

**기능 안전 요건 준수를 위한  
SW의 지속적 검증/통합**



**Synetics**

# CONTENTS

## 01. 요즘의 SW 개발 톨체인, 지속적 통합

SW ALM 환경의 변화: 지속적 통합은 배민보다 우리가 더 필요해

## 02. 빠르게 지속적 통합 구축해보기 (실습)

백 번 세미나 듣는 것보다 한 번 구축해보는 것이 낫더라구요

## 03. SW 빌드/검증 환경 12년 유지하기

ASPICE Sup.8: SW 빌드/검증 환경 베이스라인은 어떻게 유지해야 하는가?



# Synetics \_

대한민국  
오픈소스 기반 SW 빌드/검증 자동화  
최고 전문가 기업

#정적분석 #DevOps(CI) #임베디드SW #기능안전

# 한 동 준

- QA, 프로세스 개선, 오픈소스 기반 개발 환경, 기능 안전 SW, 성능 테스트
- CMMI 심사원 / (전) SPICE/A-SPICE 초급 심사원
- (현) 주식회사 시네틱스 대표
  - 삼성전자 / 삼성 디스플레이 / 현대자동차 / 현대모비스 / 미래에셋생명 SW 품질/테스팅 강사
  - 오픈소스 기반 개발 환경 구축 컨설팅 (삼성 디스플레이 / 항우연 / KAI / 앤츠 / 클립 소프트 등)
- (전) 상명대학교 SW 안전성 보증 연구센터 산학협력중점교수
  - 한국항공우주연구원, 항공우주산업 등 항공/철도/의료/자동차 개발환경 컨설팅
  - SKT CMMI 심사 / LGE ISO 26262 평가 모델 개발
- (전) LS산전 - 임베디드SW 개발 분야
  - 지속적 통합/테스트 및 빌드 가상화 체계 구축
  - CMMI Level 3 수준 전사 프로세스 구축
- (전) KTH - 웹/앱 개발 분야
  - 지속적 통합/테스트 자동화 구축
  - 웹 서비스 성능 테스트
- (전) KTDS - IT 분야
  - 정적 분석 도구 확산
  - 웹 서비스 성능 테스트
  - CMMI 기반 개발 프로세스 개선

01

# CONTENTS

요즘의 SW 개발 툴체인,  
지속적 통합

## DevOps = Development + Operations

- 단순하게는 개발과 운영의 합성어
- 개발(기획, 설계, 구현, 테스트) 과 운영(배포, 인프라, 모니터링) 의 **밀접한 연계**를 의미
- 애플리케이션과 서비스를 더 빠르게 제공하기 위한 조직의 **문화, 방법, 툴체인**

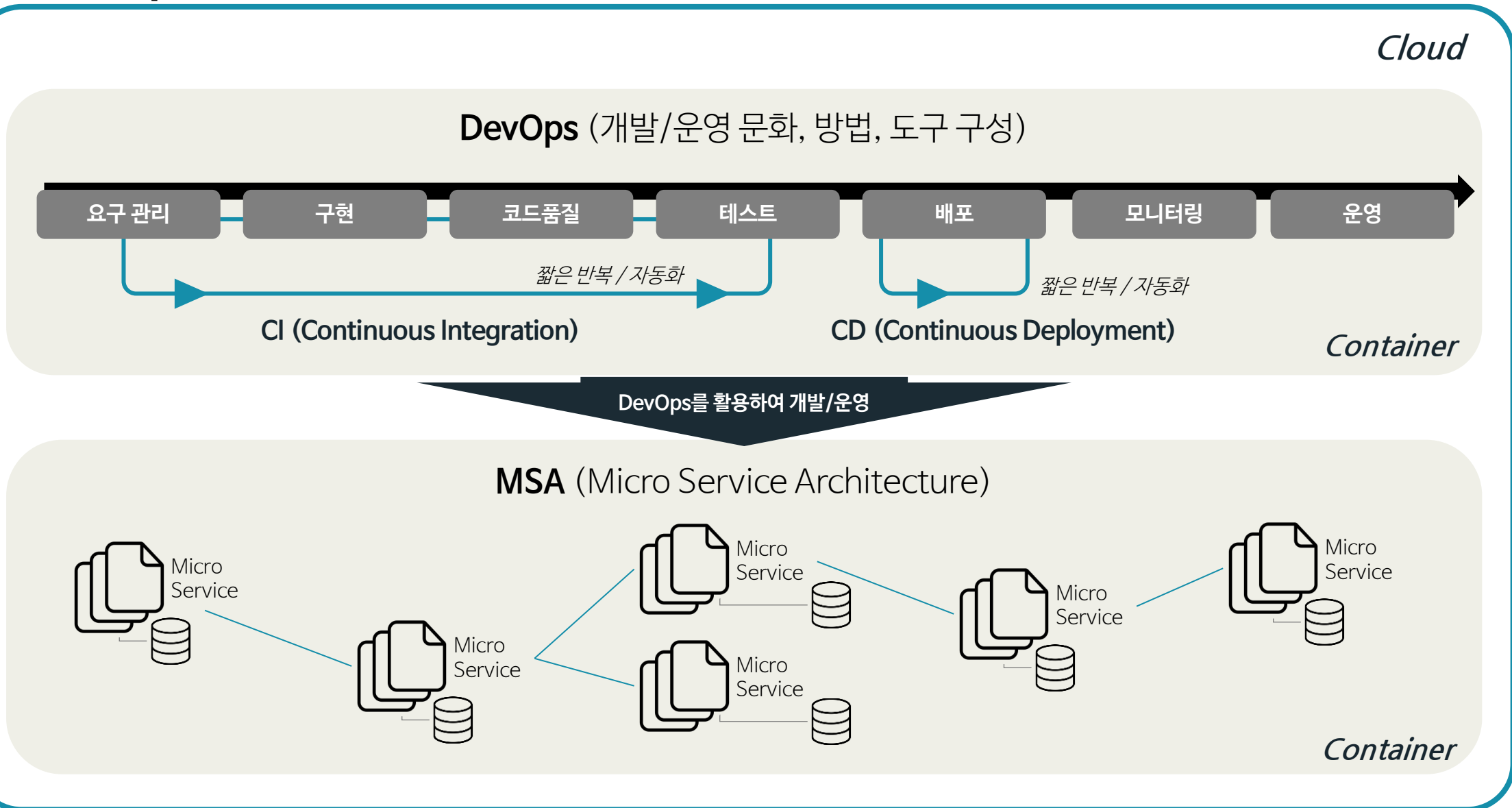
# IT 기술 트렌드의 변화

- 소프트웨어는 비즈니스 지원이 아닌 비즈니스의 중심
- 고객과 사용자는 더 빠른 대응, 더 나은 품질과 안정성을 요구
- 기존의 역할과 책임, 개발 방식, 아키텍처, 인프라의 한계

## IT 기술 트렌드의 변화

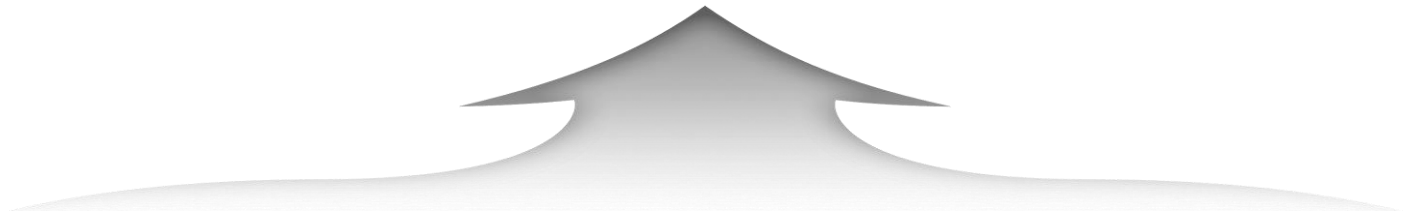


# DevOps, CI/CD, MSA의 관계





## 품질과 안정성이 확보된 소프트웨어를 더 빠르게 제공

- 
- 소프트웨어 개발/운영 담당자의 원활한 의사 소통
  - 개발/운영 프로세스의 단순화 및 자동화 (휴먼 에러 최소화)
  - 지속적인 소프트웨어 품질 및 안정성 확보

