

# JenKins 를 이용한 CI/CD Pipeline 구축



```
public static void main(String[] args)
{
    < servlet >
        < servlet-name >BoardController</servlet-name>
        < servlet-class >com.joneconsulting.controller.BoardController</servlet-class>
        < init-param >
            < param-name >user_name</param-name>
            < param-value >Kenneth Lee</param-value>
        </init-param >
    </servlet >
```

```
    < servlet >
        < servlet-name >BookController</servlet-name>
        < servlet-class >com.joneconsulting.controller.BookController</servlet-class>
        < init-param >
            < param-name >self_title, price, author</param-name>
            < param-value >self_title
                self_price
                self_author = author
            </param-value>
        </init-param >
    </servlet >
```

```
    < servlet >
        < servlet-name >UIApplicationDelegate</servlet-name>
        < servlet-class >com.joneconsulting.delegate.UIApplicationDelegate</servlet-class>
        < init-param >
            < param-name >NEXT_INNOVATION_DELEGATE</param-name>
            < param-value >@Interface NextInnovationDelegate : NSObject</param-value>
        </init-param >
    </servlet >
```

```
    < servlet >
        < servlet-name >UIProject</servlet-name>
        < servlet-class >com.joneconsulting.delegate.UIProject</servlet-class>
        < init-param >
            < param-name >NEXT_PROJECT</param-name>
            < param-value >@Interface NextProject : NSObject</param-value>
        </init-param >
    </servlet >
```



# 프로필

Dowon Lee



지식공유자 인증

5452 ★ 4.8(420)

멘토링 활성



- 홈
- 강의
- 로드맵
- 수강후기
- 블로그

최신순 ▾

## 강의 (3)



Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)

Dowon Lee

★★★★★ (174)

학습중

+3000명 독점 할인중



Spring Boot를 이용한 RESTful Web Services 개발

Dowon Lee

★★★★★ (308)

학습중

+2700명 독점 할인중



웹 애플리케이션 개발을 위한 IntelliJ IDEA 설정

Dowon Lee

★★★★★ (228)

학습중

+3500명



## 목차

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

---

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

---

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

---

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## 목차

---

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix

## Section 1.

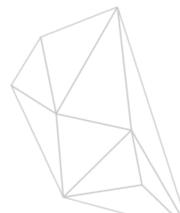
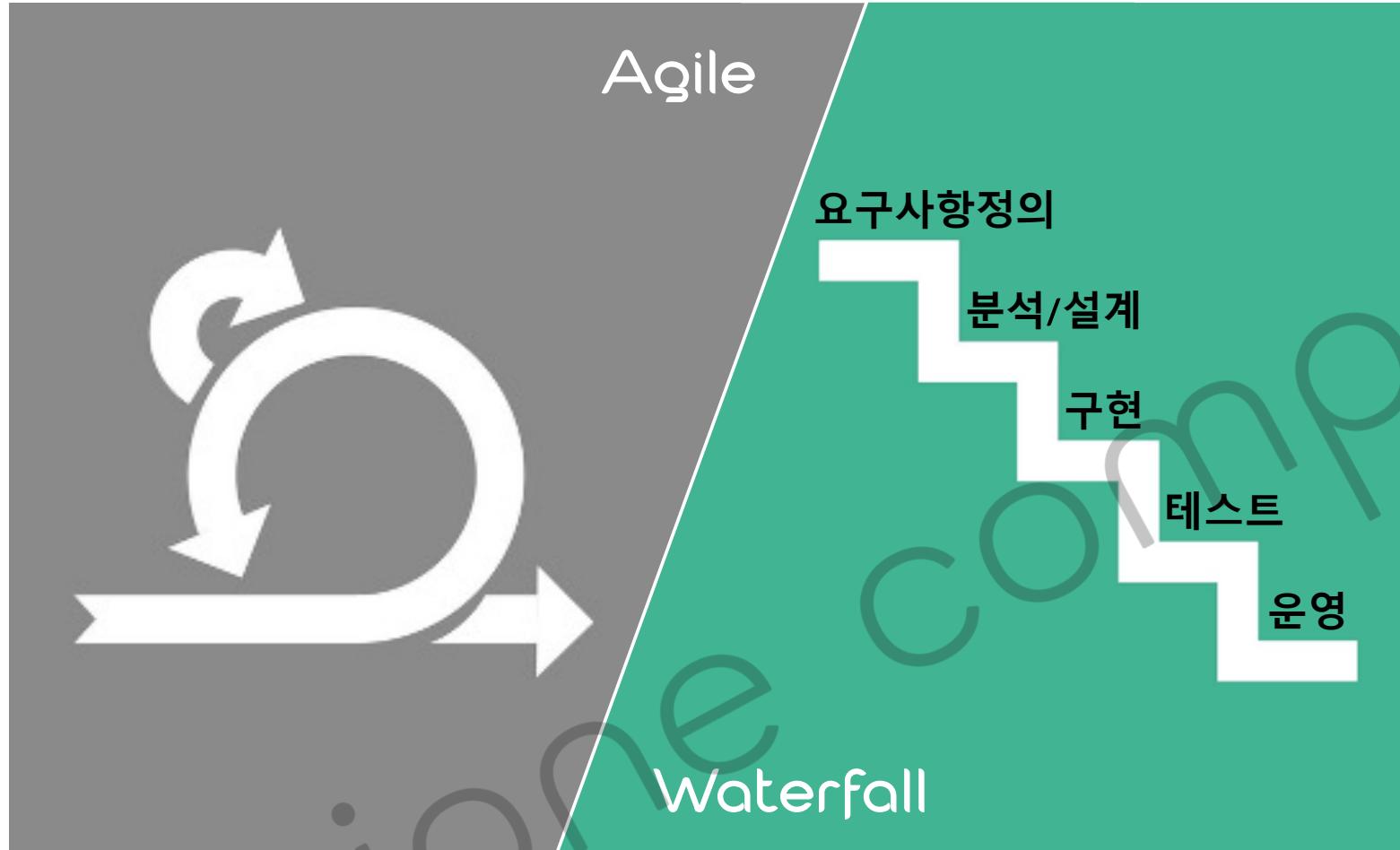
# DevOps와 CI/CD

- DevOps와 CI/CD
- Cloud Native Application
- CI/CD 작업 흐름
- Jenkins



# Waterfall vs Agile

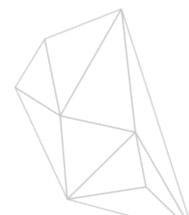
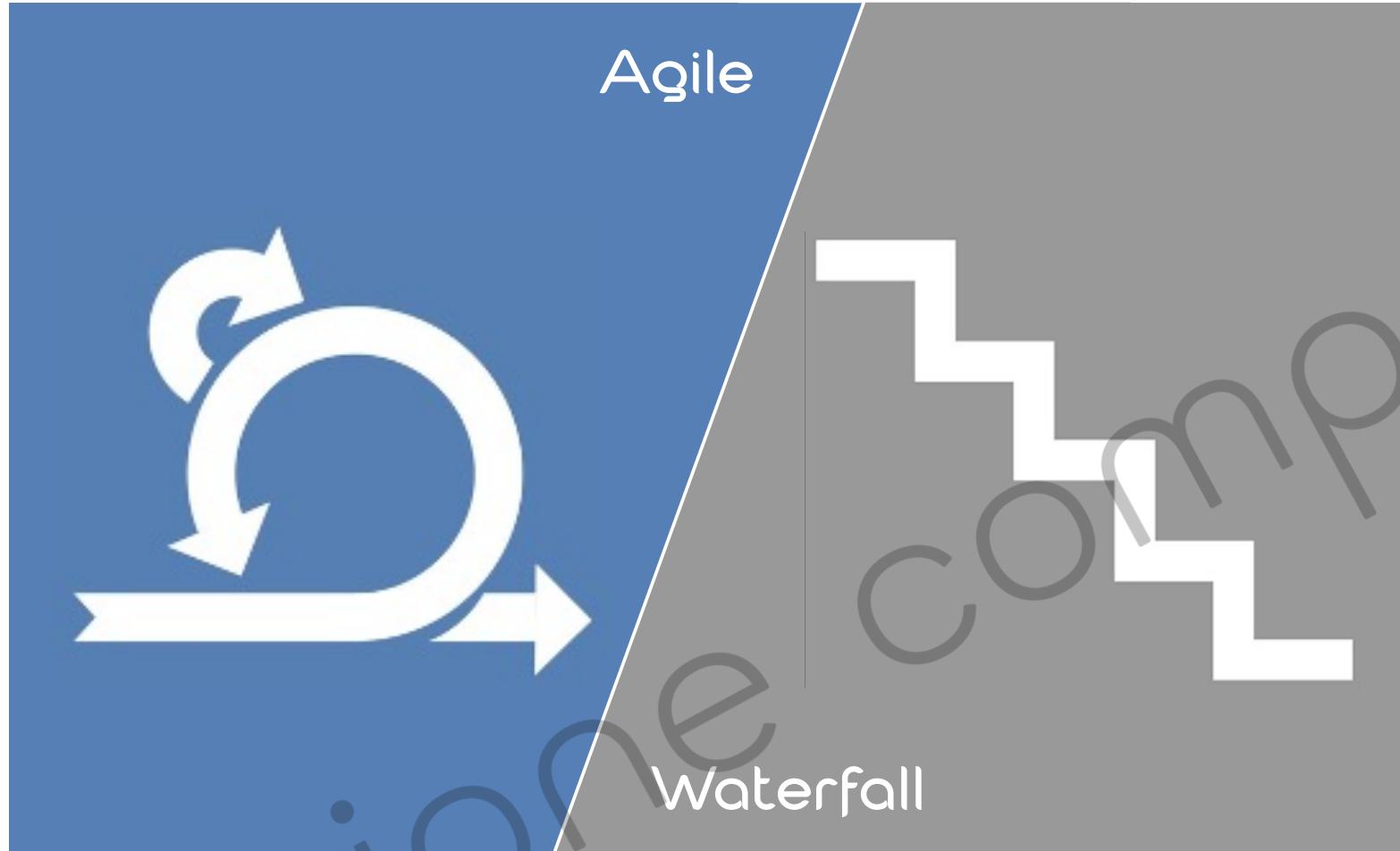
njone company





# Waterfall vs Agile

njone company



njone company



# Waterfall vs Agile

njone company

- ***Manifesto for Agile Software Development***

“

We are uncovering better ways of *developing software* by doing it and *helping others* do it.

