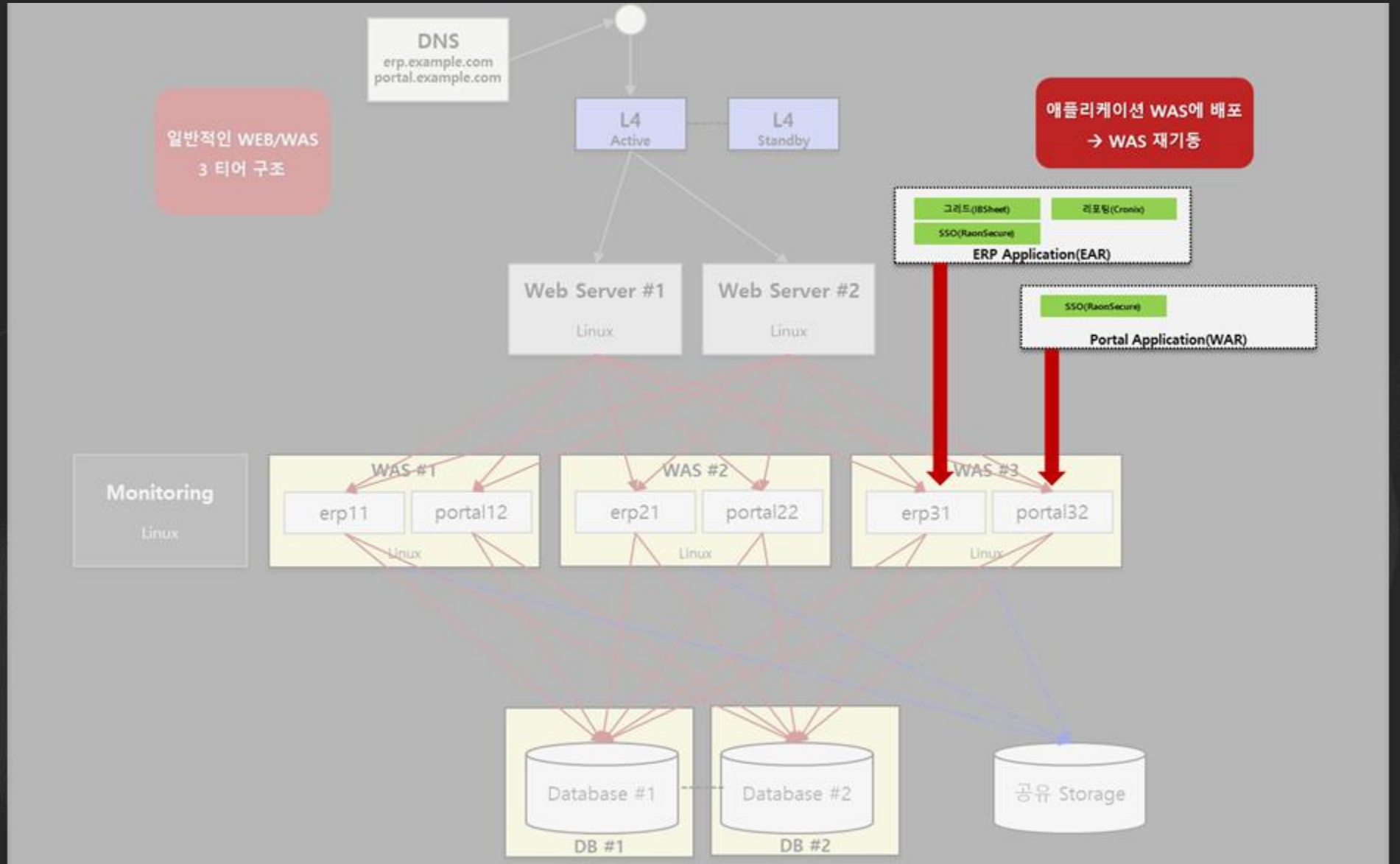
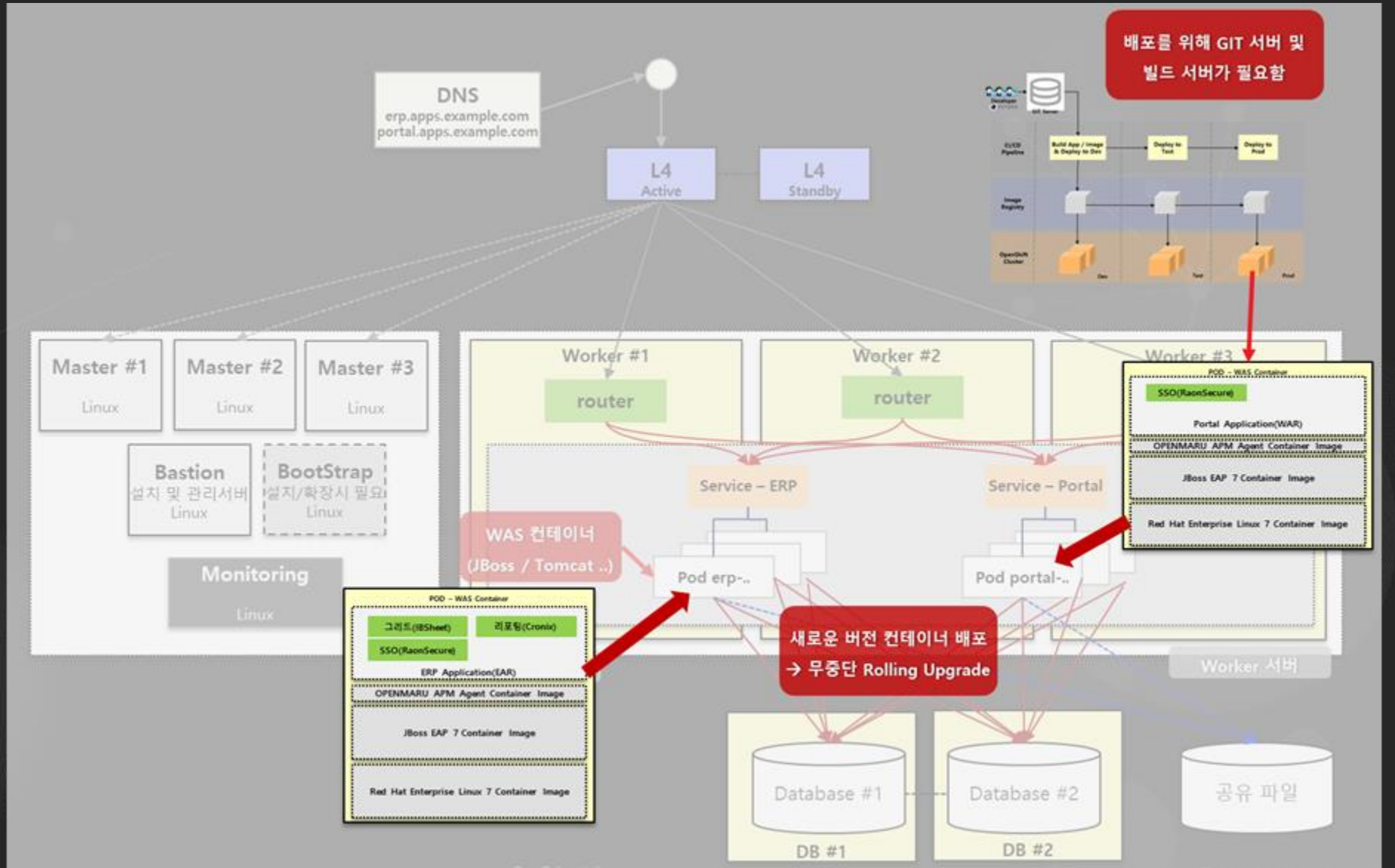


