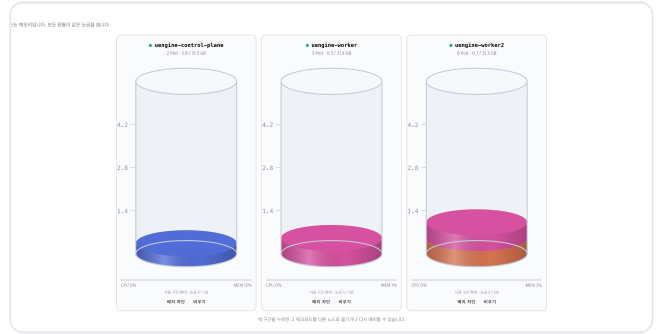


그림 5. 노드 배치 — 어떤 앱이 어느 노드에 얼마나 있었는지 봅니다.



그림 6. 카탈로그 — 템플릿을 골라 서비스를 시작합니다.

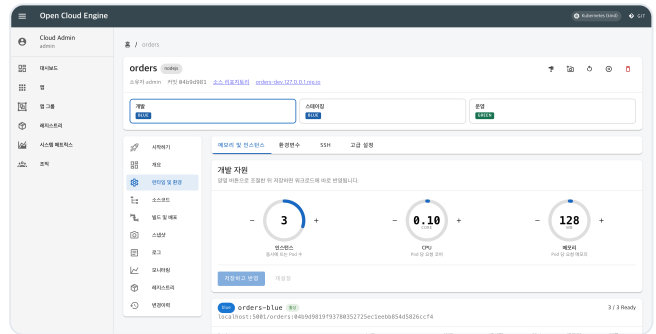


그림 7. 런타임 — 인스턴스·CPU·메모리를 조정합니다.

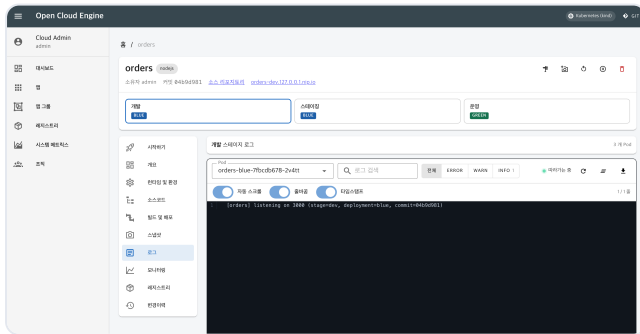


그림 8. 로그 — 실시간으로 받아 검색·필터합니다.

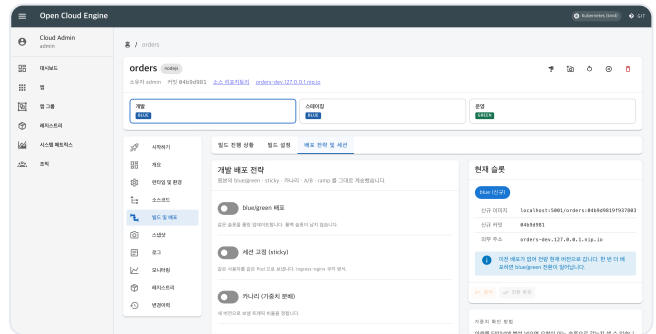


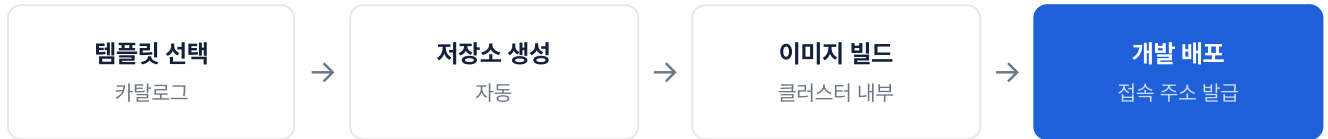
그림 9. 배포 전략 — 무중단 전환 방식을 정합니다.

화면 구성

좌측 메뉴는 대시보드 · 앱 · 앱 그룹 · 레지스트리 · 시스템 메트릭스 · 조직으로 구성됩니다. 앱을 선택하면 앞서 소개한 10개 메뉴가 펼쳐집니다.

특징점 ① — 서비스 생성부터 운영까지 한 흐름

도구를 여러 개 연결하지 않습니다. 카탈로그에서 이름을 정하는 순간 저장소·빌드·배포·접속 주소가 함께 준비됩니다.



앱을 만들면 함께 준비되는 것

- **Git 저장소** — 선택한 템플릿의 소스와 컨테이너 빌드 파일이 들어 있습니다.
- **코드 반영 연동** — 코드를 올리면 파이프라인이 시작되도록 연결됩니다.
- **단계별 접속 주소** — 개발·스테이징·운영 세 개의 주소가 정해집니다. 직접 지정할 수도 있습니다.
- **환경설정 파일** — 단계별로 분리된 설정이 만들어지고, 컨테이너에 전달됩니다.