Through this work we have come to value:

***Individuals and interactions*** over processes and tools

***Working software*** over comprehensive documentation

***Customer collaboration*** over contract negotiation

***Responding to change*** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

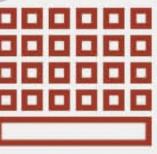
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

<http://agilemanifesto.org/iso/en/manifesto.html>



# Waterfall vs Agile vs DevOps

njone company

	Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
~ 1980	Waterfall 	Monolithic 	Physical Server 	Datacenter 
~ 1990				
~ 2000	Agile 	N-Tie 	Virtual Servers 	Hosted 
~ 2010	DevOps 	Microservices 	Containers 	Cloud 

<https://www.oracle.com/kr/cloud/cloud-native/what-is-cloud-native/>



# Cloud Native Architecture

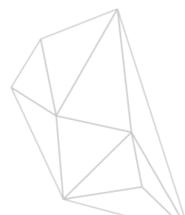
njone company

클라우드 네이티브 아키텍처 및 기술은 클라우드에서 빌드되고 클라우드 컴퓨팅 모델을 최대한 활용하는 워크로드를 디자인, 생성 및 운영하는 접근 방식입니다.

클라우드 네이티브 기술을 통해 조직은 퍼블릭, 프라이빗 및 하이브리드 클라우드와 같은 최신 동적 환경에서 확장 가능한 애플리케이션을 빌드하고 실행할 수 있습니다. 컨테이너, 서비스 메시, 마이크로 서비스, 변경할 수 없는 인프라 및 선언적 API는 이 접근 방식을 예로 들 수 있습니다.

이러한 기술을 사용하면 복원력, 관리 가능 및 관찰 가능한 느슨하게 결합된 시스템을 사용할 수 있습니다. 강력한 자동화와 결합되어 엔지니어는 최소한의 수고로 자주 예측 가능하게 높은 영향을 미치는 변경을 할 수 있습니다.

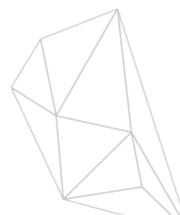
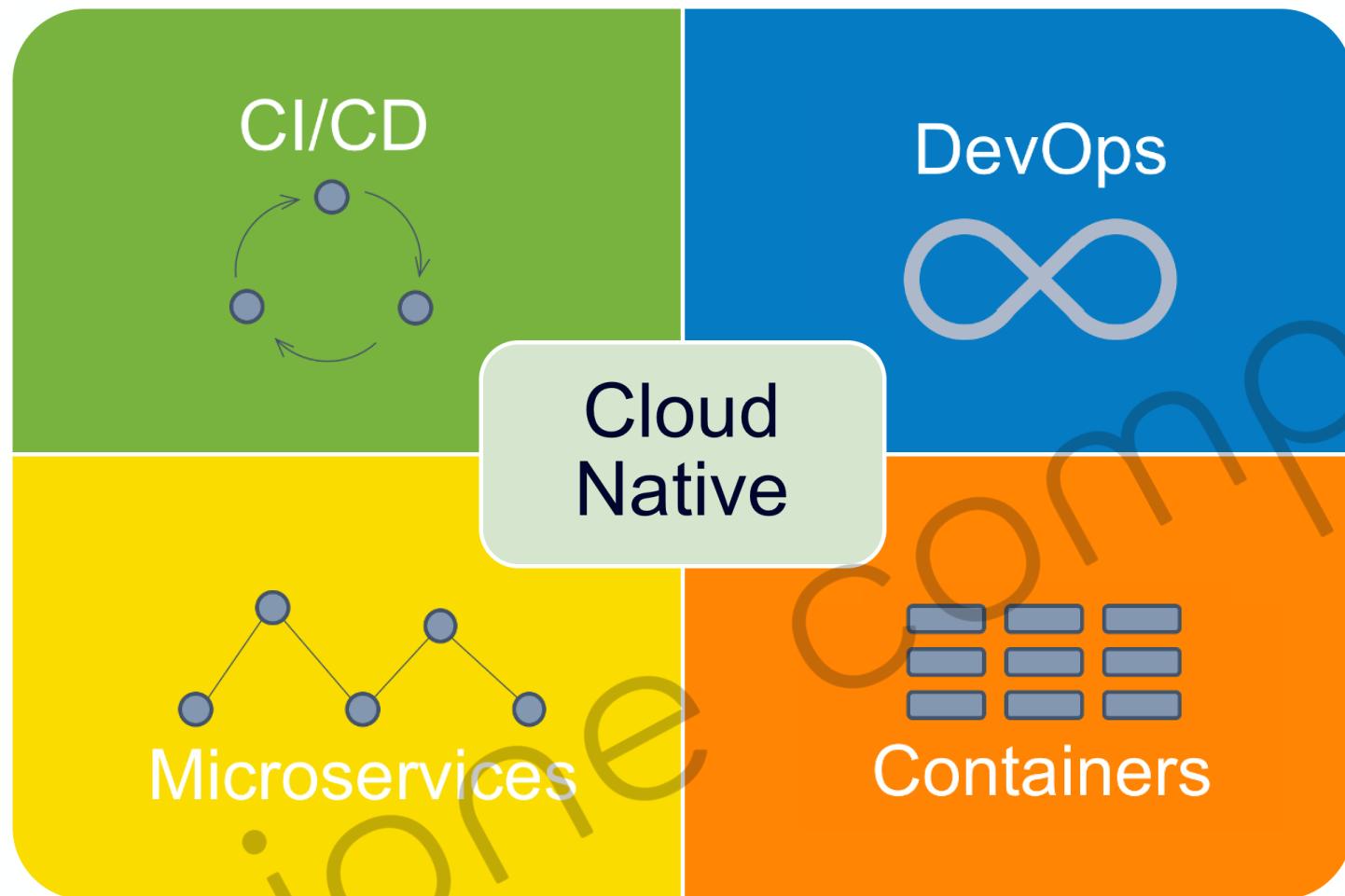
<https://docs.microsoft.com/ko-kr/dotnet/architecture/cloud-native/definition>