01 / 요즘의 SW 개발 톨체인, 지속적 통합

## 기술 관점의 DevOps 사전 조건

**자동화**된 지속적 통합(CI) / 지속적 배포(CD) 환경 구축

01/

요즘의 SW 개발 툴체인, 지속적 통합

## 지속적 통합(CI: Continuous Integration)



모든 SW 개발에 적용 가능한 정의

01/

요즘의 SW 개발 툴체인, 지속적 통합

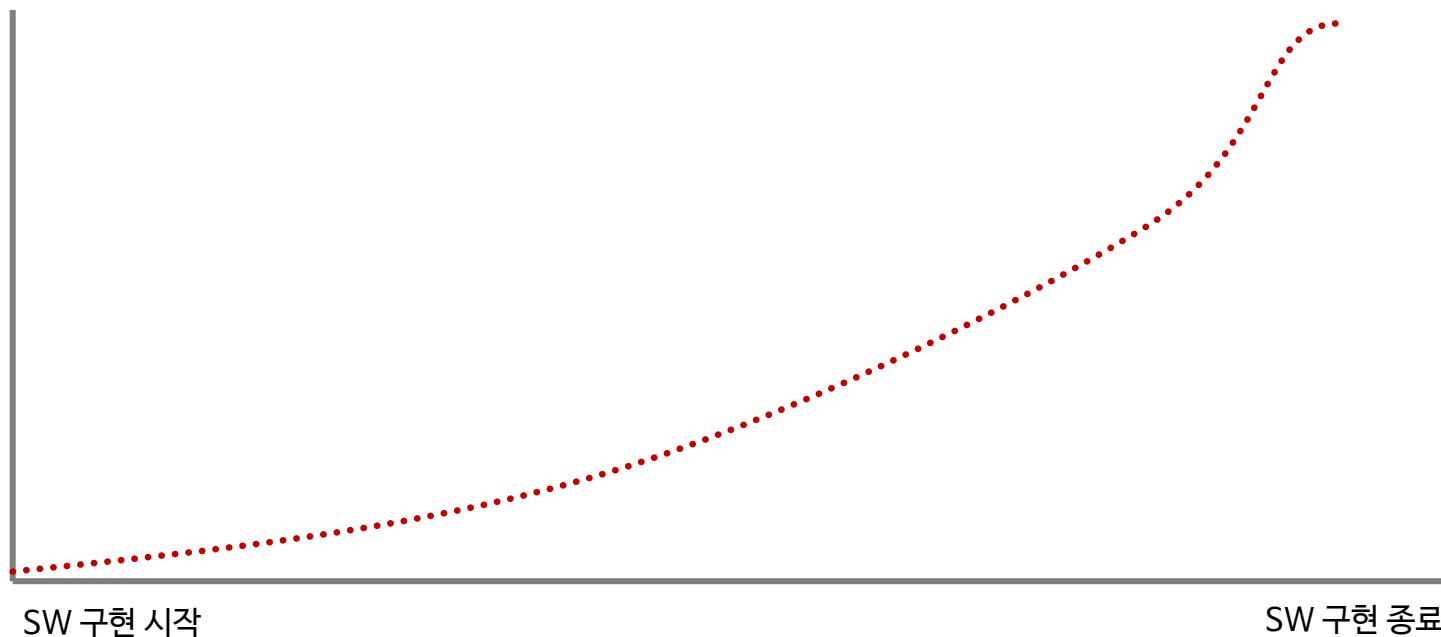
# 보통 소스코드/소프트웨어 문제점을 언제 확인하는가?

## “소프트웨어 구현 후반부에”

### 소스코드/소프트웨어 문제들

- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율

“소스코드가 ISO 26262 요건을  
**너무** 많이 위반했는데요...”



01

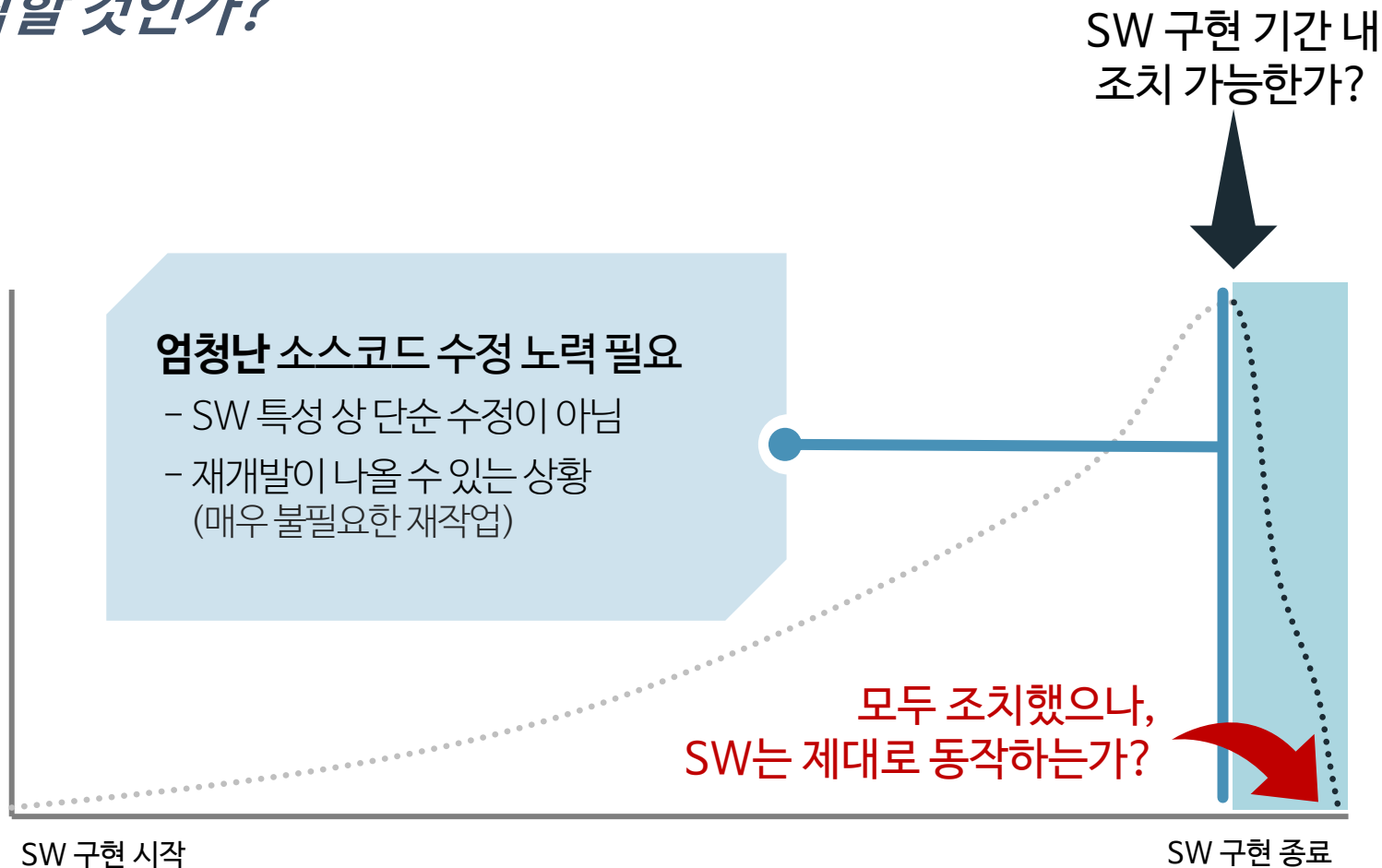
요즘의 SW 개발 툴체인, 지속적 통합

# 이 상황에서 ISO26262 요건을 만족해야 한다면

SW 구현 종료까지 **조치** 가능한가?  
조치가 끝나면 SW가 잘 동작할 것인가?