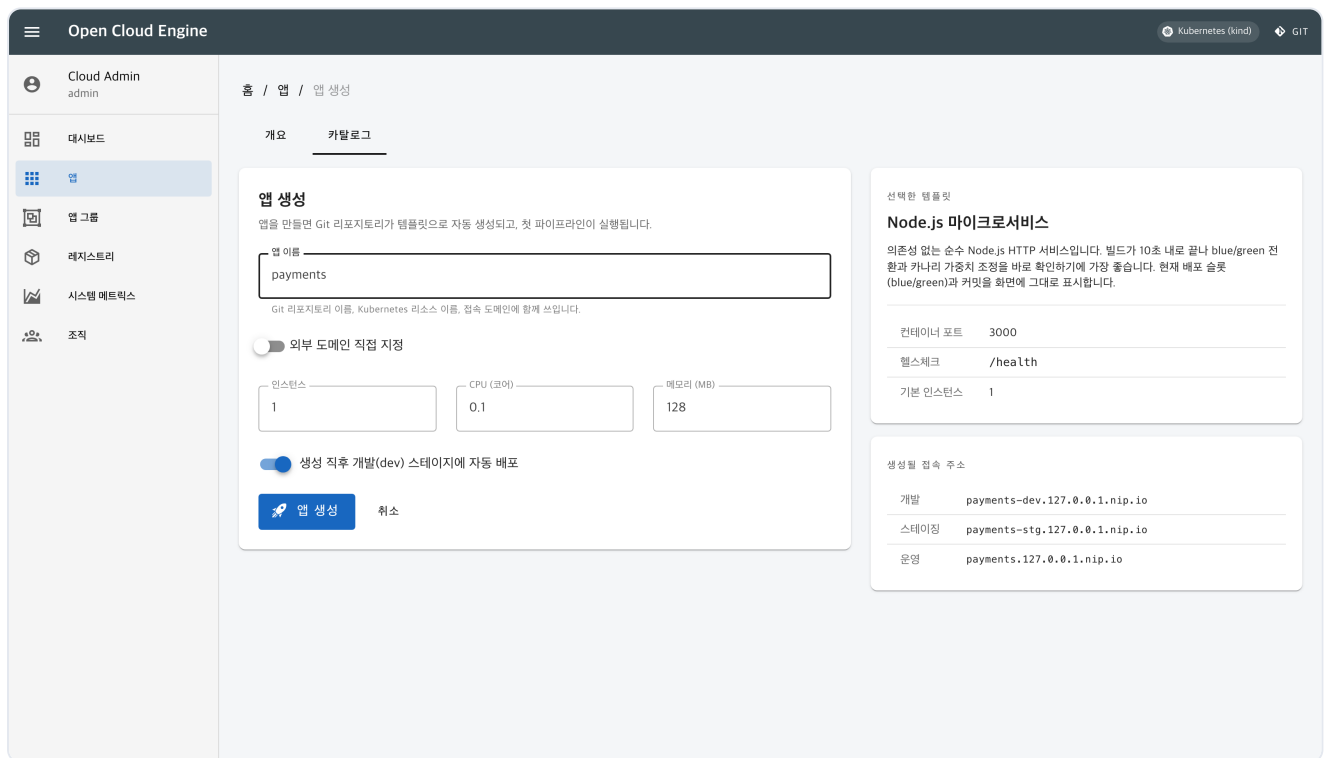


그림 10. 앱 생성 화면. 운영 도메인을 입력하면 스테이징·개발 주소가 함께 정해집니다. *데모 데이터 기준입니다.*

템플릿에 들어 있는 것

실행 가능한 예제 코드, 컨테이너 빌드 파일, 헬스체크 경로 설정, 단계별 환경설정 골격이 함께 들어 있습니다. 생성 직후 바로 빌드되고 배포되므로, 처음부터 동작하는 상태에서 개발을 시작합니다.

제공되는 템플릿은 **Node.js · Spring Boot 3 · Vue 3** 세 가지이며, 조직의 표준에 맞춰 템플릿을 추가할 수 있습니다.

특징점 ② — 무중단 배포를 화면에서

운영 단계는 두 개의 배포 슬롯으로 동작합니다. 새 버전은 사용하지 않는 슬롯에 올라가고, 트래픽 비율을 조정하면서 전환합니다. 이전 슬롯은 그대로 남아 언제든 되돌릴 수 있습니다.

제공하는 배포 전략

전략	동작
blue / green	새 버전을 반대 슬롯에 배포하고 전환합니다. 이전 슬롯은 복귀용으로 유지됩니다.
카나리	새 버전으로 보낼 트래픽 비율을 지정합니다. 화면에서 비율을 조정합니다.
A/B 테스트	지정한 요청 헤더를 가진 요청만 새 버전으로 전달합니다.
세션 고정	같은 사용자의 요청이 같은 인스턴스로 전달되도록 합니다.
롤링 속도	교체 중 동시에 띄울 인스턴스와 내릴 인스턴스의 비율을 정합니다.



그림 11. 이전 버전으로 전달된 요청.



그림 12. 신규 버전으로 전달된 요청.

실측 결과 — 신규 버전 비율을 10퍼센트로 설정한 뒤 운영 주소로 20회 요청한 결과, 이전 버전 16회 · 신규 버전 4회로 전달되었습니다. 설정한 비율대로 나뉘는 것을 확인했습니다.

특징점 ③ — 운영 도구를 포털 안에

장애를 확인할 때 화면을 여러 개 오갈 필요가 없습니다. 로그·지표·컨테이너 접속·환경설정이 모두 **같은 앱 화면 안에** 있습니다.

환경설정

단계별 환경설정을 포털의 편집기에서 직접 수정합니다. 저장하면 재배포가 필요하다는 표시가 나타나고, 재배포 시 컨테이너에 반영됩니다. 설정 항목은 환경변수로 전달되며, 파일 형태로도 함께 전달됩니다.

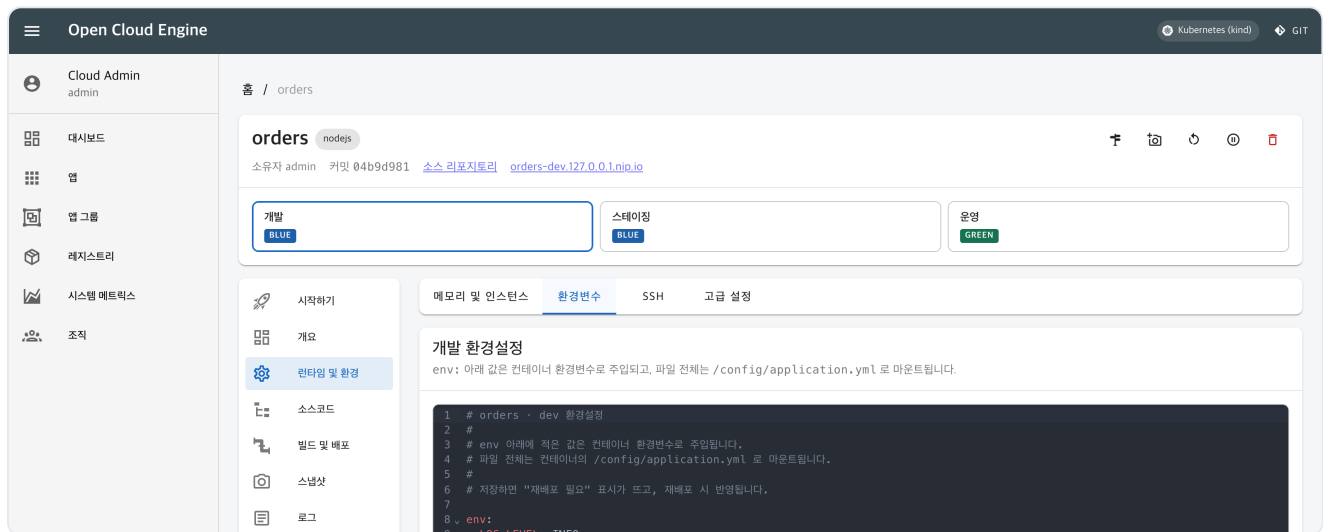


그림 13. 환경설정 편집기. 문법 오류가 있으면 저장 시점에 사유를 알려 줍니다. 데모 데이터 기준입니다.

자원 조정

인스턴스 수와 CPU·메모리는 화면의 조절기로 변경합니다. 바꾼 값은 저장하기 전에 기존 값과 함께 표시되어, 무엇이 달라지는지 확인한 뒤 반영할 수 있습니다.

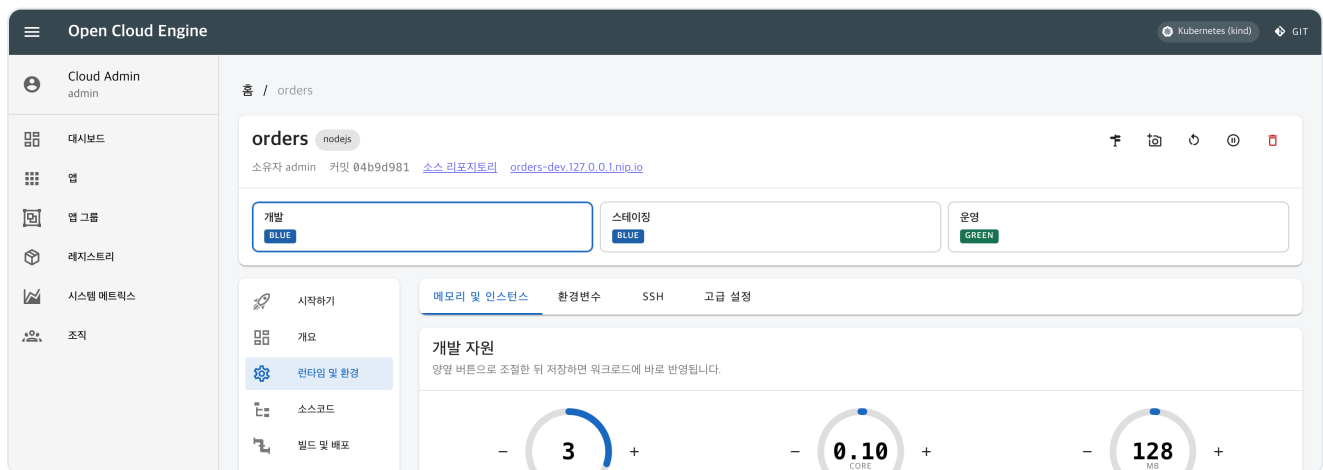


그림 14. 자원 조정 화면. 변경한 값이 기존 값과 함께 표시됩니다. 데모 데이터 기준입니다.