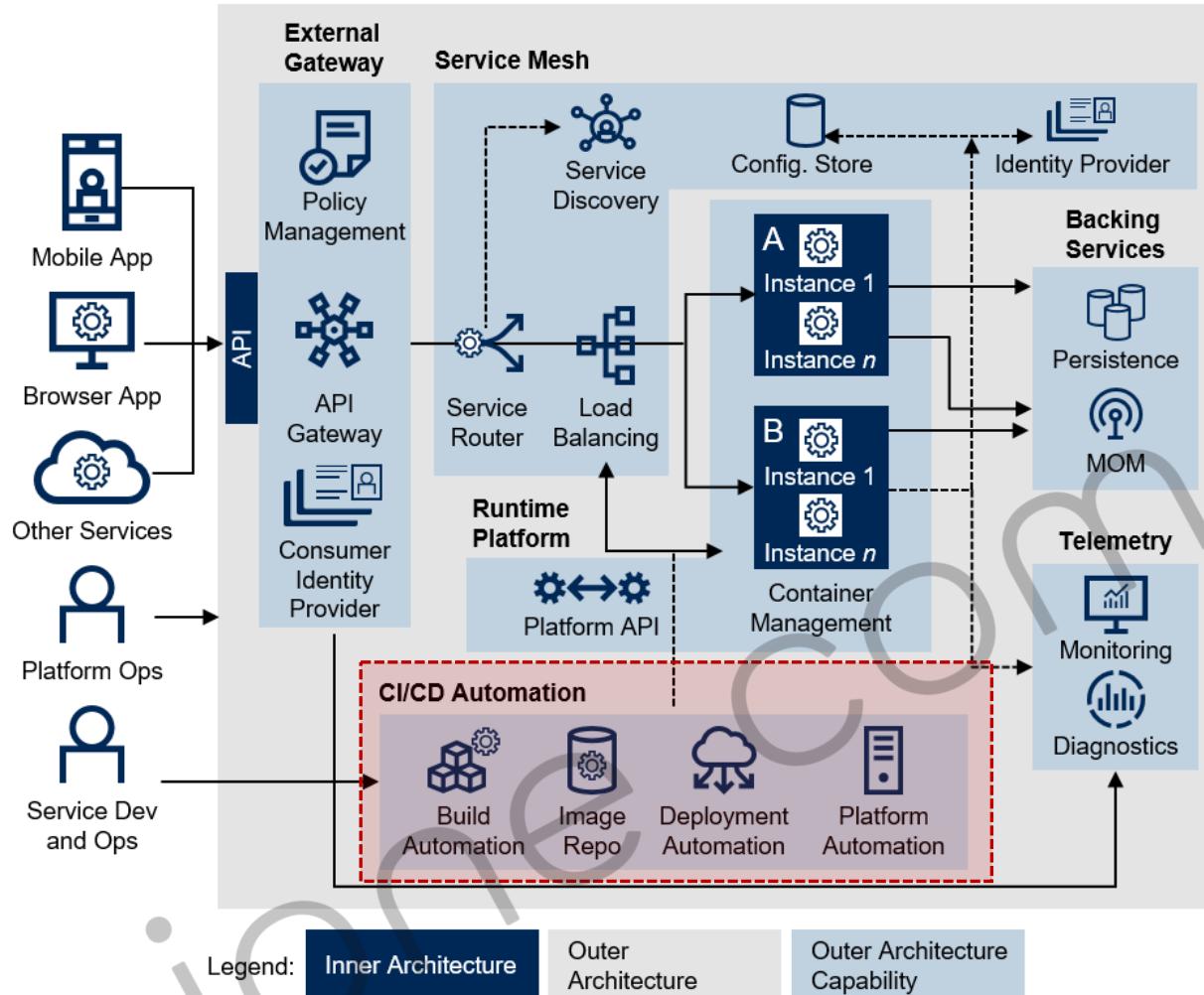
# Cloud Native Application

njone company



# Cloud Native - MSA

njone company



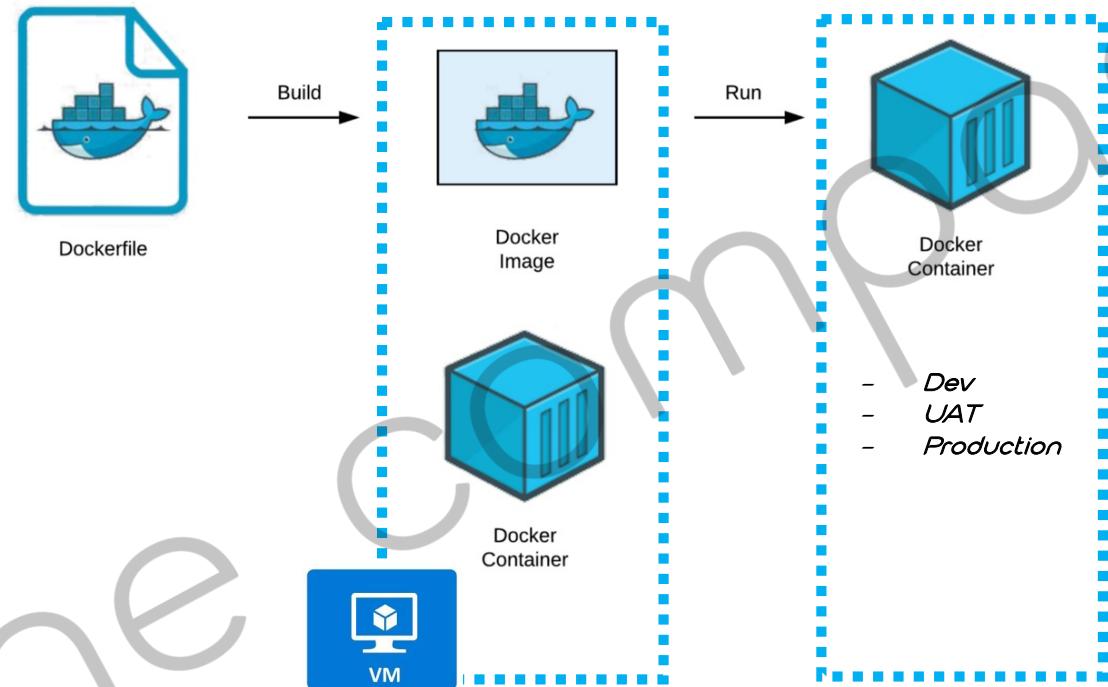
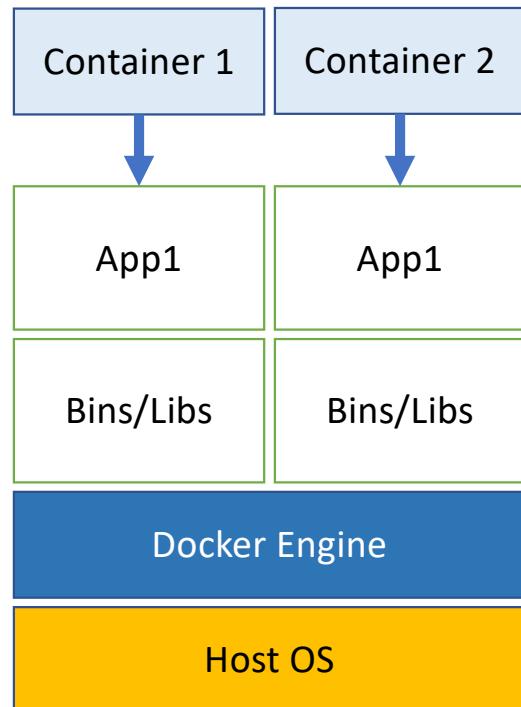
ID: 353896

© 2018 Gartner, Inc.



# Cloud Native – Containerization

njone company

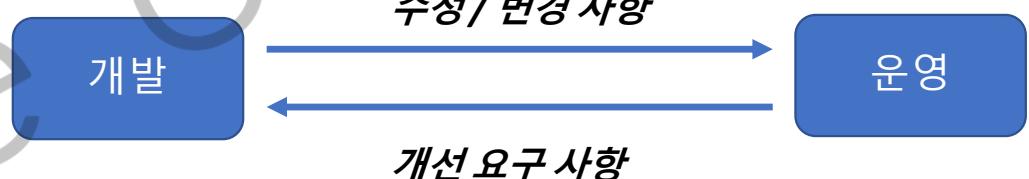




# Cloud Native – DevOps

njone company

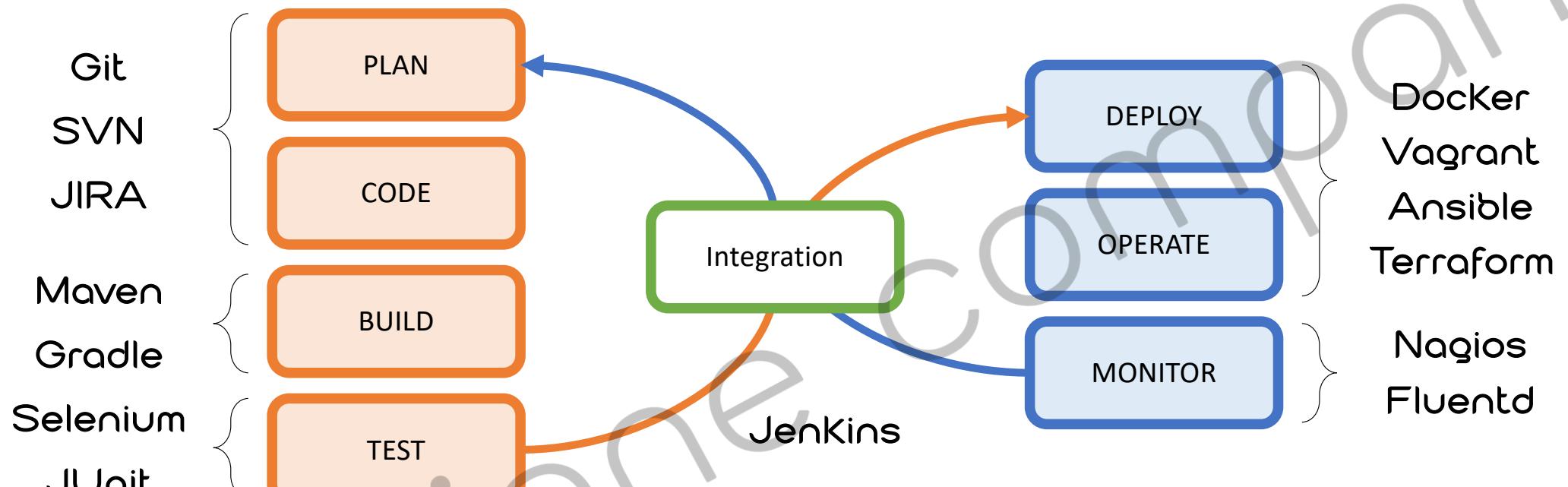
- 2007~2008년 → IT 운영 및 SW 개발에 문제점 대두
  - 개발과 배포가 다른 조직에서 관리 (다른 목표를 가짐)
- 2009년 Belgium에서 첫 DevOps 컨퍼런스 →
  - *Development + Operations*
  - 인프라로 코드 관리 (Mark Burgess and Luke Kanies)
  - 애자일 인프라 스트럭쳐 (Andrew Shafer)
  - 애자일 시스템 관리 운동 (Patrick Debois)
  - Lean Startup (Eric Ries)
  - 지속적인 통합 및 배포 운동 → CI, CD