가상화 환경에서 배포 방법과 컨테이너 환경에서 CICD 차이점

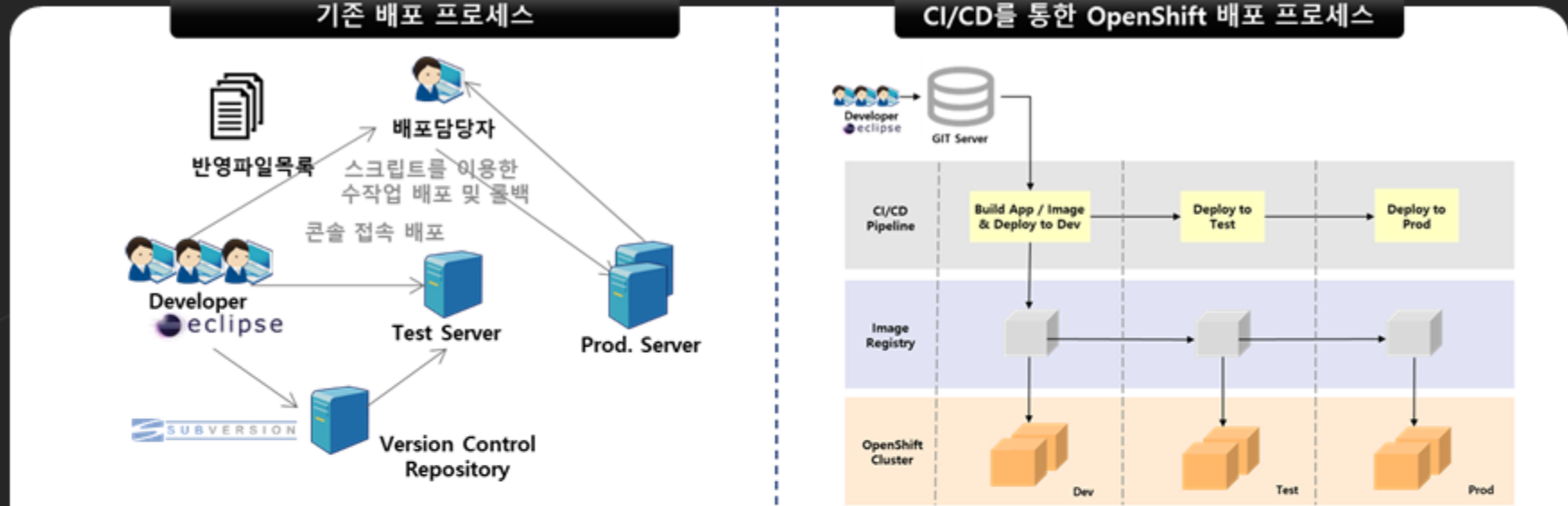
일반적인 WEB/WAS 구성시 애플리케이션 배포



OpenShift 구성시 서비스 배포방법



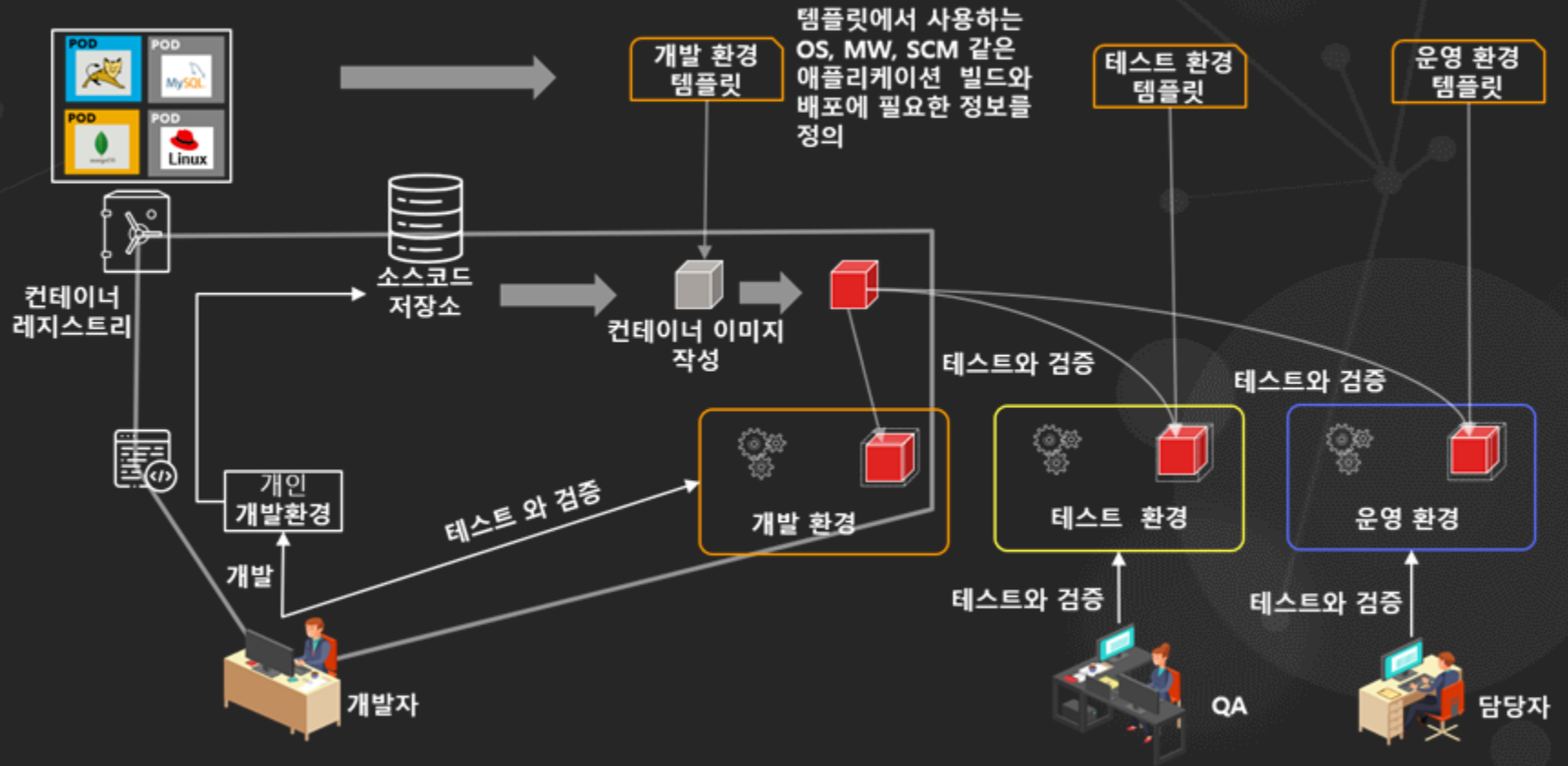
컨테이너 환경에서 CI/CD 배포 프로세스 개선



항목	AS-IS	TO-BE
자동화 여부	• 전담 인력에 의한 수작업	• CI도구(GIT)를 이용한 자동 빌드 및 자동 배포
패키징	• 배포 담당자가 수작업 취합하여 정리	• 형상관리 기준으로 자동 취합
배포 시간	• 수 십분 소요	• 수 분 소요
무중단 배포	• 배포 담당자 수작업에 의한 무중단 배포	• 자동화된 무중단 배포
롤백	• 일부 파일 및 반 자동화 된 롤백	• 기존 운영 버전으로 롤백(배포될 때 마다 버전을 기록함)
롤백 시간	• 수 십분 소요(배포 시간과 동일)	• 패키징 시간이 없으므로 약 1분이내
배포 인원	• 배포 담당자, 시스템 관리자	• 배포 담당자가 웹기반 UI에서 버튼 클릭 한번으로 가능
휴먼 장애	• 사람이 하는 일이라 파일 누락 등 실수 가능	• 없음
소스 및 설정 파일	• 서버와 형상 관리와 설정파일 다중 관리	• 형상관리로만 관리
배포검증	• 자동화 불가	• 테스트 케이스, 정적분석(시큐어 코딩), 코드 커버리지 등 다양한 기술 적용 가능