실시간 로그와 컨테이너 접속

문제를 확인하는 두 가지 방법을 포털에 넣었습니다. 로그를 실시간으로 이어 받는 화면과, 컨테이너에 직접 접속하는 화면입니다.

로그

- **실시간 수신** — 인스턴스를 선택하면 로그가 이어서 표시됩니다.
- **검색과 강조** — 찾는 문자열이 화면에서 강조됩니다.
- **심각도 구분** — 오류·경고·정보를 색으로 구분하고, 유형별로 걸러 볼 수 있습니다.
- **파일 저장** — 확인 중인 내용을 그대로 파일로 내려받습니다.

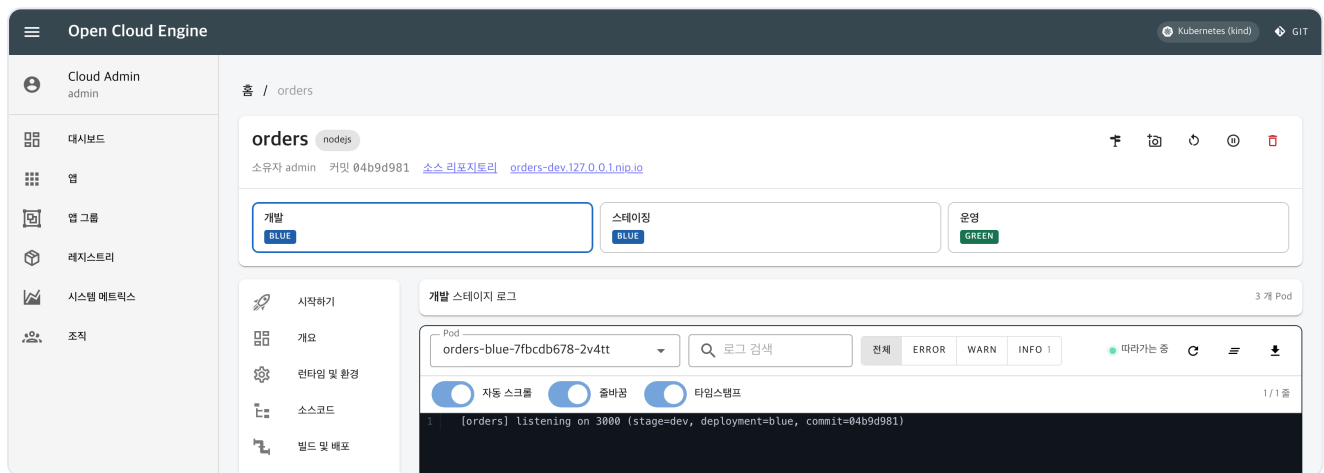


그림 15. 로그 화면. 자동 스크롤·줄바꿈·시각 표시를 각각 켜고 끌 수 있습니다. 데모 데이터 기준입니다.

컨테이너 접속

실행 중인 컨테이너에 브라우저에서 바로 접속합니다. 별도의 접속 서버를 두지 않고 쿠버네티스가 제공하는 실행 기능을 사용하므로, 접속 기록이 클러스터 감사 로그에 남습니다.

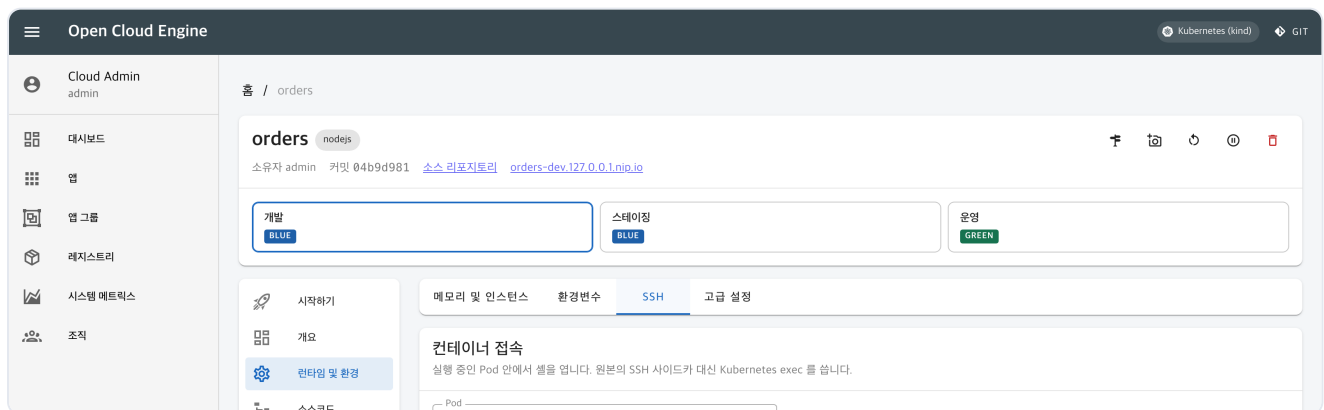


그림 16. 컨테이너 접속 화면. 데모 데이터 기준입니다.

특징점 ④ — 되돌릴 수 있는 운영

배포는 되돌릴 수 있어야 안심하고 할 수 있습니다. uEngine Cloud 는 세 가지 복귀 수단을 제공합니다.

수단 1

슬롯 복귀

이전 슬롯이 그대로 유지되므로,
문제가 확인되면 즉시 이전
버전으로 되돌립니다.

수단 2

이미지 재배포

레지스트리에 등록된 과거
이미지를 다시 빌드하지 않고
선택해서 배포합니다.

수단 3

스냅샷 복원

배포 상태와 환경설정을 함께
저장해 두고, 지정한 시점의
구성으로 되돌립니다.

스냅샷에 담기는 내용

항목	설명
배포 이미지와 커밋	단계별로 어떤 버전이 배포되어 있었는지 기록합니다.
자원 설정	인스턴스 수와 CPU·메모리 값을 함께 저장합니다.
배포 전략	전환 방식과 트래픽 비율 설정을 저장합니다.
환경설정	단계별 환경설정 내용을 함께 저장합니다.

The screenshot shows the 'orders' deployment page in the Open Cloud Engine console. The page is divided into several sections:

- Header:** Open Cloud Engine, Kubernetes (kind), and GIT.
- Left Sidebar:** Cloud Admin, 대시보드, 앱, 앱 그룹, 레지스트리, 시스템 메트릭스, 조직.
- Main Content Area:**
 - orders (nodejs):** Shows the deployment details for the 'orders' application.
 - 개발 (BLUE):** The current deployment stage.
 - 스테이징 (BLUE):** The staging deployment stage.
 - 운영 (GREEN):** The production deployment stage.
 - 스냅샷 (Snapshot):** A section indicating that snapshots are created for each deployment stage to allow for easy rollback.
 - 운영 전환 직후 (After Operation Switch):** A table showing the deployment details for each stage.