# Cloud Native – DevOps

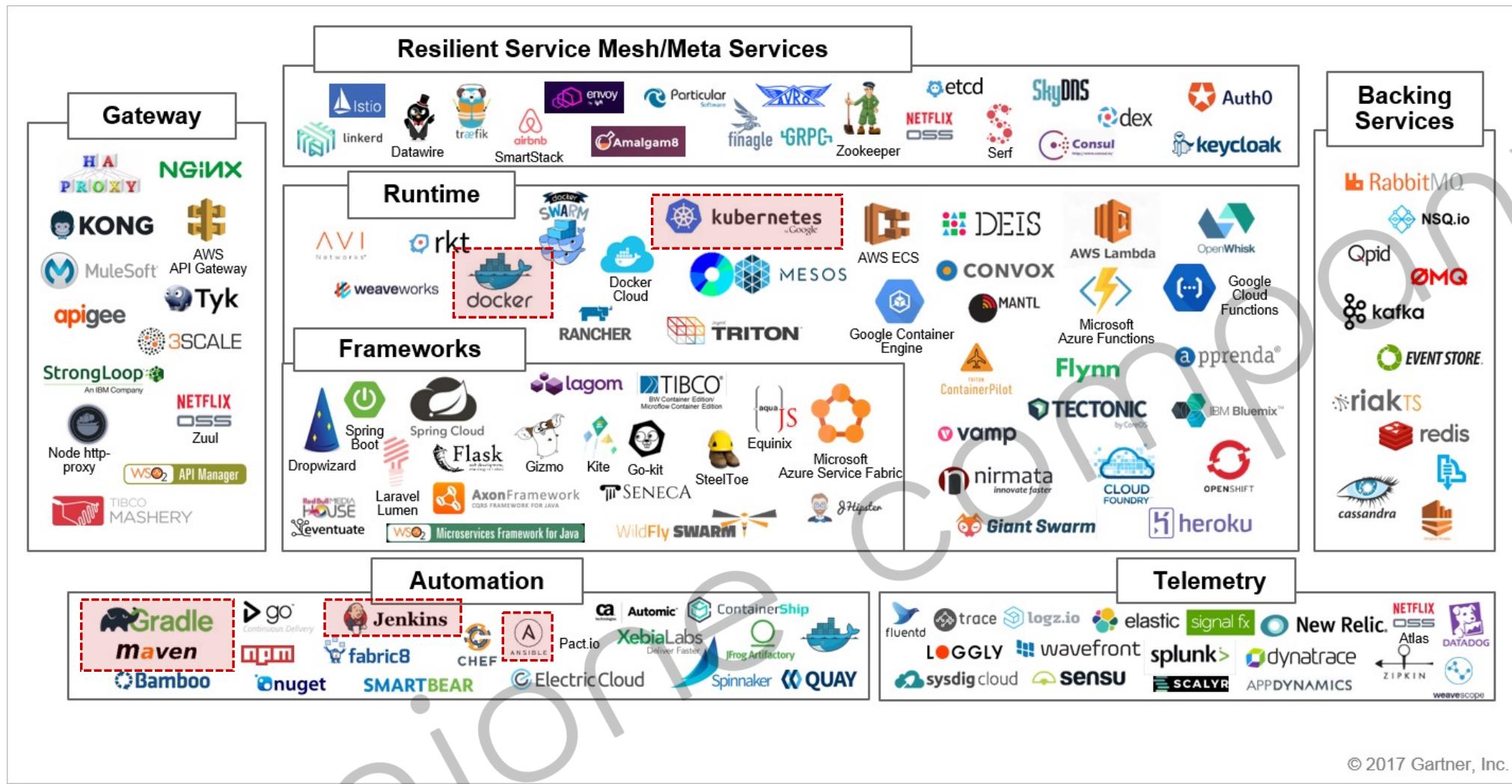
njone company

- 엔지니어가, **프로그래밍**하고, **빌드**하고, 직접 시스템에 배포 및 서비스를 **RUN**
- 사용자와 **끊임 없이** **Interaction**하면서 **서비스를 개선해** 나가는 일련의 과정, **문화**



# Cloud Native – DevOps

njone company



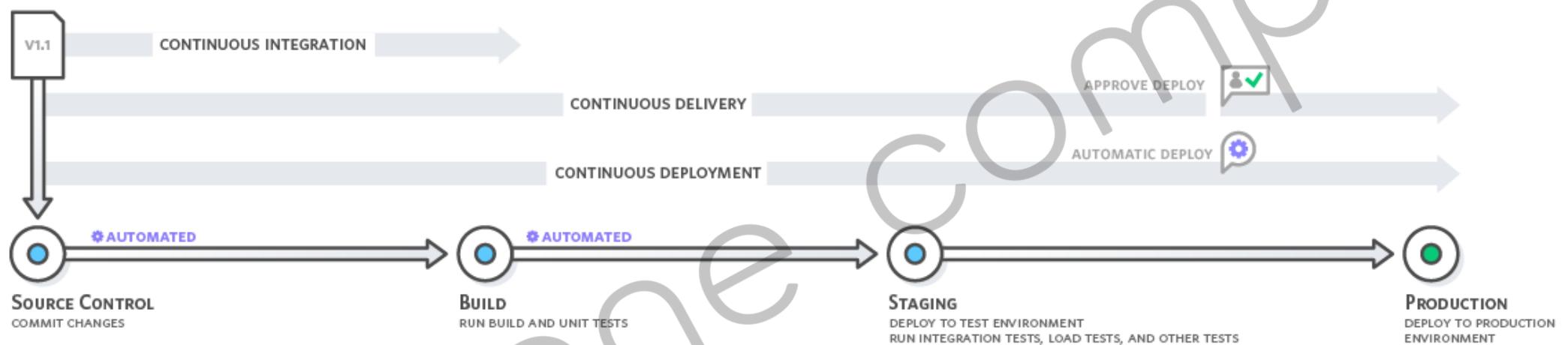
© 2017 Gartner, Inc.



# Cloud Native – CI/CD

njone company

- CI(Continuous Integration)?
- CD(Continuous Delivery)?
- CD(Continuous Deployment)?

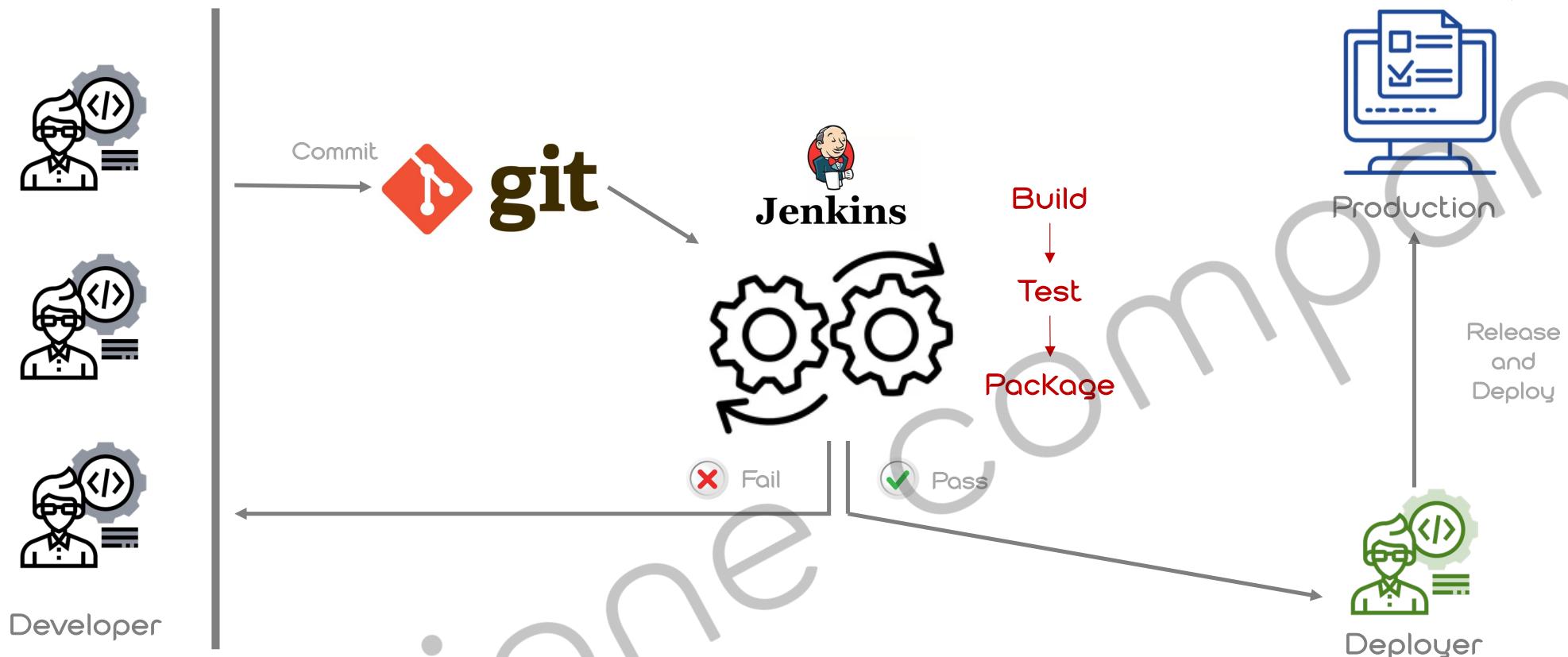


<https://aws.amazon.com/ko/devops/continuous-integration>



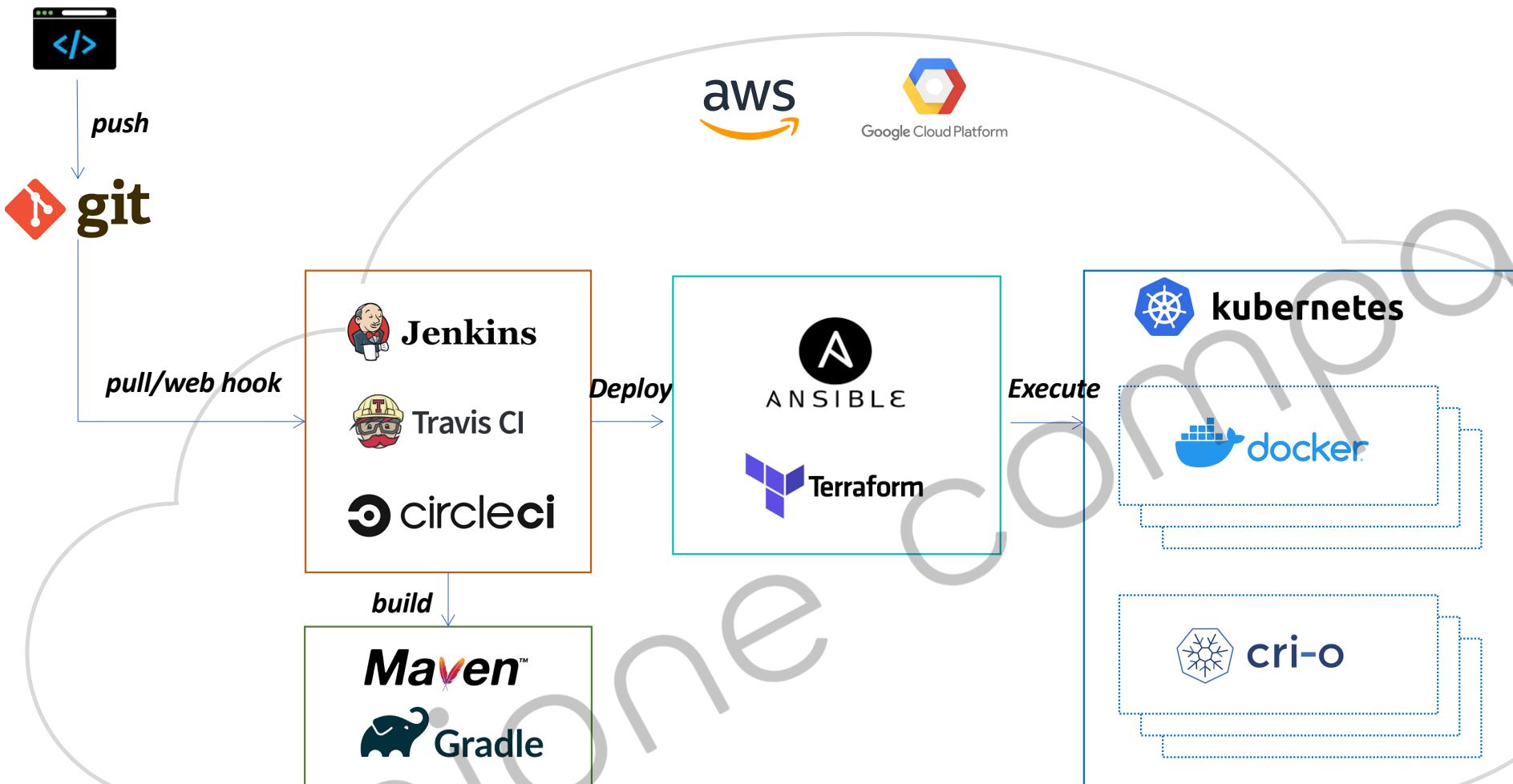
# What is Continuous Integration?