## 소스코드/소프트웨어 문제들

- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율



01

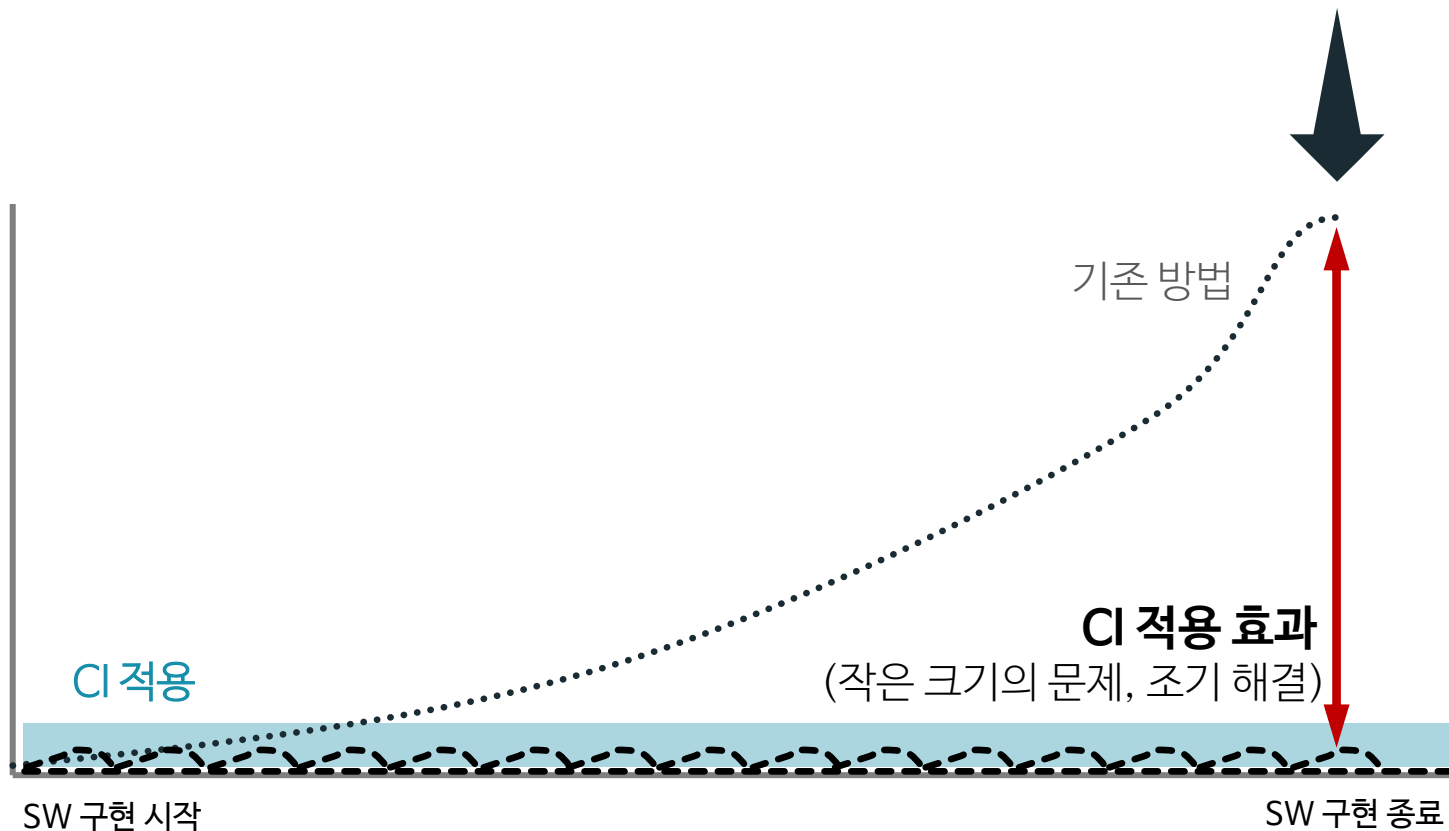
요즘의 SW 개발 툴체인, 지속적 통합

# 보다 나은 방법은?

SW 구현 시작부터 지속적 통합(CI)을 적용,  
도구가 **알아서** 검사/보고하여 품질과 안정성 조기 확보

## 소스코드/소프트웨어 문제들

- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율



01

요즘의 SW 개발 톨체인, 지속적 통합

## 지속적 통합(CI)의 장점

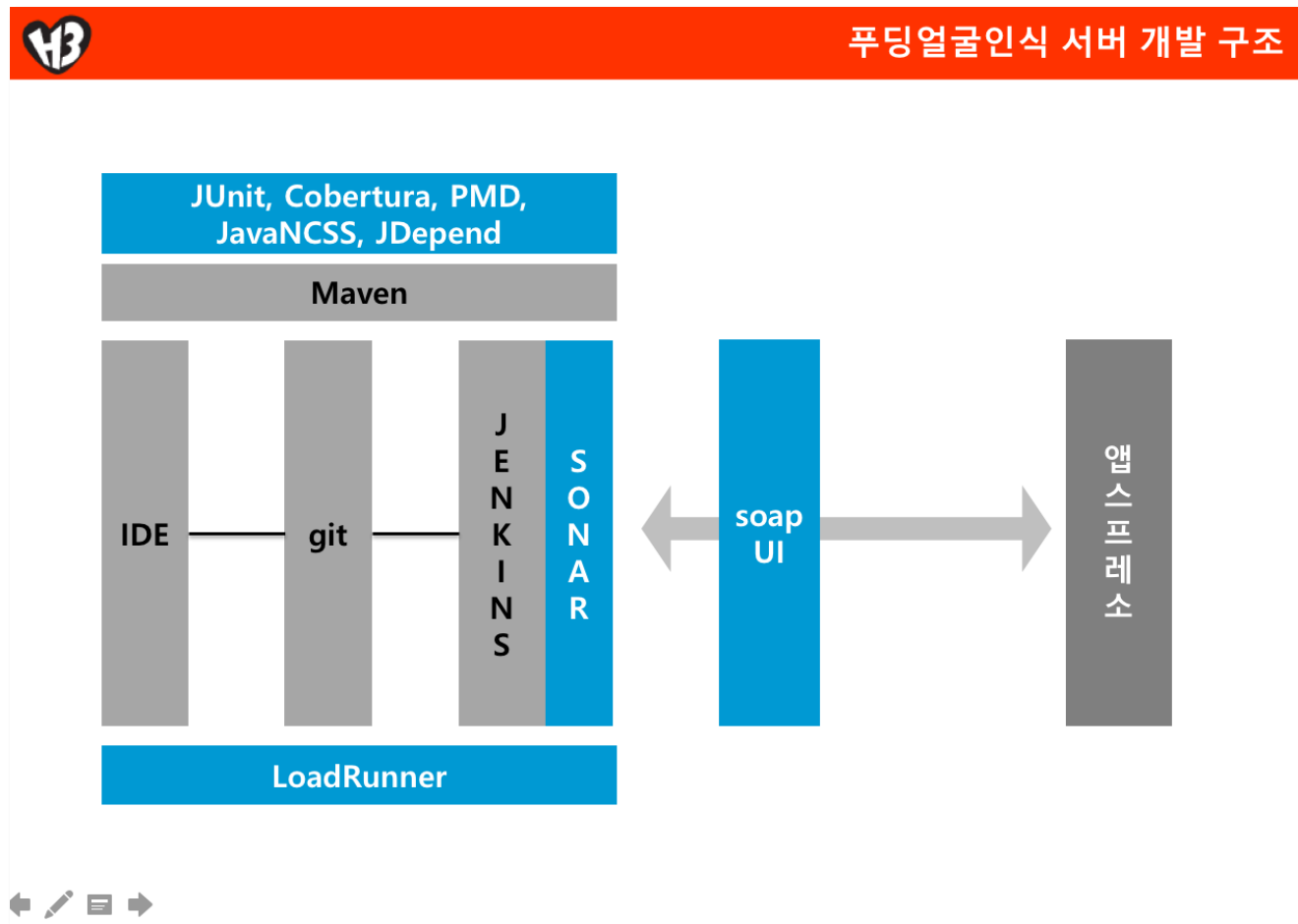
- 소스코드/소프트웨어 빌드 및 품질 문제점을 빠르게 식별하고 조치
- 빌드, 품질 검사, 문제점 보고, 배포 자동화로 휴먼 에러 및 투입 자원 감소
- 빌드 환경의 형상을 소프트웨어 폐기까지 유지
- 오픈소스를 기반으로 구축 가능 및 다양한 구축 사례 공유