스테이징	슬롯	커밋	인스턴스	이미지
개발	BLUE	04b9d981	3	localhost:5001/orders:04b9d9819f9378035272Sec1eebb854d5826ccf4
스테이징	BLUE	04b9d981	1	localhost:5001/orders:04b9d9819f9378035272Sec1eebb854d5826ccf4
운영	GREEN	04b9d981	1	localhost:5001/orders:04b9d9819f9378035272Sec1eebb854d5826ccf4

그림 17. 스냅샷 목록, 복원할 단계를 고르고, 재배포 여부를 선택합니다. 데모 데이터 기준입니다.

모든 조작은 **변경이력**에 남습니다. 누가 언제 어떤 단계를 배포하고 되돌렸는지 앱별로 확인할 수 있습니다.

클러스터 전체를 한 화면에

포털에 들어오면 가장 먼저 **클러스터 맵**이 보입니다. 어떤 서비스가 어느 단계에 몇 개 떠 있는지, 트래픽이 어떻게 나뉘고 있는지를 화면을 옮기지 않고 확인합니다.

구획으로 나눈 배포 현황

- **단계별 구획** — 개발·스테이징·운영이 각각의 영역으로 나뉘어 표시됩니다.
- **인스턴스 단위 표시** — 네모 점 하나가 실행 중인 인스턴스 하나입니다. 정상이면 초록으로 켜집니다.
- **전환 상태** — 배포 중인 워크로드에는 테두리가 깜빡이고, 무중단 전환 중이면 신규·이전 버전의 배분 비율이 함께 표시됩니다.

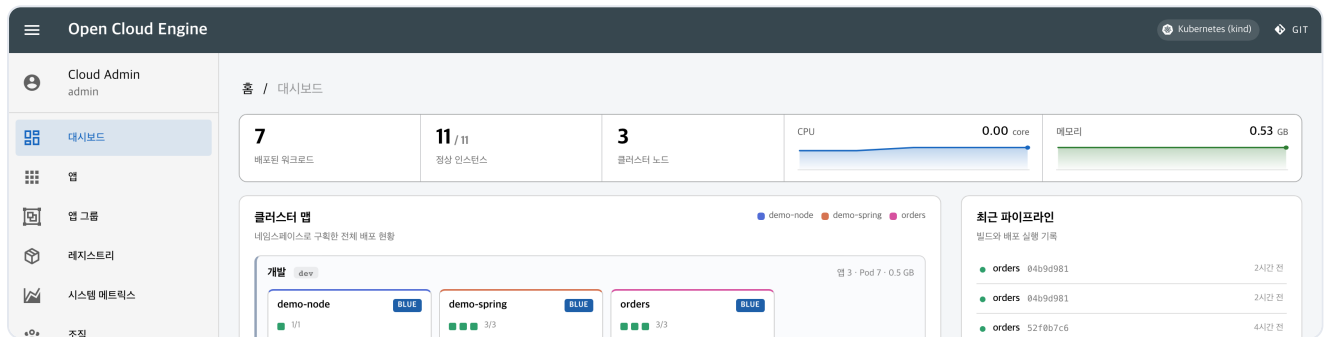


그림 18. 클러스터 맵. 운영 구획의 서비스가 신규 10퍼센트·이전 90퍼센트로 나뉘어 전달되고 있습니다. 데모 데이터 기준입니다.

노드 배치와 이동

아래쪽에는 클러스터 노드를 원통으로 그림니다. 색 구간은 각 서비스가 그 노드에서 쓰는 메모리이며, 모든 원통이 같은 눈금을 쓰므로 어느 노드가 더 찼는지 바로 비교됩니다. 색 구간을 누르면 그 서비스를 **다른 노드로 옮길** 수 있습니다.

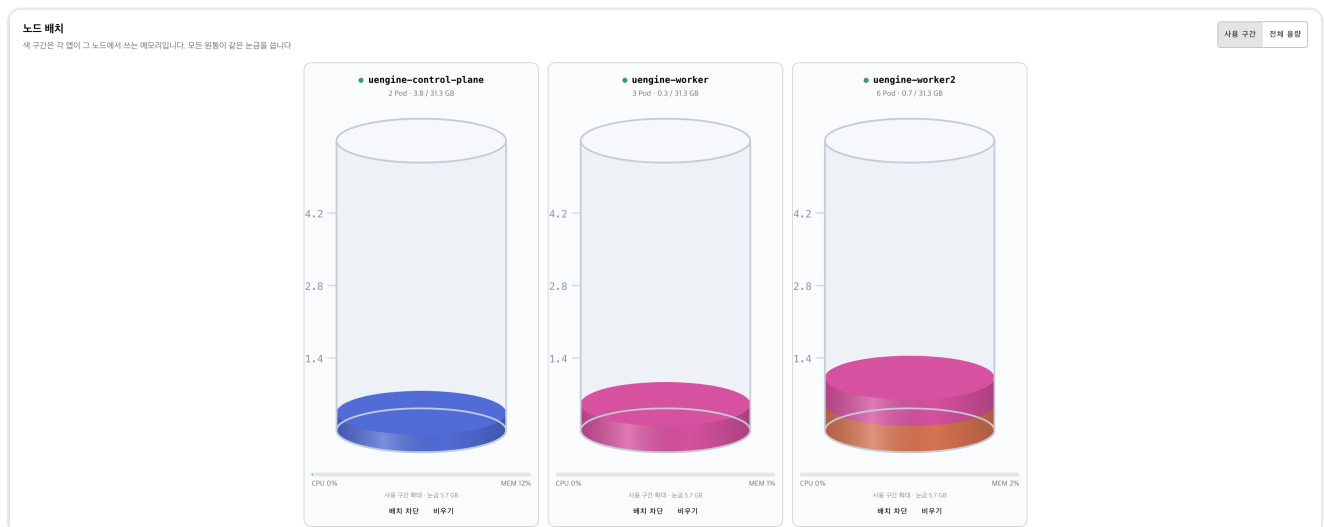


그림 19. 노드 배치. 세 노드에 서비스가 어떻게 나뉘어 있는지 보여 줍니다. 데모 데이터 기준입니다.

노드 단위 조작 — 점검 전 특정 노드에 새 배치를 막거나, 그 노드의 서비스를 다른 노드로 옮길 수 있습니다.

특징점 ⑤ — 여러 서비스를 묶어서 관리

마이크로서비스는 혼자 움직이지 않습니다. 함께 배포해야 하는 서비스를 **앱 그룹**으로 묶으면 승격과 스냅샷을 한 번에 수행할 수 있습니다.

앱 그룹

- **묶음 승격** — 그룹에 속한 서비스를 개발에서 스테이징으로, 스테이징에서 운영으로 함께 올립니다.
- **결과 확인** — 서비스별 성공·실패를 목록으로 보여 줍니다. 일부가 실패해도 나머지는 진행됩니다.
- **묶음 스냅샷** — 그룹 전체를 같은 이름의 스냅샷으로 저장합니다.
- **단계별 현황** — 그룹 안에서 몇 개 서비스가 각 단계에 배포되어 있는지 표시합니다.