njone company



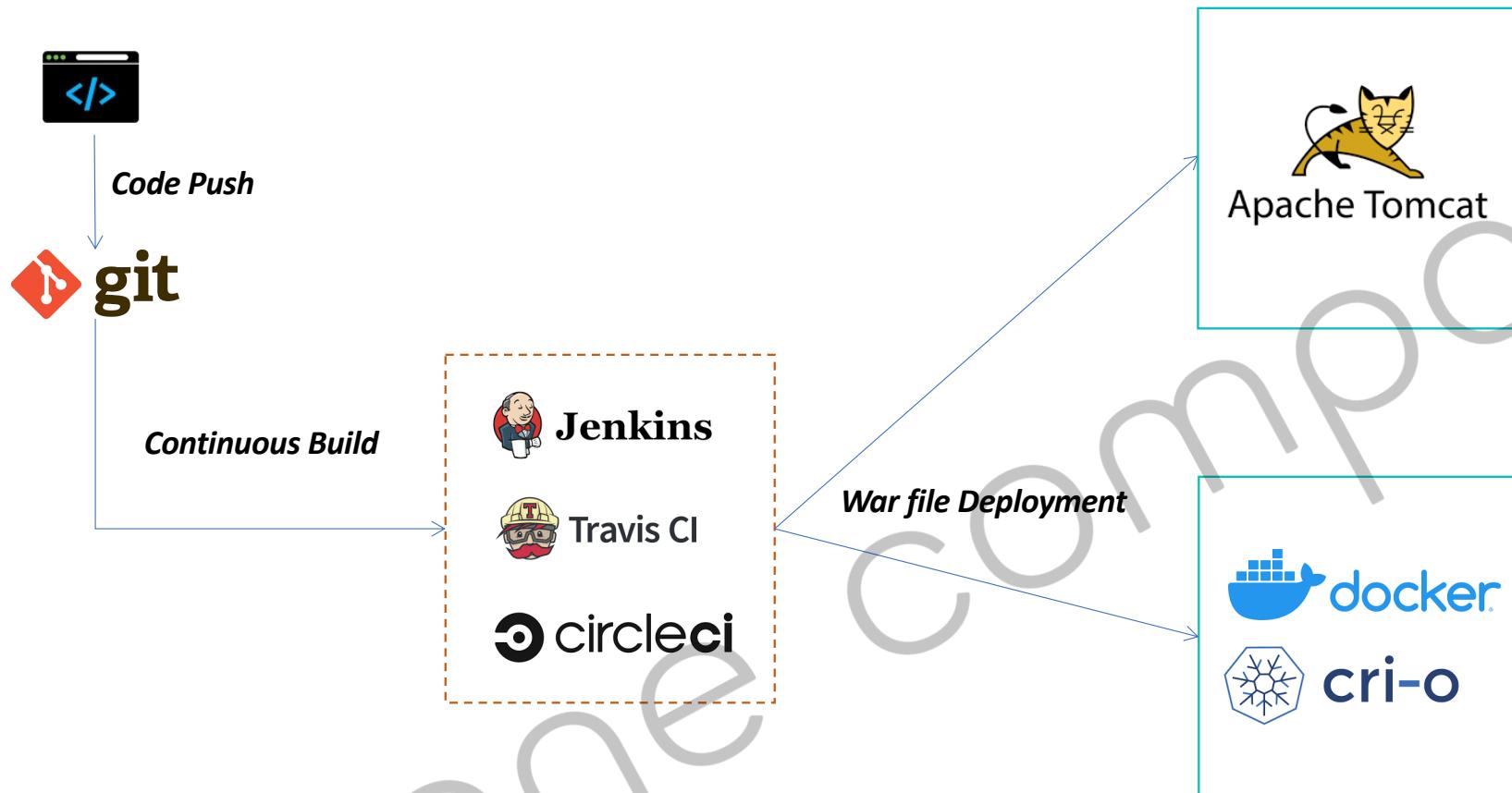
# CI/CD Flow

njone company



# Deploy on Docker using Jenkins

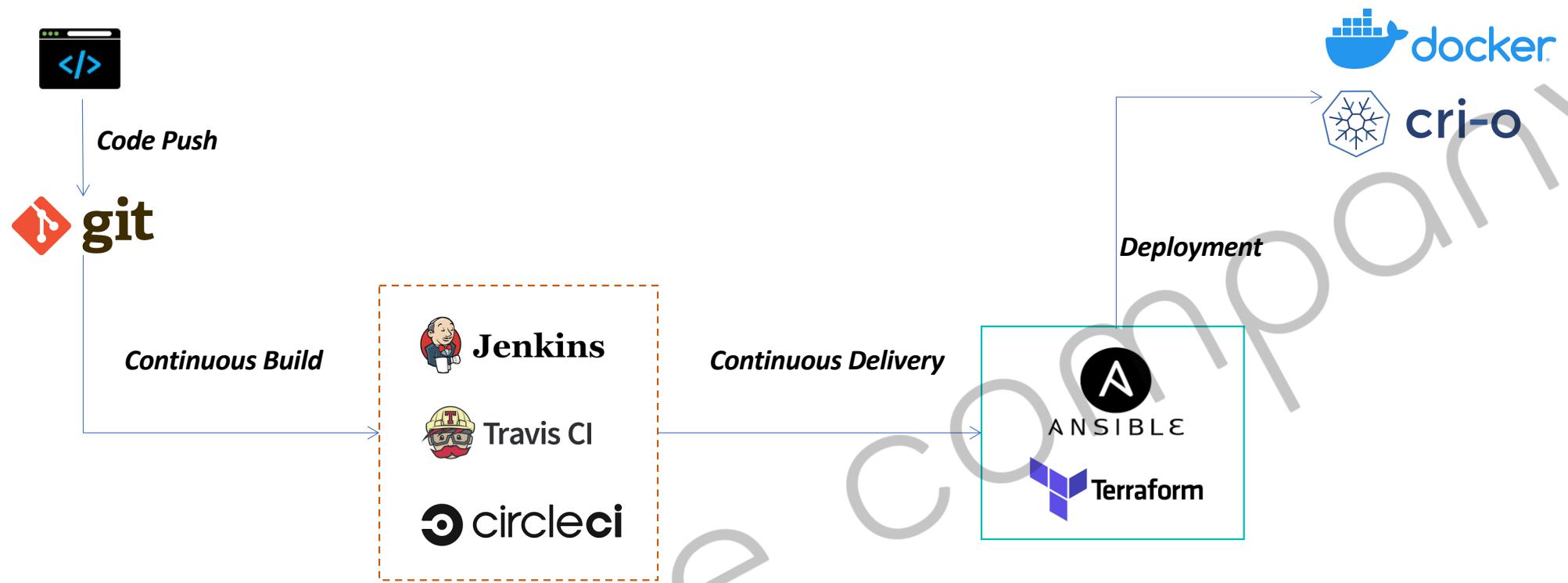
njone company





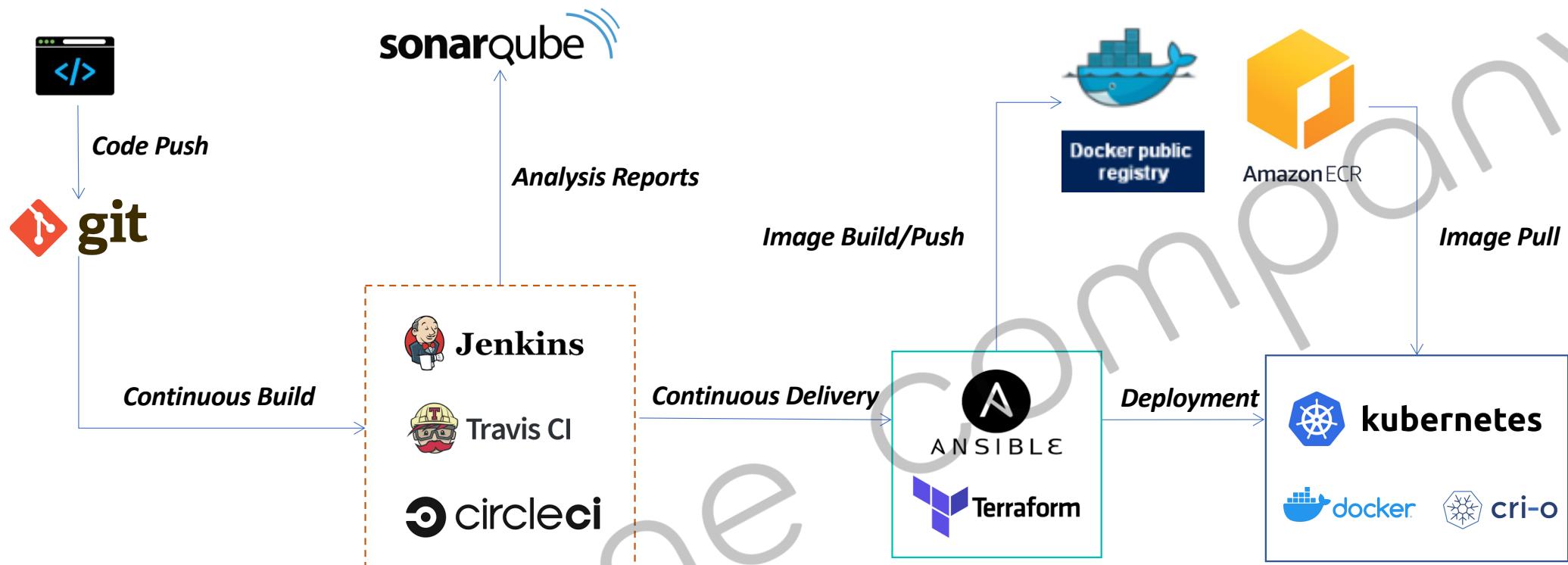