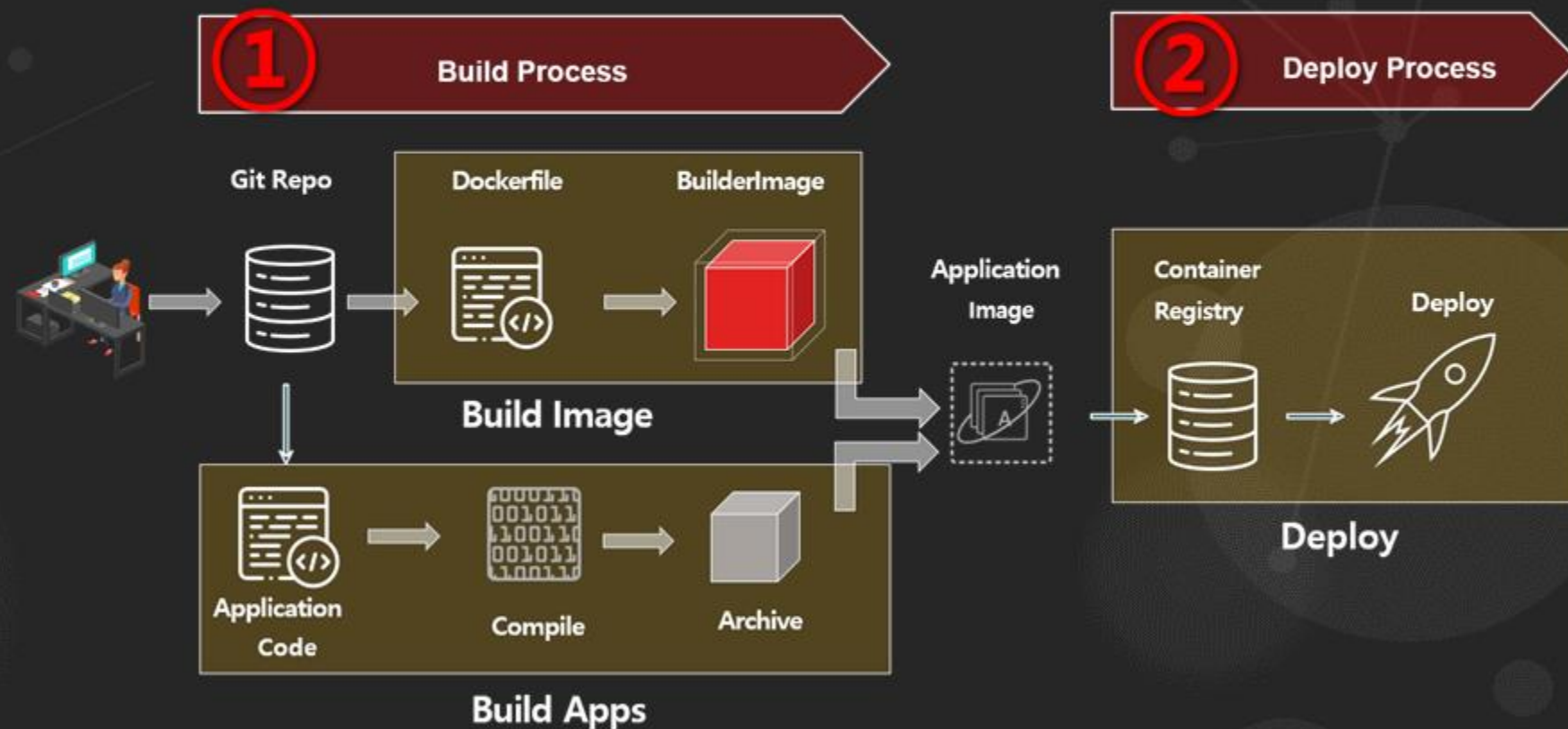
컨테이너 기반의 DevOps 워크 플로우

- OpenShift 를 통한 표준화와 자동화



Build & Deploy Application

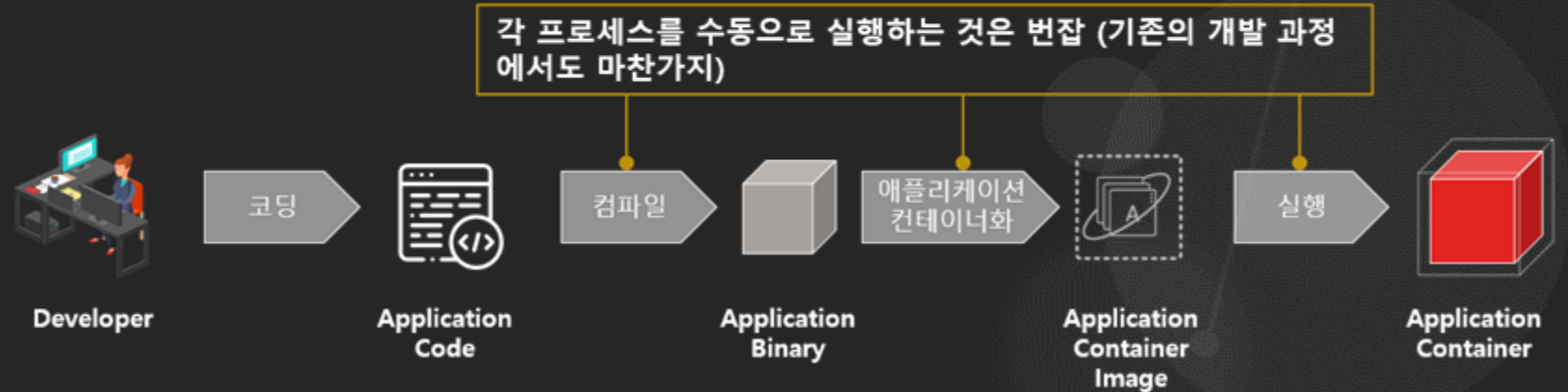
- Build Process 소스 코드 빌드 및 기반 이미지 빌드로 구성