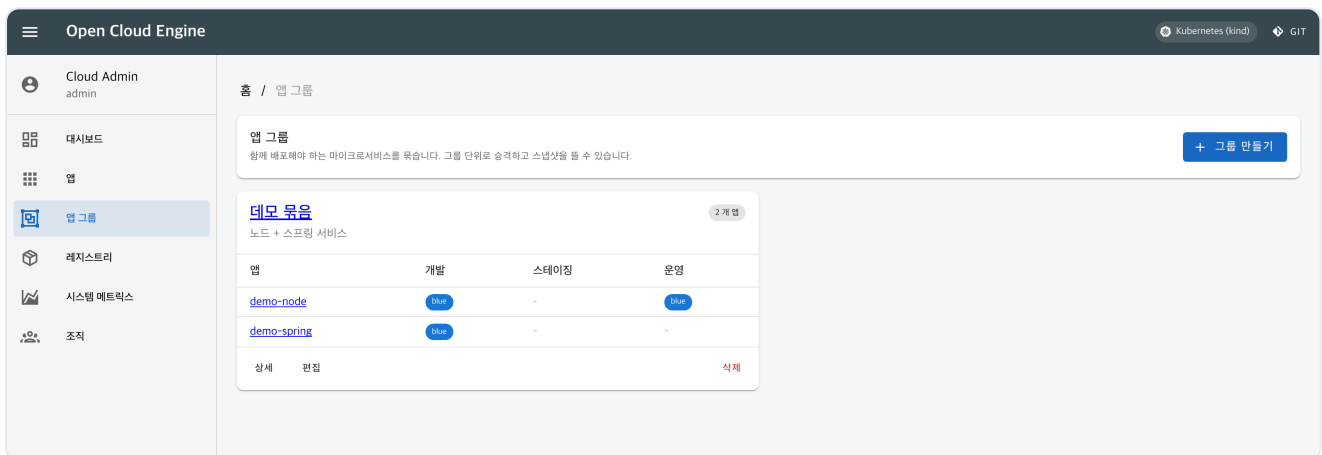


그림 20. 앱 그룹 화면. 데모 데이터 기준입니다.

사용자와 권한

포털 사용자는 조직 메뉴에서 관리합니다. 계정 추가와 비활성화, 비밀번호 재설정, 역할 지정, 그룹 소속 지정을 화면에서 처리합니다. 인증은 표준 프로토콜을 따르는 인증 서버가 담당하며, 사내 계정 체계와 연동할 수 있습니다.

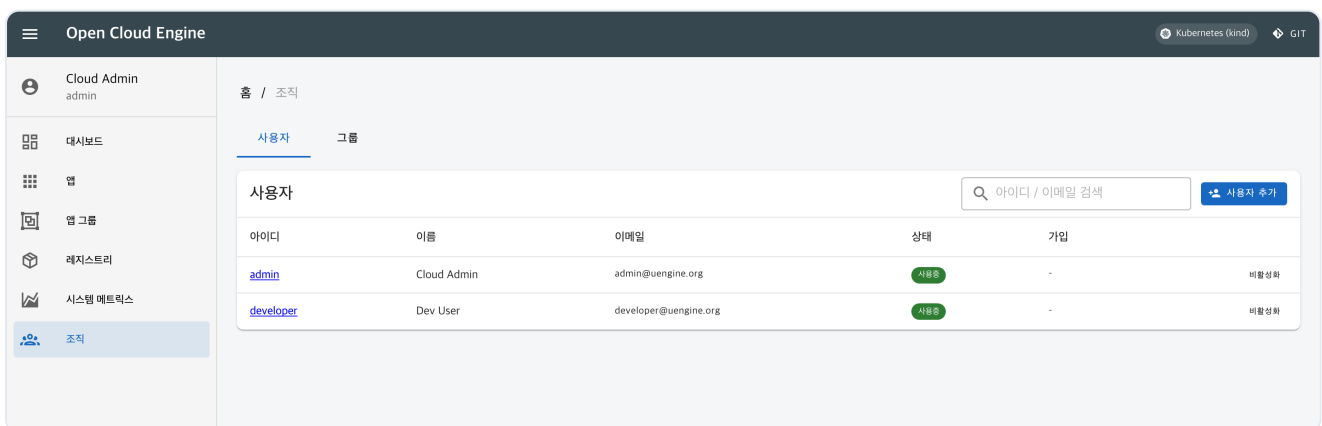


그림 21. 조직 화면. 데모 데이터 기준입니다.

특장점 ⑥ — 개방형 · 온프레미스 · 한국어

소스가 공개되어 있고, 외부 서비스에 의존하지 않습니다. 사내 클러스터와 폐쇄망에 그대로 설치할 수 있습니다.

개방형

MIT 라이선스 오픈소스

소스와 구성이 공개되어 있어 조직의 요구에 맞게 확장할 수 있습니다. 템플릿과 화면을 내부 표준에 맞춰 수정할 수 있습니다.

온프레미스

외부 연결 없이 동작

포털·저장소·레지스트리·인증 서버가 모두 클러스터 안에서 동작합니다. 외부 서비스 가입이나 사용량 과금이 없습니다.

한국어

화면과 문서가 한국어

메뉴·안내 문구·오류 메시지가 한국어로 제공됩니다. 국내 인력이 인수인계 없이 바로 운영할 수 있습니다.

표준 준수

쿠버네티스 표준 리소스

독자 규격을 쓰지 않습니다. 표준 리소스로 배포하므로 기존 클러스터 운영 도구와 함께 사용할 수 있습니다.

기존 클러스터 도구와의 관계

이미 사용 중인 도구	함께 사용하는 방식
클러스터 관리 콘솔	표준 리소스로 배포하므로 기존 콘솔에서도 그대로 조회됩니다.
모니터링 · 로그 수집	포털의 화면과 별개로 기존 수집 체계를 계속 사용할 수 있습니다.
사내 인증	표준 인증 프로토콜을 사용하므로 사내 계정 체계와 연동할 수 있습니다.
사내 저장소 · 레지스트리	설치 시 기존 저장소와 레지스트리를 사용하도록 지정할 수 있습니다.