# Deploy on Docker using Jenkins

njone company



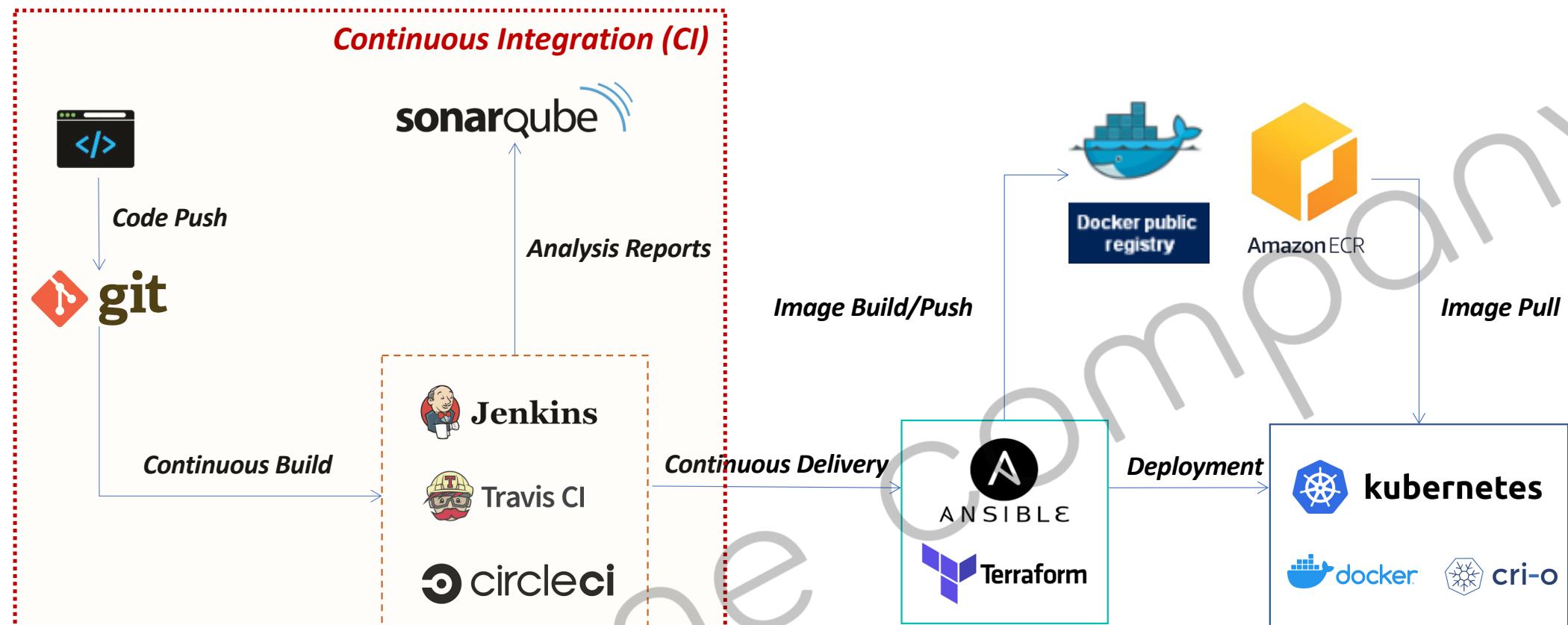
# Deploy on Kubernetes

njone company



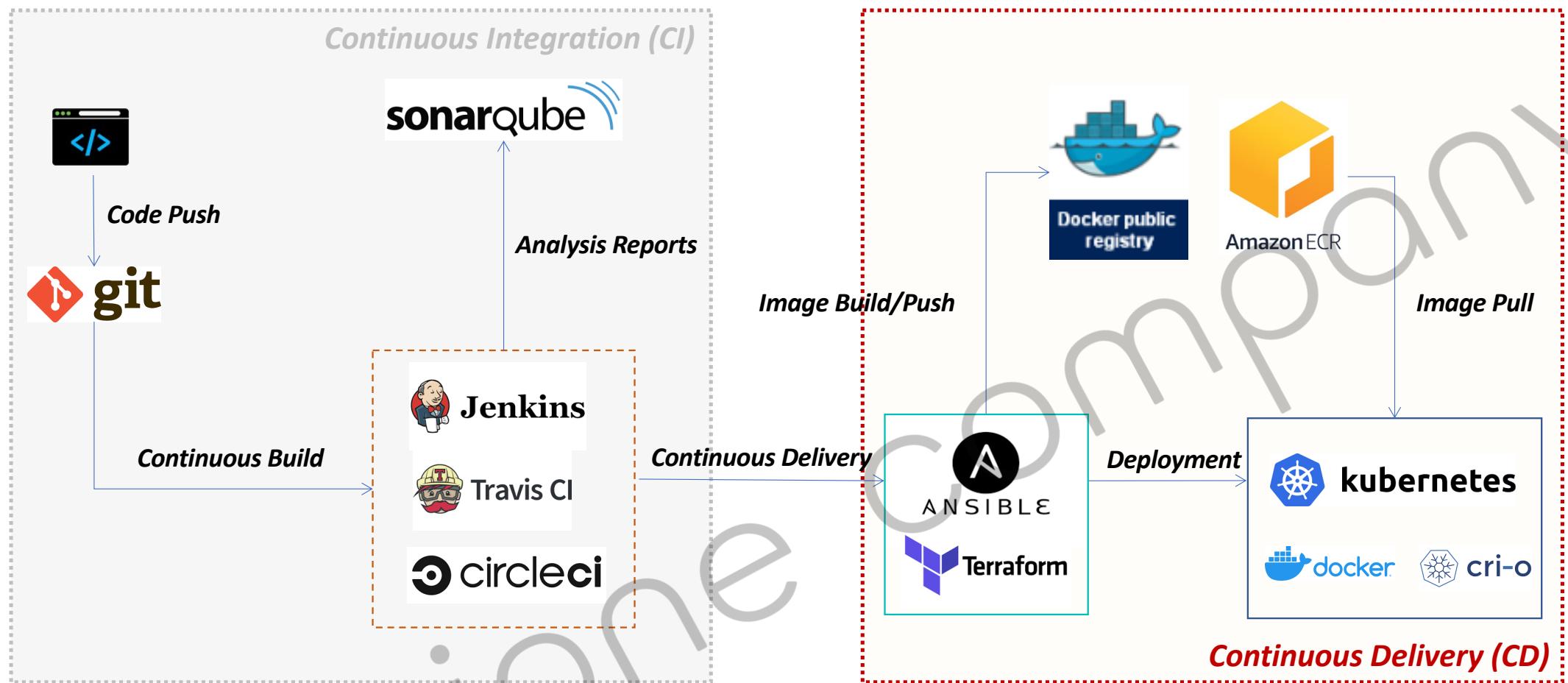
# Continuous Integration(CI)