01

요즘의 SW 개발 툴체인, 지속적 통합

# 지속적 통합(CI): 이제는 많이 하는 소프트웨어 개발 활동

[ 2012 H3 컨퍼런스 발표 ] - 10년전 한동준 대표의 지속적 통합 사용사례 발표

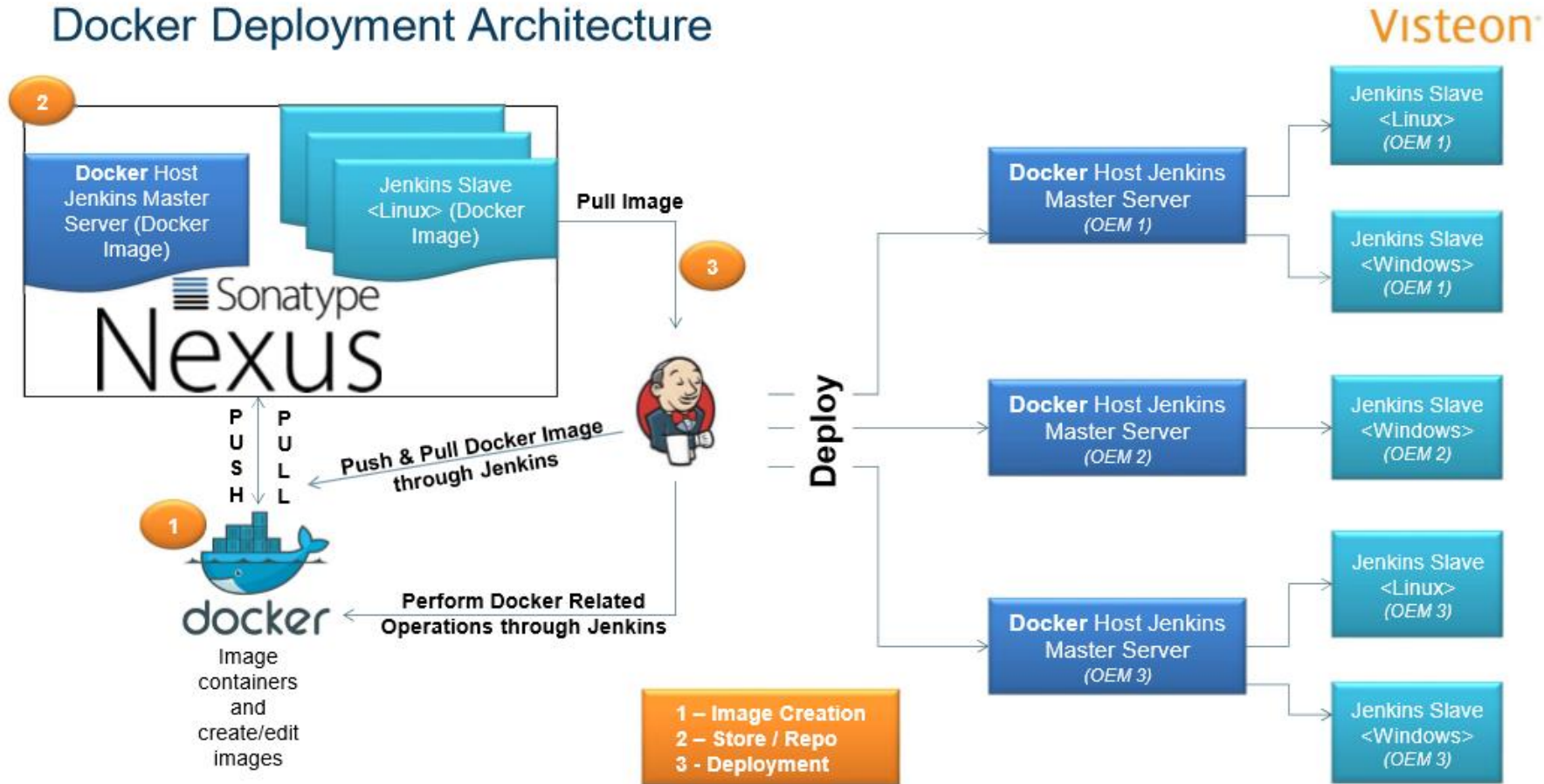




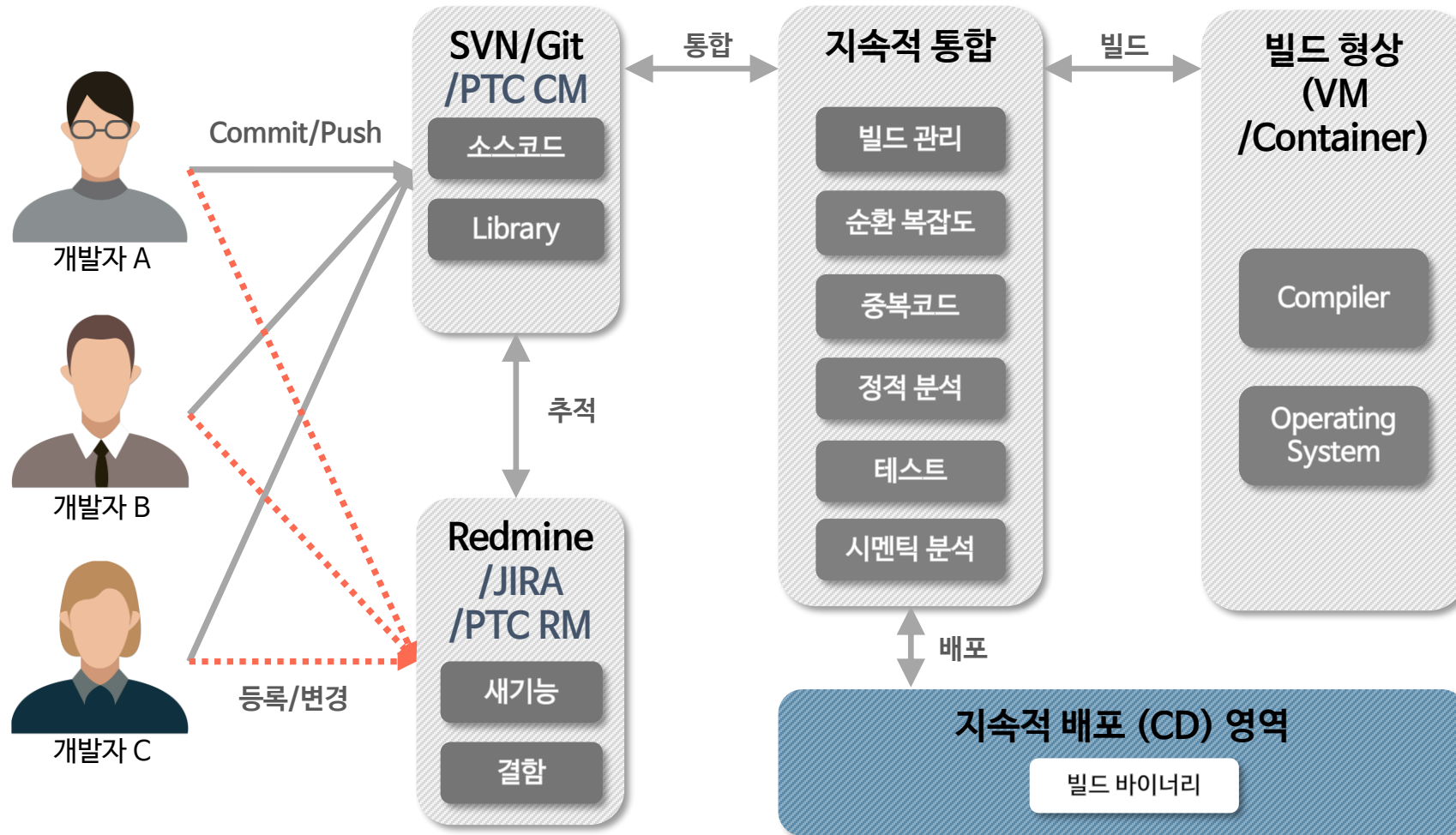
01

요즘의 SW 개발 툴체인, 지속적 통합

# 보수적인 자동차 임베디드 SW개발도 적용 확대



## 요즘의 일반적인 CI 환경 구성



01

요즘의 SW 개발 톨체인, 지속적 통합

## CI 및 CD의 동작 방식은 단순

- 1 개발자가 소스코드를 Git 저장소에 Commit/Push
- 2 Jenkins가 정해진 시간에 Git에서 지정된 소스코드 가져오기
- 3 Jenkins가 정해진 스크립트에 따라 빌드/검증 수행
- 4 문제점 발생 E-Mail, Messenger, 이슈 관리 시스템 보고
- 5 Jenkins가 정해진 스크립트에 따라 배포