- 애플리케이션 실행은 " Build Process "와" Deploy Process "로 구성



컨테이너 기반의 애플리케이션 개발 프로세스 예

1

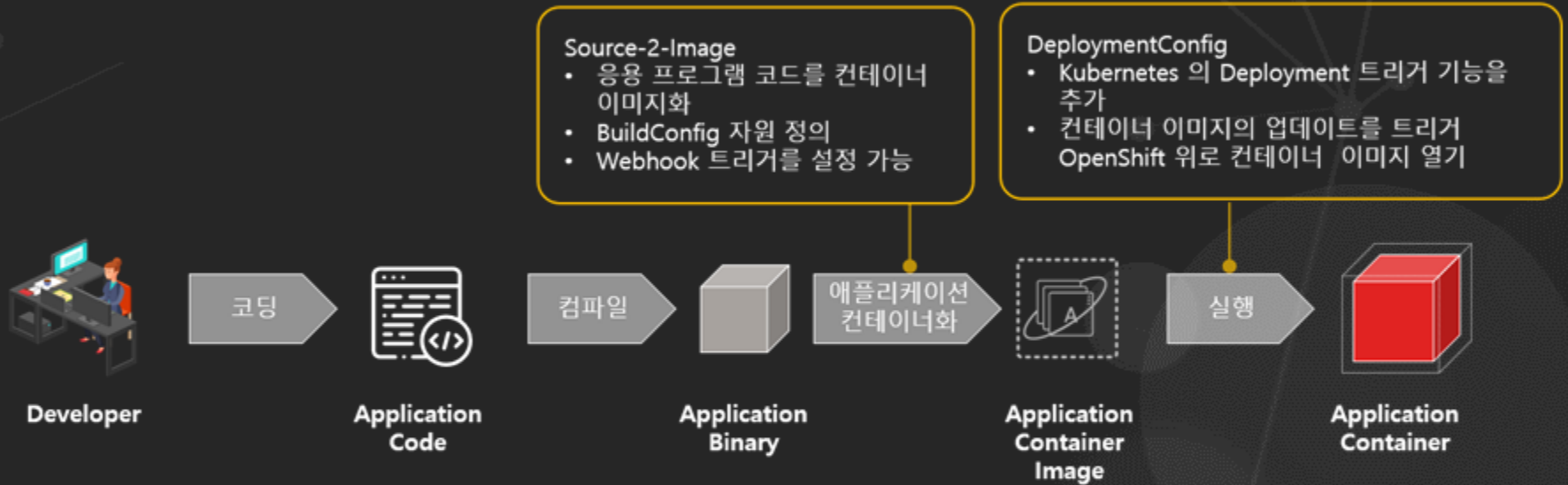
- 컨테이너 기반에서 애플리케이션을 개발하는 경우, 프로그래밍 테스트 단계의 일반적인 절차
- 컨테이너 이미지화 하여 개발 생산성 향상
 - 손쉬운 애플리케이션 환경 구성
 - 개발/스테이징/운영 환경의 차이 제거