njone company



# Continuous Delivery(CD)

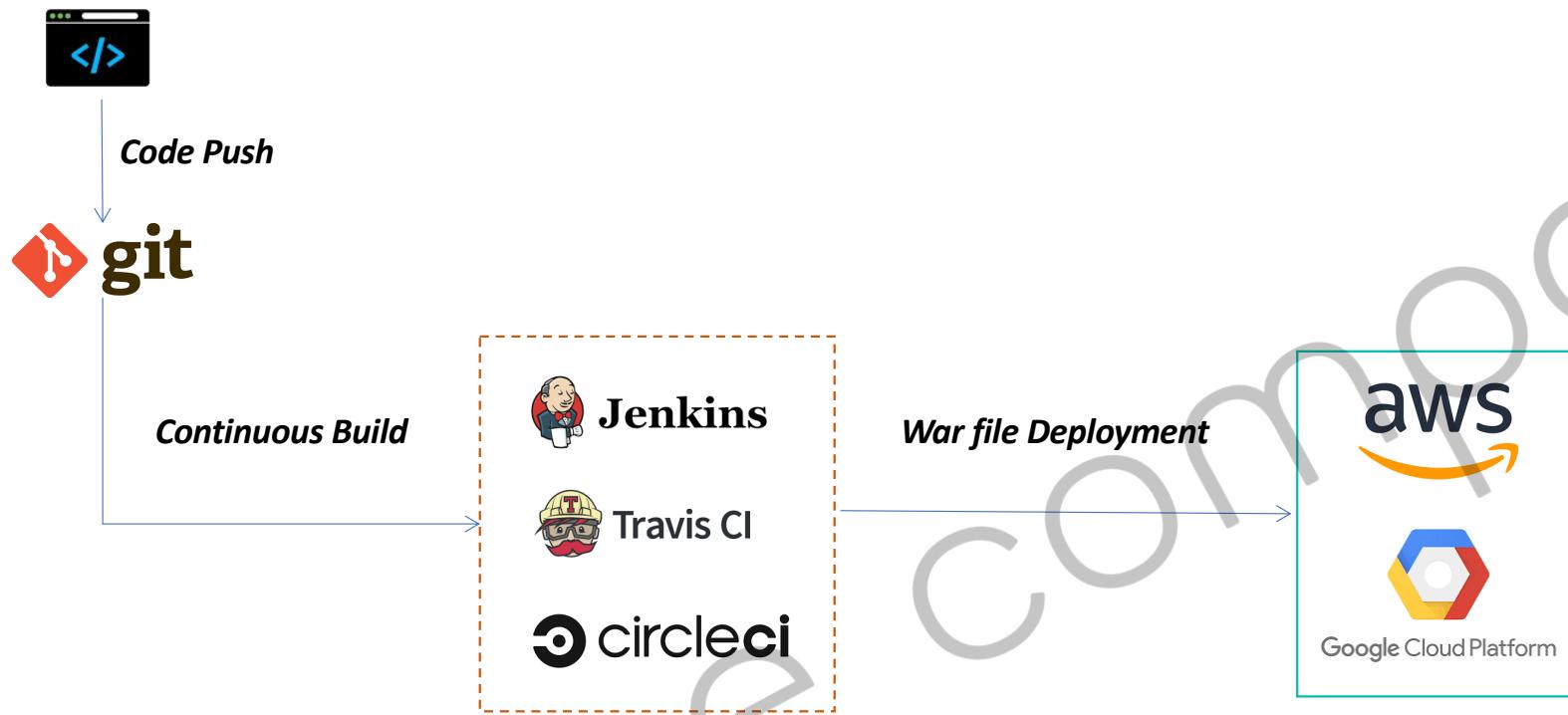
njone company





# Deploy on EC2/VM

njone company

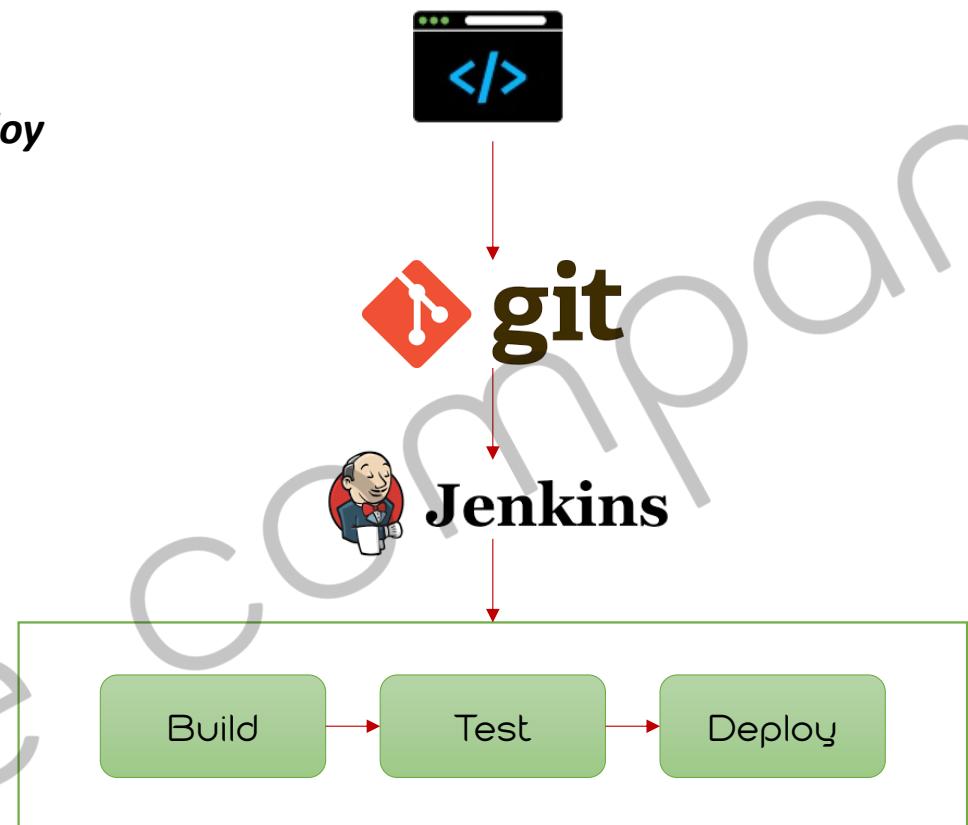


- 지속적인 통합과 배포 → **Work flow**를 제어

- *Continuous Integration Server*
    - *Continuous Development, Build, Test, Deploy*

- 다양한 Plugins 연동

- *Build Plugins: Maven, Ant, Gradle ...*
  - *VCS Plugins: Git, SVN ...*
  - *Languages Plugins: Java, Python, Node.js ...*





# Jenkins

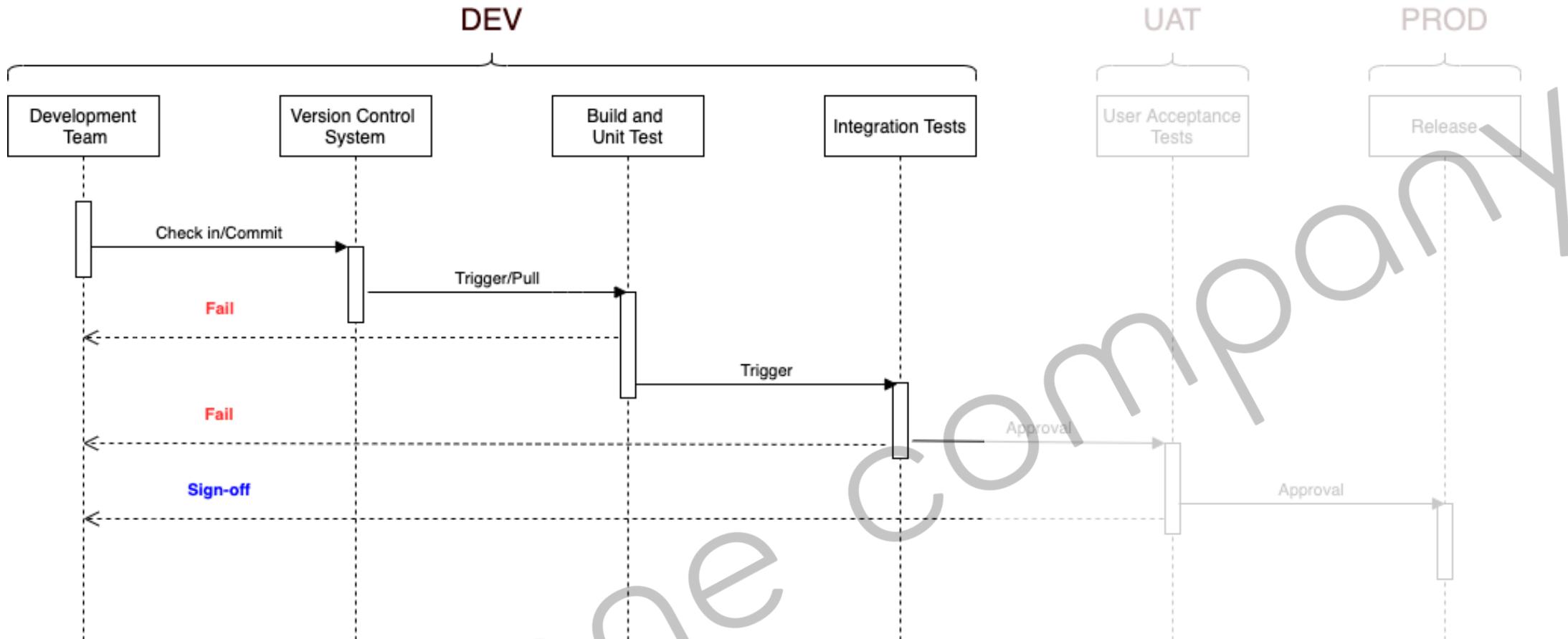
njone company

- Top 5 CI/CD tools

	Jenkins	circleci	TeamCity	Bamboo	GitLab
<b>Open source</b>	Yes	No	No	No	No
<b>Ease of use &amp; setup</b>	Medium	Medium	Medium	Medium	Medium
<b>Built-in features</b>	3/5	4/5	4/5	4/5	4/5
<b>Integration</b>	★★★★★	★★★	★★★★★	★★★	★★★★★
<b>Hosting</b>	On premise & Cloud	On premise & Cloud	On premise	On premise & Bitbucket as Cloud	On premise & Cloud
<b>Free version</b>	Yes	Yes	Yes	Yes	Yes
<b>Build Agent License Pricing</b>	Free	From \$39 per month	From \$299 one-off payment	From \$10 one-off payment	From \$4 per month per user
<b>Supported OSs</b>	Windows, Linux, macOS, Unix-like OS	Linux or MacOS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux distributions: Ubuntu, Debian, CentOS, Oracle Linux

# JenKins Pipeline

njone company



# Install Jenkins

njone company

- <https://www.jenkins.io/download/>

## Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the [Hardware and Software requirements](#) section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the [Installing Jenkins](#) section of the User Handbook.
4. You may also want to verify the package you downloaded. [Learn more about verifying Jenkins downloads.](#)

### Download Jenkins 2.303.1 LTS for:

Generic Java package (.war)  
SHA-256: 4aae135cd63e398a1f59d37978d97604cb595314f7041d2d3bac3f0bb32c065

Docker

Ubuntu/Debian

CentOS/Fedora/Red Hat

Windows

openSUSE

FreeBSD ⚙

Gentoo ⚙

macOS ⚙

OpenBSD ⚙

### Download Jenkins 2.309 for:

Generic Java package (.war)