자동화

01/

요즘의 SW 개발 툴체인, 지속적 통합

## 임베디드 SW는 덜 활성화. 그 이유는?

- SW 개발을 돕는 새로운 기술의 비교적 **신중한 접근**
- **웹/앱** 개발하는 그들만의 도구라는 **오해**
- 호환이 잘 안되는 **너무 다양한 컴파일러 및 버전**
- **민감한 빌드 환경**
- **상용 도구 위주**의 SW 품질 확보
- 이미 사용하고 있지만 외부에 **공유를 꺼려함**

# 상용 정적분석/테스트 도구의 Jenkins 지원

## [ QAC/QAF ]

**PRQA Plugin**

73 Added by [Mads Nielsen](#), last edited by [Oscar Forner Martinez](#) on Aug 26, 2016 ([view change](#))

**Jenkins**

- Home
- Mailing lists
- Source code
- Bugtracker
- Security Advisories
- Events
- Donation

**Plugin Information**

Plugin ID	Changes	In Latest Release Since Latest Release
prqa-plugin	Source Code Issue Tracking Pull Requests Maintainer(s)	GitHub Open Issues Pull Requests <a href="#">Igor Kostenko</a> (id: isanych)
Latest Release	2.1.0 ( <a href="#">archives</a> )	
Latest Release Date	Sep 26, 2016	
Required Core Dependencies	1.480.3	

## [ VectorCAST ]

**VECTOR software**

Home > Resources > Whitepapers

### Building a Flexible and Automated Testing Infrastructure Using VectorCAST/Manage and Jenkins

## [ CodeSonar ]

**CodeSonar Plugin**

8 Added by [Mads Nielsen](#), last edited by [Andrius Ordojan](#) on Sep 12, 2016 ([view change](#))

**Jenkins**

- Home
- Mailing lists
- Source code
- Bugtracker
- Security Advisories
- Events
- Donation

**Plugin Information**

Plugin ID	Changes	In Latest Release Since Latest Release
codesonar	Source Code Issue Tracking Pull Requests Maintainer(s)	GitHub Open Issues Pull Requests <a href="#">Pragma Josra</a> (id: pragma_josra)
Latest Release	2.0.4 ( <a href="#">archives</a> )	
Latest Release Date	Sep 12, 2016	
Required Core Dependencies	1.609.2 <a href="#">credentials</a> (version: 1.24) <a href="#">job-dsl</a> (version: 1.37, optional)	

## [ Polyspace ]

**Jenkins**

**Polyspace\_Jenkins\_plugin.zip**

by [Ram](#)  
21 Jul 2014 (Updated 01 Sep 2016)

Plugin to integrate Polyspace verification into your Jenkins job

[Watch this File](#)

Be the first to rate this file!

3 Downloads (last 30 days)  
File Size: 9.82 KB  
File ID: #47148  
Version: 1.1.0.1

## [ LDRA ]

**Jenkins**

**LDRA interfaces with the Jenkins Continuous Integration Platform.**

The LDRA tool suite offers support for Jenkins, a continuous integration platform that automates and manages the build process for large-scale, system-level software projects. Jenkins connects the build and version control systems with quality management and automated testing. By linking the LDRA tool suite to Jenkins, the platform can invoke the LDRA tool suite to perform analysis, track test and verification fulfilment, and assess overall improvements or regressions in code quality over time.

## [ Sparrow ]

**소프트웨어 개발 프로세스 통합**

- ✓ 기존의 개발환경을 그대로 활용하여 소스코드 분석 및 결과 확인
- ✓ GUI 분석 도구로 일괄 점검
- ✓ REST API / CLI 분석 도구로 배치 · 스케줄링 분석
- ✓ REDMINE, JIRA, Jenkins, Hudson, FRISM, AURORA, Harvest, eCAMS 등과 연동하여 보안 및 품질 기준에 부합하지 않는

설명만 그럴싸하고 제대로 지원하지 않는 도구도 있음

02

# CONTENTS

빠르게 지속적 통합  
구축해보기

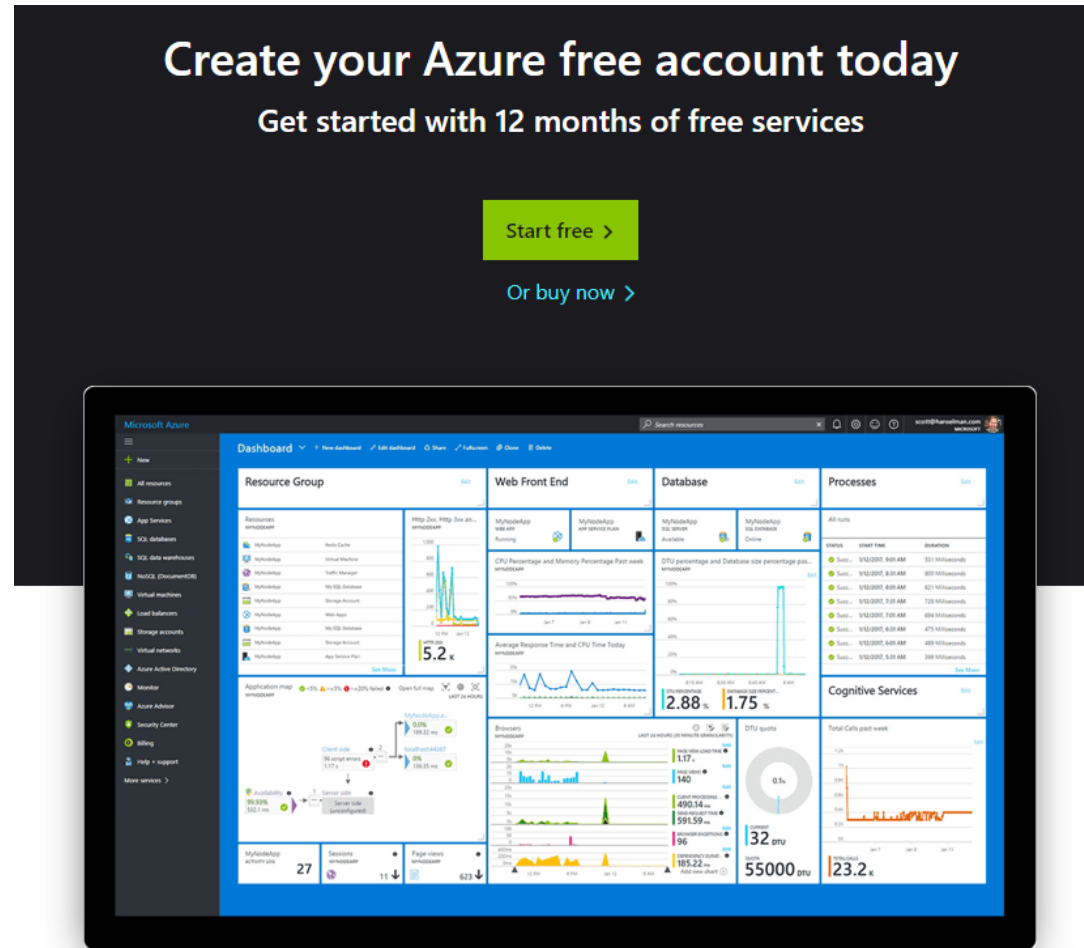
# 실습 서버 구성 (1대)