기존의 물리서버나 가상서버와는 달리 컨테이너는 이미지를 만들고 배포하는 과정이 있음.

컨테이너 환경에서 개발 및 빌드 프로세스

2

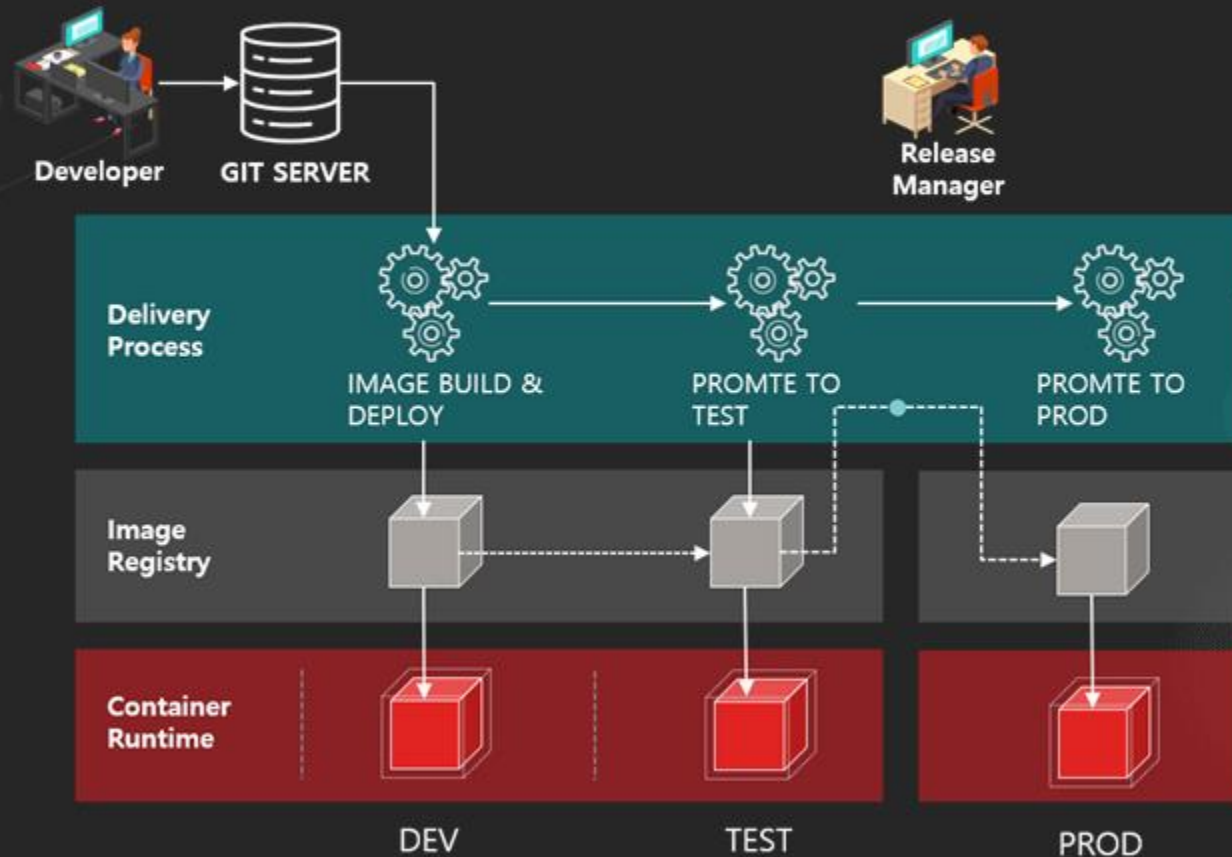


oc new-app 명령 (클라이언트 명령) / Catalog (Web 인터페이스)

- BuildConfig / DeploymentConfig 은 yaml 파일에서 정의
- oc new-app catalog 를 사용하여 여러 매개 변수를 입력하는 것으로, yaml 파일을 자동 생성