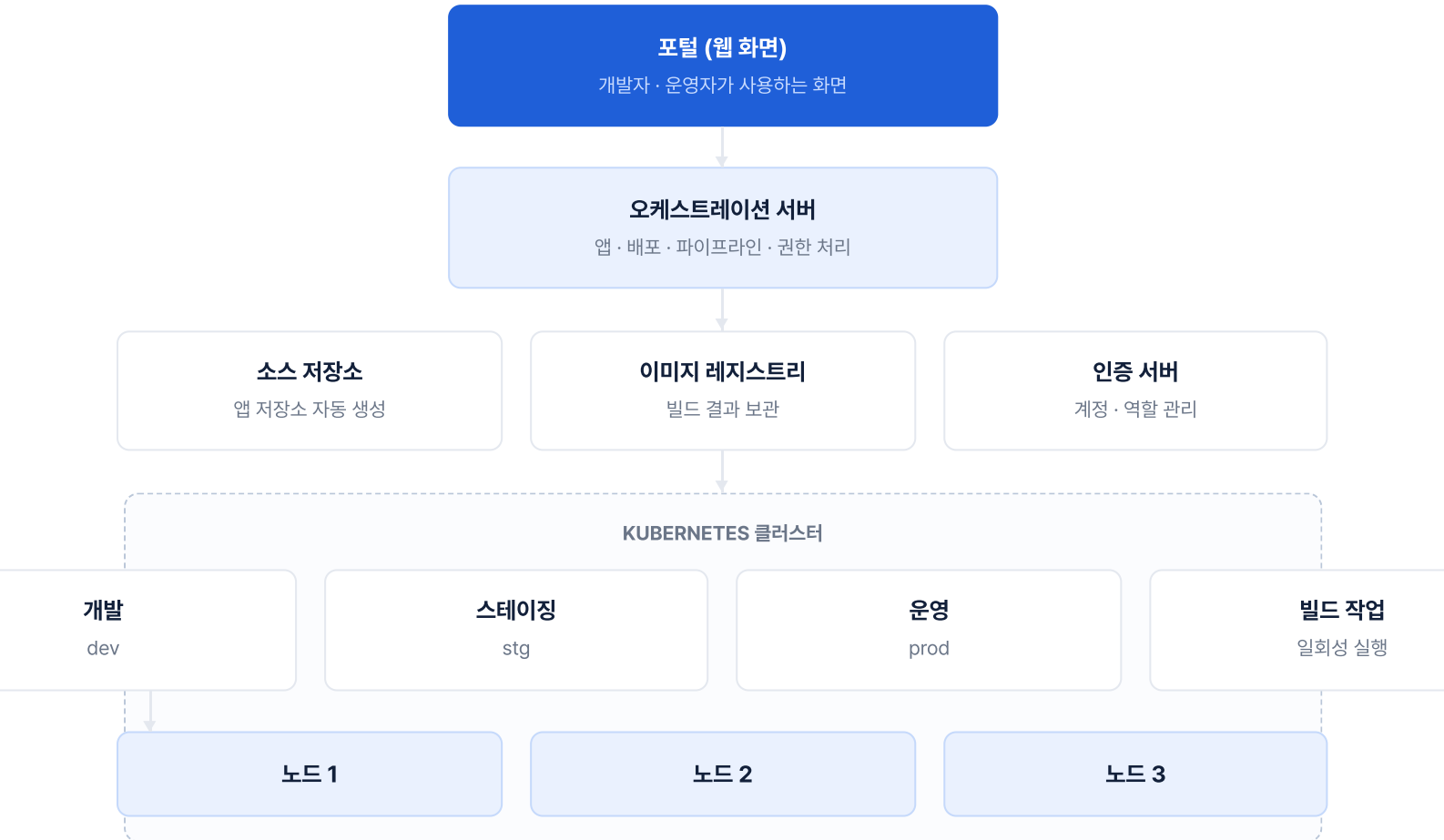
확장할 수 있는 지점

확장 대상	방법
앱 템플릿	조직의 표준 프로젝트 구조와 빌드 방식을 담은 템플릿을 추가합니다.
배포 기본값	단계별 기본 자원, 헬스체크 경로, 전환 방식의 기본값을 조정합니다.
포털 화면	사내 표기와 절차에 맞게 화면 구성과 문구를 수정합니다.

기술 종속을 만들지 않습니다. 소스가 공개되어 있어 조직이 직접 유지하고 확장할 수 있습니다.

시스템 구성과 실행 환경

포털과 오케스트레이션 서버, 그리고 클러스터 안에서 동작하는 부속 구성 요소로 이루어집니다. 모든 구성 요소가 하나의 쿠버네티스 클러스터 안에서 동작합니다.



단계별 분리와 노드 배치

개발·스테이징·운영은 각각 별도 영역으로 분리되어 배포되고, 단계마다 자원과 환경설정이 따로 유지됩니다. 배치할 노드는 스케줄러가 정하며, 필요하면 포털에서 지정할 수 있습니다.

실행 환경 요구사항

항목	요건
쿠버네티스	1.28 이상. 관리형 서비스와 자체 구축 클러스터 모두 지원합니다.
인그레스 컨트롤러	트래픽 비율 분배를 지원하는 컨트롤러가 필요합니다.
저장 공간 · 지표	저장소와 레지스트리 용량은 서비스 수에 따라 산정합니다. 사용량 표시를 위해 지표 수집 구성 요소가 필요합니다.

기술 스택

현행 버전으로 구성했습니다. 특정 상용 제품에 묶이지 않고, 각 계층을 조직의 표준 제품으로 교체할 수 있습니다.

계층	사용 기술	역할
포털 화면	Vue 3.5 · Vuetify 3.7 · Vite 6	개발자·운영자가 사용하는 웹 화면
오케스트레이션	Spring Boot 3.5 · Java 17	앱·배포·파이프라인·권한 처리
클러스터 연동	Fabric8 Kubernetes Client 7.9	표준 리소스 생성과 상태 조회
이미지 빌드	BuildKit	클러스터 안에서 컨테이너 이미지 생성
소스 저장소	Gitea	앱 저장소 자동 생성과 코드 반영 연동
인증	Keycloak (OAuth 2.0 · OIDC)	로그인, 역할·그룹 관리
트래픽 분배	Ingress NGINX	접속 주소 연결, 무중단 전환 비율 적용

설계 원칙

표준 리소스만 사용

독자 규격을 만들지 않습니다. 배포 결과는 기존 클러스터 도구에서도 그대로 확인됩니다.

앱 단위 추상화

화면과 기능이 모두 앱 단위로 구성됩니다. 개발자는 클러스터 개념을 다루지 않습니다.

교체 가능한 구성

저장소·레지스트리·인증 서버는 설정으로 지정합니다. 사내 제품으로 바꿀 수 있습니다.

버전과 유지

항목	내용
기준 버전	각 계층을 현행 정식 버전으로 구성했습니다. 장기 지원 버전을 지정해 고정할 수 있습니다.
클러스터 호환	표준 리소스만 사용하므로 클러스터 버전 갱신의 영향을 적게 받습니다.
구성 요소 교체	저장소·레지스트리·인증 서버는 설정 항목이므로 사내 제품으로 바꿀 수 있습니다.

제품 규모

오케스트레이션 서버는 12개 영역에 걸쳐 **76개의 기능 인터페이스**를 제공하고, 포털 화면은 **27개 경로**로 구성됩니다.

사용 절차 ① — 서비스 생성부터 개발 배포까지

아래 절차는 실제 클러스터에서 수행한 순서입니다. Node.js 템플릿 기준으로 전체 과정이 **5분 이내**에 끝납니다. 빌드에 시간이 더 필요한 템플릿은 그만큼 길어집니다.

- 1 카탈로그에서 템플릿 선택 — 만들려는 서비스 유형을 고릅니다.
- 2 이름과 자원 지정 — 앱 이름, 인스턴스 수, CPU·메모리를 정합니다. 접속 주소가 미리 표시됩니다.
- 3 생성 — 저장소가 만들어지고 첫 파이프라인이 시작됩니다.
- 4 빌드 확인 — 진행 상황과 빌드 로그를 화면에서 확인합니다.
- 5 개발 배포 완료 — 빌드가 끝나면 개발 단계에 배포되고 접속 주소가 열립니다.

The screenshot shows the 'Open Cloud Engine' console interface. On the left is a sidebar with navigation icons for '대시보드', '앱', '앱 그룹', '레지스트리', '시스템 메트릭스', and '조직'. The main area displays the 'orders' service configuration. At the top, it shows 'orders' with a 'nodejs' tag and a '소스 리포지토리' link. Below this, there are three tabs: '개발' (BLUE), '스테이징' (BLUE), and '운영' (GREEN). The '개발' tab is selected. The main content area shows the 'orders' service details, including a description: 'uEngine Cloud PaaS 카탈로그(Node.js 마이크로서비스)로 생성된 앱입니다.' and a list of deployment steps. On the right, there are sections for '소스 내려받기' (Source Download) with a 'git clone' command and a '명령 복사' button, and '접속 주소' (Access URL) with links for '개발', '스테이징', and '운영'.

그림 22. 시작하기 화면. 저장소 내려받기 명령과 다음에 할 일을 안내합니다. 데모 데이터 기준입니다.

이후 개발 흐름

개발자는 저장소를 내려받아 코드를 수정하고 다시 올립니다. 설정한 브랜치에 코드가 올라오면 파이프라인이 자동으로 시작되고, 개발 단계에 반영됩니다. 별도의 배포 요청 절차가 없습니다.

빌드가 실패하면 파이프라인 화면에 실패 단계와 빌드 로그가 그대로 표시됩니다. 원인을 확인하고 코드를 고쳐 다시 올리면 됩니다.