- 시스템 (다음 중 하나 선택 / Windows 10 활용)

- Azure
  - Windows Server 사용
- 개별 PC 사용 가능
- Virtual Box 가능

- S/W

- Java (또는 OpenJDK)
- **Jenkins**
- 빌드도구
- Git
- CppCheck
- Doxygen



# Jenkins 설치 시 유의할 점

- 방화벽

- 회사 외부에서 접속하려면 회사 방화벽 해제 필요
  - 내부망에서 사용은 정책 확인해야 함
- Jenkins Server 방화벽 해제
  - Windows의 경우, Windows Firewall에서 예외 규칙 추가



# Jenkins 설치: 기본 설치 버전 사용

03

# CONTENTS

SW 빌드/검증 환경  
12년 유지하기

## 해외 고객님의 요구



**“소스코드 15년간 빌드 보장해”**

## 현실과 해결 방안 고민

- 5년이면 고장나는 컴퓨터 하드웨어
- 개발자 퇴사... 빌드 환경과 소스코드는 사라지고...
- Win 10을 표준으로 사용하는데 Win XP에서 재빌드 필요



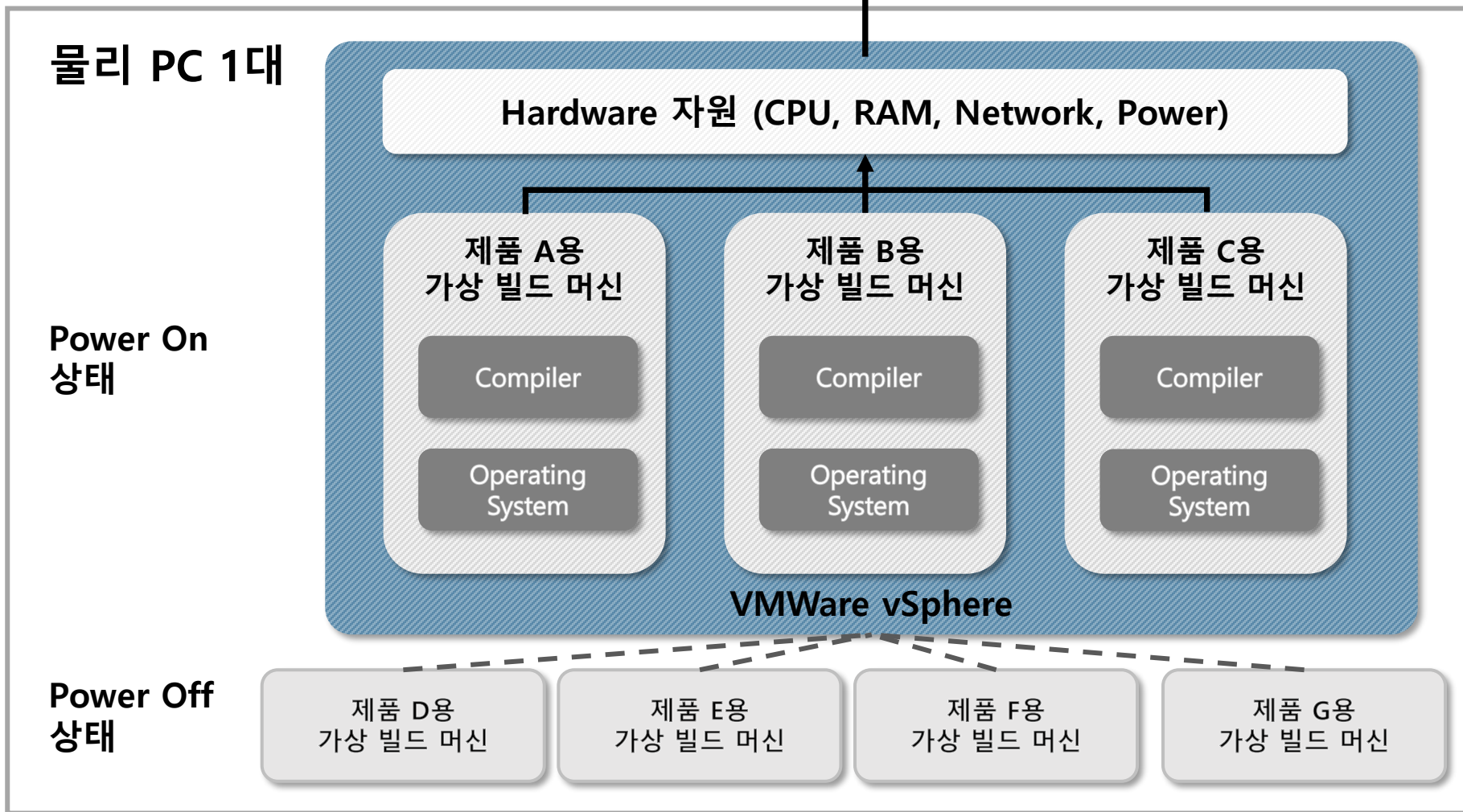
**빌드 PC를 가상화**

*언제나 2021년 09월 29일 15:00 스냅샷으로 빌드*

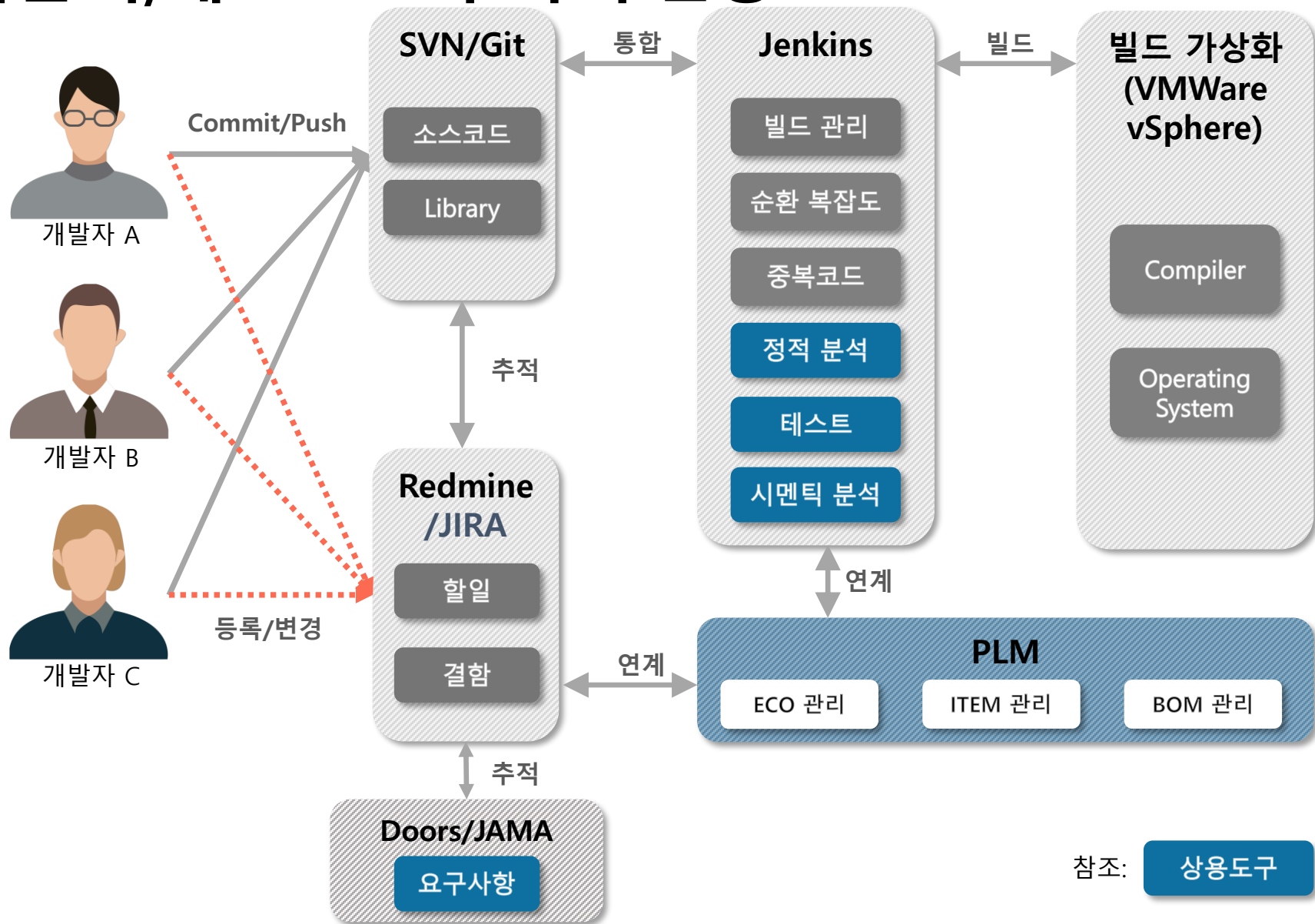
# 지속적 통합 기반의 빌드 가상화



Jenkins



# 상용 정적분석/테스트 도구와의 연동



## 새로운 문제: 비용 이슈가 발생함

- 전사 적용이 아닌 경우, 너무 많은 비용 필요
  - 대기업의 경우 팀/프로젝트 단위의 **소규모 CI 구축**이 많음
  - 가상화에 사용한 VMWare vSphere는 서버 가상화 도구 (\$\$\$\$)
  - 하드웨어(박스 1~3대) 및 SW 구입에 X천만원 필요

“데스크탑 가상화는 어떨까?”

## 해결 1: Mac + VMWare Fusion 조합

- 시스템: MacMini 2018 + 추가 램
  - 약 137만원 + 가격(OWC 직구) + 애플 공식 장착 비용 44,000원
    - 64GB(32GB \* 2) 구매: 램은 한국에서 만드는데 한국에서 팔지 않음
  - MacOS 사용 이유
    - Docker를 쓰고 싶고(Jenkins, Redmine 운영)
    - 기본 터미널로 ssh 를 자유롭게 쓰고 싶고
    - **Linux는 사용이 어려워 활용성이 떨어지고 (원도 사용자 기준)**
    - 타임머신으로 백업 걱정에서 벗어나고 싶음
- 가상화 도구: VMWare Fusion Pro
  - MacOS용 가상화