컨테이너 기반 응용 프로그램 릴리스 프로세스 예 Continuous Delivery

- 애플리케이션 코드 수정 후 컨테이너 빌드, 테스트 및 배포까지의 자동 실행 및 릴리스하는 방법

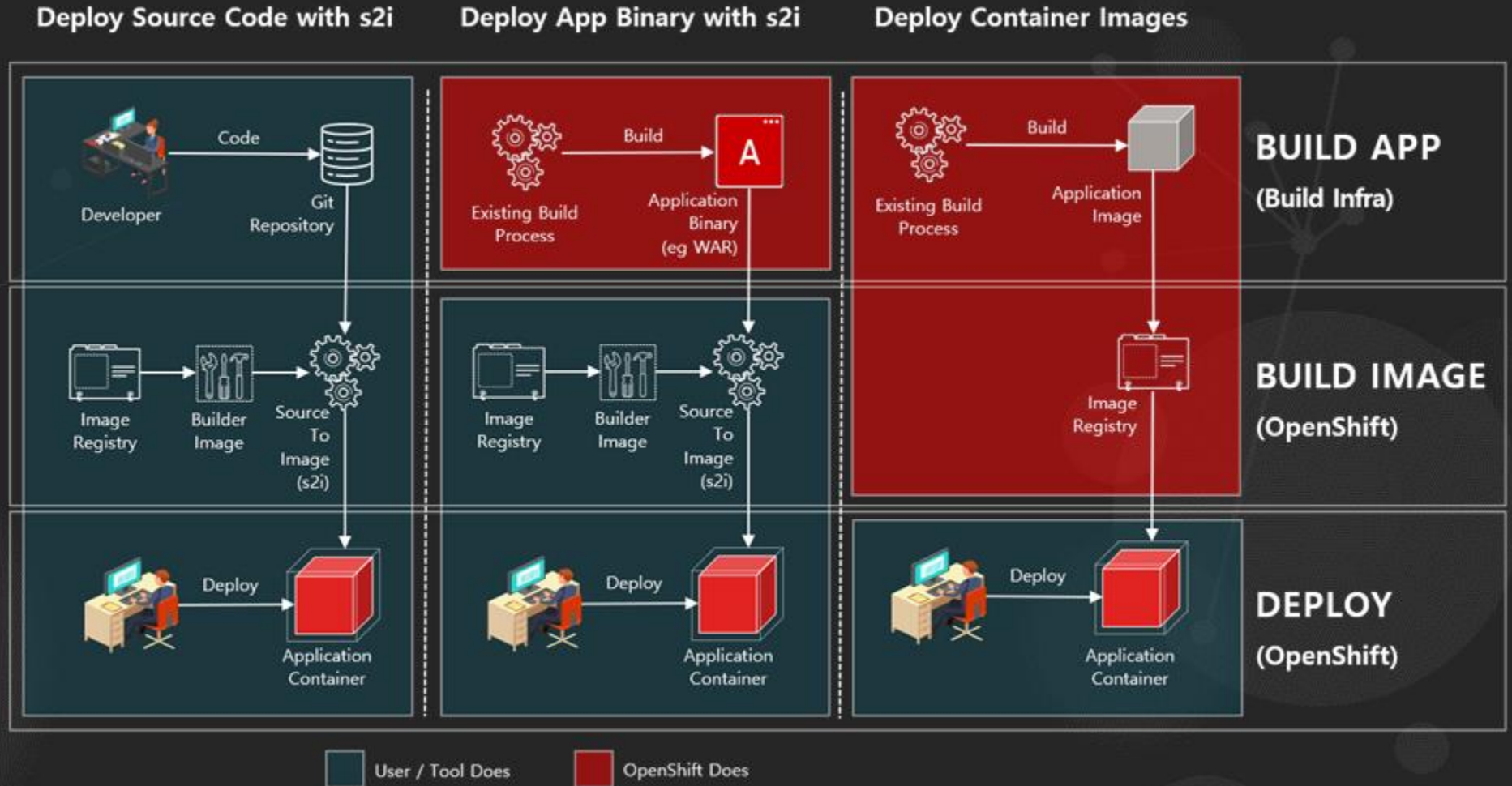


• 컨테이너와 이미지 배포

개발환경과 운영환경 통합

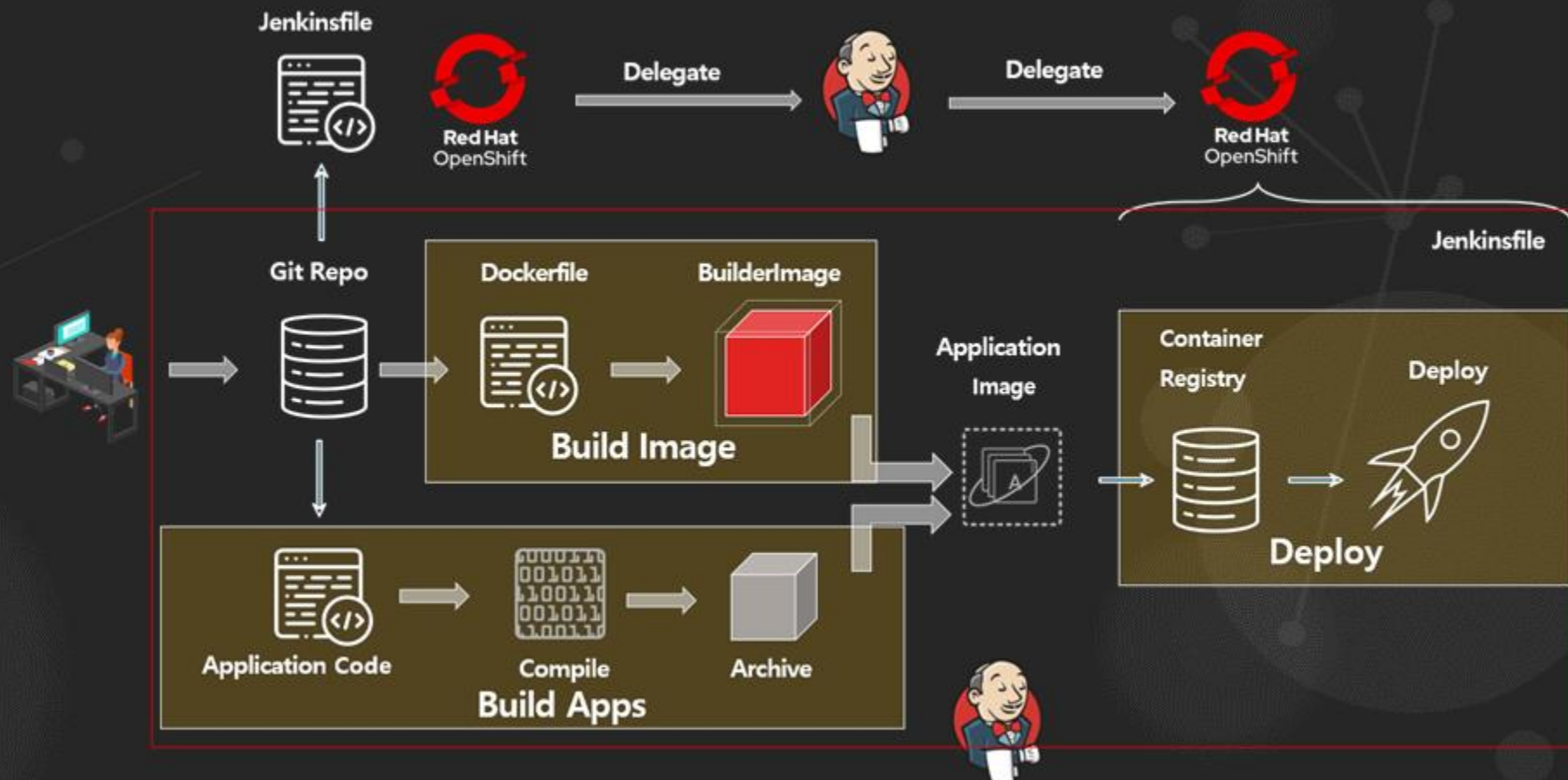


Deploy in OpenShift



OpenShift Pipeline (spec.strategy.type = JenkinsPipeline)

- Build Config 중 Strategy 로 JenkinsPipeline 을 활용



<https://blog.openshift.com/openshift-doudbees-jenkins-enterprise-devops/>

OpenShift Pipeline - Jenkins Pipeline



- OpenShift Pipelines 를 사용하면 Jenkins 파이프 라인을 통해 출시 프로세스를 정의하고 시작하고 모니터링 할 수 있습니다.

The screenshot displays the OpenShift Pipelines console interface. On the left is a navigation sidebar with categories: Workloads, Networking, Storage, Builds (expanded), Image Streams, Monitoring, Compute, and Administration. The 'Builds' section is active, showing a list of builds. The main area shows the 'Build Details' for 'tasks-pipeline-1'. It includes tabs for Overview, YAML, Environment, Logs, and Events. The 'Build Overview' section features a horizontal pipeline flow with six stages: Build App (6 minutes ago), Test (5 minutes ago), Code Analysis (5 minutes ago), Archive App (5 minutes ago), Build Image (5 minutes ago), and Deploy DEV (2 minutes ago). Below the flow, there is a 'Build 1' entry (8 minutes ago) with a 'View Logs' link and a 'Promote to ST...' button with an 'Input Required' warning (2 minutes ago). At the bottom, a table provides details for the pipeline:

NAME	STATUS
tasks-pipeline-1	Running
NAMESPACE	TYPE
cid-smx	JenkinsPipeline



openmaru

제품 / 서비스에 관한 문의

• 콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

• 전자 메일 : sales@openmaru.com