빌드 설정에서 **자동 실행 여부와 대상 브랜치**를 조정할 수 있습니다. 검토 후 배포하는 방식이 필요하다면 자동 배포 단계를 비워 두고, 포털에서 직접 실행합니다.

사용 절차 ② — 승격과 무중단 전환

개발 단계에서 확인한 이미지를 다시 빌드하지 않고 상위 단계로 올립니다. 같은 이미지가 올라가므로 단계 간 차이가 생기지 않습니다.



운영 전환 절차

- 1 **새 버전 배포** — 사용하지 않는 슬롯에 배포됩니다. 기존 슬롯은 계속 서비스합니다.
- 2 **비율 조정** — 신규 버전으로 보낼 트래픽 비율을 정합니다.
- 3 **확인** — 로그와 지표로 신규 버전의 동작을 확인합니다.
- 4 **확정 또는 복귀** — 이상이 없으면 비율을 올려 확정하고, 문제가 있으면 이전 버전으로 되돌립니다.

그림 23. 배포 전략 화면. 현재 슬롯 구성과 확인 방법을 함께 제공합니다. 데모 데이터 기준입니다.

전환 확정 전까지 이전 버전이 유지됩니다. 확정하면 이전 슬롯을 정리하고 모든 요청을 신규 버전으로 보냅니다. 확정 이후에도 레지스트리에 남은 이전 이미지를 다시 배포할 수 있습니다.

보안 · 접근 통제

인증은 표준 프로토콜을 따르는 인증 서버가 담당합니다. 클러스터 조작은 오케스트레이션 서버에 부여된 권한 범위 안에서만 수행됩니다.

인증과 권한

항목	내용
로그인	OAuth 2.0 · OIDC 를 사용합니다. 사내 계정 체계와 연동할 수 있습니다.
역할	관리자와 일반 사용자로 구분하며, 조직 메뉴에서 지정합니다.
그룹	부서·팀 단위로 사용자를 묶어 관리합니다.
토큰 검증	포털의 모든 요청은 서버에서 토큰을 검증한 뒤 처리됩니다.

기록과 추적

- **조작 이력** — 생성·배포·승격·비율 조정·복원·삭제를 앱별로 기록합니다.
- **컨테이너 접속 기록** — 접속은 쿠버네티스 실행 기능을 사용하므로 클러스터 감사 로그에 남습니다.
- **빌드 기록** — 어떤 커밋으로 어떤 이미지가 만들어졌는지 기록되고, 이미지 목록에서 확인됩니다.

망 분리 환경

포털·저장소·레지스트리·인증 서버가 모두 클러스터 안에서 동작하므로, 외부 통신 없이 운영할 수 있습니다. 설치 시 필요한 컨테이너 이미지를 사전에 반입하는 방식으로 폐쇄망에 구축합니다.

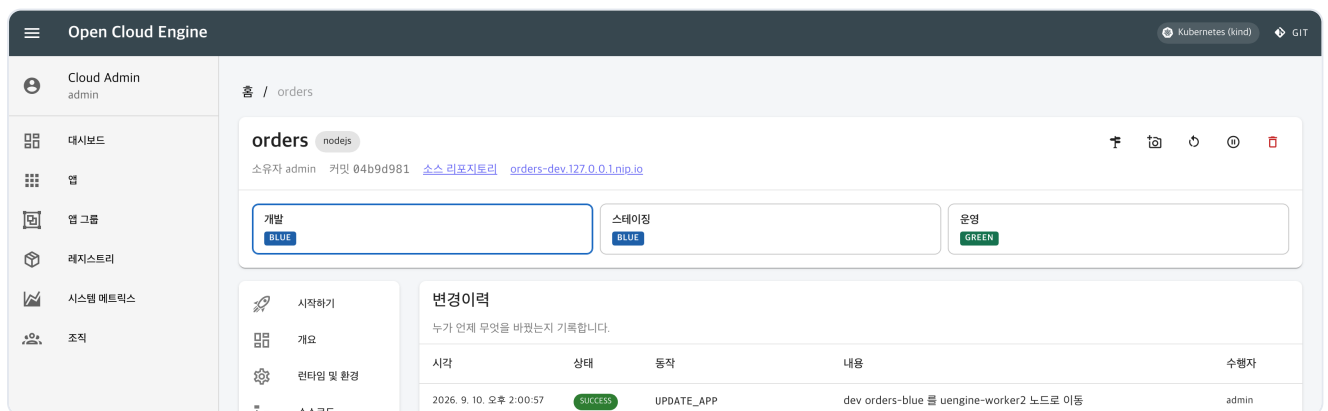


그림 24. 변경이력 화면. 시각·상태·동작·수행자가 함께 기록됩니다. 데모 데이터 기준입니다.

도입 전 확인 사항 — 사내 인증 체계 연동 방식, 이미지 반입 절차, 클러스터 권한 부여 범위는 구축 착수 시점에 함께 정합니다.

설치·납품 구성

설치는 스크립트 한 번으로 시작합니다. 클러스터 준비부터 포털 접속까지 필요한 구성 요소가 함께 설치됩니다.

납품 범위

구분	내용
제품 소스	포털·오케스트레이션 서버 전체 소스와 빌드 구성
설치 구성	클러스터 구성 파일, 부속 구성 요소 설치 정의, 설치 스크립트
앱 템플릿	Node.js · Spring Boot 3 · Vue 3 템플릿. 조직 표준에 맞춰 추가 가능
문서	설치 안내, 개념 대응표, 화면별 설명, 데모 시나리오
데모 영상	서비스 생성부터 무중단 전환·노드 배치까지 전 과정 (약 6분, 자막 포함)

설치 절차

- 1 클러스터 준비 — 기존 클러스터를 사용하거나, 제공되는 구성 파일로 새로 만듭니다.
- 2 부속 구성 요소 설치 — 저장소·인증 서버·레지스트리·트래픽 분배 구성 요소를 설치합니다.
- 3 서버 기동 — 오케스트레이션 서버를 기동하고 클러스터 접근 권한을 연결합니다.
- 4 포털 접속 — 관리자 계정으로 로그인해 첫 앱을 생성합니다.

기존 자산 연계

사내에 이미 저장소나 이미지 레지스트리를 운영 중이라면, 설치 시 해당 주소를 지정해 그대로 사용할 수 있습니다. 부속 구성 요소를 중복해서 설치할 필요가 없습니다.

The screenshot shows the 'System Metrics' page in the Open Cloud Engine interface. It displays a table of nodes with columns for Name, Role, Status, CPU, Memory, and Kubelet. The nodes listed are 'uengine-control-plane' (control-plane, Ready, 0.07 core / 16.00 core, 3.5 GB / 31 GB) and two 'uengine-worker' nodes (worker, Ready, 0.01 core / 16.00 core, 251 MB / 31 GB and 694 MB / 31 GB). All nodes are running v1.34.0 of Kubelet. Below the table, there is a section for 'Platform managed Pod' with a search bar.

이름	역할	상태	CPU	메모리	Kubelet
uengine-control-plane	control-plane	Ready	0.07 core / 16.00 core	3.5 GB / 31 GB	v1.34.0
uengine-worker	worker	Ready	0.01 core / 16.00 core	251 MB / 31 GB	v1.34.0
uengine-worker2	worker	Ready	0.02 core / 16.00 core	694 MB / 31 GB	v1.34.0

그림 25. 시스템 메트릭스 화면. 노드 상태와 제품이 배포한 인스턴스를 확인합니다. 데모 데이터 기준입니다.