  - 약 20만원



## 02/ 빠르게 지속적 통합 구축해보기

# 해결 2 게임용 Notebook + VMWare Workstation 조합

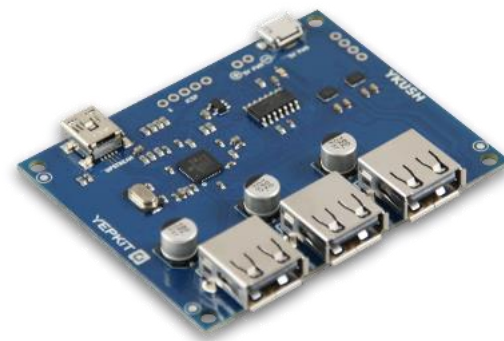
- 시스템: L/H사 노트북(240)+ 추가 램(30만원)
- 가상화 도구: VMWare Fusion Workstation Pro
  - Windows용 가상화
  - 약 30만원
- Windows/Linux VM(가상머신)을 혼용

# [참고] 공용 클라우드 사용도 고민 했으나 보안 이슈로 중지

- 빌드 가상화를 Private Cloud(VMWare)에서 Public Cloud(AWS/Azure)로 이동
  - 전문적인 인프라 관리자가 필요 없음
  - 고가용성(최대 99.99%), 신뢰성(최대 99.99999999%)
  - 초기 투자비용 불필요
  - 사용한 만큼 비용 지불
- As-is 적용 문제점
  - Jenkins AWS/Azure 플러그인으로 스냅샷을 자유롭게 Control 못함  
→ API 확인하니, 코딩 조금 하면 가능할 것 같음
- 개발 프로젝트 1개의 1달 예상 비용 (AWS: Amazon Web Service)
  - 가정: 일일 빌드. 빌드 1회에 최대 1시간 사용. 빌드 당 0.5GB 네트워크 사용
  - 가상 머신: \$2.94
  - 디스크: \$0.14
  - 네트워크 사용: \$1.89
  - 스냅샷 스토리지: \$0.5
  - 총 \$5.47 + VAT
  - 출시 후 유지보수 시에는 비용이 더 절감됨: ECR 처리 시에만 빌드

## 새로운 문제: Jenkins 테스트 시 타겟 보드를 재부팅 하자

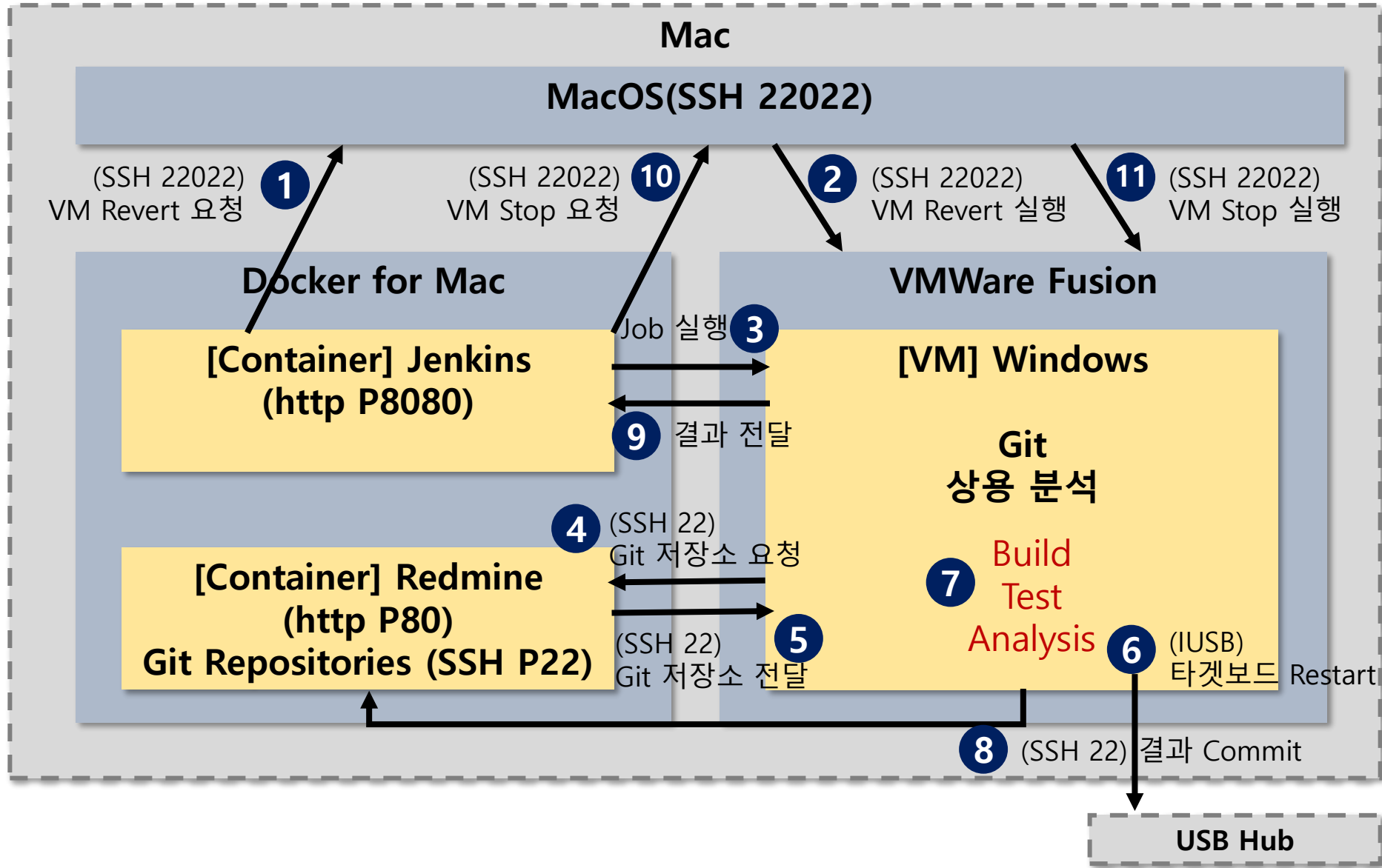
- 추가 요구사항
  - 임베디드 SW 테스트: 빌드 -> 타겟 보드에 바이너리 업로드 -> 테스트 실행
  - 테스트 신뢰성 향상을 위해 타겟 보드 재부팅 필요
  - **Jenkins가 테스트 전 자동으로 타겟 보드 재부팅**으로 메모리 삭제 필요
    - 재부팅의 의미: 전원 Off → On
- 해결방안 고민
  - 놀랍게도, Wifi나 USB를 이용해 전원(멀티탭)을 제어할 수 있는 제품이 없음
    - 통신사, 알렉사, 샤오미 등 제품이 있으나 외부 네트워크 연결 필요
  - USB의 전원 유무로 멀티탭의 전원을 제어하는 장비 발견!
    - 원래 목적은 컴퓨터 종료 시 주변기기 전원 Off (절전 멀티탭)
  - 최초안) 라즈베리파이를 이용해 USB 전원 제어
  - 결정안) 미국에서 Python으로 USB 제어 가능한 USB Hub 구매