SHA-256: 41c5de84c7afe8634212641e143ff02d75105c5697ab4114ff82b4c97916811

Docker

Ubuntu/Debian

CentOS/Fedora/Red Hat

Windows

openSUSE

Arch Linux ⚙

FreeBSD ⚙

Gentoo ⚙

macOS ⚙

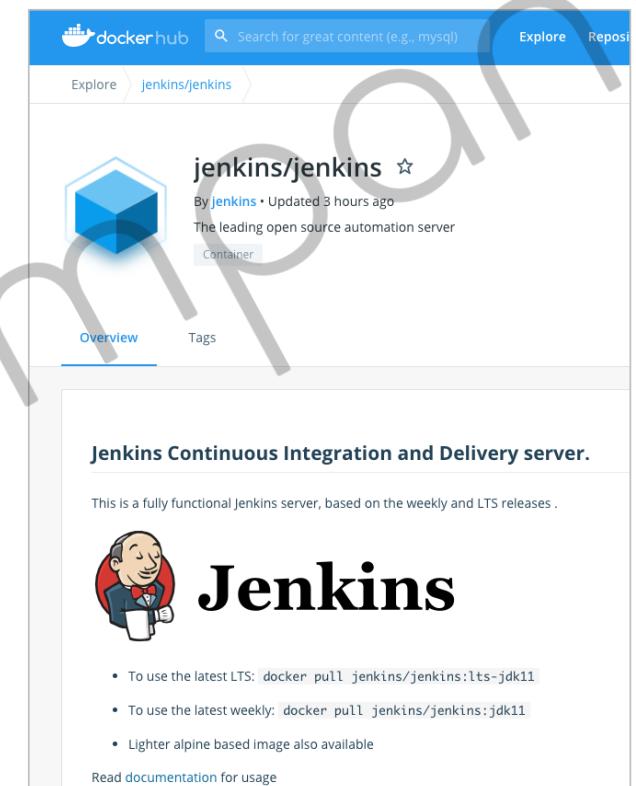
# Install Jenkins

njone company

- <https://hub.docker.com/r/jenkins/jenkins>
- <https://github.com/jenkinsci/docker/blob/master/README.md>

```
$ docker run -d -p 8088:8080 -p 50000:50000 --name jenkins-server \
--restart=on-failure \
jenkins/jenkins:lts-jdk11
```

```
$ docker run -d -p 8088:8080 -p 50000:50000 --name jenkins-server \
--restart=on-failure \
-v jenkins_home:/var/jenkins_home \
jenkins/jenkins:lts-jdk11
```





# Install Jenkins

njone company

- Check the init password

```
2021-09-02 06:35:47.922+0000 [id=29] INFO jenkins.install.SetupWizard#init:
```

```
*****
*****  
*****  
*****
```

Jenkins initial setup is required. An admin user has been created and a password generated.  
Please use the following password to proceed to installation:

```
22263a23f9ed4e0ea8ed71f976f37c7b
```

This may also be found at: /var/jenkins\_home/secrets/initialAdminPassword

```
*****
*****  
*****
```

```
2021-09-02 06:35:58.710+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2021-09-02 06:35:58.726+0000 [id=23] INFO hudson.WebAppMain$3#run: Jenkins is fully up and running
2021-09-02 06:35:59.337+0000 [id=49] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2021-09-02 06:35:59.338+0000 [id=49] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
2021-09-02 06:35:59.341+0000 [id=49] INFO hudson.model.AsyncPeriodicWork$Lambda$doRun$0: Finished Download metadata. 11,777 ms
```



# Setting Jenkins

njone company

- Access Jenkins

- <http://YOUR-SERVER-PUBLIC-IP:8088>
- Administrator Password: [Using the init password]

Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



# Setting Jenkins

njone company

## Getting Started

### Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

#### Install suggested plugins

Install plugins the Jenkins community finds most useful.

#### Select plugins to install

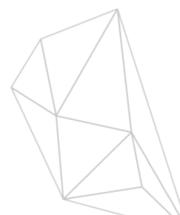
Select and install plugins most suitable for your needs

## Getting Started

### Getting Started

<input checked="" type="checkbox"/> Folders	<input type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding
<input type="checkbox"/> Timestamper	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline: Stage View
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	

\*\* SSH server  
Folders  
\*\* Trilead API





# Setting Jenkins

njone company

Getting Started

## Create First Admin User

계정명:

암호:

암호 확인:

이름:

이메일 주소:

## Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

## Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

# Setting Jenkins

njone company

- <http://127.0.0.1:8088/>

The screenshot shows the Jenkins dashboard. At the top, there is a navigation bar with the Jenkins logo, a search bar, and user information for 'Administrator'. On the left, a sidebar menu includes 'Dashboard', '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리', 'My Views', 'Lockable Resources', and 'New View'. Below the sidebar, two sections are expanded: '빌드 대기 목록' (which is empty) and '빌드 실행 상태' (which shows 1 build in progress and 2 builds pending). The main content area features a large message 'Jenkins에 오신 것을 환영합니다.' (Welcome to Jenkins), a sub-message about displaying Jenkins jobs, and several call-to-action buttons: 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

# Setting Jenkins

njone company

## ■ Jenkins 설정

The screenshot shows the Jenkins dashboard with a red dashed box highlighting the 'Jenkins 관리' (Management) link in the sidebar. The main content area displays the 'System Configuration' section, which is also highlighted with a red dashed box. Other sections visible include 'Security', 'Status Information', and 'Troubleshooting'. The top right of the dashboard includes a search bar, user information for 'Administrator', and a 'Logout' button.

**Jenkins 관리**

Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#).

**System Configuration**

시스템 설정 환경변수 및 경로 정보들을 설정합니다.

**Global Tool Configuration**  
Configure tools, their locations and automatic installers.

**Security**

Configure Global Security Secure Jenkins; define who is allowed to access/use the system.

Manage Credentials Configure credentials

Configure Credential Providers Configure the credential providers and types

**Status Information**

시스템 정보 문제 해결을 돕기위한 다양한 환경 정보를 보여줍니다.

System Log System log captures output from java.util.logging output related to Jenkins.

부하 통계 Check your resource utilization and see if you need more computers for your builds.

**Troubleshooting**

Manage Old Data Scrub configuration files to remove remnants from old plugins and earlier versions.

**Administrator**

Set up agent Set up cloud Dismiss

Set up agent Set up cloud Dismiss

노드 관리 Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Manage Users Create/delete/modify users that can log in to this Jenkins

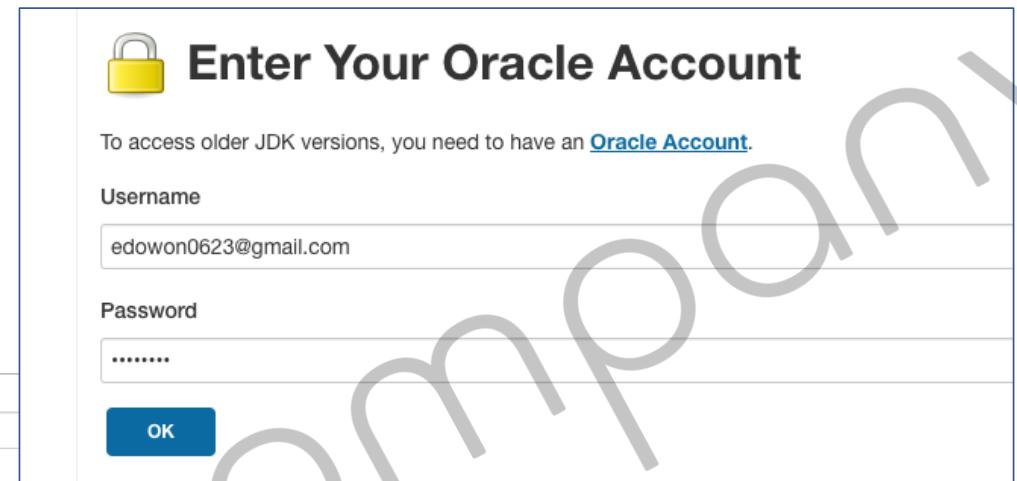
About Jenkins See the version and license information.

# Setting Jenkins

njone company

## ■ JDK

The screenshot shows the Jenkins 'JDK' configuration page. It includes sections for 'JDK installations' and 'JAVA\_HOME'. In the 'JAVA\_HOME' section, there is a field containing the path '/Users/downonlee/Library/Java/JavaVirtualMachines/openjdk-14.0.2/Contents/Home'. Below this field is a checkbox labeled 'Install automatically' which is unchecked and highlighted with a red dashed box. A red curved arrow points from this checkbox down to the 'Install Oracle Java SE Development Kit from the website' section. This section contains a dropdown menu for 'Version' set to 'Java SE Development Kit 9.0.4', a checked checkbox for 'I agree to the Java SE Development Kit License Agreement', and a note: 'Installing JDK requires Oracle account. Please enter your username/password'. At the bottom of this section is a note: 'Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license.' followed by 'Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a commercial license.' and a link to 'Oracle Java SE Licensing FAQ'. A button 'Add Installer' is located at the bottom left of this section.





# Exercise #1 Jenkins Job 1/2

njone company

- Item name → My-First-Project
  - *Freestyle project*
  - *Build > Execute shell*
  - *Save*

Enter an item name

» Required field

 **Freestyle project**  
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤

**Build**

 **Execute shell**

Command

```
echo "Welcome to my first project using Jenkins"
```

See [the list of available environment variables](#)

# Exercise #1 Jenkins Job 2/2

njone company

## ■ Build Now

The screenshot shows the Jenkins interface for the 'My-First-Project' job. On the left, a sidebar menu includes options like '대시보드로 돌아가기', '상태', '변경사항', '작업공간', 'Build Now' (which is selected and highlighted in blue), '구성', 'Project 삭제', 'Rename', and 'Build History'. The main content area displays the project name 'Project My-First-Project' and a '고정링크' section with links to '작업 공간' and '최근 변경사항'. Below this is a 'Console Output' section with a green checkmark icon and the heading '콘솔 출력'. The output log shows the following text:

```
Started by user Administrator
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/My-First-Project
Installing JDK jdk-9.0.4-oth-JPR
Downloading JDK from https://download.oracle.com/otn/java/jdk/9.0.4+11/c2514751926b451
Downloading 354635831 bytes
Installing /var/jenkins_home/tools/hudson.model.JDK/JDK9.0.4/jdk.sh
[JDK9.0.4] $ tar xzf /var/jenkins_home/tools/hudson.model.JDK/JDK9.0.4/jdk.sh
[My-First-Project] $ /bin/sh -xe /tmp/jenkins5722089168411804416.sh
+ echo Welcome to my first project using Jenkins
Welcome to my first project using Jenkins
Finished: SUCCESS
```



## 후속 강의 소개

- ~~Spring Cloud로 개발하는 마이크로서비스 애플리케이션~~
- ~~Jenkins를 이용한 CI/CD Pipeline 구축~~
- **Microservice Architecture 와 Patterns**
- **Spring Boot 와 WebFlux를 이용한 Reactive RESTful API 개발**