검증 체계

아래 항목은 실제 쿠버네티스 클러스터에서 수행해 확인한 결과입니다. 화면 동작만이 아니라 클러스터에 반영된 결과까지 함께 확인했습니다.

검증 항목	확인 방법과 결과
서비스 생성	카탈로그에서 앱 생성 → 저장소 자동 생성, 템플릿 반영, 코드 반영 연동 등록 확인
이미지 빌드	클러스터 안에서 빌드 작업 실행 → 레지스트리에 이미지 등록 확인
단계별 배포	개발 자동 배포 → 스테이징 승격 → 운영 승격. 세 단계 모두 동일 이미지로 배포 확인
환경설정 반영	포털에서 저장한 값이 실행 중인 컨테이너의 환경변수와 설정 파일에 반영된 것을 확인
자원 조정	인스턴스 수 변경 → 실제 인스턴스 수 변화 확인
중지 · 재개	중지 시 인스턴스 0, 재개 시 이전 수량으로 복귀 확인
무중단 전환	신규 비율 10퍼센트 설정 후 20회 요청 → 이전 16회 · 신규 4회 전달 확인
세션 고정	설정 시 트래픽 분배 구성에 세션 고정 항목이 적용된 것을 확인
실시간 로그	포털에서 로그가 이어서 수신되는 것을 확인
컨테이너 접속	브라우저에서 컨테이너 셸에 접속해 명령을 실행하고 결과를 확인
스냅샷	저장 후 목록에 단계별 이미지·커밋·인스턴스 수가 기록된 것을 확인
묶음 승격	두 개 서비스를 묶어 단계별 배포 현황이 집계되는 것을 확인
노드 이동	포털에서 다른 노드를 지정 → 해당 노드에서 인스턴스가 다시 뜨고 원통 표시가 옮겨간 것을 확인 (3노드 클러스터)

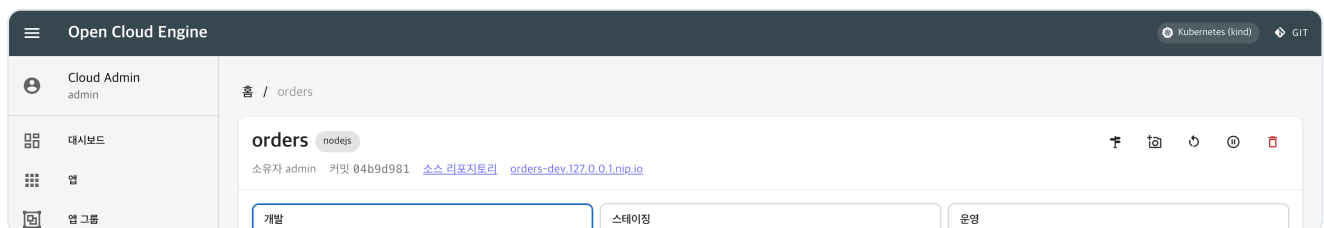


그림 26. 모니터링 화면. 앱 단위 사용량 추이와 클러스터 이벤트를 함께 확인합니다. 데모 데이터 기준입니다.

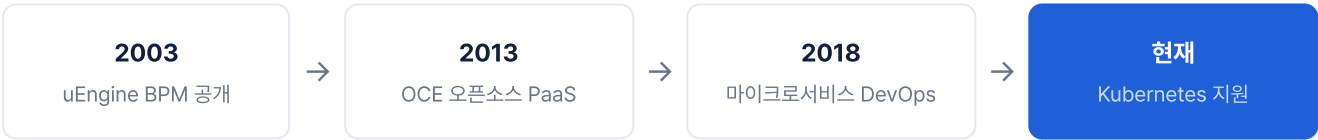
데모 영상 — 위 과정을 이어서 수행한 약 6분 분량의 화면 녹화를 함께 제공합니다.

레퍼런스

주식회사 유엔진이 수행한 **PaaS 구축 · 마이크로서비스 전환 · 업무 프로세스 관리** 분야의 주요 프로젝트입니다.

고객	주요 수행 내용
KIAT · 공공 R&D 기관	OCE 기반 클라우드 풀 · 공용 앱스토어 구축 (2013-2014)
한화그룹 · 한화시스템	업무 프로세스 관리 도입, 마이크로서비스 전환 컨설팅·교육 (2003-2025)
정부24	클라우드 네이티브 마이크로서비스 전환 컨설팅 (2025)
포스코DX · 포스코	통합물류 컨설팅 및 AI 기반 설계·현대화 (2022-2025)
SK C&C · SK텔레콤	마이크로서비스 교육 및 운영 업무 자동화 (2020-2026)
KB국민은행	KB Wallet 마이크로서비스 구현 컨설팅 (2023-2024)
LG CNS · LG에너지솔루션	마이크로서비스 개발방법론 수립, 제조·ERP 프로세스 관리 구축 (2019-2024)

제품 계보



현재 제품은 2018년 공개판에서 오케스트레이션 계층을 Kubernetes 로 교체하고, 포털과 서버를 현행 기술로 다시 구성한 버전입니다.

함께 제공하는 자산

구분	내용
마이크로서비스 전환 교재	서비스 분리, API 게이트웨이 구성, 서비스 간 통신, 도메인 구현 등 19편의 실습 문서
전환 컨설팅	클라우드 네이티브 전환 진단과 아키텍처 설계 지원
교육 과정	마이크로서비스 설계·구현 교육 과정 운영

설계 도구와의 연결 — 요구사항 분석부터 마이크로서비스 설계·구현을 담당하는 Robo Architect 의 산출물을 uEngine Cloud 에 배포해, 설계부터 운영까지 하나의 흐름으로 이룰 수 있습니다.

도입 절차 및 일정

표준 도입 일정은 **8주**입니다. 기존 클러스터와 사내 저장소를 활용하는 경우 기간을 단축할 수 있습니다.



단계별 산출물

단계	산출물
현황 진단	도입 환경 점검 결과, 연동 방식 결정서
설치 및 연동	설치 완료 환경, 구성 명세, 계정·권한 체계
템플릿 구성	조직 표준 템플릿, 빌드 설정 기준
시범 서비스 전환	전환된 서비스, 배포 절차 정의서
교육 및 인수인계	교육 자료, 운영 안내서

도입 이후 지원

운영 지원

설치 환경 점검, 장애 분석, 버전 갱신 지원을 계약 범위에 따라 제공합니다.

템플릿 추가

새로운 언어·프레임워크 표준이 생기면 템플릿을 추가로 작성해 드립니다.

전환 컨설팅

기존 서비스를 마이크로서비스로 나누는 설계 단계부터 함께 진행합니다.

사전 확인 절차 — 도입 검토 단계에서 시연 일정을 안내해 드립니다. 사내 클러스터에 시범 설치해 실제 서비스로 확인하는 방식도 가능합니다.



uEngine Cloud

SELF-SERVICE PaaS PORTAL

도입 문의

먼저 시연으로 확인해 보십시오.

서비스 생성부터 무중단 전환·노드 배치까지 전 과정을 **약 6분** 분량의 영상으로
제공합니다. 사내 클러스터에 시범 설치해 실제 서비스로 확인하는 일정도 안내해
드립니다.

02-567-8301

대표 전화

help@uengine.org

도입 문의

uengine-oss

GitHub 제품 소스

주식회사 유엔진

본사 서울 서초구 사임당로 31 공현빌딩 3층 301호 · 지사 경상남도 창원시 성산구 상남로 37

© 2026 uEngine. 본 문서 Rev. 2026.09 · 기재된 수치와 화면은 작성 시점 기준입니다.

uEngine