YKUSH

# 02/ 빠르게 지속적 통합 구축해보기

## 동작 구성도



04

CONTENTS

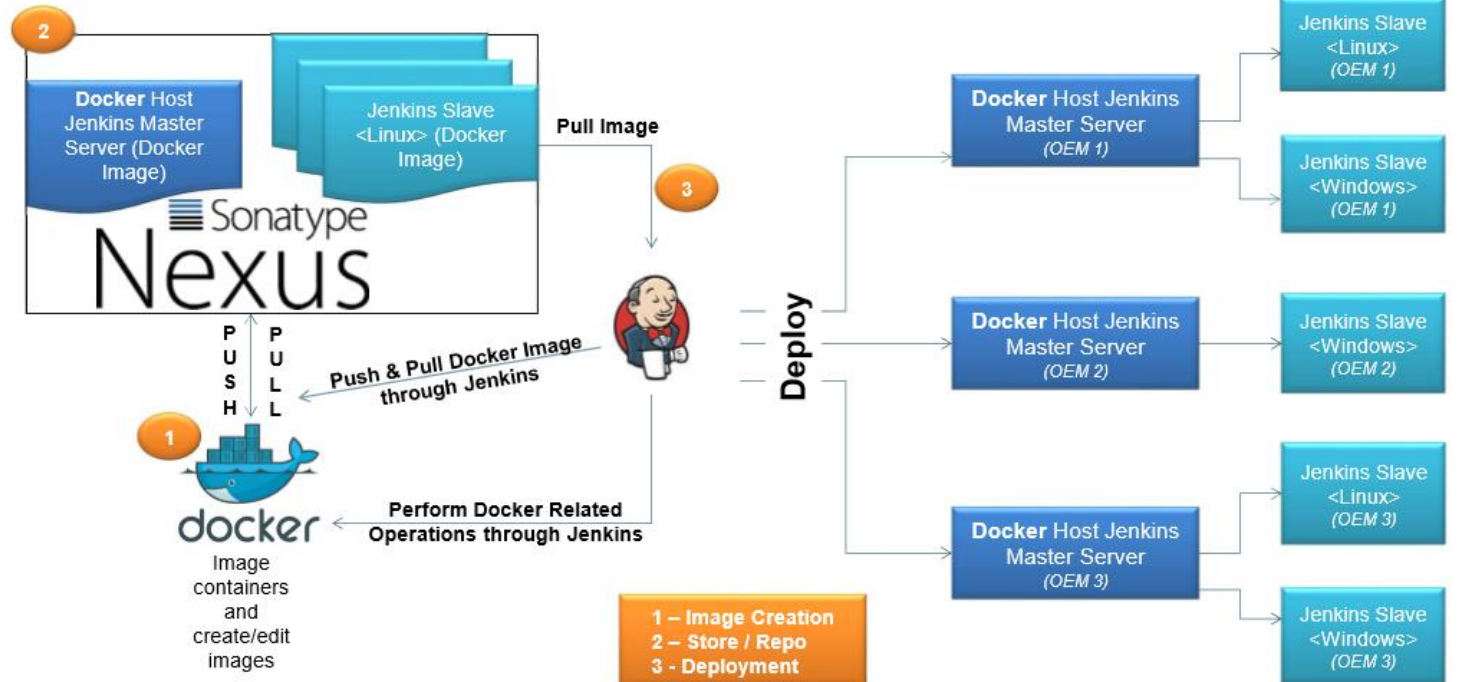
결론 및 조언

# 조심할 뿐, 관심은 점점 확대



- 자동차 분야 애자일 컨퍼런스
- 2016년 부터 독일/미국에서 개최
- BMW, Volvo 등의 애자일 문화/기술 발표

## Docker Deployment Architecture



## GitHub ♥ Auto



분야를 막론하고

SW 개발에 **지속적 통합**은 필수입니다.

기능 안전 SW처럼 표준 요건 준수가 필수인 경우  
나중에 정적 분석 MISRA 룰 위반 30,000개에 좌절 말고  
개발 시작부터 지속적 통합으로 확인합시다



## 조언 #1: 임베디드 SW에 CI를 적용하실 분께

- CI 적용 이전에 SVN/git/PTC CM 먼저 잘 쓰기
- 시스템 헤더 및 모든 SW Asset을 SVN/Git/PTC CM 에서 관리하기
- 모든 빌드 관련 요소는 project 파일의 하위 상대경로로 설정
- 컴파일러 라이선스는 네트워크(Floating) 라이선스 구입
- 다른 도구 라이선스도 네트워크(Floating) 라이선스 구입
- Jenkins Job은 2개 - 빌드 전용, 품질 Metric/상용도구 전용
- Jenkins 빌드 실패에 민감하지 않기: 금방 고치면 됨

## 조언 #2 오픈소스 사용은 무료지만 무료가 아님

- 우리는 “Copy 당 3천만원” 도구에 익숙함
  - 안되면 벤더사 전화
  - 궁금하면 벤더사 전화
  - 귀찮으면 엔지니어 호출
- 오픈소스 사용은 무료지만 도구 관리자는 무료가 아님
  - 설치/설정에 구글링(Stackoverflow): 비용(=시간)
  - 안되면 구글링(Stackoverflow): 비용(=시간)
  - 궁금하면 구글링(Stackoverflow): 비용(=시간)
  - 귀찮으면 비용 지불하고 전문가 활용: 비용(=시간)
  - 누가 Jenkins를 구축하고 운영할 것인가의 문제 발생: 갈등

**생각보다 큰 문제**

## 사내에 지속적 통합을 적용 하신다면

- 전사 SW 인프라팀 주관 Vs. 각 프로젝트 각개 전투
  - 조직/프로젝트 특성에 따라 상이
  - 전사 주관이 항상 옳지는 않음
- 사내 전문가가 필요
  - 특정 벤더에 모든 것을 맡기면 편하나, 상주가 필요함
  - 전사 주관이든, 각 프로젝트 각자 구축이든 회사 차원의 지원 전문가 양성

## 미래 준비

- V2X는 흐름

- 차와 차
- 차와 사물
- 차와 서버(백엔드)

- 곧 우리에게도 서버 SW 개발이 필요할 수 있음

- 서버(백엔드) 개발하는 도메인은 지속적 통합이 필수
- 서버는 배포 자동화까지 연결되어야 함 (사람이 서버 100대에 일일이 배포할 수 없음)

THANK YOU